

 免費電子書

學習

iOS

Free unaffiliated eBook created from
Stack Overflow contributors.

#iOS

.....	1
1: iOS	2
.....	2
.....	2
Stack Overflow	2
.....	2
Examples	3
.....	3
.....	9
.....	9
.....	13
.....	14
.....	15
.....	16
Xcode	16
.....	18
.....	18
.....	21
.....	23
Swift 3	23
.....	23
2: 3D	29
Examples	29
3DSwift	29
3 DObjective-C	30
3: AFNetworking	32
Examples	32
.....	32
4: Alamofire	33
.....	33
.....	

Examples.....	33
.....	33
.....	33
.....	33
.....	33
.....	34
.....	34
5: AppDelegate.....	35
.....	35
Examples.....	35
AppDelegate.....	35
AppDelegate.....	36
URL.....	36
.....	36
6: ARC.....	38
Examples.....	38
/ARC.....	38
7: Autolayout/.....	39
.....	39
Examples.....	39
.....	39
8: AVPlayerAVPlayerViewController.....	40
.....	40
Examples.....	40
AVPlayerViewController.....	40
Objective-C.....	40
.....	40
AVPlayerAVPlayerLayer.....	40
C.....	40
.....	40
AVPlayer.....	41

9: AVSpeechSynthesizer	42
.....	42
.....	42
Examples.....	42
.....	42
10: AWSSDK	43
Examples.....	43
AWS SDKS3.....	43
11: CAAnimation	46
.....	46
Examples.....	46
.....	46
Objective-C	46
.....	46
-.....	46
Objective-C.....	46
.....	46
.....	47
.....	47
.....	47
.....	47
C.....	47
.....	47
12: CAGradientLayer	49
.....	49
.....	49
.....	49
Examples.....	49
CAGradientLayer.....	49
CGGradientLayer.....	50
CAGradientLayer.....	51
CAGradientLayer.....	52

CAGradientLayer.....	53
13: CALayer.....	55
Examples.....	55
CALayer.....	55
CAEmitterLayer.....	55
.....	56
UIImageCALayer.....	57
.....	57
.....	61
.....	61
CALayer.....	61
.....	61
.....	61
.....	62
.....	63
.....	63
.....	64
.....	65
.....	65
.....	65
.....	65
.....	65
14: CAShapeLayer.....	67
.....	67
.....	67
Examples.....	67
CAShapeLayer.....	67
.....	71
.....	72
CAShapeLayer.....	72
15: CGContext.....	74

.....	74
Examples.....	74
.....	74
.....	74
16: CLLocation	76
Examples.....	76
.....	76
CLLocationManager.....	76
17: CloudKit	78
.....	78
.....	78
Examples.....	78
CloudKit.....	78
CloudKit.....	79
.....	79
CloudKit.....	79
.....	79
.....	79
.....	79
.....	79
Objective-C.....	79
.....	79
.....	79
.....	79
CloudKit	80
.....	80
18: CoreImage	81
Examples.....	81
.....	81
19: CTCallCenter	87
Examples.....	87

.....	95
.....	95
.....	95
.....	95
Objective-C.....	95
.....	95
.....	95
.....	95
Objective-C.....	95
.....	96
.....	96
.....	96
.....	96
Objective-C.....	96
23: FacebookSDK.....	97
Examples.....	97
FacebookSDK.....	97
“Facebook”.....	99
facebook.....	100
24: GameCenter.....	102
Examples.....	102
GameCenter.....	102
25: GameplayKit.....	105
Examples.....	105
.....	105
.....	105
.....	105
Objective-C.....	105
.....	105
Objective-C.....	105
.....	105

On	105
.....	105
Objective-C.....	105
mn	106
.....	106
Objective-C.....	106
.....	106
Objective-C.....	106
GKEntityGKComponent.....	106
GKEntity	106
GKComponent	107
GKComponentSystem	107
26: GCDGrand Central Dispatch	109
.....	109
Examples.....	109
.....	109
.....	109
.....	110
.....	110
.....	111
27: Healthkit	113
Examples.....	113
HealthKit.....	113
28: iBeacon	116
.....	116
.....	116
Examples.....	116
iBeacon.....	116
.....	117
iBeacons.....	117
29: IBOutlets	118
.....	

Examples.....	118
UIBOutlet.....	118
30: iOS - Robbie HansonXMPP.....	120
Examples.....	120
iOS XMPP Robbie HansonOpenfire.....	120
SRXMPPDemo.....	120
- https://github.com/SahebRoy92/SRXMPPDemo	120
.....	120
31: iOS 10API.....	123
Examples.....	123
.....	123
32: iOS Google Places API.....	125
Examples.....	125
.....	125
33: iOS TTS.....	127
.....	127
Examples.....	127
.....	127
C.....	127
.....	127
.....	127
34: iOSAirPrint.....	128
Examples.....	128
AirPrint.....	128
35: iOSSpotLight.....	130
Examples.....	130
.....	130
36: iOS.....	142
.....	142
Examples.....	142

URL.....	142
URL.....	142
Info.plistURL.....	142
UIApplicationDelegateURL.....	143
URL.....	143
.....	144
37: MKDistanceFormatter.....	146
Examples.....	146
.....	146
.....	146
.....	146
38: ModelPresentationStyles.....	148
.....	148
.....	148
Examples.....	148
Interface BuilderModalPresentationStyle.....	148
39: MPMediaPickerDelegate.....	150
.....	150
Examples.....	150
MPMediaPickerControllerDelegateAVAudioPlayer.....	150
40: MPVolumeView.....	152
.....	152
.....	152
Examples.....	152
MPVolumeView.....	152
41: MVP.....	153
.....	153
.....	153
Examples.....	153
Dog.swift.....	153
DoggyView.swift.....	154

DoggyService.swift	154
DoggyPresenter.swift	154
DoggyListViewController.swift	155
42: MVVM	157
Examples	157
MVVM	157
43: MyLayout	160
.....	160
Examples	160
MyLayout	160
44: NSArray	162
.....	162
.....	162
Examples	162
Arrayjson	162
45: NSAttributedString	163
.....	163
Examples	163
.....	163
.....	163
Swift	163
.....	163
.....	164
46: NSHTTPCookieStorage	165
Examples	165
NSUserDefaultcookie	165
47: NSInvocation	167
Examples	167
NSInvocation Objective-C	167
48: NSNotificationCenter	169
.....	169
.....	

.....	169
Examples.....	169
.....	169
.....	169
2.3	169
3	169
Objective-C	170
.....	170
2.3	170
3	170
Objective-C	170
.....	170
.....	170
Objective-C	170
.....	171
.....	171
Objective-C	171
.....	171
.....	171
Objective-C	171
/.....	171
.....	171
49: NSPredicate	173
.....	173
Examples.....	173
predicateWithBlockNSPredicate.....	173
Objective-C	173
.....	173
predicateWithFormatNSPredicate.....	173

Objective-C.....	173
.....	174
NSPredicate.....	174
Objective-C.....	174
.....	174
NSPredicate.....	174
Objective-C.....	174
.....	174
NSPredicate.....	175
NSPredicate`AND``OR``NOT`.....	176
Objective-C.....	176
-.....	176
-.....	177
-.....	177
50: NSURL.....	178
Examples.....	178
NSURL String.....	178
SwiftURLNSURL.....	178
51: NSURLConnection.....	179
Examples.....	179
.....	179
.....	179
.....	180
52: NSURLSession.....	181
.....	181
Examples.....	181
GET.....	181
Objective-C.....	182
.....	182
Objective-CNSURLSessionPOST.....	183
53: NSUserActivity.....	188
.....	188

.....	188
.....	188
/.....	188
.....	188
Examples.....	188
NSUserActivity.....	188
54: NSUserDefaults.....	190
.....	190
.....	190
Examples.....	190
.....	190
<3.....	190
3.....	190
Objective-C.....	190
<3.....	190
3.....	190
Objective-C.....	191
.....	191
.....	191
Objective-C.....	191
.....	191
.....	191
Objective-C.....	191
.....	192
Objective-C.....	192
.....	192
.....	192
Objective-C.....	192
.....	192
.....	192
Objective-C.....	193
.....	

193	
NSUserDefaults.....	193
.....	194
Objective-C.....	194
UserDefaultsSwift 3.....	194
55: Objective-C.....	196
Examples.....	196
.....	196
56: Objective-C.....	198
.....	198
.....	198
.....	198
.....	198
Examples.....	198
.....	198
57: OpenGL.....	200
.....	200
Examples.....	200
.....	200
58: PDFiOS.....	201
Examples.....	201
PDF.....	201
PDF.....	202
PDF.....	203
UIWebViewMicrosoft DocumentPDF.....	203
59: plist iOS.....	205
.....	205
Examples.....	205
.....	205
/Plist.....	210
60: QR.....	211
.....	

211	
Examples.....	211
UIViewControllerQR.....	211
AVFoudationQR.....	212
1.....	212
2.....	212
3.....	213
61: Safari.....	215
Examples.....	215
SFSafariViewControllerDelegate.....	215
Safari.....	215
SafariViewControllerURL.....	216
62: SLComposeViewController.....	217
Examples.....	217
TwitterFacebookSinaWelboTencentWelboSLComposeViewController.....	217
63: SqlCipher.....	219
.....	219
.....	219
Examples.....	221
.....	221
64: StoreKit.....	223
Examples.....	223
App Store.....	223
65: SwiftAppDelegaterootViewControllermainlogin / onboarding.....	224
.....	224
.....	224
.....	224
Examples.....	224
1.....	224
2.....	225
66: SwiftFCM.....	226
.....	

226	
Examples.....	226
SwiftFCM.....	226
67: SwiftObjective-C.....	228
Examples.....	228
SwiftObjective-C.....	228
1Objective-C - .m.....	228
2.....	228
3Objective-C - .h.....	229
4Objective-C.....	229
5Bridging-Header.....	230
6.....	230
Objective-CSwift.....	230
1Swift.....	230
2SwiftObjC.....	231
3.....	231
.....	231
68: SWRevealViewController.....	232
.....	232
Examples.....	232
SWRevealViewController.....	232
69: UIActivityViewController.....	237
.....	237
Examples.....	237
.....	237
Objective-C.....	237
.....	237
70: UIAlertController.....	238
.....	238
Examples.....	238
UIAlertControllerAlertViews.....	238
.....	239

.....	239
UIAlertController.....	240
.....	240
Objective-C.....	240
UIAlertController.....	241
.....	241
.....	241
Objective-C.....	241
.....	241
Objective-C.....	241
.....	241
Objective-C.....	242
.....	242
Objective-C.....	242
.....	242
.....	243
Objective-C.....	243
.....	243
.....	243
.....	244
.....	244
.....	245
.....	245
.....	246
.....	246
.....	247
.....	247
.....	247
71: UIAppearance.....	249
Examples.....	249
.....	249

.....	250
72: UIBezierPath	251
Examples.....	251
UIBezierPath.....	251
UIBezierPath.....	253
UIBezierPath + AutoLayout.....	255
UIBezierPath.....	255
.....	256
Bézier	256
.....	257
.....	257
.....	258
.....	260
.....	262
.....	262
UIBezierPath.....	262
73: UICollectionView	266
Examples.....	266
.....	266
Swift - UICollectionViewDelegateFlowLayout.....	266
UICollectionView.....	266
UICollectionView -	267
Swift.....	268
.....	268
.....	268
.....	269
.....	271
.....	271
.....	272
.....	273
.....	273

UICollectionViewDelegate.....	274
DataSourceFlowLayout.....	275
74: UIControl -	278
Examples.....	278
.....	278
75: UIFeedbackGenerator.....	281
.....	281
Examples.....	281
.....	281
.....	281
Objective-C.....	281
76: UIFont.....	283
.....	283
Examples.....	283
UIFont.....	283
.....	283
77: UIGestureRecognizer.....	284
Examples.....	284
UITapGestureRecognizer.....	284
UIPanGestureRecognizer.....	285
UITapGestureRecognizer.....	286
.....	286
UILongPressGestureRecognizer.....	286
.....	286
UISwipeGestureRecognizer.....	286
.....	288
UIPinchGestureRecognizer.....	288
.....	288
UIRotationGestureRecognizer.....	288
.....	288
Interface Builder.....	289
.....	

78: UIKit **291**

..... 291

..... 291

..... 291

Objective-C 291

Examples 291

..... 291

..... 293

..... 293

Objective-C 294

 UITextFieldBehaviors⁴⁹⁹ 296

..... 297

Objective-C 299

 UIDynamicBehavior 301

..... 302

Objective-C 302

..... 303

Objective-C 304

..... 305

Objective-C 309

 UIDynamicBehaviors 313

..... 313

Objective-C 314

..... 315

Objective-C 316

..... 316

Objective-C 320

..... 324

..... 325

Objective-C 325

.....

326	
Objective-C	326
.....	327
Objective-C	328
79: UILabelattributedString	330
.....	330
Examples	330
UILabelHTML	330
UILabel	330
80: UILabel	332
Examples	332
Objective UILabel	332
SwiftUILabel	332
81: UILocalNotification	333
.....	333
.....	333
Examples	333
.....	333
.....	333
.....	334
UUID	334
.....	334
.....	335
.....	335
.....	336
Swift 3.0iOS 10	336
iOS10UILocalNotification	337
82: UINavigationController	340
.....	340
Examples	340
.....	340
.....	340

.....	340
NavigationController.....	340
.....	341
.....	341
.....	341
83: UIPageViewController.....	342
.....	342
.....	342
.....	342
Examples.....	342
UIPageViewController.....	342
.....	343
84: UIPheonix - UI.....	348
.....	348
.....	348
Examples.....	348
UI.....	348
.....	349
85: UIPickerView.....	350
Examples.....	350
.....	350
.....	350
Objective-C.....	350
pickerView.....	351
86: UIRefreshControl TableView.....	352
.....	352
Examples.....	352
Objective-C.....	352
tableViewrefreshControl.....	352
87: UIScrollView AutoLayout.....	354
Examples.....	354

ScrollableController.....	354
Storyboard UIScrollView.....	357
88: UIScrollView.....	360
Examples.....	360
UIScrollView.....	360
.....	360
ScrollView with AutoLayout.....	360
.....	366
.....	366
.....	367
.....	367
.....	369
.....	369
/.....	369
/UIImageView.....	369
UIImageView.....	370
UIScrollView.....	370
C.....	370
.....	371
.....	371
89: UISearchController.....	372
.....	372
.....	372
.....	372
Examples.....	372
.....	372
.....	375
.....	378
Objective-CUISerachController.....	378
90: UISegmentedControl.....	380
.....	380

Examples.....	380
UISegmentedControl.....	380
91: UISlider.....	381
Examples.....	381
UISlider.....	381
SWIFT.....	381
.....	381
92: UISplitViewController.....	383
.....	383
Examples.....	383
Objective CDelegatesMasterDetail View.....	383
93: UISplitViewController.....	392
.....	392
Examples.....	392
Objective C.....	392
94: UICollectionViewController.....	396
Examples.....	396
.....	396
.....	396
UICollectionView.....	397
95: UIStoryboard.....	405
.....	405
Examples.....	405
UINavigationController.....	405
.....	405
Objective-C.....	405
.....	405
96: UISwitch.....	406
.....	406
.....	406
1. UISwitch Apple.....	406

2. Enoch Huang.....	406
Examples.....	406
/.....	406
.....	406
.....	407
/.....	407
97: UITableView.....	408
.....	408
.....	408
.....	409
Examples.....	409
.....	409
UITableView.....	410
UITableView.....	411
.....	411
.....	411
View Controller.....	412
.....	412
.....	412
.....	413
.....	413
Objective-C.....	414
.....	415
UITableViewDataSource.....	415
UITableViewDelegate.....	418
.....	420
.....	421
UITableViewCells.....	423
.....	425
.....	426
.....	427

.....	427
.....	427
.....	427
.....	428
.....	428
.....	428
.....	428
.....	428
98: UITextField	430
Examples	430
UITextField -	430
.....	430
/.....	431
99: UIView	433
Examples	433
.....	433
.....	433
100: UIWebView	435
.....	435
Examples	435
UIWebView	435
URL	435
Web	435
Web	436
.....	436
HTML	436
JavaScript	437
.pdf.txt.doc	437
UIWebview	438
webViewHTML	438
101: UI	440
.....	

Examples.....	440
Xcode.....	440
.....	440
.....	440
.....	440
.....	440
.....	440
.....	441
UITest.....	442
UIViewUIImageViewUIScrollView.....	442
UILabel.....	443
UIStackView.....	443
UITableView.....	443
UITableViewCell.....	443
UITableViewCell.....	443
UICollectionView.....	443
UIButtonUIBarButtonItem.....	443
UITextField.....	443
UITextView.....	444
UISwitch.....	444
.....	444
UI.....	444
.....	444
.....	444
.....	444
.....	444
.....	444
102: UUID.....	445
.....	445
Examples.....	445
UUID.....	445
UUID.....	445
.....	

445	
Objective-C	445
.....	445
.....	445
Objective-C	445
AppleFAIFVApple	445
iOSUUID	446
Swift 3.0	446
103: WCSSessionDelegate	447
.....	447
Examples	447
WKInterfaceController	447
104: WKWebView	448
.....	448
Examples	448
WebBrowser	448
.....	454
JavaScript	454
105: Xcode	455
.....	455
.....	455
.....	455
Examples	455
.....	455
106: XCTest -	456
Examples	456
Xcode	456
.....	456
.....	456
.....	456
Objective-C	457

StoryboardView Controller.....	457
.....	457
.....	458
StoryboardView Controller.....	458
.....	458
Objective-C.....	458
.....	458
.....	458
.....	458
Objective-C.....	458
.....	458
Objective-C.....	458
.....	459
.....	459
.....	459
.....	459
.....	459
.....	459
.....	459
.....	459
.....	459
.....	460
107: NSBundle.....	461
Examples.....	461
.....	461
.....	461
108:.....	462
.....	462
.....	462
.....	462

.....	462
Examples.....	462
-	462
.....	462
-	463
109:	464
Examples.....	464
.....	464
.....	464
.....	464
.....	464
110: CoreBluetooth	465
.....	465
.....	465
.....	465
UUID.....	465
SERVICE UUID.....	465
UInt16.....	465
Examples.....	466
BLE.....	466
.....	467
.....	468
111: CoreImage / OpenCV	470
Examples.....	470
.....	470
112: GPXiOS	473
Examples.....	473
.gpxMPS_HQ.gpx.....	473
.....	473
113: UICollectionViewUIKit Dynamics	475
.....	475

Examples.....	475
UIDynamicAnimator.....	475
.....	476
Objective-C.....	476
.....	477
Objective-C.....	478
.....	478
Objective-C.....	480
.....	481
Objective-C.....	483
114: Aseets.....	486
.....	486
Examples.....	486
.....	486
Image AssetsLaunchImage.....	489
.....	491
115:.....	492
Examples.....	492
.....	492
116: UIView.....	502
Examples.....	502
CUIView.....	502
117: XIBUIViews.....	503
.....	503
Examples.....	503
.....	503
XIBUIView.....	525
118:.....	527
.....	527
Examples.....	527
.....	527

119:	530
Examples	530
.....	530
.....	530
didSet	530
NSObject	531
.....	531
.....	531
120: .ipaApplicationloaderappstore	533
Examples	533
.ipaApplication Loaderappstore	533
121: ID	539
Examples	539
.....	539
.....	540
122:	542
Examples	542
.....	542
.....	542
123: UIStackView	543
Examples	543
UISwitch	543
124:	544
.....	544
Examples	544
.....	544
.....	544
Objective-C	544
.....	544
.....	544
Objective-C	544

WKWebView.....	544
.....	545
iOS 10.....	545
.....	546
125:	547
.....	547
Examples.....	547
Swift 4CodableJSONEncoderJSONDecoder.....	547
126: Coreplot	549
Examples.....	549
CorePlot.....	549
127: iOS	552
Examples.....	552
Swift.....	552
128:	553
Examples.....	553
Swift 2IAP.....	553
iTunesConnect.....	555
/IAP.....	556
129:	557
Examples.....	557
Segues.....	557
.....	558
.....	559
Objective-C.....	560
.....	560
Objective-C.....	560
segue.....	561
.....	562
.....	562
.....	563

130: MessageBox-Concept	564
.....	564
Examples.....	564
.....	564
131: I / O	565
Examples.....	565
Documents.....	565
132:	567
.....	567
Examples.....	567
UIView.....	567
.....	567
.....	568
133:	569
Examples.....	569
.....	569
Segue.....	569
PrepareForSegue	569
.....	569
Swift.....	569
Segue.....	569
ShouldPerformSegueWithIdentifier	569
.....	570
Swift.....	570
Segues.....	570
Segue.....	570
PerformSegueWithIdentifier.....	570
.....	570
Swift.....	571
134:	572
.....	572

Examples.....	572
.....	572
135:	576
.....	576
Examples.....	576
.....	576
.....	576
iPadiOS.....	577
136:	579
.....	579
Examples.....	579
Storyboard.....	579
137:	581
.....	581
Examples.....	581
SSL.....	581
iTunes.....	582
138: HTMLNSAttributed	584
Examples.....	584
Objective CHTMLNSAttributedStringVice Versa	584
139: NSAttributedStringUIImage	585
Examples.....	585
NSAttributedStringUIImage	585
140: UIImage	586
Examples.....	586
- C.....	586
SWIFT 3.....	587
141:	589
Examples.....	589
.....	589
SWIFT.....	589

142: StackViewUIScrollView	590
Examples	590
ScrollviewStackView	590
.....	591
StackViews	592
143:	593
.....	593
Examples	593
UIImages	593
144:	596
Examples	596
.....	596
fastlane	596
iOS	596
iOS TestFlight	596
Android	596
145:	597
Examples	597
UIViewController	597
.....	597
146:	598
.....	598
Examples	598
.....	598
.....	598
IPA	599
Application LoaderIPA	600
147: ATS	603
.....	603
.....	603
Examples	603

HTTP	603
HTTP	604
SSL	604
148:	606
.....	606
.....	606
Examples	606
.....	606
.....	606
Objective-C	606
.....	608
Objective-C	608
.....	609
Objective-C	609
.....	609
Objective-C	609
.....	609
.....	610
.....	610
.....	610
.....	610
App ID	612
.....	612
AppleApp IDAPN	612
XcodeAPN	613
.....	613
Objective-C	613
.....	613
.....	614
.....	614
.cer.pem	615

149: - iOS 10	617
.....	617
Examples.....	617
.....	617
.....	617
150:	620
.....	620
Examples.....	620
.....	620
ViewController.....	620
ViewController.....	620
151:	621
.....	621
Examples.....	621
.....	621
152:	623
Examples.....	623
UINavigationController.....	623
UINavigationController.....	623
ViewController.....	624
ViewController.....	624
.....	624
.....	624
1.....	624
2.....	625
Objective-C.....	625
153:	626
.....	626
Examples.....	626
iOS.....	626
154:	627
.....	

.....	627
.....	627
Examples.....	628
CoreLocation.....	628
.....	629
.....	629
.....	629
GPX.....	629
.....	630
155:	632
Examples.....	632
.....	632
156:	633
Examples.....	633
.....	633
.....	633
.....	633
.....	633
Objective-C.....	633
.....	633
.....	633
Objective-C.....	633
157:	635
.....	635
Examples.....	635
.....	635
158:	636
.....	636
.....	636
.....	636

.....	636
Examples	637
.....	637
3D / Force Touch.....	637
.....	637
.....	637
.....	637
.....	637
.....	637
.....	637
159:	638
.....	638
Examples	638
.....	638
160: iOS	639
Examples	639
iOS 8.....	639
.....	639
Objective-C	639
Swift 2.0.....	639
iOS.....	640
Objective-C	640
.....	640
3	640
161:	641
.....	641
.....	641
Examples	641
Reachability.....	641
.....	641
.....	641
WIFI.....	641
.....

162: SWIFT BRIDGING HEADER	644
Examples.....	644
Swift.....	644
Xcode.....	644
163:	646
.....	646
Examples.....	646
.....	646
.....	646
.....	646
.....	646
.....	646
.....	646
.....	646
.....	647
.....	647
164:	648
.....	648
Examples.....	648
.....	648
165: /	655
.....	655
Examples.....	655
/iOS.....	655
166: MKMapView	656
Examples.....	656
MKMapView.....	656
.....	656
.....	656
2.....	656
3.....	656

Objective-C.....	656
.....	657
2.....	657
3.....	657
Objective-C.....	658
.satelliteFlyover.....	658
2.....	658
3.....	659
Objective-C.....	659
.hybrid.....	659
2.....	659
3.....	659
Objective-C.....	659
.hybridFlyover.....	660
2.....	660
3.....	660
Objective-C.....	661
/.....	661
MKLocalSearch.....	661
OpenStreetMap Tile-Overlay.....	661
UserLocationUserTracking.....	663
Objective-C.....	663
.....	664
Objective-C.....	665
.....	665
Pin / Point Annotation.....	665
.....	665
.....	665
.....	665
.....	665
167: NSData.....	666
.....	

.....	666
Examples.....	666
NSData.....	666
.....	666
.....	666
Objective-C.....	666
String	666
.....	666
Objective-C.....	666
NSData.....	666
.....	666
.....	666
Objective-C.....	666
.....	666
.....	667
Objective-C.....	667
.....	667
.....	667
Objective-C.....	667
NSDataHEX.....	667
.....	667
Objective-C.....	667
168: NSDate	668
.....	668
.....	668
Examples.....	669
.....	669
.....	669
3.....	669
Objective-C.....	669

NSDateN.....	669
.....	669
3.....	669
Objective-C.....	670
.....	670
.....	670
Objective-C.....	670
.....	670
Objective-C.....	670
.....	670
Objective-C.....	671
3.....	671
Unix.....	672
.....	672
Objective-C.....	672
NSDateFormatter.....	672
1.NSDateFormatter.....	672
.....	672
3.....	672
Objective-C.....	673
2.....	673
.....	673
Objective-C.....	673
3.....	673
.....	673
3.....	673
Objective-C.....	673
.....	673
.....	673
NSDateNSDate.....	674
Objective-C.....	674

UTCTimeZoneNSDate.....	674
1224.....	674
AMPM.....	674
Objective-C.....	675
NSDateFormatter.....	675
Objective-C.....	675
.....	675
JSONNSDate“/1268123281843”.....	675
Objective-C.....	675
NSDate5s ago2m ago3h ago.....	675
Objective-C.....	676
169: NSTimer.....	677
.....	677
.....	677
Examples.....	677
.....	677
.....	677
.....	678
.....	678
.....	678
.....	678
Timer.....	679
170: UIBarButtonItem.....	680
.....	680
.....	680
Examples.....	680
UIBarButtonItem.....	680
Interface BuilderUIBarButtonItem.....	680
.....	680
.....	681
.....	681

IB	682
.....	683
.....	683
171: UIButton	684
.....	684
.....	684
.....	684
Examples.....	684
UIButton.....	684
.....	685
.....	685
.....	685
.....	686
UIButton.....	686
UIButton.....	686
.....	687
.....	687
UIButton.....	688
.....	688
.....	688
C	688
.....	688
.....	688
C	688
172: UIColor	689
Examples.....	689
UIColor.....	689
.....	690
styleString.....	690
_systemDestructiveTintColor().....	691
Alpha.....	691
.....	691

3.....	691
Objective-C.....	691
CGColor.....	691
borderUIColor.....	692
UIColor.....	692
UIColor.....	694
UIColor.....	695
UIColor.....	696
173: UIDatePicker.....	698
.....	698
Examples.....	698
.....	698
.....	698
Objective-C.....	698
-.....	698
.....	698
.....	698
.....	698
.....	698
.....	699
174: UIDevice.....	700
.....	700
.....	700
Examples.....	700
iOS.....	700
.....	702
.....	702
.....	702
.....	703
.....	704
175: UIImage.....	706
.....	706

Examples.....	706
UIImage.....	706
.....	706
.....	706
Objective-C.....	706
.....	706
NSData	706
.....	706
UIColor	706
.....	706
Objective-C.....	706
.....	707
Objective-C	707
.....	707
.....	707
.....	708
.....	708
Objective-C.....	708
UIColorUIImage.....	708
.....	708
3	709
Objective-C	709
.....	709
.....	710
UIImagebase64.....	710
UIView.....	710
UIColorUIImage.....	710
UIImage.....	711
176: UIImagePickerController	713
.....	713
Examples.....	713

UIImagePickerController	713
177: UIImageView	715
Examples	715
UIImageView	715
UIImageView	715
UIImageView	716
.....	716
Objective-C	717
.....	717
LabelUIImage	717
Objective-C	718
3	718
.....	718
Mode	718
.....	719
Aspect Fit	720
Aspect	720
.....	720
.....	721
.....	721
.....	721
.....	721
.....	722
.....	722
.....	722
.....	723
.....	723
.....	723
.....	724
178: UILabel	725
.....	725
.....	725

.....	725
Examples.....	725
.....	725
String.....	725
.....	725
.....	726
.....	726
.....	726
UILabel.....	727
.....	727
.....	727
Objective-C.....	727
.....	727
.....	727
Objective-C.....	727
Objective-c +VFL.....	727
Interface Builder.....	728
Interface BuilderView Controller.....	728
.....	729
Objective-C.....	729
.....	729
.....	729
Objective-C.....	729
.....	729
.....	729
3.....	729
Objective-C.....	729
.....	729
.....	730
Swift3.....	730
Objective-C.....	730

.....	730
Swift3.....	730
Objective-C.....	730
.....	730
.....	730
3.....	730
Objective-C.....	730
.....	730
.....	730
Objective-C.....	731
.....	731
.....	731
3.....	731
Objective-C.....	731
Swift	731
.....	731
.....	731
.....	731
Objective-C.....	731
.....	731
.....	731
Objective-C.....	731
.....	731
.....	732
Interface Builder	732
.....	732
.....	734
.....	735
.....	735
.....	735

.....	735
LineBreakMode.....	736
.....	736
.....	736
3	736
Objective-C	736
.....	736
.....	737
.....	737
.....	739
.....	739
Objective-C.....	739
storyboard“userInteractionEnabled”	739
.....	739
Objective-C.....	739
.....	740
.....	740
.....	748
.....	748
.....	749
.....	749
UILabel.....	750
.....	750
179: UITabBarController	752
Examples.....	752
.....	752
.....	752
Objective-C.....	753
.....	753
TabBar.....	753
.....	754

UITabBarController.....	754
.....	755
.....	756
Tab BarStoryboard.....	757
180: UITableViewCell.....	759
.....	759
Examples.....	759
UITableViewCellXib.....	759
181: UITableViewController.....	761
.....	761
Examples.....	761
tableViewCellStyleTableView.....	761
TableView.....	762
182: UITextField.....	764
.....	764
.....	764
Examples.....	764
.....	764
.....	764
Objective-C.....	764
Interface Builder.....	764
.....	765
.....	765
Objective-C.....	765
.....	765
.....	765
Objective-C.....	765
.....	766
.....	766
Objective-C.....	768
.....	768
.....	

Objective-C.....	768
.....	769
UITableView.....	769
.....	771
.....	771
Objective-C.....	771
.....	771
Objective-C.....	771
UIPickerView.....	771
.....	775
.....	775
.....	775
.....	776
.....	776
.....	776
.....	776
.....	777
.....	777
.....	777
Swift 2.3 <.....	777
3.....	777
Objective-C.....	777
.....	778
UITextField.....	778
.....	778
Objective-C.....	778
183: UITextView.....	779
Examples.....	779
.....	779
.....	779
.....	779

UITextViewDelegate	779
.....	779
.....	780
HTMLUITextView	780
.....	780
.....	780
.....	780
.....	781
.....	781
.....	781
.....	781
.....	781
.....	781
.....	781
.....	782
.....	782
.....	782
.....	782
.....	782
184: UIView	784
.....	784
.....	784
Examples	784
UIView	784
.....	784
.....	785
.....	785
.....	786
.....	786
IBInspectableIBDesignable	786
UIView	789
UIView	789
UIViewUIView	790
AutolayoutUIView	791
.....	

.....	795
185: UIViewController	797
Examples	797
.....	797
.....	798
.....	799
.....	799
.....	800
/.....	800
186:	801
Examples	801
.....	801
187:	802
Examples	802
UIScrollView / UITableView	802
.....	803
.....	804
.xib	804
.swift UIView	804
.....	806
.....	808
.....	808
Singleton + Delegate	808
.....	811
iOS	811
.....	811
Objective-C	811
188:	813
Examples	813
AlamofireImage	813

189:	814
.....	.814
.....	814
Examples.....	814
.....	.814
.....	814
.....	.814
Objective-C.....	.814
.....	814
.....	.814
Objective-C.....	.814
.....	814
.....	.814
.....	.815
.....	815
.....	.815
Objective-C.....	.815
.....	815
.....	.815
.....	815
.....	.815
.....	815
.....	.815
.....	.815
.....	815
.....	.815
.....	.815
.....	815
190:	817
.....	.817
Examples.....	817
.....	.817
.....	.817
.....	818

Objective-C	818
.....	818
.....	818
.....	819
191:	820
Examples	820
.....	820
192:	822
.....	822
.....	822
Examples	822
.....	822
.....	822
.....	822
.....	823
.....	823
.....	824
.....	827
UILabel	828
.....	838
UILabel	839
UILabelParentviewUILabelText	842
.....	849
.....	850
.....	851
NSLayoutConstraint	852
193: UITableViewCells	855
.....	855
Examples	855
.....	855
194: UITableViewCells	856

Examples.....	856
.....	856
195: UITextField.....	857
.....	857
Examples.....	857
UITextField.....	857
UITextField.....	857
196:	859
Examples.....	859
.....	859
.....	859
UIKit+ IBExtensions.h.....	859
UIKit+ IBExtensions.m.....	860
Storyboard.....	861
.....	862
deux.....	862
.....	862
.....	862
.....	862
.....	863
197:	864
Examples.....	864
KeyBoard.....	864
198: URL.....	871
.....	871
.....	871
.....	871
Examples.....	872
URLMail.....	872
.....	872
Objective-C.....	872

Apple URL.....	872
199:	875
Examples.....	875
.....	875
UIView.....	875
.....	875
.....	875
200:	877
.....	877
Siri	877
Examples.....	877
Siri ExtensionApp.....	877
.....	877
.....	877
Apple.....	877
.....	878
.....	878
201: UIImage	879
.....	879
Examples.....	879
.....	879
202:	880
Examples.....	880
.....	880
.....	881
.....	881
.....	881
SIGABRTEXC_BAD_INSTRUCTION.....	881
EXC_BAD_ACCESS.....	882
203:	884
.....	884

Examples.....	884
UNNotificationContentExtension.....	884
204: iOS.....	893
Examples.....	893
Mac.....	893
205:	894
.....	894
Examples.....	894
.....	894
.....	894
App-Site-Association.....	894
iOS.....	895
Objective-C.....	898
.....	899
iOS.....	899
206: iOS.....	900
.....	900
Examples.....	900
MVC.....	900
MVP.....	900
MVVM.....	901
VIPER.....	902
207:	905
.....	905
.....	905
Examples.....	905
.....	905
.....	905
.....	905
.....	905
.....	905
.....	905

210:	916
.....	.916
Examples.....	.916
RLMObject - Objective-C.....	.916
.....	917

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [ios](#)

It is an unofficial and free iOS ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official iOS.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: iOS

1-Apple iOS。 Apple ID。 Apple ID。 [App Store](#) Apple99。 TestFlightbeta。

2-Apple ID 。 <https://appleid.apple.com/>

- [iOSSwift](#)
- [Xcode](#)
- [AppStoreXcode](#)

Stack Overflow

- [xcode](#) AppleIDEiOSmacOS
- [swift-language](#)iOS。
- [objective-c-language](#)iOS。
- [cocoa](#)iOSmacOSApple API。
- [sprite-kit](#)2D。
- [core-data](#)。

iPhone OS 2	2008-07-11
iPhone OS 3	2009-06-17
iOS 4	2010-06-08
iOS 5	2011-10-12
iOS 6	2012-09-19
iOS 7	2013918
iOS 8	2014917
iOS 8.1	20141020
iOS 8.2	201539
iOS 8.3	2015-04-09
iOS 8.4	
iOS 9	2015916
iOS 9.1	20151022

iOS 9.2	2015128
iOS 9.3	2016321
iOS 10.0.1	2016913
iOS 10.1	20161024
iOS 10.2	20161212
iOS 10.2.1	2017123
iOS 10.3	2017327
iOS 10.3.3	2017719

Examples

iOSXcode。 XcodeApple。 macOS。 Xcode 8.3.3Xcode 9。

1. MacApp StoreXcode。

App StoreApple DeveloperXcode 。



2. Xcode。



Welcome to Xcode

Version 8.0 (8A218a)



Get started with a playground

Explore new ideas quickly and easily.



Create a new Xcode project

Create an app for iPhone, iPad, Mac, Apple TV, or watchOS.



Check out an existing project

Start working on something from an SCM repository.

- SwiftXcode 6。 ◦ SwiftSwift。
- **Xcode** ◦
- SVN。

3. **Xcode** Xcode

Choose a template for your new project:

ios

watchOS

tvOS

macOS

Cross-platform

Application



Single View
Application



Game



Master
App



Sticker Pack
Application



iMessage
Application

Framework & Library



Cocoa Touch
Framework



Cocoa Touch
Static Library



Metal

Cancel

- 5
 - **iOS**iOS
 - **watchOS**watchOS
 - **tvOS**tvOS
 - **macOS**macOSAppleScripts◦
 -

◦ ;UI◦

iOS ◦

1. ◦ ◦ iPad“”“”UI◦
2. ◦ ◦
3. ◦ ◦
4. ◦ UI◦ iTunes StoreiBooksApp Store◦
5. ◦ SceneKitSpriteKitOpenGL ESMetal◦

4.

Choose options for your new project:

Product Name:

Team:

None

Organization Name:

StackOver

Organization Identifier:

com.stacko

Bundle Identifier:

com.stacko

Language:

Swift

Devices:

Universal

Use Core

Include U

Include U

Cancel

- /
-
- ◦ ◦
- ◦ ◦ iTunes ConnectApp Store ◦ ◦
- ◦ Objective-C/Swift
- ◦ iPhoneiPadUniversal ◦ iPhoneiPad
- .xcdatamodel ◦ ◦
-
- **UIUIUI**

Next ◦

Create Xcode UI ◦ Storyboard ◦

◦

“iPhone 6”RUN ◦



5. ◦ ◦ ◦

◦

◦

Apple' [Jump Right In](#) ' ◦ ◦

XcodeiOS ◦

-
-
- ◦
-

Xcode**Create a new Xcode project** ◦ Xcode**File> New> Project**



Welcome to Xcode

Version ...



Get started with a playground

Explore new ideas quickly and easily.



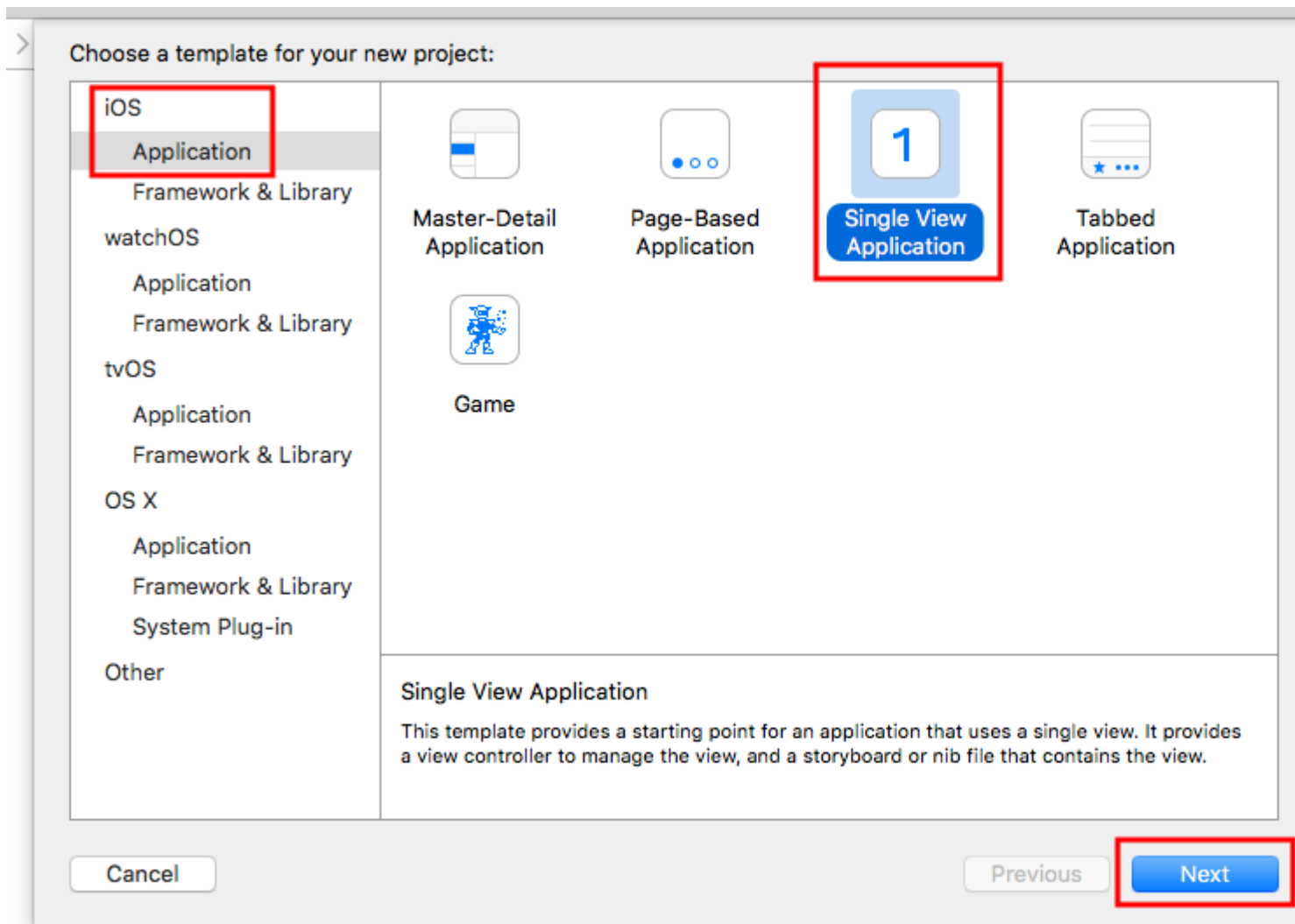
Create a new Xcode project

Start building a new iPhone, iPad or Mac application.



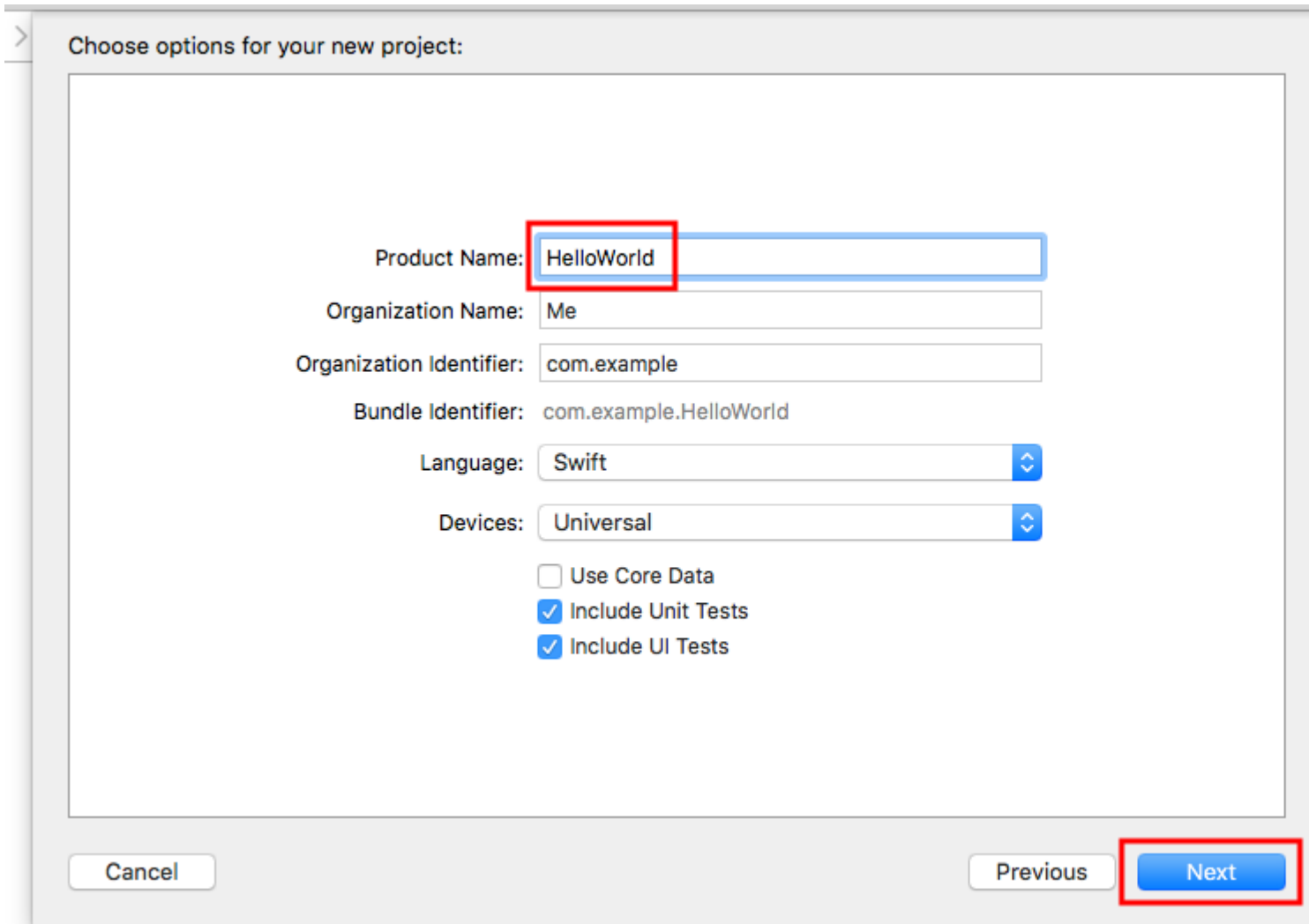
Check out an existing project

Start working on something from an SCM repository.

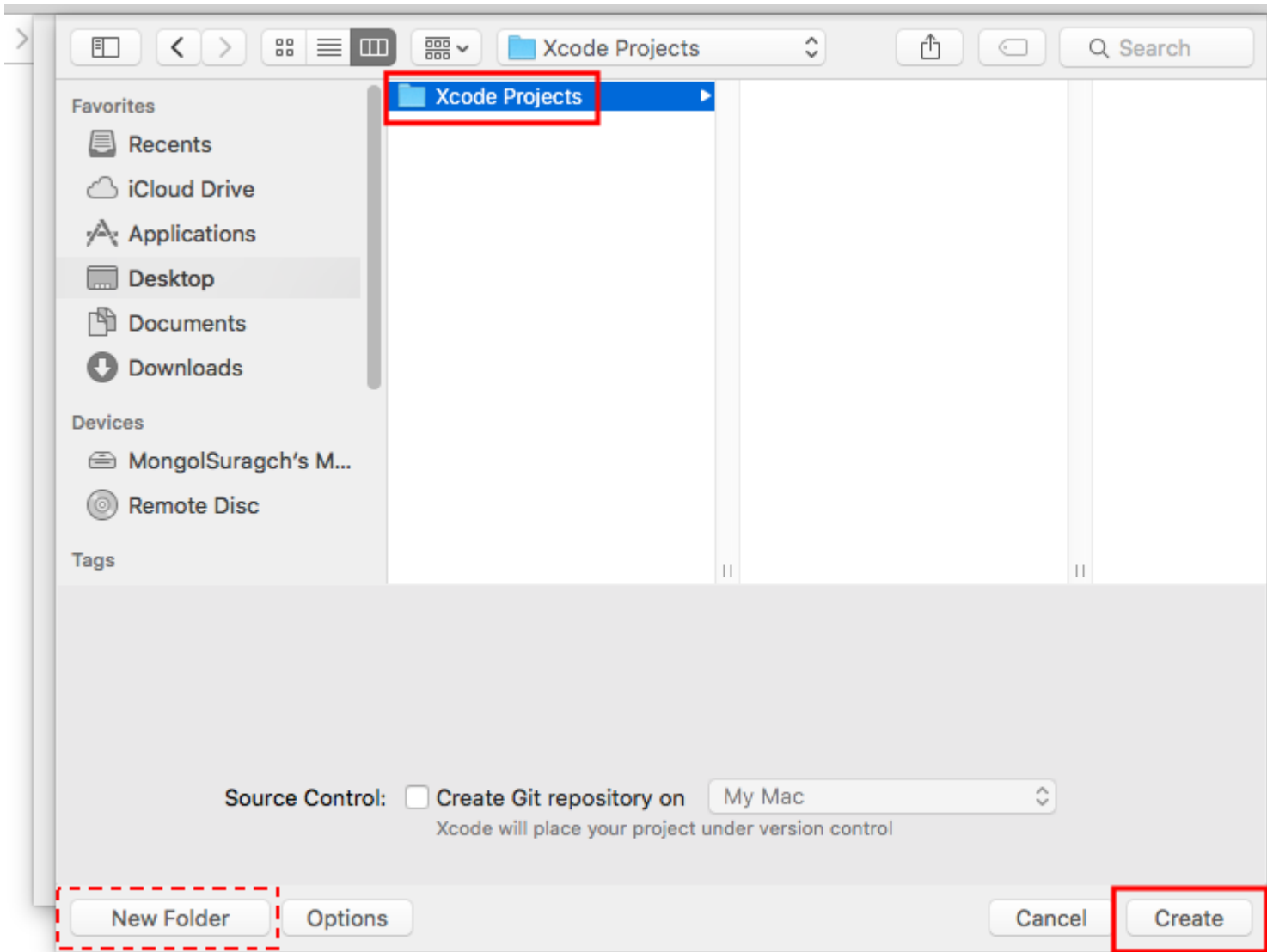


“HelloWorld”**Swift** ◦

- iPhoneiPad◦
- Hello World◦
- **UI** ◦

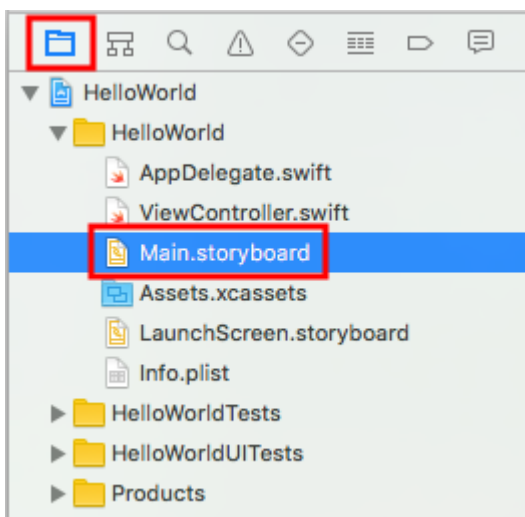


Xcode ◦ “Xcode Projects” ◦ **Create** ◦ Source Control [GitHub](#) ◦

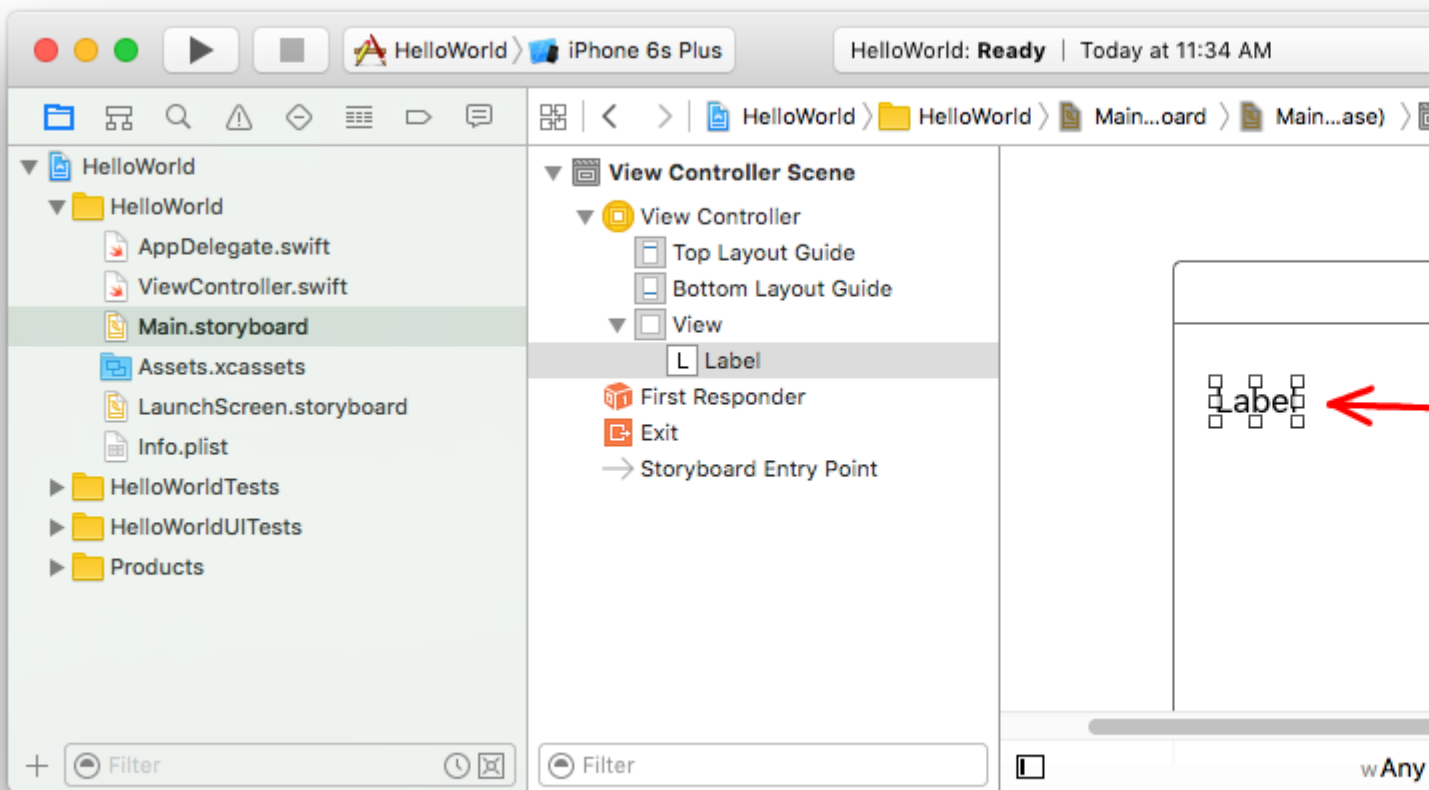


Xcode.

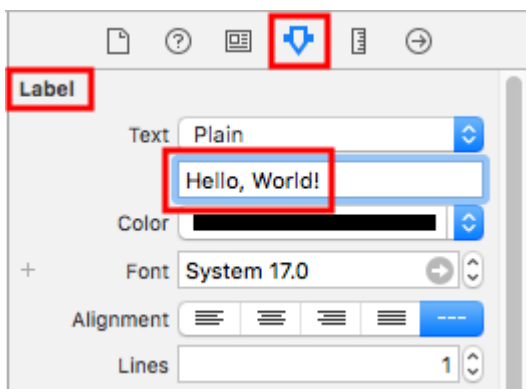
Project Navigator *Main.storyboard* .



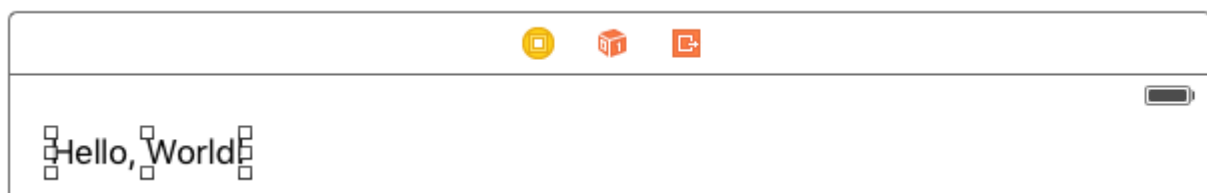
Xcode "label" . UILabel . .



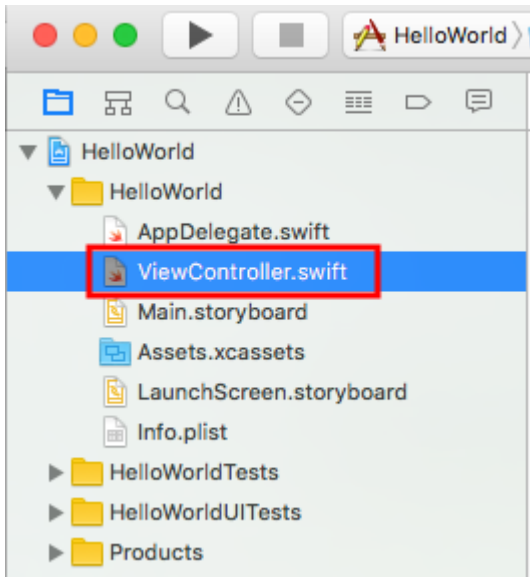
“ ”HelloWorld”。



“HelloWorld”。



Project Navigator *ViewController.swift*。



```
print("Successfully created my first iOS application.")viewDidLoad()◦ ◦
```

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

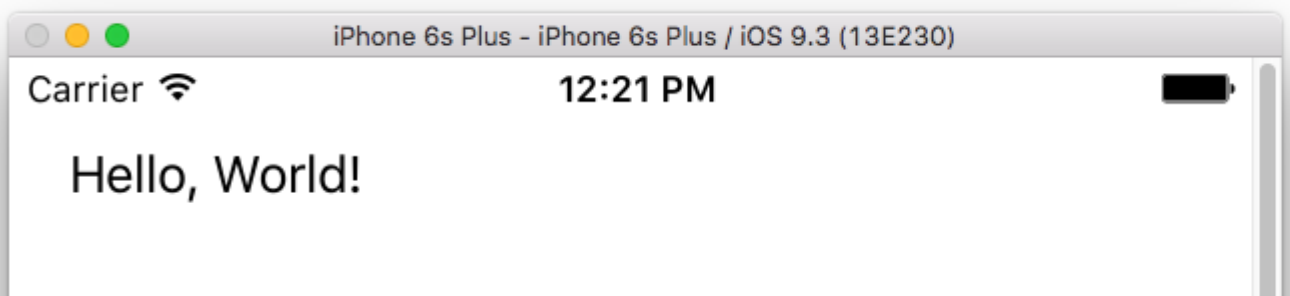
        // print to the console when app is run
        print("Successfully created my first iOS application.")
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```



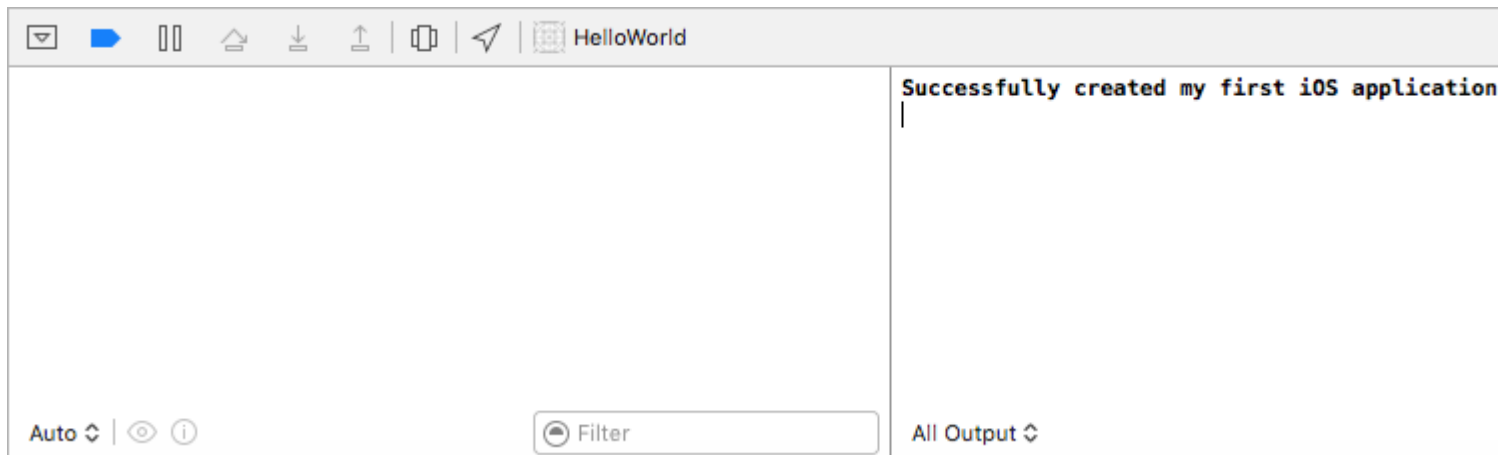
“” iPhone 6s Plus. Xcode. ◦ ◦

◦



“ ”>“ ”cmd+ 1/2/3/4/5100/ 75/ 50/ 33/ 25。

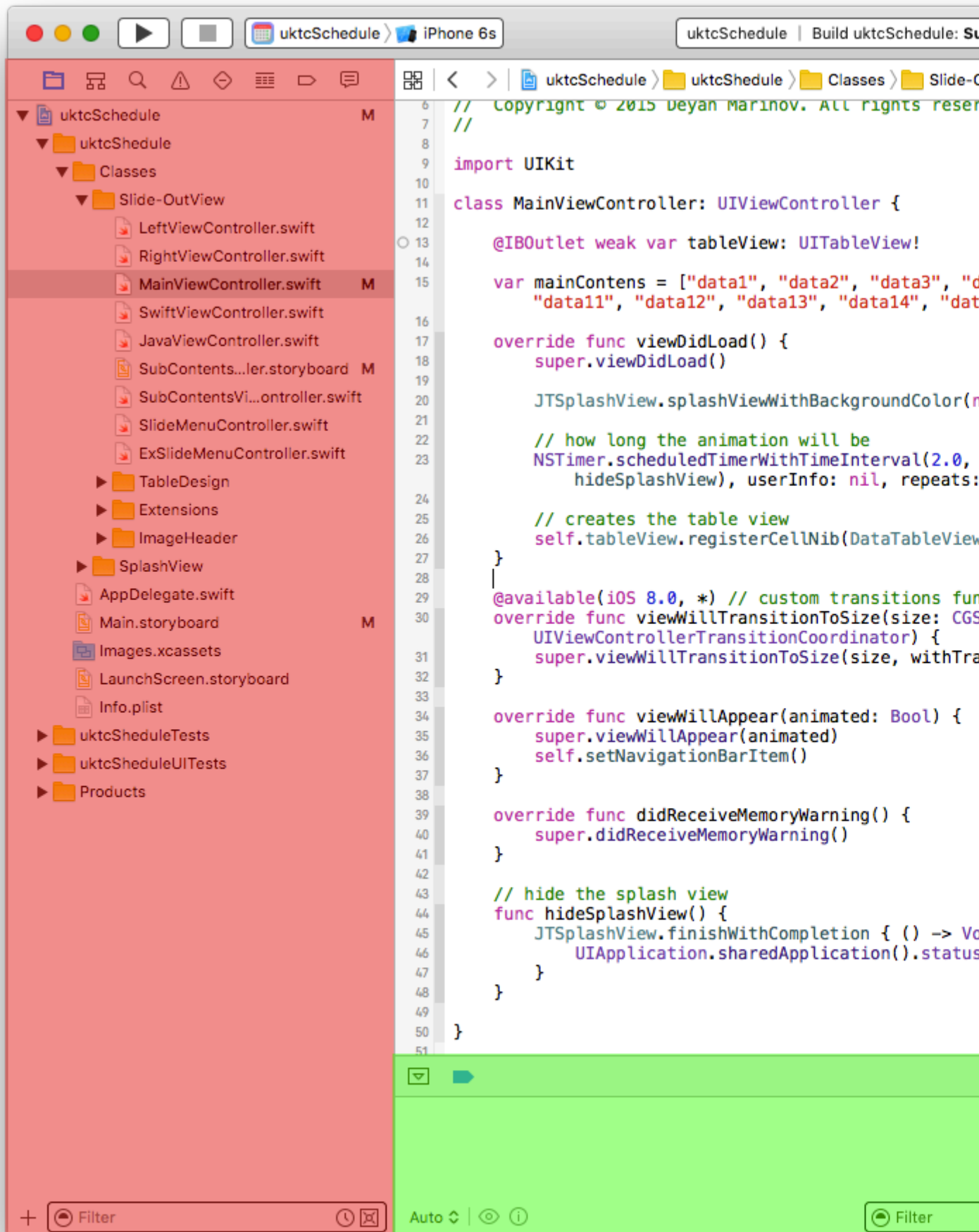
Xcode“iOS”。 。 “iOS。 ” message“ ”。



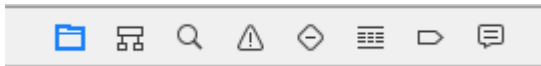
。 。

Xcode

Xcode - 。



- Xcode ◦ MainViewController.swift ◦



- ◦ ◦
- ◦ ◦ ◦
- ◦
- ◦ ◦
- ◦ ◦
- ◦ ◦
- ◦ ◦
- ◦ ◦



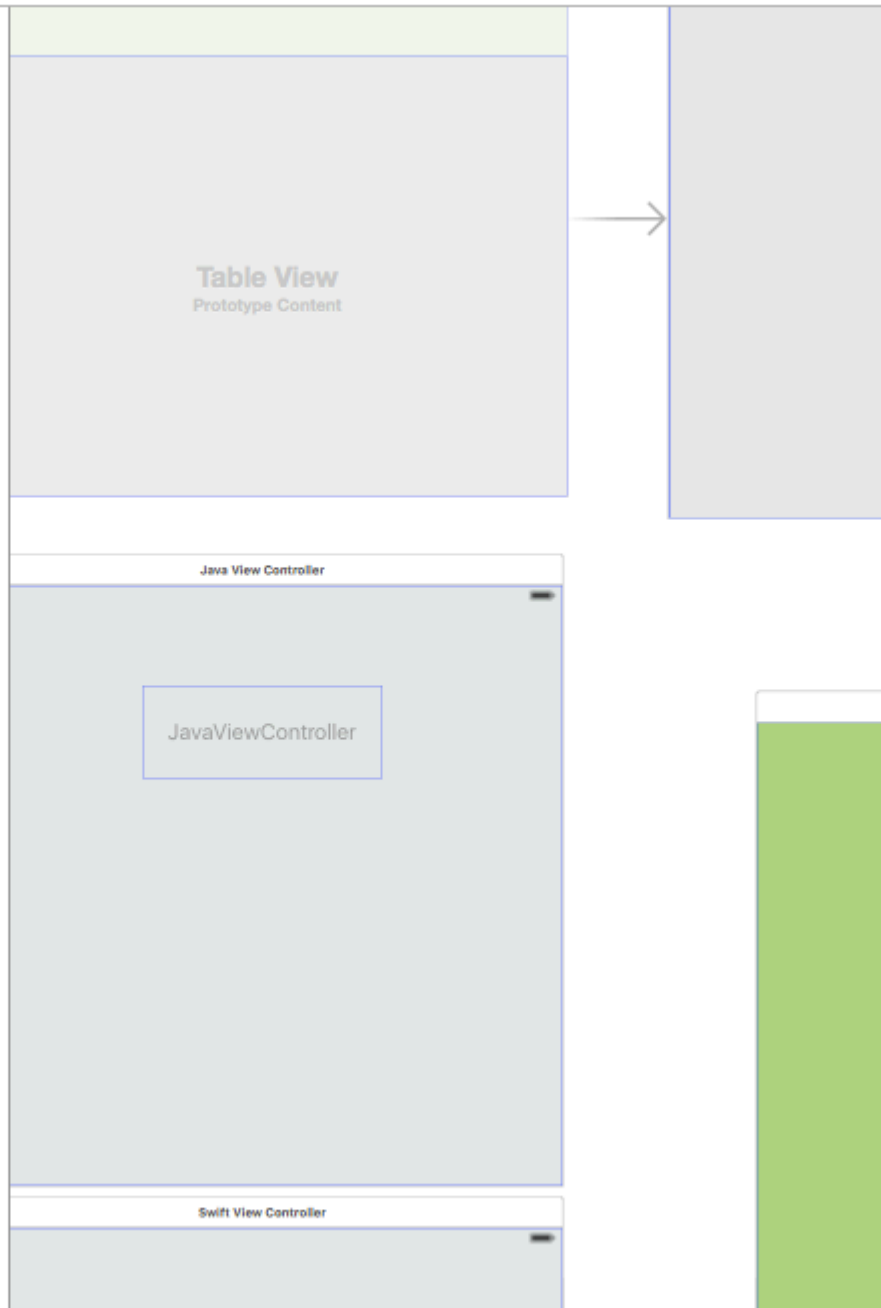
Xcode ◦

- ◦ ◦

```
uktcSchedule > uktcShedule > Classes > Slide-OutView > LeftViewController.swift > No Selection
1 //
2 // LeftViewController.swift
3 // uktcShedule
4 //
5 // Created by Deyan Marinov on 10/9/15.
6 // Copyright © 2015 Deyan Marinov. All rights reserved.
7 //
8
9 import UIKit
10
11 enum LeftMenu: Int {
12     case Main = 0
13     case Swift
14     case Java
15 }
16
17 protocol LeftMenuProtocol : class {
18     func changeViewController(menu: LeftMenu)
19 }
20
21 class LeftViewController : UIViewController, LeftMenuProtocol {
22
23     @IBOutlet weak var tableView: UITableView!
24     var menus = ["Main", "Swift", "Java"]
25     var mainViewController: UIViewController!
26     var swiftViewController: UIViewController!
27     var javaViewController: UIViewController!
28     var goViewController: UIViewController!
29     var nonMenuViewController: UIViewController!
30     var imageHeaderView: ImageHeaderView!
31
32     required init?(coder aDecoder: NSCoder) {
33         super.init(coder: aDecoder)
34     }
35
36     override func viewDidLoad() {
37         super.viewDidLoad()
38         self.tableView.separatorColor = UIColor(red: 224/255, green: 224/255, blue: 224/255,
39
40         let storyboard = UIStoryboard(name: "Main", bundle: nil)
41         let swiftViewController = storyboard.instantiateViewControllerWithIdentifier("SwiftV
42         self.swiftViewController = UINavigationController(rootViewController: swiftViewContr
43
44         let javaViewController = storyboard.instantiateViewControllerWithIdentifier("JavaVie
45         self.javaViewController = UINavigationController(rootViewController: javaViewControl
46
47         self.tableView.registerClass(RecursiveTableViewCell.self)
48     }
49 }
```

- Interface Builder. ◦

- ▼ **Right View Controller Scene**
 - ▶ Right View Controller
 - First Responder
 - Exit
- ▼ **Left View Controller Scene**
 - ▶ Left View Controller
 - First Responder
 - Exit
- ▼ **Main View Controller Scene**
 - ▶ Main View Controller
 - First Responder
 - Exit
 - Storyboard Entry Point
- ▼ **Swift View Controller Scene**
 - ▶ Swift View Controller
 - First Responder
 - Exit
- ▶ **Java View Controller Scene**
- ▶ **Non Menu Controller Scene**



Filter



wAny hAny

- ○ ○

uktcSchedule

uktcSchedule ◀ General Capabilities Resource Tags Info Build Settings Build Phases

▼ Identity

Bundle Identifier

Version

Build

Team

⚠ No matching provisioning profiles found
 No provisioning profiles with a valid signing identity (i.e. certificate and private key pair) matching the bundle identifier "com.deyanm.uktcShedule" were found.

▼ Deployment Info

Deployment Target

Devices

Main Interface

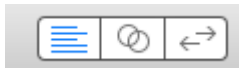
Device Orientation Portrait
 Upside Down
 Landscape Left
 Landscape Right

Status Bar Style

Hide status bar
 Requires full screen

▼ App Icons and Launch Images

App Icons Source



- ○ ○ ○
- ○ ○ ○ ○
- ○ ○ ○ ○



- ○

📄
?
📄
⬇️
📄
➡️

Simulated Metrics

Size ⌵

Orientation ⌵

Status Bar ⌵

Top Bar ⌵

Bottom Bar ⌵

View Controller

Title

Is Initial View Controller

Layout Adjust Scroll View Insets

Hide Bottom Bar on Push

Resize View From NIB

Use Full Screen (Deprecated)

Extend Edges Under Top Bars

Under Bottom Bars

Under Opaque Bars

Transition Style ⌵

Presentation ⌵

Defines Context

Provides Context


Content Size Use Preferred Explicit Size


Width Height


Key Commands

+ -

📄
{ }
🔍
📄

 **View Controller** - A controller that manages a view.

 **Storyboard Reference** - Provides a placeholder for a view controller in an external storyboard.

 **Navigation Controller** - A controller that manages navigation

📄

Inspector ◦

- ◦ ◦ ◦ ◦
- ◦ ◦ "◦"
- ◦ ◦
-

- Apple◦

- ◦ ◦

- ◦ ◦

- ;◦

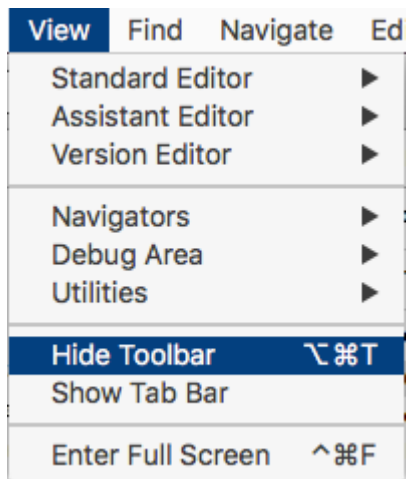
“ ” ◦ “button”“”◦

- “ ” ◦ “ ” ◦ **Scheme** ◦ ◦

-



“ ” ◦



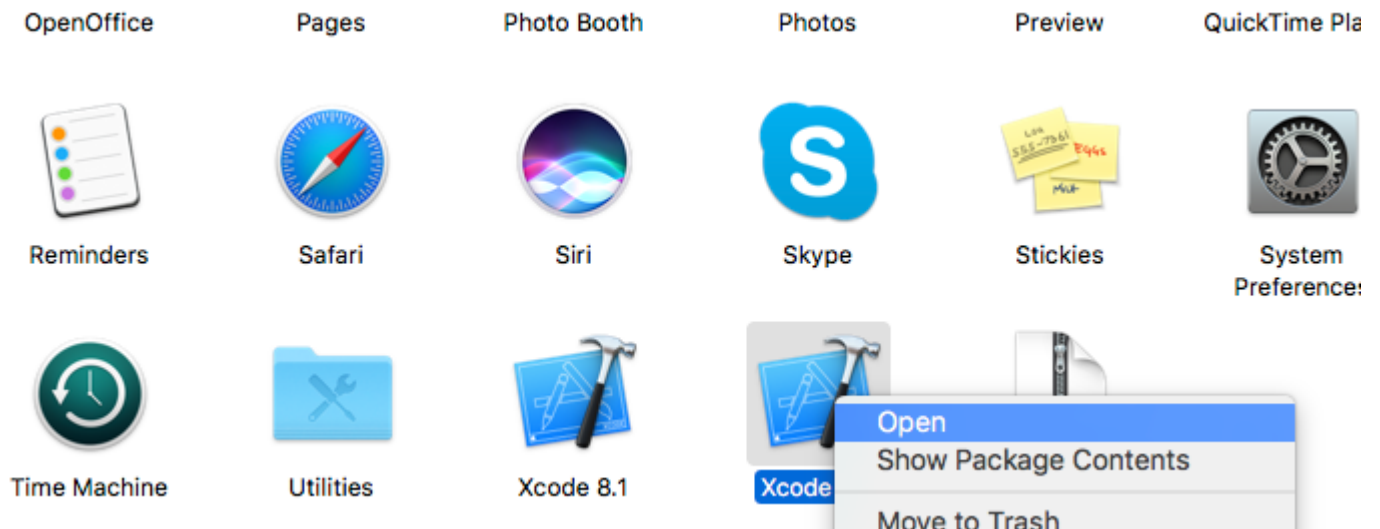
Swift 3

Swift 3◦ ◦

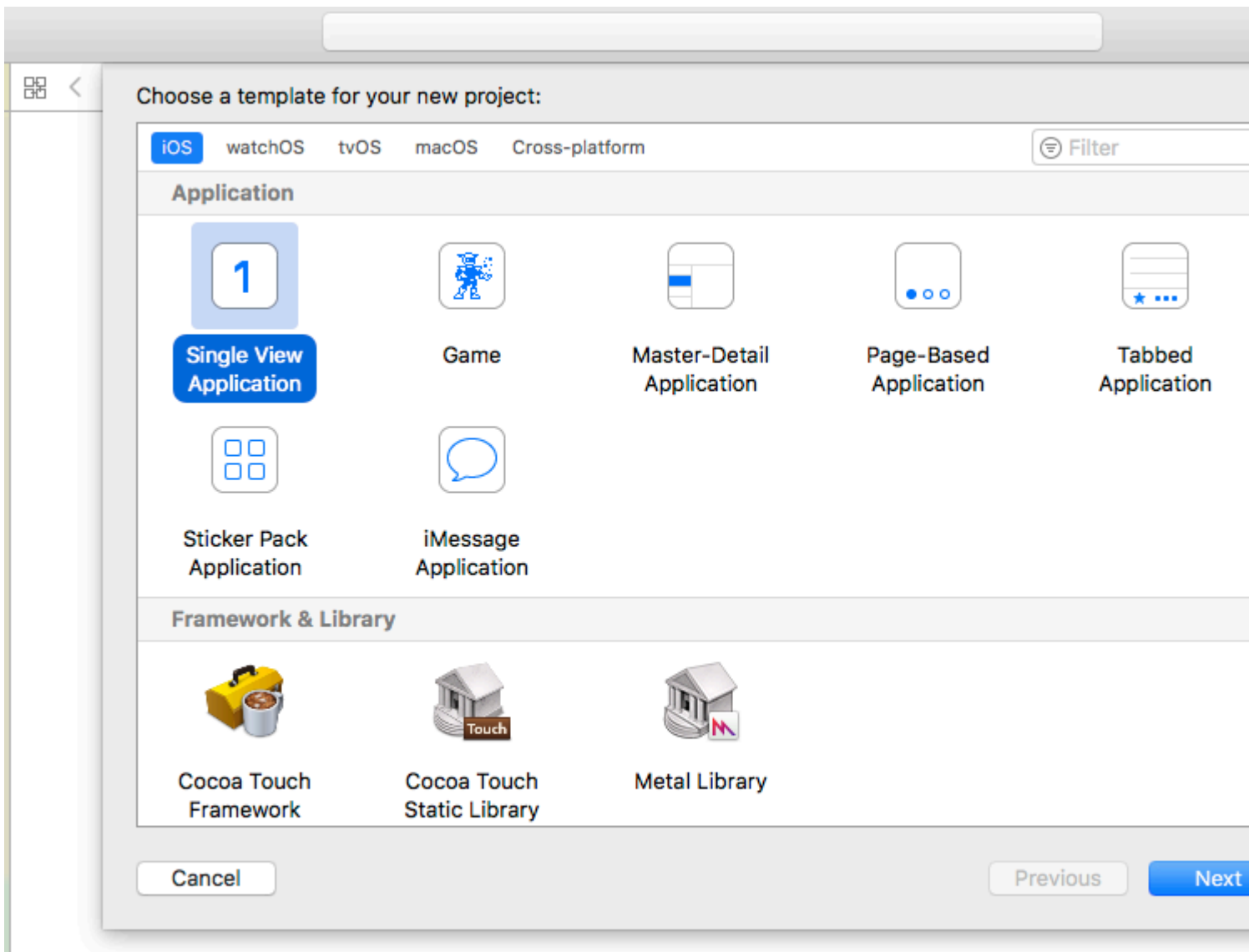
1. MAC OS - Xcode 8.210.11.6
2. Xcode - 8.2 [XcodeApple](#)◦

Xcode 8.2Swift 3iOS 10API◦

ApplicationXcode 8.2.



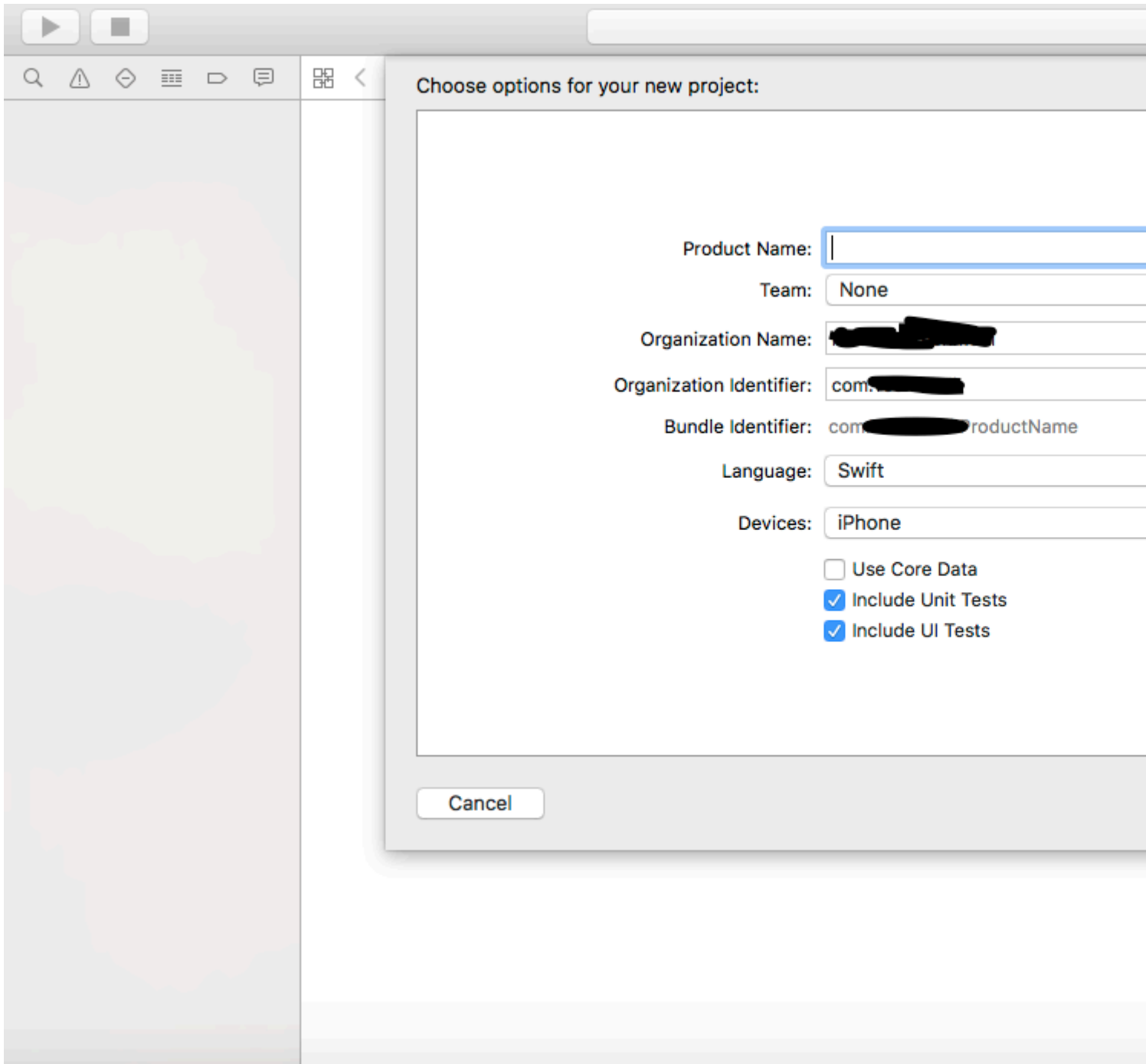
Create new Project



Xcode

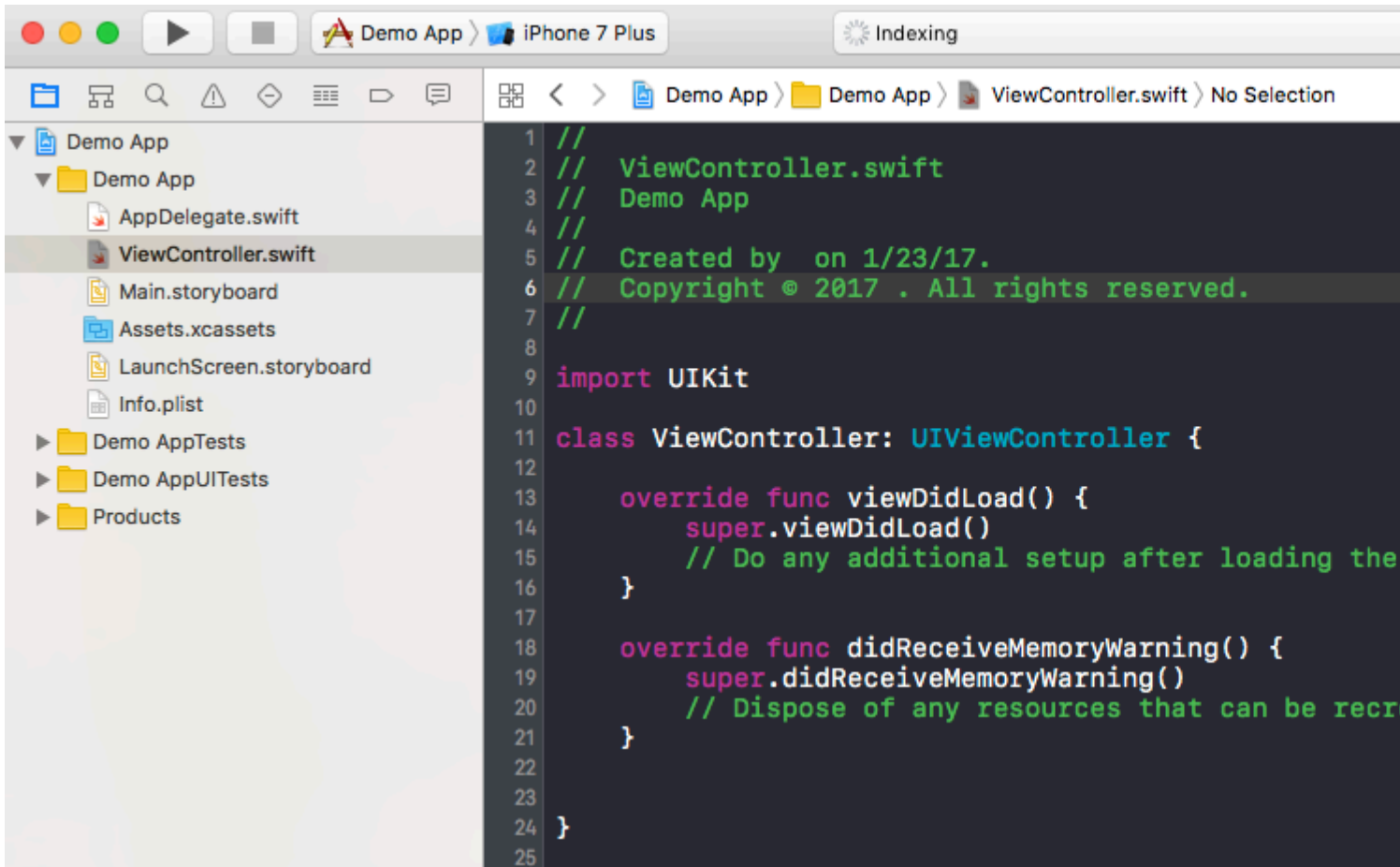
1. iOS
2. watchOS
- 3.
- 4.

iOS



Bundle.

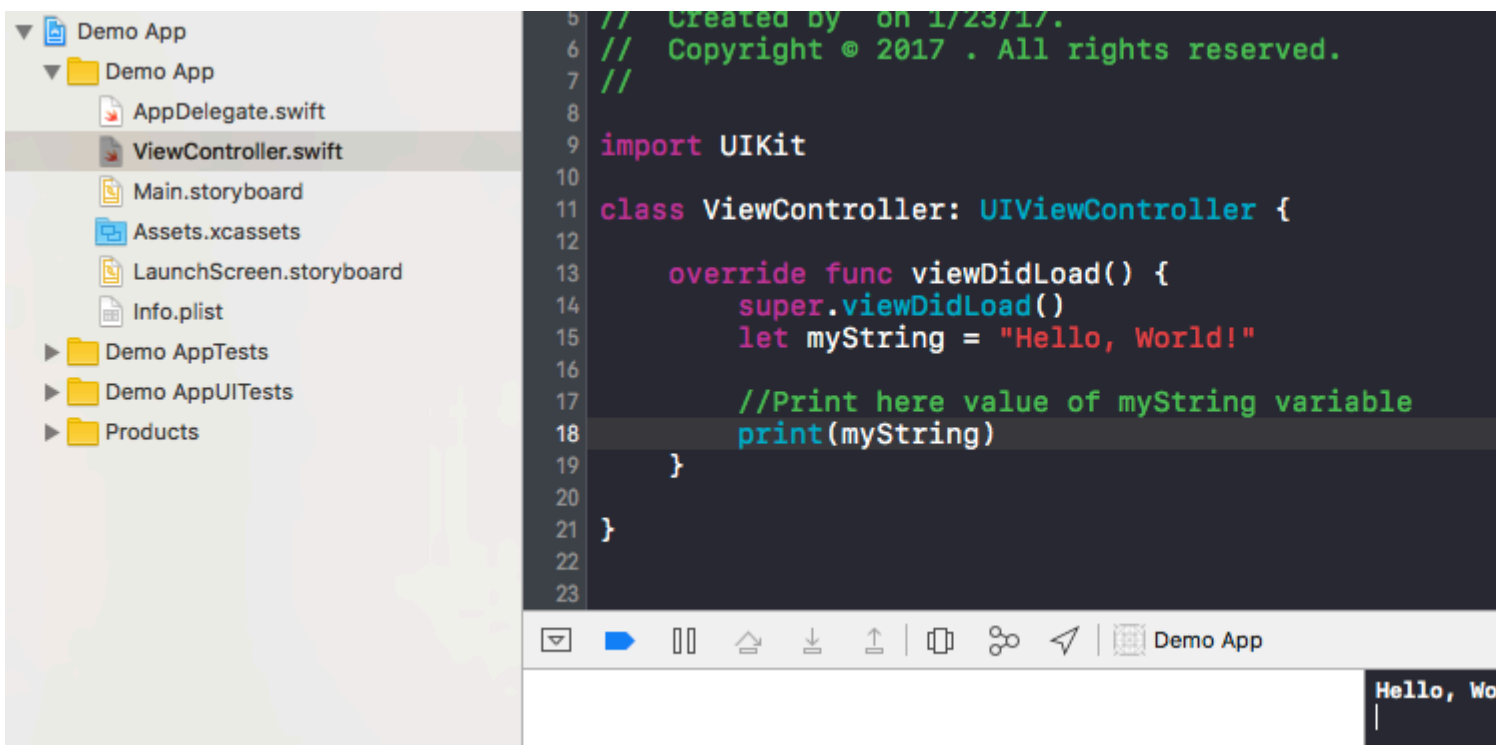
“”



ViewController.swiftViewControllerUIViewController'String'myString' . 'super.viewDidLoad'

```
let myString = "Hello, World!"
```

。 “” 。



- XcodeHello World◦

iOS <https://riptutorial.com/zh-TW/ios/topic/191/ios>

2: 3D

Examples

3DSwift

iPhone 6s Plus 3D touch PeekPop

-
-

3D

3D. *UITraitCollectionforceTouchCapability*. *UITraitCollectioniOS*.

```
if (traitCollection.forceTouchCapability == .Available) {
    registerForPreviewingWithDelegate(self, sourceView: view)
}
```

UIViewControllerPreviewingDelegate.

previewingContext.

```
func previewingContext(previewingContext: UIViewControllerPreviewing,
viewControllerForLocation location: CGPoint) -> UIViewController? {

    guard let indexPath = self.tableView.indexPathForRowAtPoint(location), cell =
self.tableView.cellForRowAtIndexPath(indexPath) as? <YourTableViewCell> else {
        return nil
    }

    guard let detailVC =
storyboard?.instantiateViewControllerWithIdentifier("<YourViewControllerIdentifier>") as?
<YourViewController> else {
        return nil
    }

    detailVC.peekActive = true
    previewingContext.sourceRect = cell.frame

    // Do the stuff

    return detailVC
}
```

poppreviewingContext :))

```
func previewingContext(previewingContext: UIViewControllerPreviewing, commitViewController
viewControllerToCommit: UIViewController) {
```

```

let balanceViewController = viewControllerToCommit as! <YourViewController>

// Do the stuff

navigationController?.pushViewController(balanceViewController, animated: true)
}

```

- 3D touch◦

Objective-C

```

//Checking for 3-D Touch availability
if ([self.traitCollection respondsToSelector:@selector(forceTouchCapability)] &&
    (self.traitCollection.forceTouchCapability == UIForceTouchCapabilityAvailable))
{
    [self registerForPreviewingWithDelegate:self sourceView:self.view];
}
//Peek
- (UIViewController *)previewingContext:(id<UIViewControllerPreviewing>)previewingContext
    viewControllerForLocation:(CGPoint)location {

    NSIndexPath *indexPath = [self.tableView indexPathForRowAtPoint:location];
    Country *country = [self countryForIndexPath:indexPath];
    if (country) {
        CountryCell *cell = [self.tableView cellForRowAtIndexPath:indexPath];
        if (cell) {
            previewingContext.sourceRect = cell.frame;
            UINavigationController *navController = [self.storyboard
instantiateViewControllerWithIdentifier:@"UYLCountryNavController"];
            [self configureNavigationController:navController withCountry:country];
            return navController;
        }
    }
    return nil;
}
//Pop
- (void)previewingContext:(id<UIViewControllerPreviewing>)previewingContext
    commitViewController:(UIViewController *)viewControllerToCommit {

    [self showDetailViewController:viewControllerToCommit sender:self];
}

```

3 DObjective-C

Objective-C

```

//Checking for 3-D Touch availability
if ([self.traitCollection respondsToSelector:@selector(forceTouchCapability)] &&
    (self.traitCollection.forceTouchCapability == UIForceTouchCapabilityAvailable))
{
    [self registerForPreviewingWithDelegate:self sourceView:self.view];
}
//Peek
- (UIViewController *)previewingContext:(id<UIViewControllerPreviewing>)previewingContext
    viewControllerForLocation:(CGPoint)location {

```

```
NSIndexPath *indexPath = [self.tableView indexPathForRowAtPoint:location];
Country *country = [self countryForIndexPath:indexPath];
if (country) {
    CountryCell *cell = [self.tableView cellForRowAtIndexPath:indexPath];
    if (cell) {
        previewingContext.sourceRect = cell.frame;
        UINavigationController *navController = [self.storyboard
instantiateViewControllerWithIdentifier:@"UYLCountryNavController"];
        [self configureNavigationController:navController withCountry:country];
        return navigationController;
    }
}
return nil;
}
//Pop
- (void)previewingContext:(id<UIViewControllerAnimatedPreviewing>)previewingContext
commitViewController:(UIViewController *)viewControllerToCommit {

    [self showDetailViewController:viewControllerToCommit sender:self];
}
```

3D <https://riptutorial.com/zh-TW/ios/topic/6705/3d>

3: AFNetworking

Examples

AFNetworkingAFNetworking。。

AFNetworking 2.xx

```
// Create dispatch_queue_t with your name and DISPATCH_QUEUE_SERIAL as for the flag
dispatch_queue_t myQueue = dispatch_queue_create("com.CompanyName.AppName.methodTest",
        DISPATCH_QUEUE_SERIAL);

// init AFHTTPRequestOperation of AFNetworking
operation = [[AFHTTPRequestOperation alloc] initWithRequest:request];

// Set the FMDB property to run off the main thread
[operation setCompletionQueue:myQueue];
```

AFNetworking 3.xx

```
AFHTTPSessionManager *manager = [[AFHTTPSessionManager alloc] init];
[self setCompletionQueue:myQueue];
```

AFNetworking <https://riptutorial.com/zh-TW/ios/topic/3002/afnetworking>

4: Alamofire

-
- responseData
- responseStringNSStringEncoding
- responseJSONNSJSONReadingOptions
- responsePropertyListoptionsNSPropertyListReadOptions

	.OPTIONS, .GET, .HEAD, .POST, .PUT, .PATCH, .DELETE, .TRACE, .CONNECT
URLString	URLStringConvertible
	[String: AnyObject]?
	ParameterEncoding
	[String: String]?

Examples

```
import Alamofire

Alamofire.request(.GET, "https://httpbin.org/get")
```

```
Alamofire.request("https://httpbin.org/get").validate().responseJSON { response in
switch response.result {
case .success:
    print("Validation Successful")
case .failure(let error):
    print(error)
}
}
```

```
Alamofire.request(.GET, "https://httpbin.org/get", parameters: ["foo": "bar"])
    .responseJSON { response in
        print(response.request) // original URL request
        print(response.response) // URL response
        print(response.data) // server data
        print(response.result) // result of response serialization

        if let JSON = response.result.value {
            print("JSON: \(JSON)")
        }
    }
```

```
Alamofire.request(.GET, "https://httpbin.org/get", parameters: ["foo": "bar"])
    .validate(statusCode: 200..<300)
    .validate(contentType: ["application/json"])
    .response { response in
        print(response)
```

```
}
```

```
Alamofire.request(.GET, "https://httpbin.org/get", parameters: ["foo": "bar"])  
    .validate()  
    .response { request, response, data, error in  
        print(request)  
        print(response)  
        print(data)  
        print(error)  
    }
```

```
Alamofire.request(.GET, "https://httpbin.org/get")  
    .validate()  
    .responseString { response in  
        print("Response String: \(response.result.value)")  
    }  
    .responseJSON { response in  
        print("Response JSON: \(response.result.value)")  
    }
```

Alamofire <https://riptutorial.com/zh-TW/ios/topic/1823/alamofire>

5: AppDelegate

AppDelegateUIApplication.

analyticsMixpanel / GoogleAnalytics / CrashliticsURL.

Examples

AppDelegate

◦

AppDidFinishLaunching

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {  
    // Write your code before app launch  
    return YES;  
}
```

App

```
- (void)applicationWillEnterForeground:(UIApplication *)application {  
    // Called as part of the transition from the background to the active state; here you can  
    undo many of the changes made on entering the background.  
}
```

App

```
- (void)applicationDidBecomeActive:(UIApplication *)application {  
    // Restart any tasks that were paused (or not yet started) while the application was  
    inactive. If the application was previously in the background, optionally refresh the user  
    interface.  
}
```

App

```
- (void)applicationDidEnterBackground:(UIApplication *)application {  
    // Use this method to release shared resources, save user data, invalidate timers, and  
    store enough application state information to restore your application to its current state in  
    case it is terminated later.  
    // If your application supports background execution, this method is called instead of  
    applicationWillTerminate: when the user quits.  
}
```

app

```
- (void)applicationWillResignActive:(UIApplication *)application {  
    // Sent when the application is about to move from active to inactive state. This can
```

occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.

```
// Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should use this method to pause the game.
}
```

```
- (void)applicationWillTerminate:(UIApplication *)application {
    // Called when the application is about to terminate. Save data if appropriate. See also
    applicationDidEnterBackground:.
}
```

AppDelegate

- **AppDelegate** startup code ◦
- statekey changes ◦ ◦
- responds to notificationsresponds to notifications ◦
- determinesstate preservationrestoration ◦
- responds to eventsresponds to events ◦ ◦

URL

URL◦

```
func application(_ app: UIApplication, open url: URL, options: [UIApplicationOpenURLOptionsKey : Any] = [:]) -> Bool {
    return SomeManager.shared.handle(
        url,
        sourceApplication: options[.sourceApplication] as? String,
        annotation: options[.annotation]
    )
}
```

```
/* Instance of your custom APNs/local notification manager */
private var pushManager: AppleNotificationManager!
```

```
func application(application: UIApplication, didRegisterUserNotificationSettings
notificationSettings: UIUserNotificationSettings) {
    // Called to tell the delegate the types of notifications that can be used to get the
    user's attention
    pushManager.didRegisterSettings(notificationSettings)
}
```

```
func application(application: UIApplication, didRegisterForRemoteNotificationsWithDeviceToken
deviceToken: NSData) {
    // Tells the delegate that the app successfully registered with Apple Push Notification
    service (APNs)
    pushManager.didRegisterDeviceToken(deviceToken)
}
```

```
func application(application: UIApplication, didFailToRegisterForRemoteNotificationsWithError
error: NSError) {
    // Sent to the delegate when Apple Push Notification service cannot successfully complete
```

```
the registration process.  
    pushManager.didFailToRegisterDeviceToken(error)  
}
```

```
func application(application: UIApplication, didReceiveRemoteNotification userInfo: [NSObject  
: AnyObject]) {  
    // Remote notification arrived, there is data to be fetched  
    // Handling it  
    pushManager.handleNotification(userInfo,  
                                   background: application.applicationState == .Background  
    )  
}
```

```
func application(application: UIApplication, didReceiveLocalNotification notification:  
UILocalNotification) {  
    pushManager.handleLocalNotification(notification, background: false)  
}
```

```
func application(application: UIApplication, handleActionWithIdentifier identifier: String?,  
forRemoteNotification userInfo: [NSObject : AnyObject],  
                    completionHandler: () -> Void) {  
    pushManager.handleInteractiveRemoteNotification(userInfo, actionIdentifier: identifier,  
completion: completionHandler)  
}
```

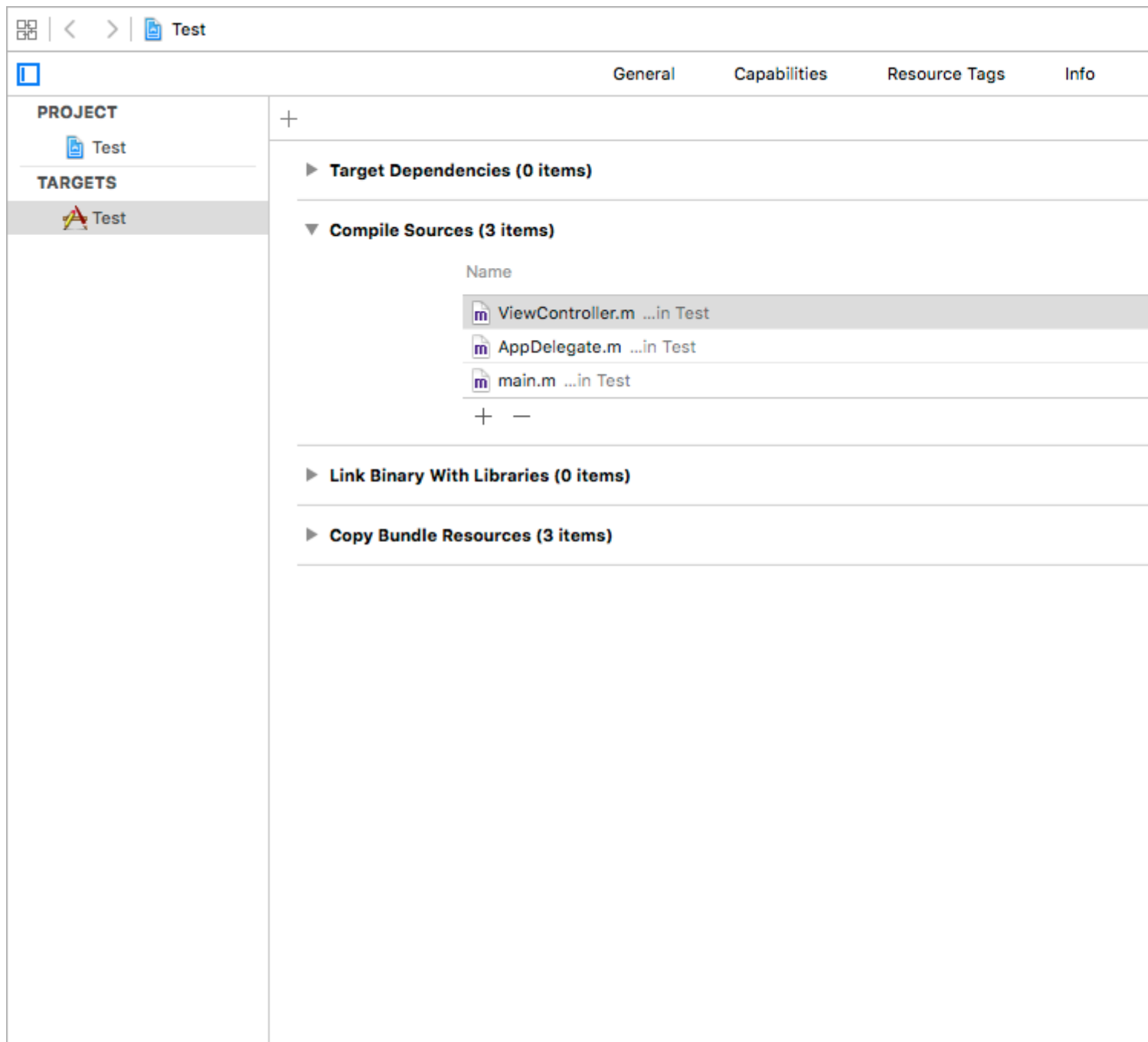
AppDelegate <https://riptutorial.com/zh-TW/ios/topic/8740/appdelegate>

6: ARC

Examples

/ARC

`-fno-objc-arc` ARC ◦ ▶▶



ARC <https://riptutorial.com/zh-TW/ios/topic/4150/arc-->

7: Autolayout/

◦ ◦

◦ “” - “” ◦ ◦

Examples

Interface Builder ◦ ◦

◦ ◦

◦

◦ ◦

intrinsic content size ◦ ◦

UIImageView ◦ ◦

<https://www.raywenderlich.com/115444/auto-layout-tutorial-in-ios-9-part-2-constraints>

[Autolayout/ https://riptutorial.com/zh-TW/ios/topic/6899/autolayout-](https://riptutorial.com/zh-TW/ios/topic/6899/autolayout-)

8: AVPlayerAVPlayerViewController

AVKitAVFoundation.

Examples

AVPlayerViewController

Objective-C

```
NSURL *url = [[NSURL alloc] initWithString:@"YOUR URL"]; // url can be remote or local

AVPlayer *player = [AVPlayer playerWithURL:url];
// create a player view controller

AVPlayerViewController *controller = [[AVPlayerViewController alloc] init];
[self presentViewController:controller animated:YES completion:nil];
controller.player = player;
[player play];
```

```
let player = AVPlayer(URL: url) // url can be remote or local

let playerViewController = AVPlayerViewController()
// creating a player view controller
playerViewController.player = player
self.presentViewController(playerViewController, animated: true) {

    playerViewController.player!.play()
}
```

AVPlayerAVPlayerLayer

C.

```
NSURL *url = [NSURL URLWithString:@"YOUR URL"];
AVPlayer *player = [AVPlayer playerWithURL:videoURL];
AVPlayerLayer *playerLayer = [AVPlayerLayer playerLayerWithPlayer:player];
playerLayer.frame = self.view.bounds;
[self.view.layer addSublayer:playerLayer];
[player play];
```

```
let url = NSURL(string: "YOUR URL")
let player = AVPlayer(URL: videoURL!)
let playerLayer = AVPlayerLayer(player: player)
playerLayer.frame = self.view.bounds
self.view.layer.addSublayer(playerLayer)
player.play()
```

AVPlayer

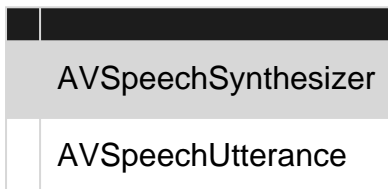
```
AVPlayer * avPlayer = [AVPlayer playerWithURL:[NSURL URLWithString:@""]];
```

```
AVPlayerViewController *avPlayerCtrl = [[AVPlayerViewController alloc] init];  
avPlayerCtrl.view.frame = self.view.frame;  
avPlayerCtrl.player = avPlayer;  
avPlayerCtrl.delegate = self;  
[avPlayer play];  
[self presentViewController:avPlayerCtrl animated:YES completion:nil
```

[AVPlayerAVPlayerViewController](https://riptutorial.com/zh-TW/ios/topic/5092/avplayer-avplayercontroller) <https://riptutorial.com/zh-TW/ios/topic/5092/avplayer-avplayercontroller>

9: AVSpeechSynthesizer

- AVSpeechSynthesizer//
- speaker.speakUtterancespeech//



Examples

speaker:AVSpeechSynthesizer° AVSpeechUtterance°

C.

```
AVSpeechSynthesizer *speaker = [[AVSpeechSynthesizer alloc] init];
AVSpeechUtterance *speech     = [AVSpeechUtterance speechUtteranceWithString:@"Hello World"];
[speaker speakUtterance:speech];
```

```
let speaker = AVSpeechSynthesizer()
let speech = AVSpeechUtterance(string: "Hello World")
speaker.speakUtterance(speech)
```

[AVSpeechSynthesizer](https://riptutorial.com/zh-TW/ios/topic/1526/avspeechsynthesizer) <https://riptutorial.com/zh-TW/ios/topic/1526/avspeechsynthesizer>

10: AWSSDK

Examples

AWS SDKS3

Singleton◦

S3

```
enum S3Configuration : String
{
    case IDENTITY_POOL_ID      = "YourIdentityPoolId"
    case BUCKET_NAME           = "YourBucketName"
    case CALLBACK_KEY          = "YourCustomStringForCallBackWhenUploadingInTheBackground"
    case CONTENT_TYPE_IMAGE    = "image/png"
    case CONTENT_TYPE_VIDEO    = "video/mp4"
}
```

didFinishLaunchingWithOptionsAppDelegateregionType

```
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[NSObject: AnyObject]?) -> Bool
{
    let credentialProvider = AWSCognitoCredentialsProvider(regionType: .EUWest1, identityPoolId:
S3Configuration.IDENTITY_POOL_ID.rawValue)
    let configuration = AWSServiceConfiguration(region: .EUWest1, credentialsProvider:
credentialProvider)
    AWSS3TransferUtility.registerS3TransferUtilityWithConfiguration(configuration, forKey:
S3Configuration.CALLBACK_KEY.rawValue)
}
```

AppDelegateAWS SDK

```
func application(application: UIApplication, handleEventsForBackgroundURLSession identifier:
String, completionHandler: () -> Void)
{
    // Will print the identifier you have set at the enum: .CALLBACK_KEY
    print("Identifier: " + identifier)
    // Stores the completion handler.
    AWSS3TransferUtility.interceptApplication(application,
                                              handleEventsForBackgroundURLSession: identifier,
                                              completionHandler: completionHandler)
}
```

◦

AWS SDKSDK◦ UIImage..

```
// Some image....
let image = UIImage()
let fileURL = NSURL(fileURLWithPath:
```

```

NSTemporaryDirectory()).URLByAppendingPathComponent(fileName)
let filePath = fileURL.path!
let imageData = UIImageJPEGRepresentation(image, 1.0)
imageData!.writeToFile(filePath, atomically: true)

```

FileURLfileName。

AWS SDK

1. AWSS3TransferUtilityUploadCompletionHandlerBlock -
AWSS3TransferUtilityUploadCompletionHandlerBlock
2. AWSS3TransferUtilityUploadProgressBlock -

Singleton。

```

var completionHandler : AWSS3TransferUtilityUploadCompletionHandlerBlock? =
    { (task, error) -> Void in

        if ((error) != nil)
        {
            print("Upload failed")
        }
        else
        {
            print("File uploaded successfully")
        }
    }

var progressBlock : AWSS3TransferUtilityUploadProgressBlock? =
    { [unowned self] (task, bytesSent: Int64, totalBytesSent: Int64,
totalBytesExpectedToSend: Int64) -> Void in

        let progressInPercentage = Float(Double(totalBytesSent) /
Double(totalBytesExpectedToSend)) * 100
        print(progressInPercentage)
    }

```

Singleton。 Singleton

```

static func uploadImageToS3(fileURL : NSURL,
                           fileName : String,
                           progressFunctionUpdater : Float -> Void,
                           resultBlock : (NSError?) -> Void)
{
    // Actual implementation .....
    // ...
    // ...
}

```

1. progressFunctionUpdater - 。
2. resultBlock - nil

```

let fileData = NSData(contentsOfFile: fileURL.relativePath!)

let expression = AWSS3TransferUtilityUploadExpression()

```

```

        expression.uploadProgress = progressBlock

        let transferUtility =
            AWSS3TransferUtility.S3TransferUtilityForKey(S3Configuration.CALLBACK_KEY.rawValue)

        transferUtility?.uploadData(fileData!,
            bucket: S3Configuration.BUCKET_NAME.rawValue,
            key: fileName,
            contentType: S3Configuration.CONTENT_TYPE_IMAGE.rawValue,
            expression: expression,
            completionHandler: completionHandler).continueWithBlock
            { (task : AWSTask) -> AnyObject? in

                if let error = task.error
                {
                    print(error)
                }
                if let exception = task.exception
                {
                    print("Exception: " + exception.description)
                }
                if let uploadTask = task.result as? AWSS3TransferUtilityUploadTask
                {
                    print("Upload started...")
                }

                return nil
            }
    }
}

```

S3:)

AWSSDK <https://riptutorial.com/zh-TW/ios/topic/4734/awssdk>

11: CAAnimation

CAAnimation ◦ **CAMediaTimingCAAction** ◦ Core AnimationScene Kit**CABasicAnimation**
CAKeyframeAnimation **CAAnimationGroup****CATransition** ◦

Examples

◦

Objective-C

```
CABasicAnimation *animation = [CABasicAnimation animationWithKeyPath:@"position.x"];
animation.fromValue = @0;
animation.toValue = @320;
animation.duration = 1;

[_label.layer addAnimation:animation forKey:@"basic"];
```

```
let animation = CABasicAnimation(keyPath: "position.x")
animation.fromValue = NSNumber(value: 0.0)
animation.toValue = NSNumber(value: 320.0)

_label.layer.addAnimation(animation, forKey: "basic")
```

0320◦ keypath

```
"position.y"
```

-

Objective-C

```
CATransition* transition = [CATransition animation];
transition.startProgress = 0;
transition.endProgress = 1.0;
transition.type = @"flip";
transition.subtype = @"fromLeft";
transition.duration = 0.8;
transition.repeatCount = 5;
[_label.layer addAnimation:transition forKey:@"transition"];
```

```
var transition = CATransition()
transition.startProgress = 0
transition.endProgress = 1.0
```

```

transition.type = "flip"
transition.subtype = "fromLeft"
transition.duration = 0.8
transition.repeatCount = 5
label.layer.addAnimation(transition, forKey: "transition")

```

```

CGRect boundingRect = CGRectMake(-150, -150, 300, 300);

CAKeyframeAnimation *orbit = [CAKeyframeAnimation animation];
orbit.keyPath = @"position";
orbit.path = CFAutorelease(CGPathCreateWithEllipseInRect(boundingRect, NULL));
orbit.duration = 4;
orbit.additive = YES;
orbit.repeatCount = HUGE_VALF;
orbit.calculationMode = kCAAnimationPaced;
orbit.rotationMode = kCAAnimationRotateAuto;

[_label.layer addAnimation:orbit forKey:@"orbit"];

```

Objective-C

```

CAKeyframeAnimation *animation = [CAKeyframeAnimation animationWithKeyPath:@"position.x"];
animation.values = @[ @0, @10, @-10, @10, @0 ];
animation.keyTimes = @[ @0, @(1 / 6.0), @(3 / 6.0), @(5 / 6.0), @1 ];
animation.duration = 0.4;
animation.additive = YES;
[_label.layer addAnimation:animation forKey:@"shake"];

```

3

```

let animation = CAKeyframeAnimation(keyPath: "position.x")
animation.values = [ 0, 10, -10, 10, 0 ]
animation.keyTimes = [ 0, NSNumber(value: (1 / 6.0)), NSNumber(value: (3 / 6.0)),
NSNumber(value: (5 / 6.0)), 1 ]
animation.duration = 0.4
animation.isAdditive = true
label.layer.add(animation, forKey: "shake")

```

C.

```

CATransition *animation = [CATransition animation];
[animation setSubtype:kCATransitionFromRight]; //kCATransitionFromLeft
[animation setDuration:0.5];
[animation setType:kCATransitionPush];
[animation setTimingFunction:[CAMediaTimingFunction
functionWithName:kCAMediaTimingFunctionEaseInEaseOut]];
[[yourView layer] addAnimation:animation forKey:@"SwitchToView1"];

```

```

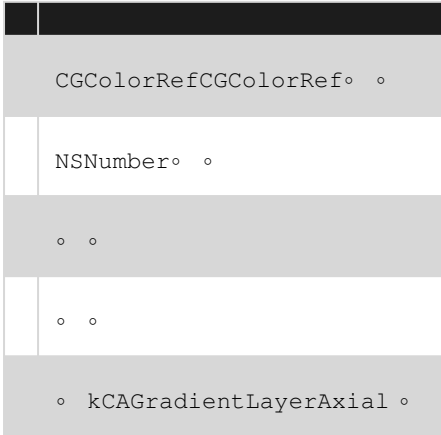
let animation = CATransition()
animation.subtype = kCATransitionFromRight //kCATransitionFromLeft
animation.duration = 0.5
animation.type = kCATransitionPush
animation.timingFunction = CAMediaTimingFunction(name: kCAMediaTimingFunctionEaseInEaseOut)
yourView.layer.addAnimation(animation, forKey: "SwitchToView1")

```


CAAnimation <https://riptutorial.com/zh-TW/ios/topic/981/caanimation>

12: CAGradientLayer

- CAGradientLayer//CALayer◦
- CAGradientLayerlayerlayer//◦



- startPointendPointCAGradientLayer◦
- locations/◦

Examples

CAGradientLayer

```
// View to hold the CAGradientLayer.
let view: UIView = UIView(frame: CGRect(x: 0, y: 0, width: 320, height: 320))

// Initialize gradient layer.
let gradientLayer: CAGradientLayer = CAGradientLayer()

// Set frame of gradient layer.
gradientLayer.frame = view.bounds

// Color at the top of the gradient.
let topColor: CGColor = UIColor.red.cgColor

// Color at the bottom of the gradient.
let bottomColor: CGColor = UIColor.yellow.cgColor

// Set colors.
gradientLayer.colors = [topColor, bottomColor]

// Set locations of the colors.
gradientLayer.locations = [0.0, 1.0]

// Insert gradient layer into view's layer heirarchy.
view.layer.insertSublayer(gradientLayer, at: 0)
```



CGGradientLayer

```
// View to hold the CAGradientLayer.
let view: UIView = UIView(frame: CGRect(x: 0, y: 0, width: 320, height: 320))

// Initialize gradient layer.
let gradientLayer: CAGradientLayer = CAGradientLayer()

// Set frame of gradient layer.
gradientLayer.frame = view.bounds

// Color at the top of the gradient.
let topColor: CGColor = UIColor.blue.cgColor

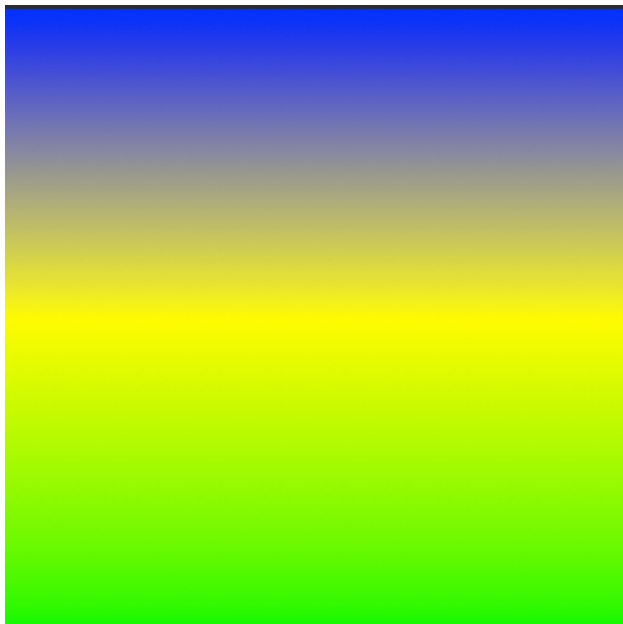
// Color at the middle of the gradient.
let middleColor: CGColor = UIColor.yellow.cgColor

// Color at the bottom of the gradient.
let bottomColor: CGColor = UIColor.green.cgColor

// Set colors.
gradientLayer.colors = [topColor, middleColor, bottomColor]

// Set locations of the colors.
gradientLayer.locations = [0.0, 0.5, 1.0]

// Insert gradient layer into view's layer heirarchy.
view.layer.insertSublayer(gradientLayer, at: 0)
```



CAGradientLayer

```
// View to hold the CAGradientLayer.
let view: UIView = UIView(frame: CGRect(x: 0, y: 0, width: 320, height: 320))

// Initialize gradient layer.
let gradientLayer: CAGradientLayer = CAGradientLayer()

// Set frame of gradient layer.
gradientLayer.frame = view.bounds

// Color at the top of the gradient.
let topColor: CGColor = UIColor.redColor().CGColor

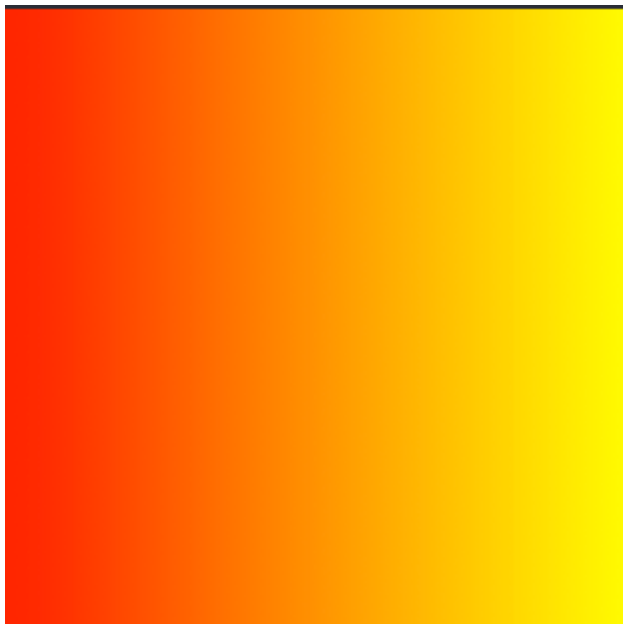
// Color at the bottom of the gradient.
let bottomColor: CGColor = UIColor.yellowColor().CGColor

// Set colors.
gradientLayer.colors = [topColor, bottomColor]

// Set start point.
gradientLayer.startPoint = CGPoint(x: 0.0, y: 0.5)

// Set end point.
gradientLayer.endPoint = CGPoint(x: 1.0, y: 0.5)

// Insert gradient layer into view's layer heirarchy.
view.layer.insertSublayer(gradientLayer, atIndex: 0)
```



CAGradientLayer

```
// View to hold the CAGradientLayer.
let view: UIView = UIView(frame: CGRect(x: 0, y: 0, width: 320, height: 320))

// Initialize gradient layer.
let gradientLayer: CAGradientLayer = CAGradientLayer()

// Set frame of gradient layer.
gradientLayer.frame = view.bounds

// Color at the top of the gradient.
let topColor: CGColor = UIColor.greenColor().CGColor

// Color at the middle of the gradient.
let middleColor: CGColor = UIColor.blueColor().CGColor

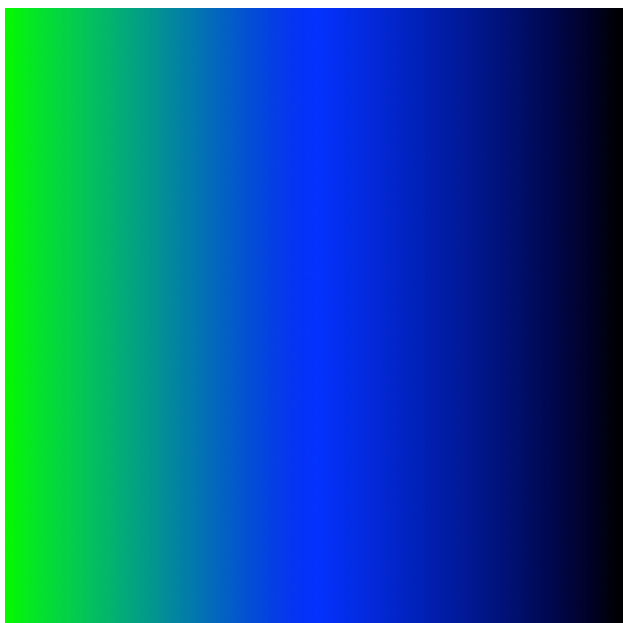
// Color at the bottom of the gradient.
let bottomColor: CGColor = UIColor.blackColor().CGColor

// Set colors.
gradientLayer.colors = [topColor, middleColor, bottomColor]

// Set start point.
gradientLayer.startPoint = CGPoint(x: 0.0, y: 0.5)

// Set end point.
gradientLayer.endPoint = CGPoint(x: 1.0, y: 0.5)

// Insert gradient layer into view's layer heirarchy.
view.layer.insertSublayer(gradientLayer, atIndex: 0)
```



CAGradientLayer

```
// Get the current colors of the gradient.
let oldColors = self.gradientLayer.colors

// Define the new colors for the gradient.
let newColors = [UIColor.red.cgColor, UIColor.yellow.cgColor]

// Set the new colors of the gradient.
self.gradientLayer.colors = newColors

// Initialize new animation for changing the colors of the gradient.
let animation: CABasicAnimation = CABasicAnimation(keyPath: "colors")

// Set current color value.
animation.fromValue = oldColors

// Set new color value.
animation.toValue = newColors

// Set duration of animation.
animation.duration = 0.3

// Set animation to remove once its completed.
animation.isRemovedOnCompletion = true

// Set receiver to remain visible in its final state when the animation is completed.
animation.fillMode = kCAFillModeForwards

// Set linear pacing, which causes an animation to occur evenly over its duration.
animation.timingFunction = CAMediaTimingFunction(name: kCAMediaTimingFunctionLinear)

// Set delegate of animation.
animation.delegate = self

// Add the animation.
self.gradientLayer.addAnimation(animation, forKey: "animateGradientColorChange")
```

Animate



[CAGradientLayer](https://riptutorial.com/zh-TW/ios/topic/1190/cagradientlayer) <https://riptutorial.com/zh-TW/ios/topic/1190/cagradientlayer>

13: CALayer

Examples

CALayer

CALayer

```
let layer = CALayer()
layer.frame = CGRect(x: 0, y: 0, width: 60, height: 80)
```

Objective-C

```
CALayer *layer = [[CALayer alloc] init];
layer.frame = CGRectMake(0, 0, 60, 80);
```

CALayer

```
existingLayer.addSublayer(layer)
```

Objective-C

```
[existingLayer addSublayer:layer];
```

QuartzCore

```
@import QuartzCore
```

Objective-C

```
#import <QuartzCore/QuartzCore.h>
```

CAEmitterLayer

CAEmitterLayerCore Animation。 CAEmitterCell。

。

```
var emitter = CAEmitterLayer()

emitter.emitterPosition = CGPoint(x: frame.size.width / 2.0, y: -20)
emitter.emitterShape = kCAEmitterLayerLine
emitter.emitterSize = CGSize(width: frame.size.width, height: 1)

emitter.emitterCells = cells
layer.addSublayer(emitter)
```



```

import QuartzCore

class ConfettiView: UIView {
    // main emitter layer
    var emitter: CAEmitterLayer!

    // array of color to emit
    var colors: [UIColor]!

    // intensity of appearance
    var intensity: Float!

    private var active :Bool!

    required init?(coder aDecoder: NSCoder) {
        super.init(coder: aDecoder)
        setup()
    }

    override init(frame: CGRect) {
        super.init(frame: frame)
        setup()
    }

    func setup() {
        // initialization
        colors = [UIColor.redColor(),
                 UIColor.greenColor(),
                 UIColor.blueColor()
                ]
        intensity = 0.2

        active = false
    }

    func startConfetti() {
        emitter = CAEmitterLayer()

        emitter.emitterPosition = CGPoint(x: frame.size.width / 2.0, y: -20)
        emitter.emitterShape = kCAEmitterLayerLine
        emitter.emitterSize = CGSize(width: frame.size.width, height: 1)

        var cells = [CAEmitterCell]()
        for color in colors {
            cells.append(confettiWithColor(color))
        }

        emitter.emitterCells = cells
        layer.addSublayer(emitter)
        active = true
    }

    func stopConfetti() {
        emitter?.birthRate = 0
        active = false
    }

    func confettiWithColor(color: UIColor) -> CAEmitterCell {

```

```

let confetti = CAEmitterCell()

confetti.birthRate = 10.0 * intensity
confetti.lifetime = 180.0 * intensity
confetti.lifetimeRange = 0
confetti.color = color.CGColor
confetti.velocity = CGFloat(350.0 * intensity)
confetti.velocityRange = CGFloat(40.0 * intensity)
confetti.emissionLongitude = CGFloat(M_PI)
confetti.emissionRange = CGFloat(M_PI_4)
confetti.spin = CGFloat(3.5 * intensity)
confetti.spinRange = CGFloat(4.0 * intensity)

// WARNING: A layer can set this property to a CGImageRef to display the image as its
contents.
confetti.contents = UIImage(named: "confetti")?.CGImage
return confetti
}

internal func isActive() -> Bool {
    return self.active
}
}

```

“confetti”**confetti.contentsRect**

UIImageCALayer

contentslayer

```
myView.layer.contents = UIImage(named: "star")?.CGImage
```

- UIImageCGImage ◦

```

let myLayer = CALayer()
let myImage = UIImage(named: "star")?.CGImage
myLayer.frame = myView.bounds
myLayer.contents = myImage
myView.layer.addSublayer(myLayer)

```

- UIView UIImage ◦



- UIImageUIView◦

contentsGravity

```
myView.layer.contents = UIImage(named: "star").CGImage
myView.layer.contentsGravity = kCAGravityTop
myView.layer.geometryFlipped = true
```

iOS [geometryFlipped true](#) [Stack Overflow](#)

contentsGravity UIView UIView UIImage

kCAGravityResize

◦



kCAGravityResizeAspect



kCAGravityResizeAspectFill



kCAGravityCenter



`kCAGravityTop`



`kCAGravityBottom`



`kCAGravityLeft`



`kCAGravityRight`



`kCAGravityTopLeft`



`kCAGravityTopRight`



`kCAGravityBottomLeft`



`kCAGravityBottomRight`

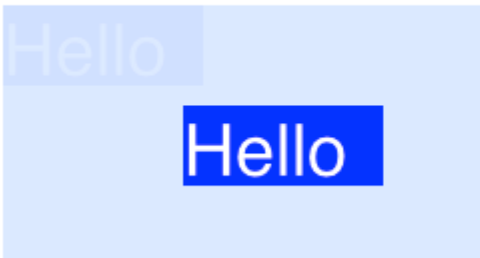


-
- [CGContextDrawImageDrawRectUIImage](#)
 - [CALayer](#)

-
- [Stack Overflow](#) ◦

CALayer

-
- -
 -



CALayertransformCATransform3D◦

```
myLayer.transform = CATransform3DMakeTranslation(20, 30, 0)
```

Make **CATransform3D Make Translation**◦ Make ◦

```
let rotation = CATransform3DMakeRotation(CGFloat(30.0 * M_PI / 180.0), 20, 20, 0)
myLayer.transform = CATransform3DTranslate(rotation, 20, 30, 0)
```

◦ ◦

UIView

◦

```
import UIKit

class ViewController: UIViewController {

    var myLayer = CATextLayer()
    @IBOutlet weak var myView: UIView!

    override func viewDidLoad() {
        super.viewDidLoad()

        // setup the sublayer
        addSubLayer()

        // do the transform
        transformExample()
    }

    func addSubLayer() {
        myLayer.frame = CGRect(x: 0, y: 0, width: 100, height: 40)
        myLayer.backgroundColor = UIColor.blueColor().CGColor
        myLayer.string = "Hello"
        myView.layer.addSublayer(myLayer)
    }

    //***** Replace this function with the examples below *****

    func transformExample() {

        // add transform code here ...

    }

}
```

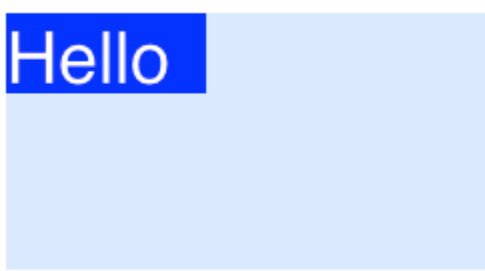
[CALayer](#) [CATextLayer](#)◦



◦

```
CATransform3DMakeTranslation(tx: CGFloat, ty: CGFloat, tz: CGFloat)
```

txX tyY tzZ◦



iOS9050

```
myLayer.transform = CATransform3DMakeTranslation(90, 50, 0)
```

- transformExample()◦
- tz0◦
- ◦ ◦

◦

```
CATransform3DMakeScale(sx: CGFloat, sy: CGFloat, sz: CGFloat)
```

sx sysz**xyz**◦



Hello



```
myLayer.transform = CATransform3DMakeScale(0.5, 3.0, 1.0)
```

- z1.0◦
- ◦ ◦ ◦

◦

```
CATransform3DMakeRotation(angle: CGFloat, x: CGFloat, y: CGFloat, z: CGFloat)
```

angle x yz◦ **0**◦



Hello



Hello

30

```
let degrees = 30.0
let radians = CGFloat(degrees * M_PI / 180)
myLayer.transform = CATransform3DMakeRotation(radians, 0.0, 0.0, 1.0)
```

- **xyz** ◦ xy0.0z1.0 ◦
- ◦ z-1.0 ◦
- ◦ ◦

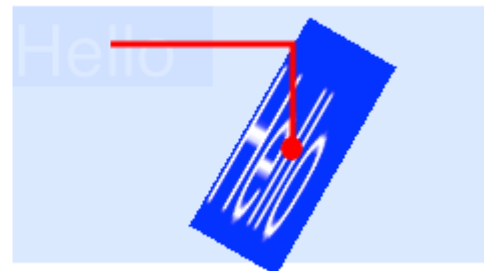
concatination

```
CATransform3DConcat(a: CATransform3D, b: CATransform3D)
```

- Make ◦ Make ◦



Hello



Hello

```
let degrees = 30.0
let radians = CGFloat(degrees * M_PI / 180)

// translate
var transform = CATransform3DMakeTranslation(90, 50, 0)

// rotate
transform = CATransform3DRotate(transform, radians, 0.0, 0.0, 1.0)

// scale
transform = CATransform3DScale(transform, 0.5, 3.0, 1.0)
```

```
// apply the transforms
myLayer.transform = transform
```

- ◦
- ◦

◦ ◦ ◦

◦ 0.5, 0.5 **superlayer** ◦

```
myLayer.anchorPoint = CGPoint(x: 0.0, y: 1.0)
myLayer.position = CGPoint(x: 50, y: 50)
```

◦ ◦ ◦ ◦

- **anchorPoint**
-

- **CALayer**
- **Bezier**

[Stack Overflow](#) ◦

CALayer ◦ ◦

```
CATransaction.begin()
CATransaction.setDisableActions(true)

// change layer properties that you don't want to animate

CATransaction.commit()
```

Objective-C

```
[CATransaction begin];
[CATransaction setDisableActions:YES];

// change layer properties that you don't want to animate

[CATransaction commit];
```

```
layer.masksToBounds = true;
layer.cornerRadius = 8;
```

5

- `shadowOffset - //`

```
self.layer.shadowOffset = CGSizeMake(-1, -1); // 1px left and up
self.layer.shadowOffset = CGSizeMake(1, 1); // 1px down and right
```

- shadowColor -

```
self.layer.shadowColor = [UIColor blackColor].CGColor;
```

- shadowOpacity - 01

```
self.layer.shadowOpacity = 0.2;
```

- shadowRadius - SketchPhotoshopblur

```
self.layer.shadowRadius = 6;
```

- shadowPath - iOSalphaalphaPNG。。

Objective-C

```
self.layer.shadowPath = [UIBezierPath bezierPathWithOvalInRect:CGRectMake(0,0,100,100)];
//this does a circular shadow
```

3

```
self.layer.shadowPath = UIBezierPath(ovalIn: CGRect(x: 0, y: 0, width: 100, height:
100)).cgPath
```

CALayer <https://riptutorial.com/zh-TW/ios/topic/1462/calayer>

14: CAShapeLayer

1. shapeLayer.fillColor
2. shapeLayer.fillRule
3. shapeLayer.lineCap
4. shapeLayer.lineDashPattern
5. shapeLayer.lineDashPhase
6. shapeLayer.lineJoin

CAShapeLayer ◦ ◦

Examples

CAShapeLayer

UIBezierPathShapeLayer

```
CAShapeLayer *circleLayer = [CAShapeLayer layer];
[circleLayer setPath:[[UIBezierPath bezierPathWithOvalInRect:
CGRectMake(50, 50, 100, 100)] CGPath]];
circleLayer.lineWidth = 2.0;
[circleLayer setStrokeColor:[[UIColor redColor] CGColor]];
[circleLayer setFillColor:[[UIColor clearColor] CGColor]];
circleLayer.lineJoin = kCALineJoinRound; //4 types are available to create a line style
circleLayer.lineDashPattern = [NSArray arrayWithObjects:
[NSNumber numberWithInt:2],[NSNumber numberWithInt:3 ], nil];
// self.origImage is parentView
[[self.view layer] addSublayer:circleLayer];
self.currentShapeLayer = circleLayer; // public value using to keep that reference of the
shape Layer
self.view.layer.borderWidth = 1.0f;
self.view.layer.borderColor = [[UIColor blueColor]CGColor]; // that will plotted in the
mainview
```

ShapeLayer

- currentShapeLayer

1

```
[self.currentShapeLayer removeFromSuperlayer];
```

2

```
self.view.layer.sublayers = nil ; //removed all earlier shapes
```

```
//Draw Square Shape
```

```
CAShapeLayer *squareLayer = [CAShapeLayer layer];
```

```

squareLayer.frame = CGRectMake(20, 20, 100, 100);
squareLayer.lineWidth = 2.0;
squareLayer.fillColor = nil;
squareLayer.strokeColor = [[UIColor redColor] CGColor];
squareLayer.path = [UIBezierPath bezierPathWithRect:squareLayer.bounds].CGPath;
[[self.view layer] addSublayer:squareLayer];

//Draw Circle Shape

CAShapeLayer *circleShape = [CAShapeLayer layer];
circleShape.frame = CGRectMake(160, 20, 120, 120);
circleShape.lineWidth = 2.0;
circleShape.fillColor = nil;
circleShape.strokeColor = [[UIColor redColor] CGColor];
circleShape.path = [UIBezierPath bezierPathWithOvalInRect:circleShape.bounds].CGPath;
[[self.view layer] addSublayer:circleShape];

//Subpaths
//UIBezierPath can have any number of "path segments" (or subpaths) so you can effectively
draw as many shapes or lines as you want in a single path object

CAShapeLayer *shapeLayer = [CAShapeLayer layer];
shapeLayer.frame = CGRectMake(20, 140, 200, 200);
shapeLayer.lineWidth = 2.0;
shapeLayer.fillColor = nil;
shapeLayer.strokeColor = [[UIColor redColor] CGColor];

CGMutablePathRef combinedPath= CGPathCreateMutableCopy(circleShape.path);
CGPathAddPath(combinedPath, NULL, squareLayer.path);

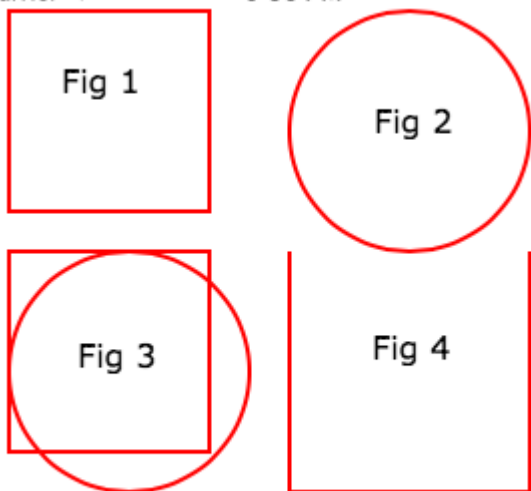
shapeLayer.path = combinedPath;
[[self.view layer] addSublayer:shapeLayer];

//Open Path
// Paths do not need to connect their end points back to their starting points. A path that
connects back to its starting point is called a closed path, and one that does not is called
an open path.

shapeLayer = [CAShapeLayer layer];
shapeLayer.frame = CGRectMake(160, 140, 300, 300);
shapeLayer.lineWidth = 2.0;
shapeLayer.fillColor = nil;
shapeLayer.strokeColor = [[UIColor redColor] CGColor];

UIBezierPath *linePath=[UIBezierPath bezierPath];
[linePath moveToPoint:CGPointZero];
[linePath addLineToPoint:CGPointMake(0 , 120)];
[linePath addLineToPoint:CGPointMake(120 , 120)];
[linePath addLineToPoint:CGPointMake(120 , 0)];
shapeLayer.path = linePath.CGPath;
[[self.view layer] addSublayer:shapeLayer];

```



//

```

CASShapeLayer *squareLayer = [CASShapeLayer layer];
squareLayer.frame = CGRectMake(20, 30, 100, 100);
squareLayer.lineWidth = 2.0;
squareLayer.fillColor = [[UIColor yellowColor]CGColor];
squareLayer.strokeColor = [[UIColor redColor] CGColor];
squareLayer.path = [UIBezierPath bezierPathWithRect:squareLayer.bounds].CGPath;
[[self.view layer] addSublayer:squareLayer];

//Fill Pattern Color
//images.jpeg

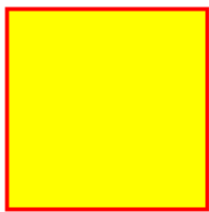
squareLayer = [CASShapeLayer layer];
squareLayer.frame = CGRectMake(140, 30, 100, 100);
squareLayer.lineWidth = 2.0;
squareLayer.fillColor = [[UIColor colorWithPatternImage:[UIImage
imageName:@"images.jpeg"]]CGColor];
squareLayer.strokeColor = [[UIColor redColor] CGColor];
squareLayer.path = [UIBezierPath bezierPathWithRect:squareLayer.bounds].CGPath;
[[self.view layer] addSublayer:squareLayer];

//Fill Rule

//Type 1: kCAFillRuleNonZero
squareLayer = [CASShapeLayer layer];
squareLayer.frame = CGRectMake(0, 140, 150, 150);
squareLayer.lineWidth = 2.0;
squareLayer.fillColor = [[UIColor yellowColor]CGColor];
squareLayer.fillRule = kCAFillRuleNonZero; // indicate the rule type
squareLayer.strokeColor = [[UIColor redColor] CGColor];
UIBezierPath *outerPath = [UIBezierPath bezierPathWithRect:CGRectInset (squareLayer.bounds,
20.0, 20.0)];
UIBezierPath *innerPath = [UIBezierPath bezierPathWithRect:CGRectInset (squareLayer.bounds,
50.0, 50.0)];
CGMutablePathRef combinedPath= CGPathCreateMutableCopy(outerPath.CGPath);
CGPathAddPath(combinedPath, NULL, innerPath.CGPath);
squareLayer.path = combinedPath;
[[self.view layer] addSublayer:squareLayer];

```

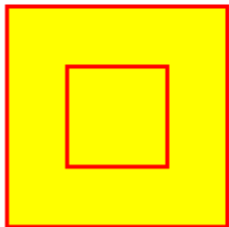
```
//Type 2: kCAFillRuleEvenOdd
squareLayer = [CAShapeLayer layer];
squareLayer.frame = CGRectMake(140, 140, 150, 150);
squareLayer.lineWidth = 2.0;
squareLayer.fillColor = [[UIColor yellowColor]CGColor];
squareLayer.fillRule = kCAFillRuleEvenOdd; // indicate the rule type
squareLayer.strokeColor = [[UIColor redColor] CGColor];
outerPath = [UIBezierPath bezierPathWithRect:CGRectInset(squareLayer.bounds, 20.0, 20.0)];
innerPath = [UIBezierPath bezierPathWithRect:CGRectInset(squareLayer.bounds, 50.0, 50.0)];
combinedPath= CGPathCreateMutableCopy(outerPath.CGPath);
CGPathAddPath(combinedPath, NULL, innerPath.CGPath);
squareLayer.path = combinedPath;
[[self.view layer] addSublayer:squareLayer];
```



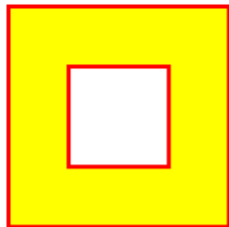
Fill color



Fill Pattern



Fill Rule Zero



Fill Rule Even

fillColor

Fill the color based on the drawn shape.

fillRule

Fill Rule the there are two rule is applied to draw the shape.

1. kCAFillRuleNonZero
2. kCAFillRuleEvenOdd

lineCap

Below type used to change the style of the line.

1. kCALineCapButt
2. kCALineCapRound
3. kCALineCapSquare

lineDashPattern

The dash pattern applied to the shape's path when stroked.

Create DashStyle while you will stroke the line.

lineDashPhase

The dash phase applied to the shape's path when stroked. Animatable.

lineJoin

Line join style for the shape path.Below style use to draw the line join style.

1. kCALineJoinMiter
2. kCALineJoinRound
3. kCALineJoinBevel

lineWidth

Which using to set the line width.

`miterLimit`

The miter limit used when stroking the shape's path. Animatable.

`strokeColor`

Set the stroke color based on the path of the line.

`strokeStart`

When the stroke will start.

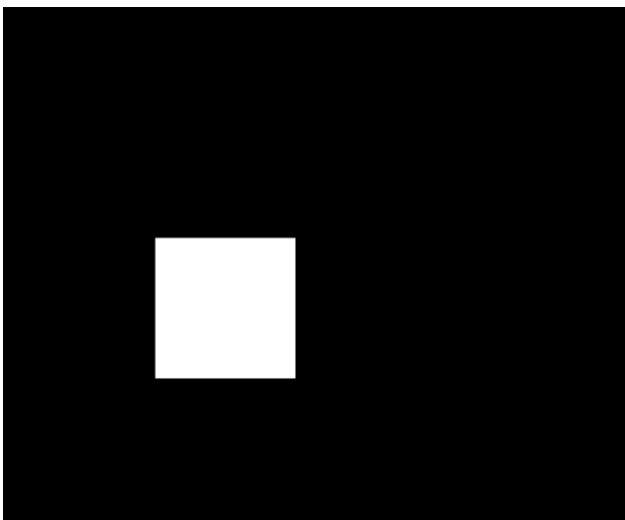
`strokeEnd`

When the stroke will end.

```
CAShapeLayer *mask = [[CAShapeLayer alloc] init];
mask.frame = CGRectMake(50, 50, 100, 100);
CGFloat width = 100;
CGFloat height = 100;
CGMutablePathRef path = CGPathCreateMutable();
CGPathMoveToPoint(path, nil, 30, 30);
CGPathAddLineToPoint(path, nil, width, 30);
CGPathAddLineToPoint(path, nil, width, height);
CGPathAddLineToPoint(path, nil, 30, height);
CGPathAddLineToPoint(path, nil, 30, 30);
CGPathCloseSubpath(path);

mask.path = path;
CGPathRelease(path);

self.view.layer.mask = mask;
```



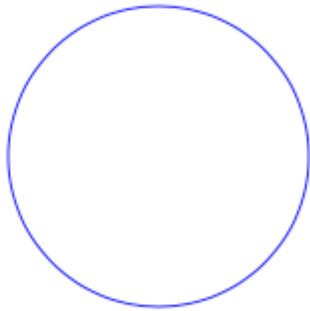

```
CAShapeLayer *circle = [CAShapeLayer layer];

[circle setPath:[[UIBezierPath bezierPathWithOvalInRect:CGRectMake(100, 100, 150, 150)]
CGPath]];

[circle setStrokeColor:[[UIColor blueColor] CGColor]];

[circle setFillColor:[[UIColor clearColor] CGColor]];

[[self.view layer] addSublayer:circle];
```



CAShapeLayer

```
CAShapeLayer *circle = [CAShapeLayer layer];

[circle setPath:[[UIBezierPath bezierPathWithOvalInRect:CGRectMake(100, 100, 150, 150)]
CGPath]];

[circle setStrokeColor:[[UIColor blueColor] CGColor]];

[circle setFillColor:[[UIColor clearColor] CGColor]];

[[self.view layer] addSublayer:circle];

CABasicAnimation *pathAnimation = [CABasicAnimation animationWithKeyPath:@"strokeEnd"];

pathAnimation.duration = 1.5f;

pathAnimation.fromValue = [NSNumber numberWithFloat:0.0f];

pathAnimation.toValue = [NSNumber numberWithFloat:1.0f];

pathAnimation.repeatCount = 10;

pathAnimation.autoreverses = YES;

[circle addAnimation:pathAnimation
forKey:@"strokeEnd"];
```

)

CShapeLayer <https://riptutorial.com/zh-TW/ios/topic/3575/cashapelayer>

15: CGContext

CGContextRef opaqueQuartz 2D。PDF。

Examples

```
CGContextRef context = UIGraphicsGetCurrentContext();

CGContextSetLineWidth(context, 5.0);
CGColorSpaceRef colorspace = CGColorSpaceCreateDeviceRGB();
CGContextMoveToPoint(context, 200, 400);
CGContextAddLineToPoint(context, 100, 100);
CGContextStrokePath(context);
CGColorSpaceRelease(colorspace);
```



Draw ToCore Text

```
[NSString* textToDraw = @"Welcome to the world Of IOS";

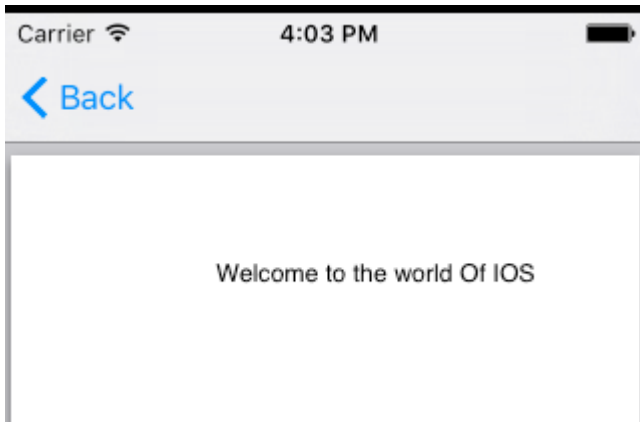
CFStringRef stringRef = (__bridge CFStringRef)textToDraw;

CFAttributedStringRef currentText = CFAttributedStringCreate(NULL, stringRef, NULL);
CTFramesetterRef framesetter = CTFramesetterCreateWithAttributedString(currentText);
CGRect frameRect = CGRectMake(0, 0, 300, 100);
CGMutablePathRef framePath = CGPathCreateMutable();
CGPathAddRect(framePath, NULL, frameRect);

CFRange currentRange = CFRangeMake(0, 0);
CTFrameRef frameRef = CTFramesetterCreateFrame(framesetter, currentRange, framePath,
NULL);
CGPathRelease(framePath);
CGContextRef currentContext = UIGraphicsGetCurrentContext();

CGContextSetTextMatrix(currentContext, CGAffineTransformIdentity);
CGContextTranslateCTM(currentContext, 200, 300);
CGContextScaleCTM(currentContext, 2, -2);
CTFrameDraw(frameRef, currentContext);

CFRelease(frameRef);
CFRelease(stringRef);
CFRelease(framesetter);
```



CGContext <https://riptutorial.com/zh-TW/ios/topic/2664/cgcontext>

16: CLLocation

Examples

```
CLLocationManager *locationManager = [[CLLocationManager alloc] init];
locationManager.delegate = self;
locationManager.desiredAccuracy = kCLLocationAccuracyBest;
locationManager.distanceFilter = 5;
```

5.

CLLocationManager

1 - CoreLocation.framework;

```
root directory -> build phases -> Link Binary With Libraries
```

+CoreLocation.framework.

2-info.plist.

```
<key>NSLocationWhenInUseUsageDescription</key>
<string>message to display when asking for permission</string>
```

3-CoreLocationViewController.

```
import CoreLocation
```

4-ViewControllerCLLocationManagerDelegate

```
class ViewController: UIViewController, CLLocationManagerDelegate {}
```

CLLocationManagerViewController.

```
var manager:CLLocationManager!
```

'let'

```
//initialize the manager
manager = CLLocationManager()

//specify delegate
manager.delegate = self

//set the minimum distance the phone needs to move before an update event is triggered (for
example: 100 meters)
manager.distanceFilter = 100
```

```

//set Accuracy to any of the following depending on your use case

//let kCLLocationAccuracyBestForNavigation: CLLocationAccuracy
//let kCLLocationAccuracyBest: CLLocationAccuracy
//let kCLLocationAccuracyNearestTenMeters: CLLocationAccuracy
//let kCLLocationAccuracyHundredMeters: CLLocationAccuracy
//let kCLLocationAccuracyKilometer: CLLocationAccuracy
//let kCLLocationAccuracyThreeKilometers: CLLocationAccuracy

manager.desiredAccuracy = kCLLocationAccuracyBest

//ask the user for permission
manager.requestWhenInUseAuthorization()

//Start collecting location information
if #available(iOS 9.0, *) {
    manager.requestLocation()
} else {
    manager.startUpdatingLocation()
}

```

distanceFilter。

```

func locationManager(manager: CLLocationManager, didUpdateLocations locations: [CLLocation])
{}

```

locationsCLLocation。 coordinate, altitude, floor, horizontalAccuracy, verticalAccuracy, timestamp, description, course, speed **and** distance(from:)

。

“”。

“”。

Plist。

CLLocation <https://riptutorial.com/zh-TW/ios/topic/2002/cllocation>

17: CloudKit

- NSData
- NSDate
- NSNumberInt / Double
- NSStringString
- NSArray
- CLLocation
- CKReference
- CKAsset

CloudKit

Examples

CloudKit

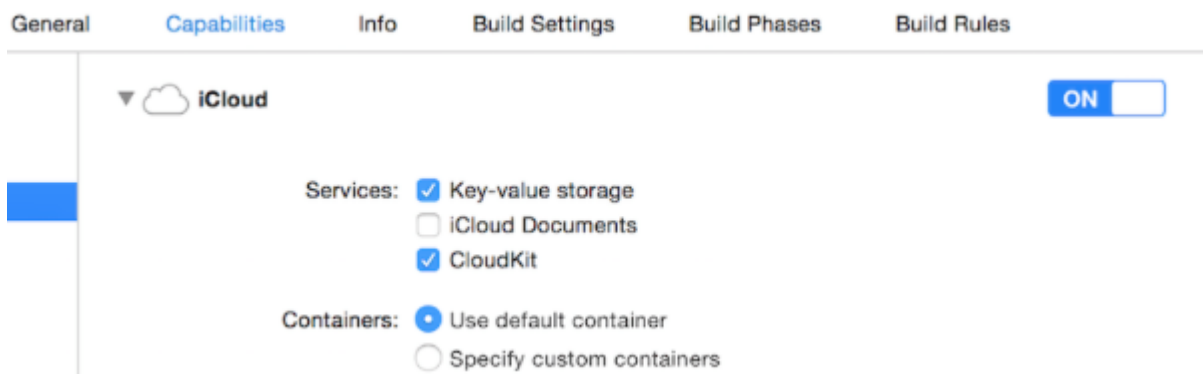
iCloudCloudKit。

iCloud

1-Project Navigator“”。

2-“Apple ID“。 Xcode - > - >。

3-“iCloud。 ““CloudKit”。



4-

- Steps:
- ✓ Add the "iCloud" entitlement to your App ID
 - ✓ Add the "iCloud containers" entitlement to your App ID
 - ✓ Add the "iCloud" entitlement to your entitlements file
 - ✓ Link CloudKit.framework

CloudKit。

CloudKit

CloudKitCloudKit。 CloudKit。

-
-
- [AppleAPN](#)

◦ CloudKitUsers。

◦ iOS SDK。 ◦

CloudKit

CloudKit

- CKRecordID
- CKRecord

◦ `NSDateTimeIntervalSinceReferenceDate()` ◦ **###** ◦ **###**are numbers。

```
let timestamp = String(format: "%f", NSDate.timeIntervalSinceReferenceDate())
let timestampParts = timestamp.componentsSeparatedByString(".")
let recordID = CKRecordID(recordName: timestampParts[0])
```

CloudKitID。

```
let record = CKRecord(recordType: "Users", recordID: recordID)
record.setObject("Some Text", forKey: "text")
record.setObject(CKAsset(fileURL: someValidImageURL), forKey: "image")
record.setObject(NSDate(), forKey: "date")
```

Objective-C

```
CKRecord *record = [[CKRecord alloc] initWithRecordType: "Users" recordID: recordID];
[record setObject: "Some Text" forKey: "text"];
[record setObject: [CKAsset assetWithURL: someValidImageURL] forKey: "image"];
[record setObject: [[NSDate alloc] init] forKey: "date"];
```

UIImageCloudKitUIImageCKAsset。


```
let container = CKContainer.defaultContainer()
let database = container.privateCloudDatabase // or container.publicCloudDatabase
```

CloudKit

```
database.saveRecord(record, completionHandler: { (_, error) -> Void in
    print(error ?? "")
})
```

CloudKit <https://riptutorial.com/zh-TW/ios/topic/4946/cloudkit>

18: CoreImage

Examples

Objective-C

```
NSArray *properties = [CIFilter filterNamesInCategory:kCICategoryBuiltIn];

for (NSString *filterName in properties)
{
    CIFilter *fltr = [CIFilter filterWithName:filterName];
    NSLog(@"%@", [fltr attributes]);
}
```

CISepiaTone

```
CIAttributeFilterDisplayName = "Sepia Tone";
CIAttributeFilterName = CISepiaTone;
CIAttributeReferenceDocumentation = "http://developer.apple.com/cgi-bin/apple_ref.cgi?apple_ref=//apple_ref/doc/filter/ci/CISepiaTone";
inputImage = {
    CIAttributeClass = CIImage;
    CIAttributeDescription = "The image to use as an input image. For filters that also use a background image, this is the foreground image.";
    CIAttributeDisplayName = Image;
    CIAttributeType = CIAttributeTypeImage;
};
inputIntensity = {
    CIAttributeClass = NSNumber;
    CIAttributeDefault = 1;
    CIAttributeDescription = "The intensity of the sepia effect. A value of 1.0 creates a monochrome sepia image. A value of 0.0 has no effect on the image.";
    CIAttributeDisplayName = Intensity;
    CIAttributeIdentity = 0;
    CIAttributeMin = 0;
    CIAttributeSliderMax = 1;
    CIAttributeSliderMin = 0;
    CIAttributeType = CIAttributeTypeScalar;
};
}
```

```
CIImage *beginImage = [CIImage imageWithCGImage:[myImageView.image CGImage]];
CIContext *context = [CIContext contextWithOptions:nil];
//select Filter Name and Intensity
CIFilter *filter = [CIFilter filterWithName:@"CISepiaTone" keysAndValues:
kCIInputImageKey, beginImage, @"inputIntensity", [NSNumber numberWithInt:0.8], nil];
CIImage *outputImage = [filter outputImage];

CGImageRef cgimg = [context createCGImage:outputImage fromRect:[outputImage extent]];
UIImage *newImg = [UIImage imageWithCGImage:cgimg];

[myImageView1 setImage:newImg];

CGImageRelease(cgimg);
```



```
UIImageView *imageView1=[[UIImageView alloc]initWithFrame:CGRectMake(0, 0,
self.view.frame.size.width, self.view.frame.size.height/2)];
UIImageView *imageView2=[[UIImageView alloc]initWithFrame:CGRectMake(0,
self.view.frame.size.height/2, self.view.frame.size.width, self.view.frame.size.height/2)];
imageView1.image=[UIImage imageNamed:@"image.png"];

CIImage *beginImage = [CIImage imageWithCGImage:[imageView1.image CGImage]];
CIText *context = [CIText contextWithOptions:nil];
//select Filter Name and Intensity

CIFilter *filter = [CIFilter filterWithName:@"CIColorPosterize"];
[filter setValue:beginImage forKey:kCIInputImageKey];
[filter setValue:[NSNumber numberWithInt:8.0] forKey:@"inputLevels"];
CIImage *outputImage = [filter outputImage];

CGImageRef cgimg = [context createCGImage:outputImage fromRect:[outputImage extent]];
UIImage *newImg = [UIImage imageWithCGImage:cgimg];

[imageView2 setImage:newImg];

CGImageRelease(cgimg);
[self.view addSubview:imageView1];
[self.view addSubview:imageView2];
```



```
/* CIAccordionFoldTransition,  
  CIAdditionCompositing,  
  CIAffineClamp,  
  CIAffineTile,  
  CIAffineTransform,  
  CIColorAverage,  
  CIColorHistogram,  
  CIColorMaximum,  
  CIColorMaximumAlpha,  
  CIColorMinimum,  
  CIColorMinimumAlpha,  
  CIAztecCodeGenerator,  
  CIBarsSwipeTransition,  
  CIBlendWithAlphaMask,  
  CIBlendWithMask,  
  CIBloom,  
  CIBoxBlur,  
  CIBumpDistortion,  
  CIBumpDistortionLinear,  
  CIColorCheckerboardGenerator,  
  CICircularSplashDistortion,  
  CICircularScreen,  
  CICircularWrap,  
  CICMYKHalftone,  
  CIColorCode128BarcodeGenerator,  
  CIColorBlendMode,  
  CIColorBurnBlendMode,  
  CIColorClamp,  
  CIColorControls,  
  CIColorCrossPolynomial,
```

CIColorCube,
CIColorCubeWithColorSpace,
CIColorDodgeBlendMode,
CIColorInvert,
CIColorMap,
CIColorMatrix,
CIColorMonochrome,
CIColorPolynomial,
CIColorPosterize,
CIColumnAverage,
CIComicEffect,
CIConstantColorGenerator,
CIConvolution3X3,
CIConvolution5X5,
CIConvolution7X7,
CIConvolution9Horizontal,
CIConvolution9Vertical,
CICopyMachineTransition,
CICrop,
CICrystallize,
CIDarkenBlendMode,
CIDepthOfField,
CIDifferenceBlendMode,
CIDiscBlur,
CIDisintegrateWithMaskTransition,
CIDisplacementDistortion,
CIDissolveTransition,
CIDivideBlendMode,
CIDotScreen,
CIDroste,
CIEdges,
CIEdgeWork,
CIEightfoldReflectedTile,
CIExclusionBlendMode,
CIExposureAdjust,
CIFalseColor,
CIFlashTransition,
CIFourfoldReflectedTile,
CIFourfoldRotatedTile,
CIFourfoldTranslatedTile,
CIGammaAdjust,
CIGaussianBlur,
CIGaussianGradient,
CIGlassDistortion,
CIGlassLozenge,
CIGlideReflectedTile,
CIGloom,
CIHardLightBlendMode,
CIHatchedScreen,
CIHeightFieldFromMask,
CIHexagonalPixellate,
CIHighlightShadowAdjust,
CIHistogramDisplayFilter,
CIHoleDistortion,
CIHueAdjust,
CIHueBlendMode,
CIKaleidoscope,
CILanczosScaleTransform,
CILenticularHaloGenerator,
CILightenBlendMode,
CILightTunnel,

CILinearBurnBlendMode,
CILinearDodgeBlendMode,
CILinearGradient,
CILinearToSRGBToneCurve,
CILineOverlay,
CILineScreen,
CILuminosityBlendMode,
CIMaskedVariableBlur,
CIMaskToAlpha,
CIMaximumComponent,
CIMaximumCompositing,
CIMedianFilter,
CIMinimumComponent,
CIMinimumCompositing,
CIModTransition,
CIMotionBlur,
CIMultiplyBlendMode,
CIMultiplyCompositing,
CINoiseReduction,
CIOpTile,
CIOverlayBlendMode,
CIPageCurlTransition,
CIPageCurlWithShadowTransition,
CIParallelogramTile,
CIPDF417BarcodeGenerator,
CIPerspectiveCorrection,
CIPerspectiveTile,
CIPerspectiveTransform,
CIPerspectiveTransformWithExtent,
CIPhotoEffectChrome,
CIPhotoEffectFade,
CIPhotoEffectInstant,
CIPhotoEffectMono,
CIPhotoEffectNoir,
CIPhotoEffectProcess,
CIPhotoEffectTonal,
CIPhotoEffectTransfer,
CIPinchDistortion,
CIPinLightBlendMode,
CIPixellate,
CIPointillize,
CIQRCodeGenerator,
CIRadialGradient,
CIRandomGenerator,
CIRippleTransition,
CIRowAverage,
CISaturationBlendMode,
CIScreenBlendMode,
CISepiaTone,
CIShadedMaterial,
CISharpenLuminance,
CISixfoldReflectedTile,
CISixfoldRotatedTile,
CISmoothLinearGradient,
CISoftLightBlendMode,
CISourceAtopCompositing,
CISourceInCompositing,
CISourceOutCompositing,
CISourceOverCompositing,
CISpotColor,
CISpotLight,

```
CISSRGBToneCurveToLinear,  
CIStarShineGenerator,  
CIStraightenFilter,  
CIStretchCrop,  
CIStripesGenerator,  
CISubtractBlendMode,  
CISunbeamsGenerator,  
CISwipeTransition,  
CITemperatureAndTint,  
CIToneCurve,  
CITorusLensDistortion,  
CITriangleKaleidoscope,  
CITriangleTile,  
CITwelvefoldReflectedTile,  
CITwirlDistortion,  
CIUnsharpMask,  
CIVibrance,  
CIVignette,  
CIVignetteEffect,  
CIVortexDistortion,  
CIWhitePointAdjust,  
CIZoomBlur*/
```

CoreImage <https://riptutorial.com/zh-TW/ios/topic/7278/coreimage>

19: CTCallCenter

Examples

Apple

CTCallCenter。

CTCallCenter。

Objective-C

```
@property (atomic, strong) CTCallCenter *callCenter;
```

init

```
[self setCallCenter:[CTCallCenter new]];
```

```
- (void)registerPhoneCallListener
{
    [[self callCenter] setCallEventHandler:^(CTCall * _Nonnull call) {
        NSLog(@"CallEventHandler called - interception in progress");

        if ([call.callState isEqualToString: CTCallStateConnected])
        {
            NSLog(@"Connected");
        }
        else if ([call.callState isEqualToString: CTCallStateDialing])
        {
            NSLog(@"Dialing");
        }
        else if ([call.callState isEqualToString: CTCallStateDisconnected])
        {
            NSLog(@"Disconnected");
        }
        else if ([call.callState isEqualToString: CTCallStateIncoming])
        {
            NSLog(@"Incomming");
        }
    }];
}
```

。

4

```
CTCallStateDialing
CTCallStateIncoming
CTCallStateConnected
CTCallStateDisconnected
```



```
self.callCenter = CTCallCenter()
self.callCenter.callEventHandler = { call in
    // Handle your interception
    if call.callState == CTCallStateConnected
    {
    }
}
```

2VOIP

-> -> -> IP

CallKit - ios 10

```
//Header File
<CallKit/CXCallObserver.h>
CXCallObserver *callObserver = [[CXCallObserver alloc] init];
// If queue is nil, then callbacks will be performed on main queue
[callObserver setDelegate:self queue:nil];
// Don't forget to store reference to callObserver, to prevent it from being released
self.callObserver = callObserver;
// get call status
- (void)callObserver:(CXCallObserver *)callObserver callChanged:(CXCall *)call {
    if (call.hasConnected) {
        // perform necessary actions
    }
}
```

CTCallCenter <https://riptutorial.com/zh-TW/ios/topic/3007/ctcallcenter>

20: CydiaSubstrate

iPhonecydia◦

◦

Theos

<https://github.com/theos/theos/wiki/Installation>

Examples

Theos

nic

```
$THEOS/bin/nic.pl
```

```
NIC 2.0 - New Instance Creator
```

```
-----
```

```
[1.] iphone/activator_event
[2.] iphone/application_modern
[3.] iphone/cyidget
[4.] iphone/flipswitch_switch
[5.] iphone/framework
[6.] iphone/ios7_notification_center_widget
[7.] iphone/library
[8.] iphone/notification_center_widget
[9.] iphone/preference_bundle_modern
[10.] iphone/tool
[11.] iphone/tweak
[12.] iphone/xpc_service
```

```
Choose a Template (required):
```

```
[11.] iphone/tweak
```

```
-rw-r--r--@ 1 gkpln3  staff  214B Jun 12 15:09 Makefile
-rw-r--r--@ 1 gkpln3  staff   89B Jun 11 22:58 TorchonFocus.plist
-rw-r--r--  1 gkpln3  staff  2.7K Jun 12 16:10 Tweak.xm
-rw-r--r--  1 gkpln3  staff  224B Jun 11 16:17 control
drwxr-xr-x  3 gkpln3  staff  102B Jun 11 16:18 obj
drwxr-xr-x 16 gkpln3  staff  544B Jun 12 16:12 packages
```

iOS

Tweak.xm◦

o

```
%hook SBScreenShotter
- (void) saveScreenshot:(BOOL) screenshot
{
    %orig;
    NSLog(@"saveScreenshot: is called");
}
%end
```

```
%hook SBScreenShotter
- (void) saveScreenshot:(BOOL) screenshot
{
    NSLog(@"saveScreenshot: is called");
}
%end
```

o

CydiaSubstrate <https://riptutorial.com/zh-TW/ios/topic/10533/cydiastrate>

21: DispatchGroup

Grand Central Dispatch

Examples

◦ ◦ mainThread◦

DispatchGroup ◦

DispatchGroup enterleave◦

notify ◦

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        let dispatchGroup = DispatchGroup() //Create a group for the tasks.
        let session: URLSession = URLSession.shared

        dispatchGroup.enter() //Enter the group for the first task.

        let firstTask = session.dataTask(with: URLRequest(url: URL(string:
"https://stackoverflow.com"!)) { (data, response, error) in

            //Process Response..

            dispatchGroup.leave() //Leave the group for the first task.
        }

        dispatchGroup.enter() //Enter the group for the second task.

        let secondTask = session.dataTask(with: URLRequest(url: URL(string:
"https://google.ca"!)) { (data, response, error) in

            //Process Response..

            dispatchGroup.leave() //Leave the group for the second task.
        }

        //Get notified on the main thread/queue.. when ALL of the tasks above has been
        completed.
        dispatchGroup.notify(queue: DispatchQueue.main) {

            print("Every task is complete")

        }
    }
}
```

```

        //Start the tasks.
        firstTask.resume()
        secondTask.resume()
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
    }
}

```

wait° ° °

ordered°

```

import UIKit

//Locking mechanism..
func synchronized(_ lock: AnyObject, closure: () -> Void) {
    objc_sync_enter(lock)
    closure()
    objc_sync_exit(lock)
}

class ViewController: UIViewController {

    let lock = NSObject() //Object to lock on.
    var responseArray = Array<Data?>() //Array of responses.

    override func viewDidLoad() {
        super.viewDidLoad()

        let dispatchGroup = DispatchGroup()
        let session: URLSession = URLSession.shared

        dispatchGroup.enter() //Enter the group for the first task.

        let firstTask = session.dataTask(with: URLRequest(url: URL(string:
"https://stackoverflow.com")!)) { (data, response, error) in

            //Process Response..

            synchronized(self.lock, closure: { () -> Void in
                self.responseArray[0] = data ?? nil
            })

            dispatchGroup.leave() //Leave the group for the first task.
        }

        dispatchGroup.enter() //Enter the group for the second task.

        let secondTask = session.dataTask(with: URLRequest(url: URL(string:
"https://google.ca")!)) { (data, response, error) in

            //Process Response..

            synchronized(self.lock, closure: { () -> Void in

```

```

        self.responseArray[1] = data ?? nil
    })

    dispatchGroup.leave() //Leave the group for the second task.
}

//Get notified on the main thread.. when ALL of the requests above has been completed.
dispatchGroup.notify(queue: DispatchQueue.main) {

    print("Every task is complete..")

    for i in 0..

```

DispatchGroup◦ enterleave◦ ◦

enterleave◦

DispatchGroup <https://riptutorial.com/zh-TW/ios/topic/4624/dispatchgroup>

22: EventKit

Examples

◦ /◦

EventKit

```
import EventKit
```

Objective-C

```
#import <EventKit/EventKit.h>
```

EKEventStore

EKEventStore◦

```
let eventStore = EKEventStore()
```

Objective-C

```
EKEventStore *eventStore = [[EKEventStore alloc] init];
```

EKEventStore◦ ◦

◦ Not Determined◦

EKEventStore.authorizationStatusForEntityType()

```
switch EKEventStore.authorizationStatusForEntityType(EKEntityTypeEvent) {  
    case .Authorized: //...  
    case .Denied: //...  
    case .NotDetermined: //...  
    default: break  
}
```

Objective-C

```
switch ([EKEventStore authorizationStatusForEntityType:EKEntityTypeEvent]) {  
    case EKAuthorizationStatus.Authorized:
```

```
        //...
        break;
    case EKAuthorizationStatus.Denied:
        //...
        break;
    case EKAuthorizationStatus.NotDetermined:
        //...
        break;
    default:
        break;
}
```

NotDetermined

```
eventStore.requestAccessToEntityType(EKEntityTypeEvent, completion: { [weak self]
(userGrantedAccess, _) -> Void in
    if userGrantedAccess{
        //access calendar
    }
})
```

EKCalendarcalendarsForEntityType

```
let calendarsArray = eventStore.calendarsForEntityType(EKEntityType.Event) as! [EKCalendar]
```

for

```
for calendar in calendarsArray{
    //...
}
```

```
let calendarColor = UIColor(CGColor: calendar.CGColor)
let calendarTitle = calendar.title
```

Objective-C

```
UIColor *calendarColor = [UIColor initWithCGColor: calendar.CGColor];
NSString *calendarTitle = calendar.title;
```

```
var event = EKEvent(eventStore: eventStore)
```

Objective-C


```
EKEvent *event = [EKEvent initWithEventStore:eventStore];
```

```
event.calendar = calendar  
event.title = "Event Title"  
event.startDate = startDate //assuming startDate is a valid NSDate object  
event.endDate = endDate //assuming endDate is a valid NSDate object
```

```
try {  
    do eventStore.saveEvent(event, span: EKSpan.ThisEvent)  
} catch let error as NSError {  
    //error  
}
```

Objective-C

```
NSError *error;  
BOOL *result = [eventStore saveEvent:event span:EKSpanThisEvent error:&error];  
if (result == NO){  
    //error  
}
```

EventKit <https://riptutorial.com/zh-TW/ios/topic/5854/eventkit>

23: FacebookSDK

Examples

FacebookSDK

1SDK

CocoaPodsSDK ◦ ◦

Podfile

```
target 'MyApp' do
  use_frameworks!

  pod 'FBSDKCoreKit'
  pod 'FBSDKLoginKit'
  pod 'FBSDKShareKit'
end
```

pod install .xcworkspace.xcodeproj ◦

FBSDKLoginKitFBSDKShareKit ◦ ◦

2Facebook

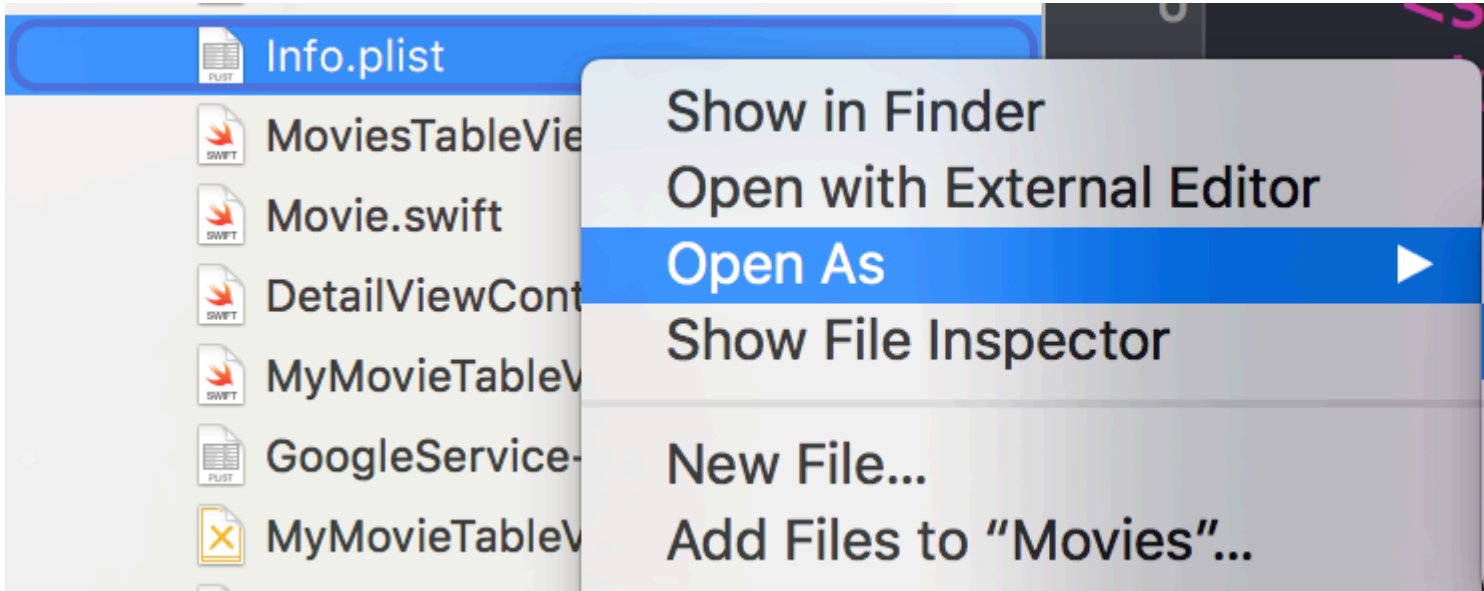
- [Facebook for Developers](#) ◦

FacebookSDK ◦ CocoaPodsSDK ◦

3.plist

◦ Facebook“” .plist ◦ Facebook“” ◦

.plist ◦



C ◦ ◦ <dict>◦

```

<plist version="1.0">
<dict>
  // ...
  //some default settings
  // ...
  <key>CFBundleURLTypes</key>
  <array>
    <dict>
      <key>CFBundleURLSchemes</key>
      <array>
        <string>fb{FBAppId}</string>
      </array>
    </dict>
  </array>
  <key>FacebookAppID</key>
  <string>{FBAppId}</string>
  <key>FacebookDisplayName</key>
  <string>{FBAppName}</string>
  <key>LSApplicationQueriesSchemes</key>
  <array>
    <string>fbapi</string>
    <string>fb-messenger-api</string>
    <string>fbauth2</string>
    <string>fbshareextension</string>
  </array>
  <key>NSAppTransportSecurity</key>
  <dict>
    <key>NSExceptionDomains</key>
    <dict>
      <key>facebook.com</key>
      <dict>
        <key>NSIncludesSubdomains</key>
        <true/>
        <key>NSEnforcementRequiresForwardSecrecy</key>
        <false/>
      </dict>
      <key>fbcdn.net</key>
      <dict>
        <key>NSIncludesSubdomains</key>

```

```

        <true/>
        <key>NSExceptionRequiresForwardSecrecy</key>
        <false/>
    </dict>
    <key>akamaihd.net</key>
    <dict>
        <key>NSIncludesSubdomains</key>
        <true/>
        <key>NSExceptionRequiresForwardSecrecy</key>
        <false/>
    </dict>
</dict>
</plist>

```

◦

4 Facebook

=>

5 AppDelegate.swift

◦

```
import FBSDKCoreKit
```

```

func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[NSObject: AnyObject]?) -> Bool {
    FBSDKApplicationDelegate.sharedInstance().application(application,
didFinishLaunchingWithOptions: launchOptions)
    return true
}

func application(application: UIApplication, openURL url: NSURL, sourceApplication: String?,
annotation: AnyObject) -> Bool {
    return FBSDKApplicationDelegate.sharedInstance().application(application, openURL: url,
sourceApplication: sourceApplication, annotation: annotation)
}

```

“Facebook”

“Facebook”FacebookSDKUI

1. UIButton
2. CtrlIBAction
3. IBActionFacebook

```

let loginButton = FBSDKLoginButton()
loginButton.delegate = self
// Your Custom Permissions Array
loginButton.readPermissions =
[
    "public_profile",

```

```

        "email",
        "user_about_me",
        "user_photos"
    ]
    // Hiding the button
    loginButton.hidden = true
    self.view.addSubview(loginButton)
    // Simulating a tap for the actual Facebook SDK button
    loginButton.sendActionsForControlEvents(UIControlEvents.TouchUpInside)

```

Objective-C

```

FBSDKLoginButton *FBButton = [FBSDKLoginButton new];

// Your Custom Permissions Array
FBButton.readPermissions = @[@"public_profile",
                             @"email",
                             @"user_about_me",
                             @"user_photos"
                             ];
FBButton.loginBehavior = FBSDKLoginBehaviorNative;
[FBButton setDelegate:self];
[FBButton setHidden:true];
[loginButton addSubview:FBButton];

[FBButton sendActionsForControlEvents:UIControlEventTouchUpInside];

```

o

facebook

Facebook

```

enum FacebookParametesField : String
{
    case FIELDS_KEY = "fields"
    case FIELDS_VALUE = "id, email, picture, first_name, last_name"
}

if FBSDKAccessToken.currentAccessToken() != nil
{
    // Getting user facebook data
    FBSDKGraphRequest(graphPath: "me",
                     parameters: [FacebookParametesField.FIELDS_KEY.rawValue :
FacebookParametesField.FIELDS_VALUE.rawValue])
    .startWithCompletionHandler({ (graphConnection : FBSDKGraphRequestConnection!, result :
AnyObject!, error : NSError!) -> Void in

        if error == nil
        {
            print("Facebook Graph phaze")

            let email = result["email"]
            let facebookToken = FBSDKAccessToken.currentAccessToken().tokenString
            let userFacebookId = result["id"]
            let firstName = result["first_name"]

```

```
let lastName = result["last_name"]

if let result = result as? Dictionary<String, AnyObject>
{
    if let picture = result["picture"] as? Dictionary<String,AnyObject>
    {
        if let data = picture["data"] as? Dictionary <String,AnyObject>
        {
            if let url = data["url"] as? String
            {
                // Profile picture URL
                let profilePictureURL = url
            }
        }
    }
}
})
}
```

FacebookSDK <https://riptutorial.com/zh-TW/ios/topic/2972/facebooksdk>

24: GameCenter

Examples

GameCenter

1. Apple
2. iTunesConnectGameCenter

GameCenter

1. *iTunesConnect*
2. ◦ ◦
- 3.
4. ◦
5. ◦
6. ◦
7. *ID* ◦
8. *Integer*
- 9.
10. ◦

LeaderboardID Xcode◦

Xcode

4◦

- 1.
2. GameCenter
3. GameCenter
- 4.
5. GameKit import GameKit Protocols GKGameCenterControllerDelegate
6. GameCenter

```
func authenticateLocalPlayer() {  
  
    let localPlayer = GKLocalPlayer.localPlayer()  
    localPlayer.authenticateHandler = { (viewController, error) -> Void in  
  
        if viewController != nil {  
            //If the user is not signed in to GameCenter, we make them sign in  
            let vc:UIViewController = self.view!.window!.rootViewController!  
            vc.presentViewController(viewController!, animated: true, completion: nil)  
        }  
    }  
}
```

```

        } else {

            //Do something here if you want
        }
    }
}

```

3. ◦

2. ◦

Identifier `iTunesConnectleaderboardID`.

score `Int` iTunesConnect

```

func saveHighScore(identifier:String, score:Int) {

    if GKLocalPlayer.localPlayer().authenticated {

        let scoreReporter = GKScore(leaderboardIdentifier: identifier)

        scoreReporter.value = Int64(score)

        let scoreArray:[GKScore] = [scoreReporter]

        GKScore.reportScores(scoreArray, withCompletionHandler: {
            error -> Void in

            if error != nil {
                print("Error")
            } else {

            }

        })
    }
}

```

4. ◦

```

//This function will show GameCenter leaderboards and Achievements if you call this function.
func showGameCenter() {

    let gameCenterViewController = GKGameCenterViewController()
    gameCenterViewController.gameCenterDelegate = self

    let vc:UIViewController = self.view!.window!.rootViewController!
    vc.presentViewController(gameCenterViewController, animated: true, completion:nil)

}

//This function closes gameCenter after showing.
func gameCenterViewControllerDidFinish(gameCenterViewController:
GKGameCenterViewController) {

    gameCenterViewController.dismissViewControllerAnimated(true, completion: nil)
    self.gameCenterAchievements.removeAll()
}

```



```
}
```

GameCenter <https://riptutorial.com/zh-TW/ios/topic/6720/gamecenter>

25: GameplayKit

Examples

GameplayKit iOS 9 SDK。

```
GKRandomSource.sharedRandomGKRandomSource。
```

- **GKARC4RandomSource**ARC4
- **GKLinearCongruentialRandomSource**GKRandomSource
- **GKMersenneTwisterRandomSource**MersenneTwister。。

```
GKRandomSource.nextInt()。 nextBool() -> Bool.nextUniform() -> Float
```

GameplayKit

```
import GameplayKit
```

Objective-C

```
#import <GameplayKit/GameplayKit.h>
```

```
let randomNumber = GKRandomSource.sharedRandom().nextInt()
```

Objective-C

```
int randomNumber = [[GKRandomSource sharedRandom] nextInt];
```

```
nextInt-2,147,483,6482,147,483,647。
```

On

```
nextIntWithUpperBound()
```

```
let randomNumber = GKRandomSource.sharedRandom().nextInt(upperBound: 10)
```

Objective-C

```
int randomNumber = [[GKRandomSource sharedRandom] nextIntWithUpperBound: 10];
```

010。

mn

GKRandomDistribution GKRandomSource ◦ GKRandomDistribution GKGaussianDistribution
GKShuffledDistribution ◦

GKRandomSource GKRandom ◦

```
let randomizer = GKRandomDistribution(randomSource: GKRandomSource(), lowestValue: 0,  
highestValue: 6)  
let randomNumberInBounds = randomizer.nextInt()
```

Objective-C

```
int randomNumber = [[GKRandomSource sharedRandom] nextIntWithUpperBound: n - m] + m;
```

310

```
let randomNumber = GKRandomSource.sharedRandom().nextInt(upperBound: 7) + 3
```

Objective-C

```
int randomNumber = [[GKRandomSource sharedRandom] nextIntWithUpperBound: 7] + 3;
```

GKEntity GKComponent

◦ ◦ **apple** GKEntity GKComponent ◦

```
class Player: GKEntity{}  
class PlayerSpriteComponent: GKComponent {}
```

GKEntity

◦

GKEntity ◦

◦ GKEntity GKEntity

```
let otherComponent = PlayerSpriteComponent()
var player = Player()
player.addComponent(PlayerSpriteComponent())
player.addComponent(otherComponent)
print(player.components.count) //will print 1
print(player.components[0] === otherComponent) // will print true
```

◦ component (for: T.Type) ◦

```
let component = player.component(ofType: PlayerSpriteComponent.self)
```

update◦

```
var player = Player()
player.addComponent(PlayerSpriteComponent())
player.update(deltaTime: 1.0) // will call the update method of the PlayerSpriteComponent
added to it
```

GKComponent

◦

◦ ◦

```
class PlayerSpriteComponent: GKComponent {
    override func update(deltaTime seconds: TimeInterval) {
        //move the sprite depending on the update time
    }
}
```

didAddToEntitywillRemoveFromEntity◦

GKEntity◦

```
override func update(deltaTime seconds: TimeInterval) {
    let controller = self.entity?.component(ofType: PlayerControlComponent.self)
    //call methods on the controller
}
```

◦

GKComponentSystem

GKEntityGKComponentsGKComponents GKComponentSystem ◦

◦

GKComponentSystem ◦

```
let system = GKComponentSystem(componentClass: PlayerSpriteComponent.self)
```

add

```
system.addComponent(PlayerSpriteComponent())
```

GKComponentSystem ◦

```
system.addComponent(foundIn: player)
```

```
system.update(deltaTime: delta)
```

GKComponentSystem GKComponentSystem ◦

GameplayKit <https://riptutorial.com/zh-TW/ios/topic/4966/gameplaykit>

26: GCD Grand Central Dispatch

Grand Central Dispatch GCD Apple CPU.

Examples

dispatch_queue_create

Objective-C

```
dispatch_queue_t queue = dispatch_queue_create("com.example.myqueue", DISPATCH_QUEUE_SERIAL);
```

```
// Before Swift 3
let queue = dispatch_queue_create("com.example.myqueue", DISPATCH_QUEUE_SERIAL)
// Swift 3
let queue = DispatchQueue(label: "com.example.myqueue") //default is serial queue, unless
.concurrent is specified as an attribute otherwise
```

UIUI.

NSURLSessionUI

synchronous asynchronous ◦ synchronously ◦ Asynchronous ◦

-C

Synchronous

```
dispatch_queue_t queue = dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
```

Asynchronous

```
dispatch_async(dispatch_get_main_queue(), ^{
    // do work here to Usually to update the User Interface
});
```

SWIFT 3

Asynchronous

```
DispatchQueue.main.async {
}
```

Synchronous

```
DispatchQueue.main.sync {
}
```

DispatchGroup

webservice Web

3

```
func doLongTasksAndWait () {
    print("starting long running tasks")
    let group = DispatchGroup() //create a group for a bunch of tasks we are about to
do
    for i in 0...3 { //launch a bunch of tasks (eg a bunch of webservice
calls that all need to be finished before proceeding to the next ViewController)
        group.enter() //let the group know that something is being added
        DispatchQueue.global().async { //run tasks on a background thread
            sleep(arc4random() % 4) //do some long task eg webservice or database lookup
(here we are just sleeping for a random amount of time for demonstration purposes)
            print("long task \(i) done!")
            group.leave() //let group know that the task is finished
        }
    }
    group.wait() //will block whatever thread we are on here until all
the above tasks have finished (so maybe dont use this function on your main thread)
    print("all tasks done!")
}
```

notifygroup.wait()

```
group.notify(queue: DispatchQueue.main) { //the queue: parameter is which queue this block
will run on, if you need to do UI updates, use the main queue
    print("all tasks done!") //this will execute when all tasks have left the
group
}
```

```
starting long running tasks
long task 0 done!
long task 3 done!
long task 1 done!
long task 2 done!
all tasks done!
```

Apple Docs

DispatchSemaphore

// O

3

```
func do2TasksAtATime () {
    print("starting long running tasks (2 at a time)")
    let sem = DispatchSemaphore(value: 2) //this semaphore only allows 2 tasks to
run at the same time (the resource count)
    for i in 0...7 { //launch a bunch of tasks
        DispatchQueue.global().async { //run tasks on a background thread
            sem.wait() //wait here if no resources available
        }
    }
}
```

```

        sleep(2) //do some long task eg file access (here
we are just sleeping for a 2 seconds for demonstration purposes)
        print("long task \(i) done! \(Date())")
        sem.signal() //let the semaphore know this resource is
now available
    }
}
}

```

:(

```

starting long running tasks (2 at a time)
long task 0 done! 2017-02-16 07:11:53 +0000
long task 1 done! 2017-02-16 07:11:53 +0000
long task 2 done! 2017-02-16 07:11:55 +0000
long task 3 done! 2017-02-16 07:11:55 +0000
long task 5 done! 2017-02-16 07:11:57 +0000
long task 4 done! 2017-02-16 07:11:57 +0000
long task 6 done! 2017-02-16 07:11:59 +0000
long task 7 done! 2017-02-16 07:11:59 +0000

```

Apple Docs

3

```

func serialQueues () {
    let serialQueue = DispatchQueue(label: "com.example.serial") //default queue type is a
serial queue
    let start = Date ()
    for i in 0...3 { //launch a bunch of tasks
        serialQueue.async { //run tasks on a background
thread, using our serial queue
            sleep(2) //do some long task eg
webservice or database lookup
            let timeTaken = Date().timeIntervalSince(start)
            print("serial long task \(i) done! total time taken: \(timeTaken)")
        }
    }
}

```

```

serial long task 0 done! total time taken: 2.07241100072861
serial long task 1 done! total time taken: 4.16347700357437
serial long task 2 done! total time taken: 6.23209798336029
serial long task 3 done! total time taken: 8.30682599544525

```

```

func concurrentQueues () {
    let concurrentQueue = DispatchQueue(label: "com.example.concurrent", attributes:
.concurrent) //explicitly specify the queue to be a concurrent queue
    let start = Date ()
    for i in 0...3 { //launch a bunch of tasks
        concurrentQueue.async { //run tasks on a background thread, using our concurrent queue
            sleep(2) //do some long task eg webservice or database lookup
            let timeTaken = Date().timeIntervalSince(start)
            print("concurrent long task \(i) done! total time taken: \(timeTaken)")
        }
    }
}

```



```
}
```

```
concurrent long task 3 done! total time taken: 2.07092100381851  
concurrent long task 0 done! total time taken: 2.07087397575378  
concurrent long task 2 done! total time taken: 2.07086700201035  
concurrent long task 1 done! total time taken: 2.07089096307755
```

◦ ◦ ;◦

◦

GCDGrand Central Dispatch <https://riptutorial.com/zh-TW/ios/topic/4626/gcd-grand-central-dispatch->

27: Healthkit

Examples

HealthKit

Objective-C

Target->CapabilitiesHealthKit ◦ info.plist◦

NSObjectCocoaClassGSHealthKitManager

GSHealthKitManager.h

```
#import <Foundation/Foundation.h>
#import <HealthKit/HealthKit.h>
@interface GSHealthKitManager : NSObject

+ (GSHealthKitManager *)sharedManager;

- (void)requestAuthorization;

- (NSDate *)readBirthDate;
- (void)writeWeightSample:(double)weight;
- (NSString *)readGender;

@end
```

GSHealthKitManager.m

```
#import "GSHealthKitManager.h"
#import <HealthKit/HealthKit.h>

@interface GSHealthKitManager ()

@property (nonatomic, retain) HKHealthStore *healthStore;

@end

@implementation GSHealthKitManager

+ (GSHealthKitManager *)sharedManager {
    static dispatch_once_t pred = 0;
    static GSHealthKitManager *instance = nil;
    dispatch_once(&pred, ^{
        instance = [[GSHealthKitManager alloc] init];
        instance.healthStore = [[HKHealthStore alloc] init];
    });
    return instance;
}

- (void)requestAuthorization {
```

```

if ([HKHealthStore isHealthDataAvailable] == NO) {
    // If our device doesn't support HealthKit -> return.
    return;
}

NSArray *readTypes = @[HKObjectType
characteristicTypeForIdentifier:HKCharacteristicTypeIdentifierDateOfBirth],[HKObjectType
characteristicTypeForIdentifier:HKCharacteristicTypeIdentifierBiologicalSex]];

[self.healthStore requestAuthorizationToShareTypes:nil readTypes:[NSSet
setWithArray:readTypes] completion:nil];
}

- (NSDate *)readBirthDate {
    NSError *error;
    NSDate *dateOfBirth = [self.healthStore dateOfBirthWithError:&error];    // Convenience
method of HKHealthStore to get date of birth directly.

    if (!dateOfBirth) {
        NSLog(@"Either an error occured fetching the user's age information or none has been
stored yet. In your app, try to handle this gracefully.");
    }

    return dateOfBirth;
}

- (NSString *)readGender
{
    NSError *error;
    HKBiologicalSexObject *gen=[self.healthStore biologicalSexWithError:&error];
    if (gen.biologicalSex==HKBiologicalSexMale)
    {
        return(@"Male");
    }
    else if (gen.biologicalSex==HKBiologicalSexFemale)
    {
        return(@"Female");
    }
    else if (gen.biologicalSex==HKBiologicalSexOther)
    {
        return(@"Other");
    }
    else{
        return(@"Not Set");
    }
}

@end

```

ViewController

```

- (IBAction)pressed:(id)sender {

    [[GSHealthKitManager sharedManager] requestAuthorization];
    NSDate *birthDate = [[GSHealthKitManager sharedManager] readBirthDate];
    NSLog(@"birthdate %@", birthDate);
}

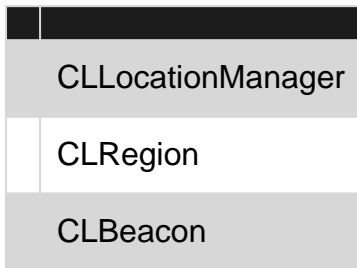
```

```
NSLog(@"gender 2131321 %@", [[GSHealthKitManager sharedInstance] readGender]);  
  
}
```

```
2016-10-13 14:41:39.568 random[778:26371] birthdate 1992-11-29 18:30:00 +0000  
2016-10-13 14:41:39.570 random[778:26371] gender 2131321 Male
```

Healthkit <https://riptutorial.com/zh-TW/ios/topic/7412/healthkit>

28: iBeacon



◦ iBeaconApple ◦ 3

1. UUID
- 2.
- 3.

CLLocationiBeaconsUUID ◦ UUID ◦

CLLocation ◦

Examples

iBeacon

- 1.

```
func initiateRegion(ref:BeaconHandler){
    let uuid: NSUUID = NSUUID(UUIDString: "<UUID>")
    let beacon = CLBeaconRegion(proximityUUID: uuid, identifier: "")
    locationManager?.requestAlwaysAuthorization() //CLLocation manager obj.
    beacon?.notifyOnEntry = true
    beacon?.notifyOnExit = true
    beacon?.notifyEntryStateOnDisplay = true
    locationManager?.startMonitoringForRegion(beacon!)
    locationManager?.delegate = self;
    // Check if beacon monitoring is available for this device
    if (!CLLocationManager.isMonitoringAvailableForClass(CLBeaconRegion)) {
        print("error")
    }
    locationManager!.startRangingBeaconsInRegion(self.beacon!)
}
```

- 2.

```
func locationManager(manager: CLLocationManager, didEnterRegion region: CLRegion) {
    if(region.isKindOfClass(CLBeaconRegion)) {
        locationManager!.startRangingBeaconsInRegion(self.beacon!)
    }
}

func locationManager(manager: CLLocationManager, didExitRegion region: CLRegion) {
    if(region.isKindOfClass(CLBeaconRegion)) {
```

```
        locationManager!.stopRangingBeaconsInRegion(self.beacon!)
    }
}
```

3.

```
func locationManager(manager: CLLocationManager, didRangeBeacons beacons: [CLBeacon], inRegion
region: CLBeaconRegion) {
    print (beacons.first.major)
}
```

```
    beacon = CLBeaconRegion(proximityUUID: <#NSUUID#>, major: <#CLBeaconMajorValue#>, identifier:
<#String#>) // listening to all beacons with given UUID and major value
    beacon = CLBeaconRegion(proximityUUID: <##NSUUID#>, major: <##CLBeaconMajorValue#>, minor:
<##CLBeaconMinorValue#>, identifier: <##String#>) // listening to all beacons with given UUID
and major and minor value
```

iBeacons

```
let locationManager = CLLocationManager()
locationManager.delegate = self
locationManager.requestWhenInUseAuthorization()
// OR locationManager.requestAlwaysAuthorization()
```

didRangeBeacons iBeacons

```
func locationManager(manager: CLLocationManager, didRangeBeacons beacons: [CLBeacon], inRegion
region: CLBeaconRegion) {
    for beacon in beacons {
        print (beacon.major)
        print (beacon.minor)
    }
}
```

iBeacon <https://riptutorial.com/zh-TW/ios/topic/1958/ibeacon>

29: IBOutlets

IBOutletInterface Builder。 Objective-C。

Swift。

```
<UIKit/UINibDeclarations.h>
```

```
#ifndef IBOutlet
#define IBOutlet
#endif
```

Examples

UIIBOutlet

IBOutletsUIViewController。 。 .IBOutlet""。 UIViewController

```
//ViewController.h
#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

//This is the declaration of the outlet
@property (nonatomic, weak) IBOutlet UILabel *myLabel;

@end

//ViewController.m
#import "ViewController.h"

@implementation ViewController

@synthesize myLabel;

-(void) viewDidLoad {

    [super viewDidLoad];
    //Editing the properties of the outlet
    myLabel.text = @"TextHere";

}

@end
```

```
import UIKit
class ViewController: UIViewController {
    //This is the declaration of the outlet
    @IBOutlet weak var myLabel: UILabel!

    override func viewDidLoad() {
        super.viewDidLoad()
```

```
    //Editing the properties of the outlet
    myLabel.text = "TextHere"
}
}
```

.h ◦ ◦

IBOutlets <https://riptutorial.com/zh-TW/ios/topic/4713/iboutlets>

30: iOS - Robbie Hanson XMPP

Examples

iOS XMPP Robbie Hanson Openfire

SRXMPPDemo

- <https://github.com/SahebRoy92/SRXMPPDemo>

Objective CXMPP. XMPP "" xmpp. -

SRXMPP - Singleton.

-
-
- Robbie Hanson XMLCore Data vCard.
- //

1. Openfire - openfire.

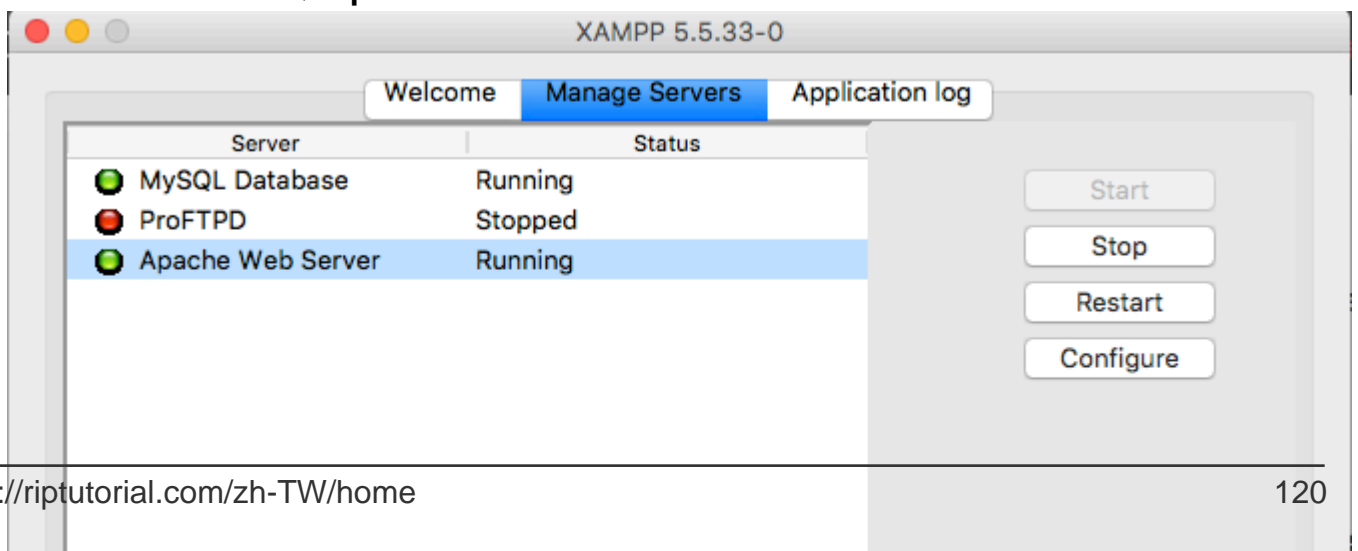
2. - 3

• Java -

- Java for Mac.

XAMPP -

- XAMPP.
- XAMPP Database SQL Apache Server .



- URL [[http:// localhost / phpmyadmin /](http://localhost/phpmyadmin/)]
- ◦ DB◦
- **DBChatDB**

C◦ Openfire -

- Openfire“Openfire”



- URL - [[http:// localhost9090 / setup / index.jsp](http://localhost9090/setup/index.jsp)]
- ◦ >
- ◦ >
- ◦ “”
- ◦ - “◦ **ChatDB** ◦
- ◦ * “MySQL”◦ JDBC Driver Class◦ URL◦ XAMPPHostname“localhost” “ChatDB” DB◦ ◦

Database Settings - Standard Connection

Specify a JDBC driver and connection properties to connect to your database. If you need more information about this pro

Note: Database scripts for most popular databases are included in the server distribution at [`Openfire_HOME`]/resc

Database Driver Presets: MySQL

JDBC Driver Class: ?

Database URL: ?

Username: ?

Password: ?

Minimum Connections: ?

Maximum Connections: ?

- - ◦ Openfire◦
-

◦

--**SRXMPP.m** NSString extern **SRXMPP_Hostname**

- OpenFireIP
- - **“localhost”** ◦

◦

XMPPXMPP◦

XMPP - <https://xmpp.org/rfc/rfc3920.html>

- 1.
- 2.

◦

iOS - Robbie HansonXMPP <https://riptutorial.com/zh-TW/ios/topic/1475/ios----robbie-hansonxmpp>

31: iOS 10API

Examples

```
//import Speech
//import AVFoundation

// create a text field to show speech output
@IBOutlet weak var transcriptionTextField: UITextView!
// we need this audio player to play audio
var audioPlayer: AVAudioPlayer!

override func viewDidLoad()
{
    super.viewDidLoad()
}

// this function is required to stop audio on audio completion otherwise it will play same
audio again and again
func audioPlayerDidFinishPlaying(_ player: AVAudioPlayer, successfully flag: Bool)
{
    player.stop()
}

// this function is required to get a speech recognizer and after that make and request to
speech recognizer
func requestSpeechAuth()
{
    SFSpeechRecognizer.requestAuthorization { authStatus in
        if authStatus == SFSpeechRecognizerAuthorizationStatus.authorized {
            if let path = Bundle.main.url(forResource: "mpthreetest", withExtension: "m4a") {
                do {
                    let sound = try AVAudioPlayer(contentsOf: path)
                    self.audioPlayer = sound
                    self.audioPlayer.delegate = self
                    sound.play()
                } catch {
                    print("error")
                }
            }

            let recognizer = SFSpeechRecognizer()
            let request = SFSpeechURLRecognitionRequest(url:path)
            recognizer?.recognitionTask(with: request) { (result, error) in
                if let error = error {
                    print("there is a error\(error)")
                } else {
                    // here you are printing out the audio output basically showing it on uitext field
                    self.transcriptionTextField.text =
                    result?.bestTranscription.formattedString
                }
            }
        }
    }
}

// here you are calling requestSpeechAuth function on UIButton press
```

```
@IBAction func playButtonPress(_ sender: AnyObject)
{
    requestSpeechAuth()
}
```

iOS 10API <https://riptutorial.com/zh-TW/ios/topic/5986/ios-10api>

32: iOS Google Places API

Examples

1. pod
2. GooglePlaces SDK
- 3.

◦

1. GooglePlacesGooglePlacePicker

```
import GooglePlaces
import GooglePlacePicker
```

2. CLLocationManagerDelegate

```
class ViewController: UIViewController, CLLocationManagerDelegate {
}
}
```

3. CLLocationManager

```
var currentLocation = CLLocationManager()
```

4.

```
currentLocation = CLLocationManager()
currentLocation.requestAlwaysAuthorization()
```

5. GooglePlacePicker

@IBAction func placePickerAction(sender: AnyObject){

```
if CLLocationManager.authorizationStatus() == .AuthorizedAlways {

    let center =
CLLocationCoordinate2DMake((currentLocation.location?.coordinate.latitude)!,
(currentLocation.location?.coordinate.longitude)!)
    let northEast = CLLocationCoordinate2DMake(center.latitude + 0.001, center.longitude +
0.001)
    let southWest = CLLocationCoordinate2DMake(center.latitude - 0.001, center.longitude -
0.001)
    let viewport = GMSCoordinateBounds(coordinate: northEast, coordinate: southWest)
    let config = GMSPlacePickerConfig(viewport: viewport)
    placePicker = GMSPlacePicker(config: config)

    placePicker?.pickPlaceWithCallback({ (place: GMSPlace?, error: NSError?) -> Void in
        if let error = error {
            print("Pick Place error: \(error.localizedDescription)")
            return
        }
    })
}
```

```
    }

    if let place = place {
        print("Place name: \(place.name)")
        print("Address: \(place.formattedAddress)")
    } else {
        print("Place name: nil")
        print("Address: nil")
    }
})
}
```

iOS Google Places API <https://riptutorial.com/zh-TW/ios/topic/6908/ios-google-places-api>

33: iOS TTS

iOS

Examples

C.

```
AVSpeechSynthesizer *synthesizer = [[AVSpeechSynthesizer alloc] init];
AVSpeechUtterance *utterance = [AVSpeechUtterance speechUtteranceWithString:@"Some text"];
[utterance setRate:0.2f];
[synthesizer speakUtterance:utterance];
```

```
let synthesizer = AVSpeechSynthesizer()
let utterance = AVSpeechUtterance(string: "Some text")
utterance.rate = 0.2
```

```
utterance.voice = AVSpeechSynthesisVoice(language: "fr-FR")
```

- **Swift 2** synthesizer.speakUtterance(utterance)
- **Swift 3** synthesizer.speak(utterance)

AVFoundation

```
- (BOOL) pauseSpeakingAtBoundary: (AVSpeechBoundary) boundary;
- (BOOL) stopSpeakingAtBoundary: (AVSpeechBoundary) boundary;
```

AVSpeechBoundary AVSpeechBoundaryImmediate AVSpeechBoundaryWord ◦

iOS TTS <https://riptutorial.com/zh-TW/ios/topic/8909/ios-tts>

34: iOS AirPrint

Examples

AirPrint

Objective-C

ViewController.h

```
@interface ViewController : UIViewController <UIPrintInteractionControllerDelegate> {
    UISimpleTextPrintFormatter *_textFormatter;
}
```

ViewController.m

```
#define DefaultFontSize 48
#define PaddingFactor 0.1f
```

```
-(IBAction)print:(id)sender;
{
    /* Get the UIPrintInteractionController, which is a shared object */
    UIPrintInteractionController *controller = [UIPrintInteractionController
sharedPrintController];
    if(!controller){
        NSLog(@"Couldn't get shared UIPrintInteractionController!");
        return;
    }

    /* Set this object as delegate so you can use the
printInteractionController:cutLengthForPaper: delegate */
    controller.delegate = self;

    UIPrintInfo *printInfo = [UIPrintInfo printInfo];
    printInfo.outputType = UIPrintInfoOutputGeneral;

    /* Use landscape orientation for a banner so the text print along the long side of the
paper. */
    printInfo.orientation = UIPrintInfoOrientationLandscape;

    printInfo.jobName = self.textField.text;
    controller.printInfo = printInfo;

    /* Create the UISimpleTextPrintFormatter with the text supplied by the user in the text
field */
    _textFormatter = [[UISimpleTextPrintFormatter alloc] initWithText:self.textField.text];

    /* Set the text formatter's color and font properties based on what the user chose */
    _textFormatter.color = [self chosenColor];
    _textFormatter.font = [self chosenFontWithSize:DefaultFontSize];

    /* Set this UISimpleTextPrintFormatter on the controller */
```

```

controller.printFormatter = _textFormatter;

/* Set up a completion handler block.  If the print job has an error before spooling, this
is where it's handled. */
void (^completionHandler)(UIPrintInteractionController *, BOOL, NSError *) =
^(UIPrintInteractionController *printController, BOOL completed, NSError *error) {
    if(completed && error)
        NSLog( @"Printing failed due to error in domain %@ with error code %lu. Localized
description: %@, and failure reason: %@", error.domain, (long)error.code,
error.localizedDescription, error.localizedFailureReason );
    };

    if (UI_USER_INTERFACE_IDIOM() == UIUserInterfaceIdiomPad)
        [controller presentFromRect:self.printButton.frame inView:self.view animated:YES
completionHandler:completionHandler];
    else
        [controller presentAnimated:YES completionHandler:completionHandler]; // iPhone
}

```

```

- (CGFloat)printInteractionController:(UIPrintInteractionController
*)printInteractionController cutLengthForPaper:(UIPrintPaper *)paper {

    /* Create a font with arbitrary size so that you can calculate the approximate
font points per screen point for the height of the text. */
    UIFont *font = _textFormatter.font;
    CGSize size = [self.textField.text sizeWithAttributes:@{NSFontAttributeName: font}];

    float approximateFontPointPerScreenPoint = font.pointSize / size.height;

    /* Create a new font using a size that will fill the width of the paper */
    font = [self chosenFontWithSize: paper.printableRect.size.width *
approximateFontPointPerScreenPoint];

    /* Calculate the height and width of the text with the final font size */
    CGSize finalTextSize = [self.textField.text sizeWithAttributes:@{NSFontAttributeName:
font}];

    /* Set the UISimpleTextFormatter font to the font with the size calculated */
    _textFormatter.font = font;

    /* Calculate the margins of the roll. Roll printers may have unprintable areas
before and after the cut. We must add this to our cut length to ensure the
printable area has enough room for our text. */
    CGFloat lengthOfMargins = paper.paperSize.height - paper.printableRect.size.height;

    /* The cut length is the width of the text, plus margins, plus some padding */
    return finalTextSize.width + lengthOfMargins + paper.printableRect.size.width *
PaddingFactor;
}

```

iOS AirPrint <https://riptutorial.com/zh-TW/ios/topic/7395/iosairprint>

35: iOSSpotLight

Examples

Objective-C

1. `iOSCoreSpotlightMobileCoreServices`



General

Capabilities

PROJECT



CoreSpotlighSample

TARGETS



CoreSpotlighSample



CoreSpotlighSampl...



CoreSpotlighSampl...



▶ **Target Dependencies (0 it**

▶ **Compile Sources (3 item...**

▼ **Link Binary With Libraries**

Name



CoreSp



Mobile



▶ **Copy Bundle Resources (4**

2. CSSearchableItemuniqueIdentifierdomainIdentifierattributeSet. [[CSSearchableIndex defaultSearchableIndex] ...]CSSearchableItem.

[[CSSearchableIndex defaultSearchableIndex] ...]CSSearchableItem。

3.

36: iOS

Apple

Examples

URL

URLtodolist://

Objective-C

```
NSURL *myURL = [NSURL URLWithString:@"todolist://there/is/something/to/do"];  
[[UIApplication sharedApplication] openURL:myURL];
```

```
let urlString = "todolist://there/is/something/to/do"  
if let url = NSURL(string: urlString) {  
    UIApplication.shared().openURL(url)  
}
```

HTML

```
<a href="todolist://there/is/something/to/do">New SMS Message</a>
```

◦ canOpenURL:◦

URL

MyTasksURL◦ URL

mytasks://create?title=hello&body=world

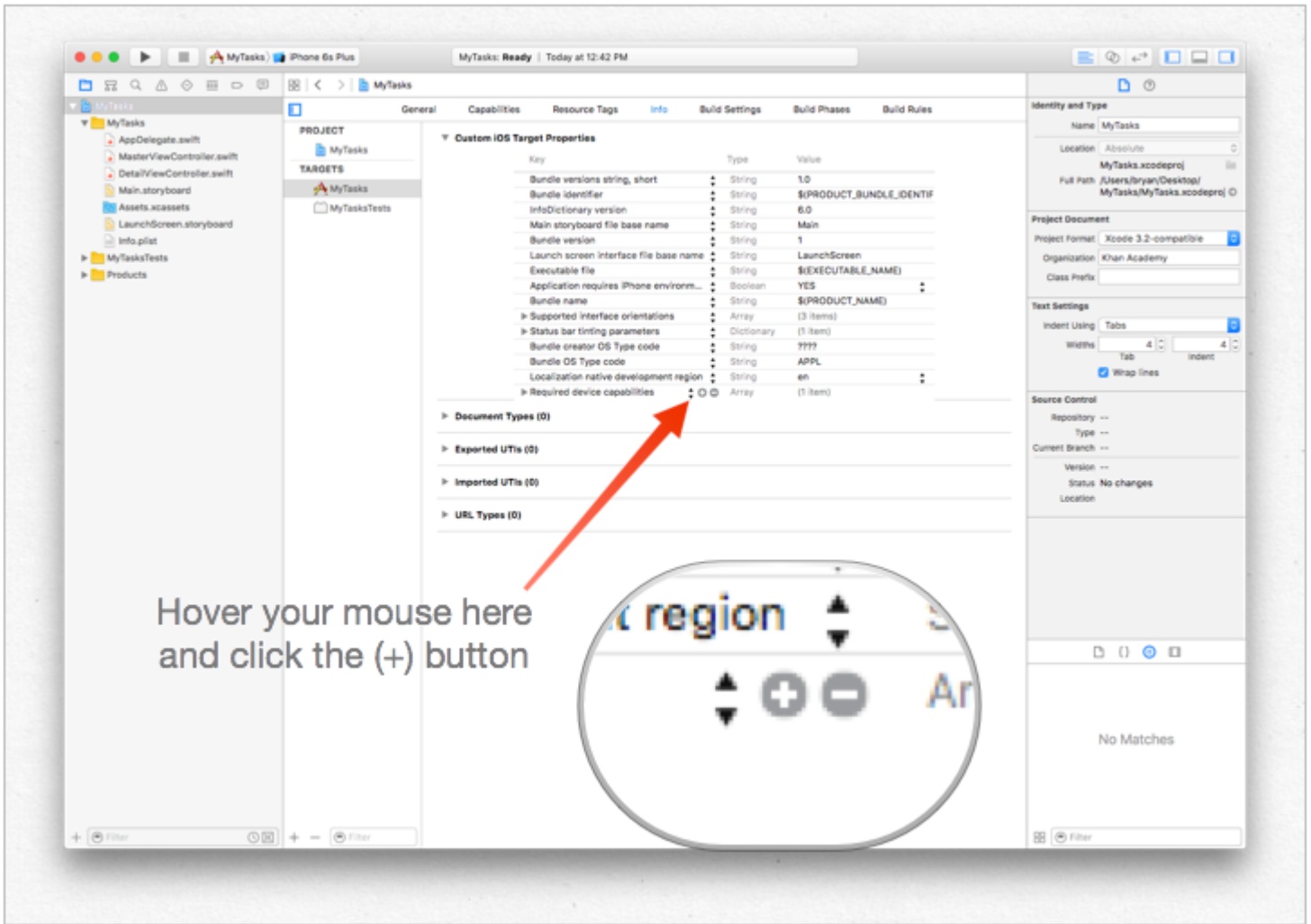
tbody

URL

1. Info.plistURLURL◦
2. UIApplicationDelegateURL◦
3. URL◦

Info.plistURL

Info.plist“URL”◦ +



Hover your mouse here and click the (+) button

▼ URL types	▲	Array	(5 items)
▼ Item 0	▲	Dictionary	(2 items)
URL identifier	▲	String	com.mycompany
▼ URL Schemes	▲	Array	(1 item)
Item 0	⊕ ⊖	String	mytasks
▼ Item 1	▲	Dictionary	(1 item)

... URL.

UIApplicationDelegateURL

UIApplicationDelegateapplication:openURL:options: .URL

```
func application(app: UIApplication, openURL url: NSURL, options: [String : AnyObject]) -> Bool {
    if url.scheme == "mytasks" && url.host == "create" {
        let title = // get the title out of the URL's query using a method of your choice
        let body = // get the title out of the URL's query using a method of your choice
        self.rootViewController.createTaskWithTitle(title, body: body)
        return true
    }

    return false
}
```


URL。

URL。 -

```
self.rootViewController.createTaskWithTitle(:body:) - AppDelegate
```

◦ ◦
◦

1. DeepLinkPOC。

2. ◦

3. ""。

4. URL

5. "+"。

6. URL。 URL“ DeepLinking ”。

Safari“DeepLinking//”。

```
[scheme]://[host]/[path] --> DeepLinking://path/Page1
```

Scheme“DeepLinking”“Page1”

◦ ◦

7. appDelegate。

```
func application(application: UIApplication, openURL url: NSURL, sourceApplication: String?,  
annotation: AnyObject) -> Bool
```

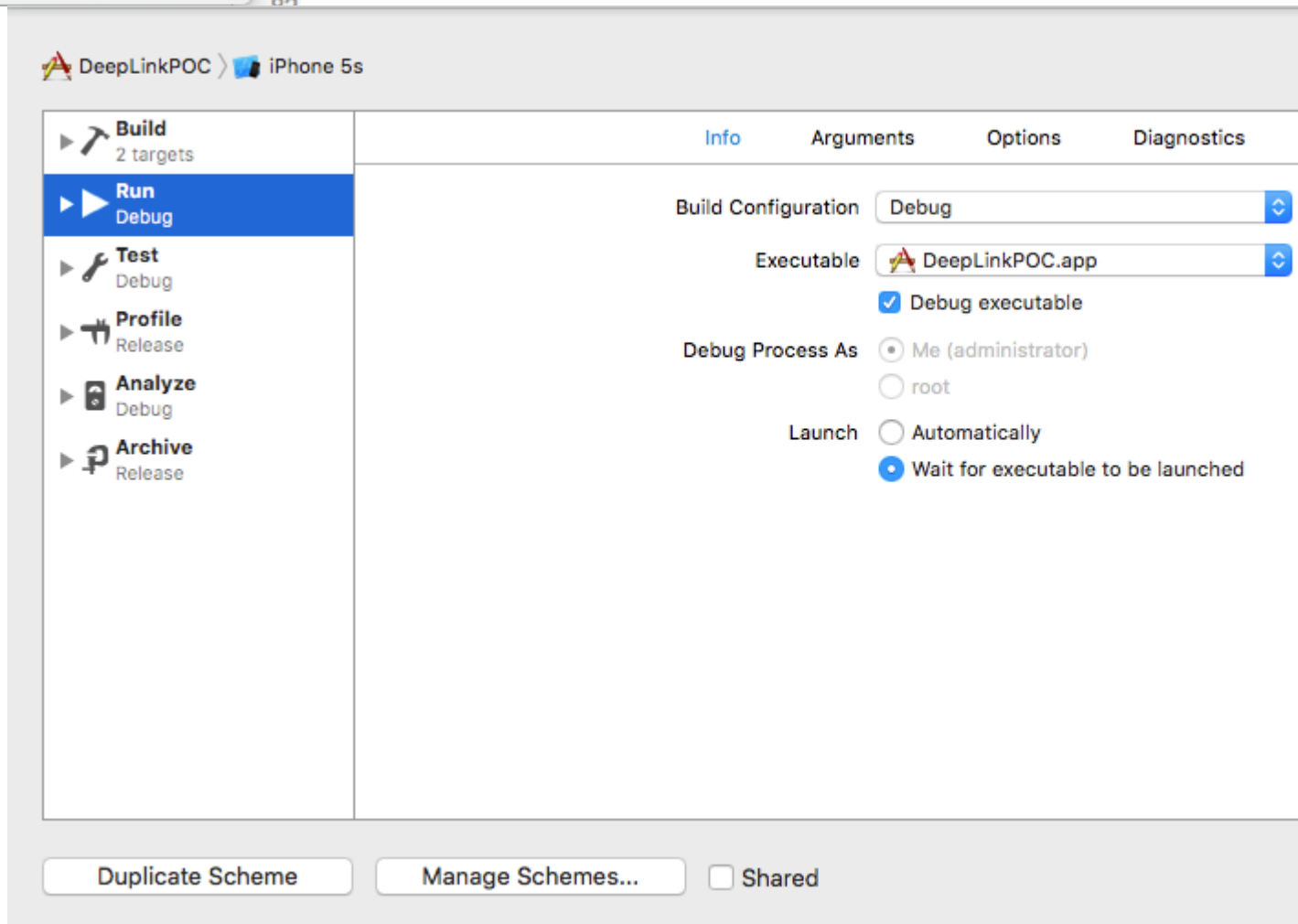
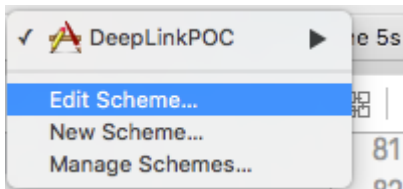
Objective-C

```
-(BOOL)application:(UIApplication *)application  
openURL:(NSURL *)url  
sourceApplication:(NSString *)sourceApplication  
annotation:(id)annotation
```

◦

8. ◦ schemeapp-launch。

•



9. RurididFinishLaunchingWithOptionsopenURL

10. "DeepLinkPOC".

11. safari" **DeepLinking//** ""DeepLinkPOC""".

:)

iOS <https://riptutorial.com/zh-TW/ios/topic/5173/ios>

37: MKDistanceFormatter

Examples

CLLocationDistance Double

```
let distance = CLLocationDistance(42)
let formatter = MKDistanceFormatter()
let answer = formatter.stringFromDistance(distance)
// answer = "150 feet"
```

Objective-C

```
CLLocationDistance distance=42;
MKDistanceFormatter *formatter=[[MKDistanceFormatter alloc]init];
NSString *answer=[formatter stringFromDistance:distance];
// answer = "150 feet"
```

o

import MapKitUnits.Default, .Metric, .Imperial, .ImperialWithYards

```
formatter.units = .Metric
var answer = formatter.stringFromDistance(distance)
// "40 m"

formatter.units = .ImperialWithYards
answer = formatter.stringFromDistance(distance)
// "50 yards"
```

Objective-C

```
MKDistanceFormatter *formatter=[[MKDistanceFormatter alloc]init];
formatter.units=MKDistanceFormatterUnitsMetric;
NSString *answer=[formatter stringFromDistance:distance];
//40 m

formatter.units=MKDistanceFormatterUnitsImperialWithYards;
NSString *answer=[formatter stringFromDistance:distance];
//50 yards
```

unitStyle.Default, .Abbreviated, .Full

```
formatter.unitStyle = .Full
var answer = formatter.stringFromDistance(distance)
// "150 feet"

formatter.unitStyle = .Abbreviated
answer = formatter.stringFromDistance(distance)
// "150 ft"
```

Objective-C

```
formatter.unitStyle=MKDistanceFormatterUnitStyleFull;
NSString *answer=[formatter stringFromDistance:distance];
// "150 feet"

formatter.unitStyle=MKDistanceFormatterUnitStyleAbbreviated;
NSString *answer=[formatter stringFromDistance:distance];
// "150 ft"
```

MKDistanceFormatter <https://riptutorial.com/zh-TW/ios/topic/6677/mkdistanceformatter>

38: ModalPresentationStyles

◦ ◦ **Interface Builder** ◦ `modalPresentationStyle` ◦ `UIModalPresentationStyle` ◦
`modalPresentationStyle` ◦ `UIViewController` ◦ `UIViewController` ◦

Apple ◦

◦ ◦

Examples

Interface Builder `UIModalPresentationStyle`

iOS `UIModalPresentationStyle` ◦ `UIModalPresentationStyle` 9

1. `fullScreen`
2. `pageSheet`
3. `formSheet`
4. `currentContext`
5. `custom`
6. `overFullScreen`
7. `overCurrentContext`
8. `popover`
9. `none`

iOS 2 `UIViewController` ◦ `UIViewController` ◦ `UIButton` ◦ `Target` -> `ActionViewController` ◦ `UIViewController` ◦
`UIViewController` ◦ **Interface Builder**



Button

39: MPMediaPickerDelegate

Apple ◦

iPhone ◦ ◦

Examples

MPMediaPickerControllerDelegateAVAudioPlayer

- “NSAppleMusicUsageDescription”Info.plist◦
- iPhone◦ ◦

iOS 10.0.1

```
import UIKit
import AVFoundation
import MediaPlayer

class ViewController: UIViewController, MPMediaPickerControllerDelegate {

    var avMusicPlayer: AVAudioPlayer!
    var mpMediaPicker: MPMediaPickerController!
    var mediaItems = [MPMediaItem]()
    let currentIndex = 0

    override func viewDidLoad() {
        super.viewDidLoad()
    }

    func audioPlayerDidFinishPlaying(_ player: AVAudioPlayer, successfully flag: Bool){
        //What to do?
    }

    func mediaPicker(_ mediaPicker: MPMediaPickerController, didPickMediaItems
mediaItemCollection: MPMediaItemCollection) {
        mediaItems = mediaItemCollection.items
        updatePlayer()
        self.dismiss(animated: true, completion: nil)
    }

    func updatePlayer(){
        let item = mediaItems[currentIndex]
        // DO-TRY-CATCH try to setup AVAudioPlayer with the path, if successful, sets up the
AVMusicPlayer, and song values.
        if let path: NSURL = item.assetURL as NSURL? {
            do
            {
                avMusicPlayer = try AVAudioPlayer(contentsOf: path as URL)
                avMusicPlayer.enableRate = true
                avMusicPlayer.rate = 1.0
                avMusicPlayer.numberOfLoops = 0
                avMusicPlayer.currentTime = 0
            }
            catch

```

```
        {
            avMusicPlayer = nil
        }
    }
}

@IBAction func Play(_ sender: AnyObject) {
    //AVMusicPlayer.deviceCurrentTime
    avMusicPlayer.play()
}

@IBAction func Stop(_ sender: AnyObject) {
    avMusicPlayer.stop()
}

@IBAction func picker(_ sender: AnyObject) {
    mpMediapicker = MPMediaPickerController.self(mediaTypes:MPMediaType.music)
    mpMediapicker.allowsPickingMultipleItems = false
    mpMediapicker.delegate = self
    self.present(mpMediapicker, animated: true, completion: nil)
}
}
```

MPMediaPickerDelegate <https://riptutorial.com/zh-TW/ios/topic/7299/mpmediapickerdelegate>

40: MPVolumeView

MPVolumeView。

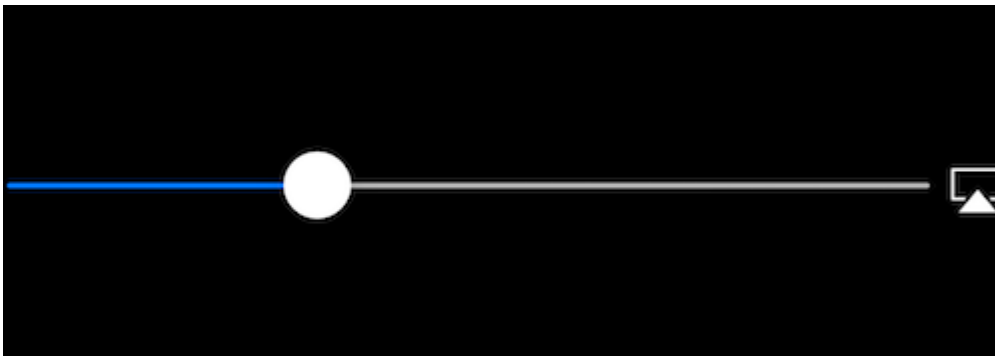
MPVolumeViewiOS。

Examples

MPVolumeView

```
// Add MPVolumeView in a holder view
let mpVolumeHolderView = UIView(frame: CGRect(x: 0, y: view.bounds.midY, width:
view.bounds.width, height: view.bounds.height))
// Set the holder view's background color to transparent
mpVolumeHolderView.backgroundColor = .clear
let mpVolume = MPVolumeView(frame: mpVolumeHolderView.bounds)
mpVolume.showsRouteButton = true
mpVolumeHolderView.addSubview(mpVolume)
view.addSubview(mpVolumeHolderView)
// the volume view is white, set the parent background to black to show it better in this
example
view.backgroundColor = .black
```

!!!MPVolumeView。

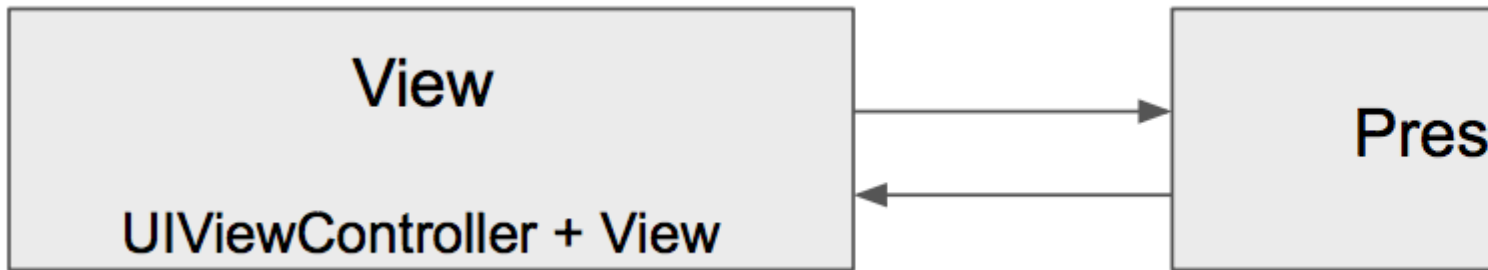


[MPVolumeView](https://riptutorial.com/zh-TW/ios/topic/9038/mpvolumeview) <https://riptutorial.com/zh-TW/ios/topic/9038/mpvolumeview>

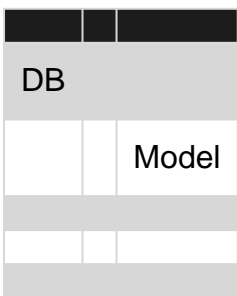
41: MVP

MVP - - ◦ ModelViewPresenter ◦ ◦

MVP ◦



- **ModelGUI**
- **ViewGUI**
- “” ◦ ViewView



MVCMVP

- MVCControllerMVPViewUIViewsUIViewController
- MVP ViewMVVMMVC View
- MVPPresenterMVCController
- MVPMVC
- MVC ControllerUIKitMVP Presenter

- MVPUIViewControllerView.....;
- ◦ ◦
- ◦

- ◦
- ◦

Examples

Dog.swift

```
import Foundation
```

```

enum Breed: String {
    case bulldog = "Bulldog"
    case doberman = "Doberman"
    case labrador = "Labrador"
}

struct Dog {
    let name: String
    let breed: String
    let age: Int
}

```

DoggyView.swift

```

import Foundation

protocol DoggyView: NSObjectProtocol {
    func startLoading()
    func finishLoading()
    func setDoggies(_ doggies: [DoggyViewData])
    func setEmpty()
}

```

DoggyService.swift

```

import Foundation

typealias Result = ([Dog]) -> Void

class DoggyService {

    func deliverDoggies(_ result: @escaping Result) {

        let firstDoggy = Dog(name: "Alfred", breed: Breed.labrador.rawValue, age: 1)
        let secondDoggy = Dog(name: "Vinny", breed: Breed.doberman.rawValue, age: 5)
        let thirdDoggy = Dog(name: "Lucky", breed: Breed.labrador.rawValue, age: 3)

        let delay = DispatchTime.now() + Double(Int64(Double(NSEC_PER_SEC)*2)) /
Double(NSEC_PER_SEC)

        DispatchQueue.main.asyncAfter(deadline: delay) {
            result([firstDoggy,
                    secondDoggy,
                    thirdDoggy])
        }
    }
}

```

DoggyPresenter.swift

```

import Foundation

class DoggyPresenter {

    // MARK: - Private
}

```

```

fileprivate let dogService: DoggyService
weak fileprivate var dogView: DoggyView?

init(dogService: DoggyService){
    self.dogService = dogService
}

func attachView(_ attach: Bool, view: DoggyView?) {
    if attach {
        dogView = nil
    } else {
        if let view = view { dogView = view }
    }
}

func getDogs(){
    self.dogView?.startLoading()

    dogService.deliverDoggies { [weak self] doggies in
        self?.dogView?.finishLoading()

        if doggies.count == 0 {
            self?.dogView?.setEmpty()
        } else {
            self?.dogView?.setDoggies(doggies.map {
                return DoggyViewData(name: "\($0.name) \($0.breed)",
                    age: "\($0.age)")
            })
        }
    }
}

struct DoggyViewData {
    let name: String
    let age: String
}

```

DoggyListViewController.swift

```

import UIKit

class DoggyListViewController: UIViewController, UITableViewDataSource {

    @IBOutlet weak var emptyView: UIView?
    @IBOutlet weak var tableView: UITableView?
    @IBOutlet weak var spinner: UIActivityIndicatorView?

    fileprivate let dogPresenter = DoggyPresenter(dogService: DoggyService())
    fileprivate var dogsToDisplay = [DoggyViewData]()

    override func viewDidLoad() {
        super.viewDidLoad()

        tableView?.dataSource = self
        spinner?.hidesWhenStopped = true
        dogPresenter.attachView(true, view: self)
        dogPresenter.getDogs()
    }
}

```

```

// MARK: DataSource
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return dogsToDisplay.count
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
UITableViewCell {
    let cell = UITableViewCell(style: .subtitle, reuseIdentifier: "Cell")
    let userViewData = dogsToDisplay[indexPath.row]
    cell.textLabel?.text = userViewData.name
    cell.detailTextLabel?.text = userViewData.age
    return cell
}
}

extension DoggyListViewController: DoggyView {

    func startLoading() {
        spinner?.startAnimating()
    }

    func finishLoading() {
        spinner?.stopAnimating()
    }

    func setDoggies(_ doggies: [DoggyViewData]) {
        dogsToDisplay = doggies
        tableView?.isHidden = false
        emptyView?.isHidden = true;
        tableView?.reloadData()
    }

    func setEmpty() {
        tableView?.isHidden = true
        emptyView?.isHidden = false;
    }
}
}

```

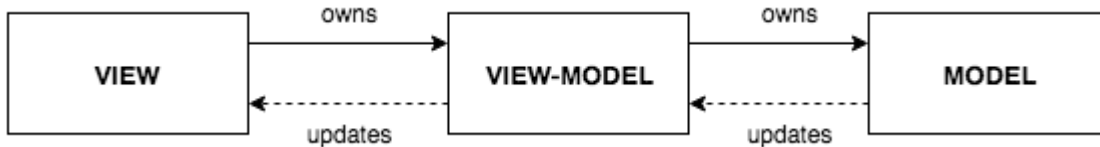
MVP <https://riptutorial.com/zh-TW/ios/topic/9467/mvp>

42: MVVM

Examples

MVVM

iOSModel-View-ViewModelMVVM。 iOSAppleMVC - - 。 - MVC。 ViewController。 MVCiOS - -
- MVVM。



MVVM。 ViewController + ViewstoryboardXIBCodeMVVMView - ViewMVVMView。 ViewModel。
ViewModelView。 ViewModelModel。

MVVM。 。

MVVMViewModelViewViewModelView。

MVVMViewModelViewReactiveCocoaReactiveSwiftRxSwif。 ReactiveMVVM。 ReactiveMVVM。

MVVM。

MVVMExampleViewControllerViewController。 "Hello"。 ViewModel。 MVVMView。

```
class MVVMExampleViewController: UIViewController {  
  
    @IBOutlet weak var helloLabel: UILabel!  
  
    var viewModel: MVVMExampleViewModel?  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
  
    @IBAction func sayHelloButtonPressed(_ sender: UIButton) {  
        viewModel?.userTriggeredSayHelloButton()  
    }  
}
```

MVVMExampleViewModelViewModel。

```
class MVVMExampleViewModel {  
  
    func userTriggeredSayHelloButton() {  
        // How to update View's label when there is no reference to the View??  
    }  
}
```

ViewViewModel ◦ ViewController ◦ AppDelegate

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    if let rootVC = window?.rootViewController as? MVVMEExampleViewController {
        let viewModel = MVVMEExampleViewModel()
        rootVC.viewModel = viewModel
    }

    return true
}
```

ViewModelViewViewModelView Reactive Programming iOS

KVO ◦ ◦ MVVM ◦

DynamicMVVM ◦

```
class Dynamic<T> {
    typealias Listener = (T) -> Void
    var listener: Listener?

    func bind(_ listener: Listener?) {
        self.listener = listener
    }

    func bindAndFire(_ listener: Listener?) {
        self.listener = listener
        listener?(value)
    }

    var value: T {
        didSet {
            listener?(value)
        }
    }

    init(_ v: T) {
        value = v
    }
}
```

DynamicGenericsClosuresViewModelView ◦ ◦ MVVMEExampleViewControllerMVVMEExampleViewModel ◦

MVVMEExampleViewController

```
class MVVMEExampleViewController: UIViewController {

    @IBOutlet weak var helloLabel: UILabel!

    var viewModel: MVVMEExampleViewModel?

    override func viewDidLoad() {
        super.viewDidLoad()
        bindViewModel()
    }

    func bindViewModel() {
```

```

    if let viewModel = viewModel {
        viewModel.helloText.bind({ (helloText) in
            DispatchQueue.main.async {
                // When value of the helloText Dynamic variable
                // is set or changed in the ViewModel, this code will
                // be executed
                self.helloLabel.text = helloText
            }
        })
    }
}

@IBAction func sayHelloButtonPressed(_ sender: UIButton) {
    viewModel?.userTriggeredSayHelloButton()
}
}

```

MVVMExampleViewModel

```

class MVVMExampleViewModel {

    // we have to initialize the Dynamic var with the
    // data type we want
    var helloText = Dynamic("")

    func userTriggeredSayHelloButton() {
        // Setting the value of the Dynamic variable
        // will trigger the closure we defined in the View
        helloText.value = "Hello"
    }
}

```

- ViewModelViewView ◦
- MVVMMVCMVVM ◦ ◦

MVVM <https://riptutorial.com/zh-TW/ios/topic/8775/mvvm>

43: MyLayout

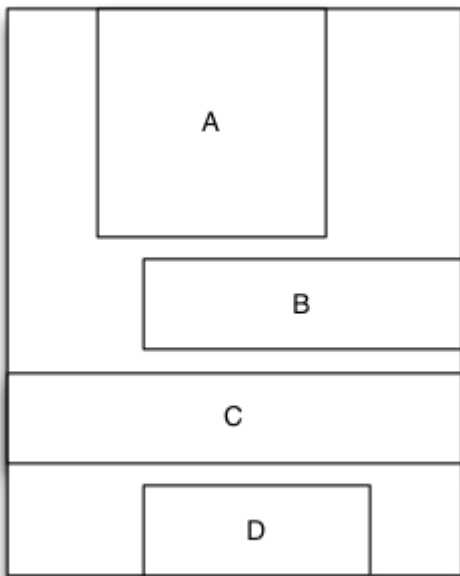
MyLayoutiOS。 MyLayout。 iOSAutolayoutSizeClassAndroidHTML / CSS。

Objective-C <https://github.com/youngsoft/MyLinearLayout> Swift <https://github.com/youngsoft/TangramKit>

Examples

MyLayout

1. S100。 ABCD。
2. AS20S30A。
3. B40S40.CS
40.
4. D20S5040



```
MyLinearLayout *S = [MyLinearLayout  
linearLayoutWithOrientation:MyLayoutViewOrientation_Vert];  
S.subviewSpace = 10;  
S.widthSize.equalTo(@100);  
  
UIView *A = UIView.new;  
A.leftPos.equalTo(@0.2);  
A.rightPos.equalTo(@0.3);  
A.heightSize.equalTo(A.widthSize);  
[S addSubview:A];  
  
UIView *B = UIView.new;  
B.leftPos.equalTo(@40);  
B.widthSize.equalTo(@60);  
B.heightSize.equalTo(@40);
```

```
[S addSubview:B];

UIView *C = UIView.new;
C.leftPos.equalTo(@0);
C.rightPos.equalTo(@0);
C.heightSize.equalTo(@40);
[S addSubview:C];

UIView *D = UIView.new;
D.rightPos.equalTo(@20);
D.widthSize.equalTo(S.widthSize).multiply(0.5);
D.heightSize.equalTo(@40);
[S addSubview:D];
```

MyLayout <https://riptutorial.com/zh-TW/ios/topic/9692/mylayout>

44: NSArray

/Array。

。。

Examples

Arrayjson

“any”。 json。 JsonWebSwift。

// -----

```
let array = [{"one" : 1}, {"two" : 2}, {"three" : 3}, {"four" : 4}]

let jsonString = convertIntoJSONString(arrayObject: array)
print("jsonString - \(jsonString)")
```

// -----

```
func convertIntoJSONString(arrayObject: [Any]) -> String? {

    do {
        let jsonData: Data = try JSONSerialization.data(withJSONObject: arrayObject,
options: [])
        if let jsonString = NSString(data: jsonData, encoding:
String.Encoding.utf8.rawValue) {
            return jsonString as String
        }
    } catch let error as NSError {
        print("Array convertIntoJSON - \(error.description)")
    }
    return nil
}
```

NSArray <https://riptutorial.com/zh-TW/ios/topic/9248/nsarray>

45: NSAttributedString

NSAttributedString

Examples

```
NSAttributedString NSMutableAttributedString ◦
```

```
/◦
```

UILabel “kerning”

```
var attributedString = NSMutableAttributedString("Apply kerning")
attributedString.addAttribute(attribute: NSKernAttributeName, value: 5, range: NSRange(6, 7))
label.attributedString = attributedString
```

Objective-C

```
NSMutableAttributedString *attributedString;
attributedString = [[NSMutableAttributedString alloc] initWithString:@"Apply kerning"];
[attributedString addAttribute:NSKernAttributeName value:@5 range:NSMakeRange(6, 7)];
[label setAttributedString:attributedString];
```

Objective-C

```
NSMutableAttributedString *attributeString = [[NSMutableAttributedString alloc]
initWithString:@"Your String here"];
[attributeString addAttribute:NSStrikethroughStyleAttributeName
value:@2
range:NSMakeRange(0, [attributeString length])];
```

```
let attributeString: NSMutableAttributedString = NSMutableAttributedString(string: "Your
String here")
attributeString.addAttribute(NSStrikethroughStyleAttributeName, value: 2, range:
NSMakeRange(0, attributeString.length))
```

UILabel

```
yourLabel.attributedString = attributeString;
```

Swift

```
let someValue : String = "Something the user entered"
let text = NSMutableAttributedString(string: "The value is: ")
text.appendAttributedString(NSAttributedString(string: someValue, attributes:
[NSFontAttributeName:UIFont.boldSystemFontOfSize(UIFont.systemFontSize())]))
```

Objective-C

```
UIColor *color = [UIColor redColor];
NSString *textToFind = @"redword";

NSMutableAttributedString *attrsString = [[NSMutableAttributedString alloc]
initWithAttributedString:yourLabel.attributedString];

// search for word occurrence
NSRange range = [yourLabel.text rangeOfString:textToFind];
if (range.location != NSNotFound) {
    [attrsString addAttribute:NSForegroundColorAttributeName value:color range:range];
}

// set attributed text
yourLabel.attributedString = attrsString;
```

```
let color = UIColor.red;
let textToFind = "redword"

let attrsString = NSMutableAttributedString(string:yourlabel.text!);

// search for word occurrence
let range = (yourlabel.text! as NSString).range(of: textToFind)
if (range.length > 0) {
    attrsString.addAttribute(NSForegroundColorAttributeName, value:color, range:range)
}

// set attributed text
yourlabel.attributedString = attrsString
```

NSMutableAttributedStringaddAttribute:value:rangeNSForegroundColorAttributeName

```
NSMutableAttributedString *attrsString = [[NSMutableAttributedString alloc]
initWithAttributedString:label.attributedString];
[attrsString addAttribute:NSForegroundColorAttributeName value:color range:range];
```

NSRegularExpression

Objective-C

```
NSMutableAttributedString *mutAttString = @"string goes here";
NSRange range = NSMakeRange(0, mutAttString.length);
[mutAttString setAttributes:@{ } range:originalRange];
```

Apple setAttributesaddAttribute

```
mutAttString.setAttributes([:], range: NSRange(0..
```

NSAttributedString <https://riptutorial.com/zh-TW/ios/topic/979/nsattributedString>

46: NSHTTPCookieStorage

Examples

NSUserDefaultscookie

```
import Foundation

class CookiesSingleton {

static let instance : CookiesSingleton = CookiesSingleton()
static var enableDebug = true

func loadCookies() {
    if let cookiesDetails =
NSUserDefaults.standardUserDefaults().objectForKey("customeWebsite") {
        for (keys,_) in cookiesDetails as! NSDictionary{
            if let cookieDict = NSUserDefaults.standardUserDefaults().objectForKey(keys
as! String){
                if let cookie = NSHTTPCookie(properties:cookieDict as! [String:AnyObject])
{
                    NSHTTPCookieStorage.sharedHTTPCookieStorage().setCookie(cookie)
                    if(CookiesSingleton.enableDebug){
                        print("Each Cookies",cookieDict)
                    }
                }
            }
        }
    }
}

func removeCookies(){
    NSURLCache.sharedURLCache().removeAllCachedResponses()
    NSURLCache.sharedURLCache().diskCapacity = 0
    NSURLCache.sharedURLCache().memoryCapacity = 0

    let storage : NSHTTPCookieStorage = NSHTTPCookieStorage.sharedHTTPCookieStorage()
    for cookie in storage.cookies! {
        storage.deleteCookie(cookie as NSHTTPCookie)
    }

    NSUserDefaults.standardUserDefaults().setValue("", forKey: "customeWebsite")
    NSUserDefaults.standardUserDefaults().synchronize()

    if(CookiesSingleton.enableDebug){
        print("Cookies Removed")
    }
}

func saveCookies() {

    let cookieArray = NSMutableArray()
    let savedC = NSHTTPCookieStorage.sharedHTTPCookieStorage().cookies

    let allCookiesDic:NSMutableDictionary = NSMutableDictionary()
```

```

for c : NSHTTPCookie in savedC! {

    let cookieProps = NSMutableDictionary()
    cookieArray.addObject(c.name)
    cookieProps.setValue(c.name, forKey: NSHTTPCookieName)
    cookieProps.setValue(c.value, forKey: NSHTTPCookieValue)
    cookieProps.setValue(c.domain, forKey: NSHTTPCookieDomain)
    cookieProps.setValue(c.path, forKey: NSHTTPCookiePath)
    cookieProps.setValue(c.version, forKey: NSHTTPCookieVersion)
    cookieProps.setValue(NSDate().dateByAddingTimeInterval(2629743), forKey:
NSHTTPCookieExpires)

    allCookiesDic.setValue(cookieProps, forKey: c.name)

}
NSUserDefaults.standardUserDefaults().setValue(allCookiesDic, forKey: "customeWebsite")
NSUserDefaults.standardUserDefaults().synchronize()

if(CookiesSingleton.enableDebug){
    print("Cookies Saved")
}
}
}

```

NSHTTPCookieStorage <https://riptutorial.com/zh-TW/ios/topic/7312/nshttpcookiestorage>

47: NSInvocation

Examples

NSInvocation Objective-C

[e.James](#)

[AppleNSInvocation](#)

NSInvocationObjective-C

C

- NSInvocation
- NSInvocation

- addObject:

```
[myArray addObject:myString];
```

NSInvocation

NSInvocationNSMutableArrayaddObject: selector

```
NSString * mySignature = [NSMutableArray  
    instanceMethodSignatureForSelector:@selector(addObject:)];  
NSInvocation * myInvocation = [NSInvocation  
    invocationWithMethodSignature:mySignature];
```

```
[myInvocation setTarget:myArray];
```

```
[myInvocation setSelector:@selector(addObject:)];
```

```
[myInvocation setArgument:&myString atIndex:2];
```

- [Ryan McCuaigApple](#)

myInvocation

```
[myInvocation invoke];
```

```
[myArray addObject:myString];
```

-

- NSInvocationselector invoke

[NSInvocation](#)

NSUndoManagerNSInvocation° NSInvocation“”° NSInvocationNSUndoManager ° “Undo” NSUndoManager
NSInvocation°

°

NSInvocation <https://riptutorial.com/zh-TW/ios/topic/8276/nsinvocation>

48: NotificationCenter

iOS ◦ ◦

- [Swift](#)

	◦ nil◦
OBJ	◦ nil◦
	◦ nil◦
	◦ ◦

NSNotificationCenter ◦ NotificationCenter ◦

Apple

[SwiftNotificationCenterNSNotificationCenter](#)

Examples

NSString

Name of associated class + Did | Will + UniquePartOfName + Notification

- UIApplicationDidBecomeActiveNotification
- UIWindowDidMiniaturizeNotification
- NSTextViewDidChangeSelectionNotification
- NSColorPanelColorDidChangeNotification

2.3

```
NSNotificationCenter.defaultCenter().addObserver(self,
                                                    selector:
#selector(self.testNotification(_:)),
                                                    name: "TestNotification",
                                                    object: nil)
```

3

```
NSNotificationCenter.default.addObserver(self,
                                       selector: #selector(self.testNotification(_:)),
                                       name: NSNotification.Name(rawValue:
"TestNotification"),
                                       object: nil)
```

Objective-C

```
[[NSNotificationCenter defaultCenter] addObserver:self
                                       selector:@selector(testNotification:)
                                       name:@"TestNotification"
                                       object:nil];
```

PS. `UIViewController` `viewWillAppear:` `viewDidLoad:`

2.3

```
//Remove observer for single notification
NSNotificationCenter.defaultCenter().removeObserver(self, name: "TestNotification", object:
nil)

//Remove observer for all notifications
NSNotificationCenter.defaultCenter().removeObserver(self)
```

3

```
//Remove observer for single notification
NSNotificationCenter.default.removeObserver(self, name: NSNotification.Name(rawValue:
"TestNotification"), object: nil)

//Remove observer for all notifications
NSNotificationCenter.default.removeObserver(self)
```

Objective-C

```
//Remove observer for single notification
[[NSNotificationCenter defaultCenter] removeObserver:self name:@"TestNotification"
object:nil];

//Remove observer for all notifications
[[NSNotificationCenter defaultCenter] removeObserver:self];
```

```
NSNotificationCenter.defaultCenter().postNotificationName("TestNotification", object: self)
```

Objective-C

```
[[NSNotificationCenter defaultCenter] postNotificationName:@"TestNotification" object:nil];
```

```
let userInfo: [String: AnyObject] = ["someKey": myObject]
NSNotificationCenter.defaultCenter().postNotificationName("TestNotification", object: self,
userInfo: userInfo)
```

Objective-C

```
NSDictionary *userInfo = [NSDictionary dictionaryWithObject:myObject forKey:@"someKey"];
[[NSNotificationCenter defaultCenter] postNotificationName: @"TestNotification" object:nil
userInfo:userInfo];
```

```
func testNotification(notification: NSNotification) {
    let userInfo = notification.userInfo
    let myObject: MyObject = userInfo["someKey"]
}
```

Objective-C

```
- (void)testNotification:(NSNotification *)notification {
    NSDictionary *userInfo = notification.userInfo;
    MyObject *myObject = [userInfo objectForKey:@"someKey"];
}
```

```
id testObserver = [[NSNotificationCenter defaultCenter] addObserverForName:@"TestNotification"
                                                                    object:nil
                                                                    queue:nil
                                                                    usingBlock:^(NSNotification*
notification) {
    NSDictionary *userInfo = notification.userInfo;
    MyObject *myObject = [userInfo objectForKey:@"someKey"];
}];
```

```
[[NSNotificationCenter defaultCenter] removeObserver:testObserver
                                             name:@"TestNotification"
                                             object:nil];
```

```
// Add observer
```

```
let observer =
NotificationCenter.defaultCenter().addObserverForName("nameOfTheNotification", object: nil,
queue: nil) { (notification) in
    // Do operations with the notification in this block
}

// Remove observer
NotificationCenter.defaultCenter().removeObserver(observer)
```

NSNotificationCenter <https://riptutorial.com/zh-TW/ios/topic/1601/nsnotificationcenter>

49: NSPredicate

- - Cdsf
 - @
 - K
- - ===
 - > ==>
 - <== <
 - >
 - <
 - =<>
 - BETWEEN - BETWEEN {0,5}
- - AND&&AND
 - OR||
 -
- - BEGINSWITH
 - ENDSWITH
 - CONTAINS
 - LIKE
 - *
 -

Examples

predicateWithBlockNSPredicate

Objective-C

```
NSPredicate *predicate = [NSPredicate predicateWithBlock:^(id item,
                                                         NSDictionary *bindings) {
    return [item isKindOfClass:[UILabel class]];
}];
```

```
let predicate = NSPredicate { (item, bindings) -> Bool in
    return item.isKindOfClass(UILabel.self)
}
```

UILabel◦

predicateWithFormatNSPredicate

Objective-C

```
NSPredicate *predicate = [NSPredicate predicateWithFormat:@"%self[SIZE] = %d", 5];
```

```
let predicate = NSPredicate(format: "self[SIZE] >= %d", 5)
```

5.

NSPredicate

NSPredicate°

Objective-C

```
NSPredicate *template = [NSPredicate predicateWithFormat: @"self BEGINSWITH $letter"];  
NSDictionary *variables = @{@"letter": @"r"};  
NSPredicate *beginsWithR = [template predicateWithSubstitutionVariables: variables];
```

```
let template = NSPredicate(format: "self BEGINSWITH $letter")  
let variables = ["letter": "r"]  
let beginsWithR = template.predicateWithSubstitutionVariables(variables)
```

`predicateWithSubstitutionVariables` `predicateWithSubstitutionVariables` ° °

NSPredicate

Objective-C

```
NSArray *heroes = @[@"tracer", @"bastion", @"reaper", @"junkrat", @"roadhog"];  
  
NSPredicate *template = [NSPredicate predicateWithFormat:@"self BEGINSWITH $letter"];  
  
NSDictionary *beginsWithRVariables = @{@"letter": @"r"};  
NSPredicate *beginsWithR = [template predicateWithSubstitutionVariables:  
beginsWithRVariables];  
  
NSArray *beginsWithRHeroes = [heroes filteredArrayUsingPredicate: beginsWithR];  
// ["reaper", "roadhog"]  
  
NSDictionary *beginsWithTVariables = @{@"letter": @"t"};  
NSPredicate *beginsWithT = [template predicateWithSubstitutionVariables: beginsWithTVariables];  
  
NSArray *beginsWithTHeroes = [heroes filteredArrayUsingPredicate: beginsWithT];  
// ["tracer"]
```

```
let heroes = ["tracer", "bastion", "reaper", "junkrat", "roadhog"]  
  
let template = NSPredicate(format: "self BEGINSWITH $letter")  
  
let beginsWithRVariables = ["letter": "r"]  
let beginsWithR = template.predicateWithSubstitutionVariables(beginsWithRVariables)  
  
let beginsWithRHeroes = heroes.filter { beginsWithR.evaluateWithObject($0) }
```

```
// ["reaper", "roadhog"]

let beginsWithTVVariables = ["letter": "t"]
let beginsWithT = template.predicateWithSubstitutionVariables(beginsWithTVVariables)

let beginsWithTHeroes = heroes.filter { beginsWithT.evaluateWithObject($0) }
// ["tracer"]
```

NSPredicate

```
NSString *emailRegex = @"[A-Z0-9a-z]([A-Z0-9a-z._-]{0,64})+[A-Z0-9a-z]+@[A-Z0-9a-z]+([A-Za-z0-9.-]{0,64})+([A-Z0-9a-z])+\.[A-Za-z]{2,4}";    NSString *firstNameRegex = @"[0-9A-Za-z\"'-]{2,32}$";
NSString *firstNameRegex = @"[ 0-9A-Za-z]{2,32}$";
NSString *lastNameRegex = @"[0-9A-Za-z\"'-]{2,32}$";
NSString *mobileNumberRegex = @"^[0-9]{10}$";
NSString *zipcodeRegex = @"^[0-9]{5}$";
NSString *SSNRegex = @"^\d{3}-?\d{2}-?\d{4}$";
NSString *addressRegex = @"^[ A-Za-z0-9]{2,32}$";
NSString *cityRegex = @"^[ A-Za-z0-9]{2,25}$";
NSString *PINRegex = @"^[0-9]{4}$";
NSString *driversLiscRegex = @"^[0-9a-zA-Z]{5,20}$";

-(BOOL)validateEmail {
    //Email address field should give an error when the email address begins with ".", "-", "_"
    .
    NSPredicate *emailPredicate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@",
emailRegex];
    return ([emailPredicate evaluateWithObject:self.text] && self.text.length <= 64 &&
([self.text rangeOfString:@".."].location == NSNotFound));
}

-(BOOL)validateFirstName {
    NSPredicate *firstNamePredicate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@",
firstNameRegex];
    return [firstNamePredicate evaluateWithObject:self.text];
}

-(BOOL)validateLastName {
    NSPredicate *lastNamePredicate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@",
lastNameRegex];
    return [lastNamePredicate evaluateWithObject:self.text];
}

-(BOOL)validateAlphaNumericMin2Max32 {
    NSPredicate *firstNamePredicate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@",
firstNameRegex];
    return [firstNamePredicate evaluateWithObject:self.text];
}

-(BOOL)validateMobileNumber {
    NSString *strippedMobileNumber = [[[[self.text stringByReplacingOccurrencesOfString:@"("
withString:@""]
                                stringByReplacingOccurrencesOfString:@")"
withString:@""]
                                stringByReplacingOccurrencesOfString:@"-"
withString:@""]
                                stringByReplacingOccurrencesOfString:@" "
withString:@""]];
```



```

    NSPredicate *mobileNumberPredicate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@",
mobileNumberRegex];

    return [mobileNumberPredicate evaluateWithObject:strippedMobileNumber];
}

- (BOOL)validateZipcode {
    NSPredicate *zipcodePredicate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@",
zipcodeRegex];

    return [zipcodePredicate evaluateWithObject:self.text];
}

- (BOOL)validateSSN {
NSPredicate *predicate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@", SSNRegex];

return [predicate evaluateWithObject:self.text];
}

- (BOOL)validateAddress {
    NSPredicate *predicate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@",
addressRegex];

    return [predicate evaluateWithObject:self.text];
}

- (BOOL)validateCity {
    NSPredicate *predicate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@", cityRegex];
    return [predicate evaluateWithObject:self.text];
}

- (BOOL)validatePIN {
    NSPredicate *predicate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@", PINRegex];

    return [predicate evaluateWithObject:self.text];
}

- (BOOL)validateDriversLiscNumber {
    if([self.text length] > 20) {
        return NO;
    }
    NSPredicate *driversLiscPredicate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@",
driversLiscRegex];

    return [driversLiscPredicate evaluateWithObject:self.text];
}
}

```

NSPredicate`AND`OR`NOT`

NSCompoundPredicate◦

Objective-C

-

```
NSPredicate *predicate = [NSPredicate predicateWithFormat:@"samplePredicate"];
NSPredicate *anotherPredicate = [NSPredicate predicateWithFormat:@"anotherPredicate"];
NSPredicate *combinedPredicate = [NSCompoundPredicate andPredicateWithSubpredicates:
@[predicate, anotherPredicate]];
```

-

```
NSPredicate *predicate = [NSPredicate predicateWithFormat:@"samplePredicate"];
NSPredicate *anotherPredicate = [NSPredicate predicateWithFormat:@"anotherPredicate"];
NSPredicate *combinedPredicate = [NSCompoundPredicate orPredicateWithSubpredicates:
@[predicate, anotherPredicate]];
```

-

```
NSPredicate *predicate = [NSPredicate predicateWithFormat:@"samplePredicate"];
NSPredicate *anotherPredicate = [NSPredicate predicateWithFormat:@"anotherPredicate"];
NSPredicate *combinedPredicate = [NSCompoundPredicate notPredicateWithSubpredicate:
@[predicate, anotherPredicate]];
```

NSPredicate <https://riptutorial.com/zh-TW/ios/topic/5796/nspredicate>

50: NSURL

Examples

NSURL String

```
NSURL *url = [NSURL URLWithString:@"http://www.example.com/images/apple-tree.jpg"];
NSString *fileName = [url lastPathComponent];
// fileName = "apple-tree.jpg"
```

SwiftURLNSURL

2.3

```
let url = NSURL(string: "http://google.com/lastPath")
let lastPath = url?.lastPathComponent
```

Swift 3.0

```
let url = URL(string: "http://google.com/lastPath")
let lastPath = url?.lastPathComponent
```

NSURL <https://riptutorial.com/zh-TW/ios/topic/4610/nsurl>

51: NSURLConnection

Examples

//NSURLConnectionDelegate.

```
@interface ViewController : UIViewController<NSURLConnectionDelegate>
{
    NSMutableData *_responseData;
}
```

// NSURLConnection.

```
#pragma mark NSURLConnection Delegate Methods

- (void)connection:(NSURLConnection *)connection didReceiveResponse:(NSURLResponse *)response
{
    // A response has been received, this is where we initialize the instance var you created
    // so that we can append data to it in the didReceiveData method
    // Furthermore, this method is called each time there is a redirect so reinitializing it
    // also serves to clear it
    _responseData = [[NSMutableData alloc] init];
}

- (void)connection:(NSURLConnection *)connection didReceiveData:(NSData *)data {
    // Append the new data to the instance variable you declared
    [_responseData appendData:data];
}

- (NSCachedURLResponse *)connection:(NSURLConnection *)connection
    willCacheResponse:(NSCachedURLResponse*)cachedResponse {
    // Return nil to indicate not necessary to store a cached response for this connection
    return nil;
}

- (void)connectionDidFinishLoading:(NSURLConnection *)connection {
    // The request is complete and data has been received
    // You can parse the stuff in your instance variable now
}

- (void)connection:(NSURLConnection *)connection didFailWithError:(NSError *)error {
    // The request has failed for some reason!
    // Check the error var
}
```

```
NSURLRequest * urlRequest = [NSURLRequest requestWithURL:[NSURL
URLWithString:@"http://google.com"]];
NSURLResponse * response = nil;
NSError * error = nil;
NSData * data = [NSURLConnection sendSynchronousRequest:urlRequest
                                returningResponse:&response
                                error:&error];
```

```
if (error == nil)
{
    // Parse data here
}
```

```
// Create the request instance.
NSURLRequest *request = [NSURLRequest requestWithURL:[NSURL
URLWithString:@"http://google.com"]];

// Create url connection and fire request
NSURLConnection *conn = [[NSURLConnection alloc] initWithRequest:request delegate:self];
```

NSURLConnection <https://riptutorial.com/zh-TW/ios/topic/6004/nsurlconnection>

52: NSURLSession

NSURLSessionAPI. APIiOS.

NSURLSession. URLURL. .

- NSURLConnection. .

. . Web.

NSURLSession. iOSTLS.

- NSURLConnection.

- .

- cookie.

. . . .

.

- NSData. .

- . .

- NSData. POST. /NSURLRequest/.

- .

NSURLSessionTLS.

Examples

GET

```
// define url
let url = NSURL(string: "https://urlToGet.com")

//create a task to get data from a url
let task = NSURLSession.sharedSession().dataTaskWithURL(url!)
{
    /*inside this block, we have access to NSData *data, NSURLResponse *response, and
    NSError *error returned by the dataTaskWithURL() function*/
    (data, response, error) in

    if error == nil
    {
        // Data from the request can be manipulated here
    }
    else
    {
        // An error occurred
    }
}
```

```

    }
}

//make the request
task.resume()

```

Objective-C

```

NSURL *url = [NSURL URLWithString:@"http://www.example.com/"];
NSURLSessionConfiguration *configuration = [NSURLSessionConfiguration
defaultSessionConfiguration];

// Configure the session here.

NSURLSession *session = [NSURLSession sessionWithConfiguration:configuration];

[[session dataTaskWithURL:url
    completionHandler:^(NSData *data, NSURLResponse *response, NSError *error)
{
    // The response object contains the metadata (HTTP headers, status code)

    // The data object contains the response body

    // The error object contains any client-side errors (e.g. connection
    // failures) and, in some cases, may report server-side errors.
    // In general, however, you should detect server-side errors by
    // checking the HTTP status code in the response object.
}] resume];

```

```

// Swift:
let mySessionID = "com.example.bgSession"
let bgSessionConfig =
NSURLSessionConfiguration.backgroundSessionConfigurationWithIdentifier(mySessionID)

let session = NSURLSession(configuration: bgSessionConfig)

// add tasks here

// Objective-C:
NSString *mySessionID = @"com.example.bgSession";
NSURLSessionConfiguration *configuration =
    [NSURLSessionConfiguration backgroundSessionConfigurationWithIdentifier: mySessionID];
NSURLSession *session = [NSURLSession sessionWithConfiguration:configuration
                        delegate:self]

```

iOS application:handleEventsForBackgroundURLSession:completionHandler: **Objective-C**

application(_:handleEventsForBackgroundURLSession:completionHandler:) **Swift**

◦ ◦

URLSessionDidFinishEventsForBackgroundURLSession:

URLSessionDidFinishEventsForBackgroundURLSession: **Obj-C/**

URLSessionDidFinishEventsForBackgroundURLSession **Swift** ◦ ◦

application:application:didFinishLaunchingWithOptions: **call**

- NSURLSession
- NSXML

ID。 ◦ handleEventsForBackgroundURLSession◦

Objective-C NSURLSession POST

POSTURLapplication / x-www-form-urlencodedmultipart / form-data◦ ◦

URL

iOS◦

◦

== ABCDE

◦

◦ NSURLSession POST◦

```
// Create the configuration, which is necessary so we can cancel cacheing amongst other things.
NSURLSessionConfiguration * defaultConfigObject = [NSURLSessionConfiguration
defaultSessionConfiguration];
// Disables cacheing
defaultConfigObject.requestCachePolicy = NSURLRequestReloadIgnoringLocalCacheData;
NSURLSession * defaultSession = [NSURLSession sessionWithConfiguration:defaultConfigObject
delegate:self delegateQueue:[NSOperationQueue mainQueue]];

NSString * scriptURL = [NSString stringWithFormat:@"https://server.io/api/script.php"];
//Converts the URL string to a URL usable by NSURLSession
NSMutableURLRequest * urlRequest = [NSMutableURLRequest requestWithURL:[NSURL
URLWithString:scriptURL]];
NSString * postDataString = [NSString stringWithFormat:@"name=%@&password=%@", [self
nameString], [self URLEncode:passwordString]];
[urlRequest setHTTPMethod:@"POST"];
[urlRequest setHTTPBody:[postDataString dataUsingEncoding:NSUTF8StringEncoding]];

NSURLSessionDataTask * dataTask = [defaultSession dataTaskWithRequest:urlRequest];
// Fire the data task.
[dataTask resume];
```

POST◦ URLPOST◦ ◦

URL

```
- (NSString *)URLEncode:(NSString *)originalString encoding:(NSStringEncoding)encoding
{
    return (__bridge_transfer NSString *)CFURLCreateStringByAddingPercentEscapes(
        kCFAllocatorDefault,
        (__bridge CFStringRef)originalString,
        NULL,
        CFSTR(":/?#[]@!$&'()*+,-;="),
```



```
        CFStringConvertNSStringEncodingToEncoding(encoding));
    }
```

iOS。

NSURLSession。 ◦ 5ENTIRE

```
- (void)URLSession:(NSURLSession *)session dataTask:(NSURLSessionDataTask *)dataTask
didReceiveResponse:(NSURLResponse *)response
completionHandler:(void (^)(NSURLSessionResponseDisposition disposition))completionHandler;

- (void)URLSession:(NSURLSession *)session dataTask:(NSURLSessionDataTask *)dataTask
didReceiveData:(NSData *)data;

- (void)URLSession:(NSURLSession *)session task:(NSURLSessionTask *)task
didCompleteWithError:(NSError *)error;

- (void)URLSession:(NSURLSession *)session didReceiveChallenge:(NSURLAuthenticationChallenge
*)challenge completionHandler:(void (^)(NSURLSessionAuthChallengeDisposition, NSURLCredential
*))completionHandler;

- (void)URLSession:(NSURLSession *)session task:(NSURLSessionTask *)task
didReceiveChallenge:(NSURLAuthenticationChallenge *)challenge completionHandler:(void
(^)(NSURLSessionAuthChallengeDisposition, NSURLCredential * _Nullable))completionHandler;
```

◦ Apple

```
// Response handling delegates
- (void)URLSession:(NSURLSession *)session dataTask:(NSURLSessionDataTask *)dataTask
didReceiveResponse:(NSURLResponse *)response
completionHandler:(void (^)(NSURLSessionResponseDisposition disposition))completionHandler{
    // Handler allows us to receive and parse responses from the server
    completionHandler(NSURLSessionResponseAllow);
}

- (void)URLSession:(NSURLSession *)session dataTask:(NSURLSessionDataTask *)dataTask
didReceiveData:(NSData *)data{

    // Parse the JSON that came in into an NSDictionary
    NSError * err = nil;
    NSDictionary * jsonDict = [NSJSONSerialization JSONObjectWithData:data
options:NSJSONReadingAllowFragments error:&err];

    if (!err){ // if no error occurred, parse the array of objects as normal
        // Parse the JSON dictionary 'jsonDict' here
    }else{ // an error occurred so we need to let the user know
        // Handle your error here
    }
}

// Error handling delegate
- (void)URLSession:(NSURLSession *)session task:(NSURLSessionTask *)task
didCompleteWithError:(NSError *)error{
    if(error == nil){
        // Download from API was successful
        NSLog(@"Data Network Request Did Complete Successfully.");
    }else{
        // Describes and logs the error preventing us from receiving a response
    }
}
```

```

        NSLog(@"Error: %@", [error userInfo]);

        // Handle network error, letting the user know what happened.
    }
}

// When the session receives a challenge (because of iOS 9 App Transport Security blocking
non-valid SSL certificates) we use the following methods to tell NSURLSession "Chill out, I
can trust me".
// The following is not necessary unless your server is using HTTP, not HTTPS

- (void)NSURLSession:(NSURLSession *)session didReceiveChallenge:(NSURLAuthenticationChallenge
*)challenge completionHandler:(void (^)(NSURLSessionAuthChallengeDisposition, NSURLCredential
*))completionHandler{
    if([challenge.protectionSpace.authenticationMethod
isEqualToString:NSURLAuthenticationMethodServerTrust]){
        if([challenge.protectionSpace.host isEqualToString:@"DomainNameOfServer.io"]){
            NSURLCredential * credential = [NSURLCredential
credentialForTrust:challenge.protectionSpace.serverTrust];
            completionHandler(NSURLSessionAuthChallengeUseCredential,credential);
        }
    }
}

- (void)NSURLSession:(NSURLSession *)session task:(NSURLSessionTask *)task
didReceiveChallenge:(NSURLAuthenticationChallenge *)challenge completionHandler:(void
(^)(NSURLSessionAuthChallengeDisposition, NSURLCredential * _Nullable))completionHandler{
    if([challenge.protectionSpace.authenticationMethod
isEqualToString:NSURLAuthenticationMethodServerTrust]){
        if([challenge.protectionSpace.host isEqualToString:@"DomainNameOfServer.io"]){
            NSURLCredential * credential = [NSURLCredential
credentialForTrust:challenge.protectionSpace.serverTrust];
            completionHandler(NSURLSessionAuthChallengeUseCredential,credential);
        }
    }
}
}

```

iOS 9API.....◦ ◦

URL◦ ASCII◦ ◦

◦ ◦ **NSURLSession**

```

UIImage * imgToSend;

// 2nd parameter of UIImageJPEGRepresentation represents compression quality. 0 being most
compressed, 1 being the least
// Using 0.4 likely stops us hitting the servers upload limit and costs us less server space
NSData * imageData = UIImageJPEGRepresentation(imgToSend, 0.4f);

// Alternatively, if the photo is on disk, you can retrieve it with
// [NSData dataWithContentsOfURL:...]
```

// Set up the body of the POST request.

```

// This boundary serves as a separator between one form field and the next.
// It must not appear anywhere within the actual data that you intend to
// upload.
NSString * boundary = @"-----14737809831466499882746641449";

```

```

// Body of the POST method
NSMutableData * body = [NSMutableData data];

// The body must start with the boundary preceded by two hyphens, followed
// by a carriage return and newline pair.
//
// Notice that we prepend two additional hyphens to the boundary when
// we actually use it as part of the body data.
//
[body appendData:[NSString stringWithFormat:@"\r\n--%@\r\n",boundary]
dataUsingEncoding:NSUTF8StringEncoding]];

// This is followed by a series of headers for the first field and then
// TWO CR-LF pairs.
[body appendData:[NSString stringWithFormat:@"Content-Disposition: form-data;
name=\"%tag_name%\r\n\r\n"] dataUsingEncoding:NSUTF8StringEncoding]];

// Next is the actual data for that field (called "tag_name") followed by
// a CR-LF pair, a boundary, and another CR-LF pair.
[body appendData:[strippedCompanyName dataUsingEncoding:NSUTF8StringEncoding]];
[body appendData:[NSString stringWithFormat:@"\r\n--%@\r\n", boundary]
dataUsingEncoding:NSUTF8StringEncoding]];

// Encode the filename and image data as the "userfile" CGI parameter.
// This is similar to the previous field, except that it is being sent
// as an actual file attachment rather than a blob of data, which means
// it has both a filename and the actual file contents.
//
// IMPORTANT: The filename MUST be plain ASCII (and if encoded like this,
//             must not include quotation marks in the filename).
//
NSString * picFileName = [NSString stringWithFormat:@"photoName"];
NSString * appendDataString = [NSString stringWithFormat:@"Content-Disposition: form-data;
name=\"userfile\"; filename=\"%@.jpg\"\r\n", picFileName];
[body appendData:[appendDataString dataUsingEncoding:NSUTF8StringEncoding]];
[body appendData:[@"Content-Type: application/octet-stream\r\n\r\n"
dataUsingEncoding:NSUTF8StringEncoding]];
[body appendData:[NSData dataWithData:imageData]];

// Close the request body with one last boundary with two
// additional hyphens prepended **and** two additional hyphens appended.
[body appendData:[NSString stringWithFormat:@"\r\n--@--\r\n", boundary]
dataUsingEncoding:NSUTF8StringEncoding]];

// Create the session
// We can use the delegate to track upload progress and disable cacheing
NSURLSessionConfiguration * defaultConfigObject = [NSURLSessionConfiguration
defaultSessionConfiguration];
defaultConfigObject.requestCachePolicy = NSURLRequestReloadIgnoringLocalCacheData;
NSURLSession * defaultSession = [NSURLSession sessionWithConfiguration: defaultConfigObject
delegate: self delegateQueue: [NSOperationQueue mainQueue]];

// Data uploading task.
NSURL * url = [NSURL URLWithString:@"https://server.io/api/script.php"];
NSMutableURLRequest * request = [NSMutableURLRequest requestWithURL:url];
NSString * contentType = [NSString stringWithFormat:@"multipart/form-data;
boundary=%@",boundary];
[request addValue:contentType forHTTPHeaderField:@"Content-Type"];
request.HTTPMethod = @"POST";
request.HTTPBody = body;

```

```
NSURLSessionDataTask * uploadTask = [defaultSession dataTaskWithRequest:request];  
[uploadTask resume];
```

NSURLSession。 urlurl。

NSURLSession <https://riptutorial.com/zh-TW/ios/topic/2009/nsurlsession>

53: NSUserActivity

NSUserActivity。 iOSmacOS。 Spotlight。 iOS 10SiriSiriKit。

NSUserActivityTypesInfo.plist。 IDID“Safari”“Chrome”。

becomeCurrentHandoffSpotlight Indexing。 。 resignCurrent。

invalidateinvalidate。

SiriKit。

。 。 CoreSpotlight。

Examples

NSUserActivity

NSUserActivityInfo.plist。 。 “com.companyName.productName.activityName”。 Info.plist

NSUserActivityTypes	[]
- item0	com.companyName.productName.activityName01
- item1	com.companyName.productName.activityName02

。

NSUserActivity

```
// Initialize the activity object and set its type from one of the ones specified in your
app's plist
NSUserActivity *currentActivity = [[NSUserActivity alloc]
initWithActivityType:@"com.companyName.productName.activityName01"];

// Set the title of the activity.
// This title may be displayed to the user, so make sure it is localized and human-readable
currentActivity.title = @"Current Activity";

// Configure additional properties like userInfo which will be included in the activity
currentActivity.userInfo = @{@"informationKey" : @"value"};
```

```
// Configure the activity so the system knows what may be done with it
// It is important that you only set YES to tasks that your application supports
// In this example, we will only enable the activity for use with Handoff
[currentActivity setEligibleForHandoff:YES];
[currentActivity setEligibleForSearch:NO]; // Defaults to NO
[currentActivity setEligibleForPublicIndexing:NO]; // Defaults to NO

// Set this activity as the current user activity
// Only one activity may be current at a time on a device. Calling this method invalidates any
other current activities.
[currentActivity becomeCurrent];
```

Handoff”。

NSUserActivity <https://riptutorial.com/zh-TW/ios/topic/10716/nsuseractivity>

54: NSUserDefaults

- `UserDefaults.standard.set(dic, forKey: "LoginSession") //Save value inside userdefaults`
 - `UserDefaults.standard.object(forKey: "LoginSession") as? [String:AnyObject] ?? [:]`
`//Get value from UserDefaults`

NSUserDefaultsDataTypeapp◦ [NSUserDefaults](#)

Examples

NSUserDefaults

<3

```
setBool(_:forKey:)
setFloat(_:forKey:)
setInteger(_:forKey:)
setObject(_:forKey:)
setDouble(_:forKey:)
setURL(_:forKey:)
```

3

3setinstedsetfolloed◦

```
set(_:forKey:)
```

Objective-C

```
-(void)setBool:(BOOL)value forKey:(nonnull NSString *)defaultName;
-(void)setFloat:(float)value forKey:(nonnull NSString *)defaultName;
-(void)setInteger:(NSInteger)value forKey:(nonnull NSString *)defaultName;
-(void)setObject:(nullable id)value forKey:(nonnull NSString *)defaultName;
-(void)setDouble:(double)value forKey:(nonnull NSString *)defaultName;
-(void)setURL:(nullable NSURL *)value forKey:(nonnull NSString *)defaultName;
```

<3

```
NSUserDefaults.standardUserDefaults.setObject("Netherlands", forKey: "HomeCountry")
```

3

```
UserDefaults.standard.set("Netherlands", forKey: "HomeCountry")
```

Objective-C

```
[[NSUserDefaults standardUserDefaults] setObject:@"Netherlands" forKey:@"HomeCountry"];
```

↳ `NSUserDefaults` CustomClass `NSCoding`。

```
public func encodeWithCoder(aCoder: NSCoder) {
    aCoder.encodeObject(name, forKey:"name")
    aCoder.encodeObject(unitId, forKey: "unitId")
}

required public init(coder aDecoder: NSCoder) {
    super.init()
    name = aDecoder.decodeObjectForKey("name") as? String
    unitId = aDecoder.decodeIntegerForKey("unitId") as? NSInteger
}
```

Objective-C

```
- (id)initWithCoder:(NSCoder *)coder {
    self = [super init];
    if (self) {
        name = [coder decodeObjectForKey:@"name"];
        unitId = [coder decodeIntegerForKey:@"unitId"];
    }
    return self;
}

- (void)encodeWithCoder:(NSCoder*)coder {
    [coder encodeObject:name forKey:@"name"];
    [coder encodeInteger:unitId forKey:@"unitId"];
}
```

NSUserDefaults

```
arrayForKey(_:)
boolForKey(_:)
dataForKey(_:)
dictionaryForKey(_:)
floatForKey(_:)
integerForKey(_:)
objectForKey(_:)
stringArrayForKey(_:)
stringForKey(_:)
doubleForKey(_:)
URLForKey(_:)
```

Objective-C


```

-(nullable NSArray *)arrayForKey:(nonnull NSString *)defaultName;
-(BOOL)boolForKey:(nonnull NSString *)defaultName;
-(nullable NSData *)dataForKey:(nonnull NSString *)defaultName;
-(nullable NSDictionary<NSString *, id> *)dictionaryForKey:(nonnull NSString *)defaultName;
-(float)floatForKey:(nonnull NSString *)defaultName;
-(NSInteger)integerForKey:(nonnull NSString *)defaultName;
-(nullable id)objectForKey:(nonnull NSString *)key;
-(nullable NSArray<NSString *> *)stringArrayForKey:(nonnull NSString *)defaultName;
-(nullable NSString *)stringForKey:(nonnull NSString *)defaultName;
-(double)doubleForKey:(nonnull NSString *)defaultName;
-(nullable NSURL *)URLForKey:(nonnull NSString *)defaultName;

```

```
let homeCountry = UserDefaults.standardUserDefaults().stringForKey("HomeCountry")
```

Objective-C

```
NSString *homeCountry = [[NSUserDefaults standardUserDefaults] stringForKey:@"HomeCountry"];
```

```
NSUserDefaults° synchronize°
```

```
NSUserDefaults.standardUserDefaults().synchronize()
```

Objective-C

```
[[NSUserDefaults standardUserDefaults] synchronize];
```

```
NSUserDefaultsNSUserDefaults °
```

```
NSUserDefaults ° setHighScorehighScore ° °
```

```

public class ScoreManager: NSObject {

    let highScoreDefaultKey = "HighScoreDefaultKey"

    var highScore = {
        set {
            // This method includes your implementation for saving the high score
            // You can use UserDefaults or any other data store like CoreData or
            // SQLite etc.

            UserDefaults.standardUserDefaults().setInteger(newValue, forKey:
highScoreDefaultKey)
            UserDefaults.standardUserDefaults().synchronize()
        }
        get {
            //This method includes your implementation for reading the high score

            let score =
NSUserDefaults.standardUserDefaults().objectForKey(highScoreDefaultKey)

            if (score != nil) {

```



```
let bundleIdentifier = NSBundle.mainBundle().bundleIdentifier()

NSUserDefaults.standardUserDefaults().removePersistentDomainForName(bundleIdentifier)
```

Objective-C

```
NSString *bundleIdentifier = [[NSBundle mainBundle] bundleIdentifier];

[[NSUserDefaults standardUserDefaults] removePersistentDomainForName: bundleIdentifier];
```

UserDefaultsSwift 3

UserDefault。 *UserDefaults*。

3

```
import Foundation

public struct Session {

    fileprivate static let defaults = UserDefaults.standard

    enum userValues: String {
        case auth_token
        case email
        case fname
        case mobile
        case title
        case userId
        case userType
        case OTP
        case isApproved
    }

    //MARK: - Getting here User Details
    static func getUserSessionDetails()->[String:AnyObject]? {
        let dictionary = defaults.object(forKey: "LoginSession") as? [String:AnyObject]
        return dictionary
    }

    //MARK: - Saving Device Token
    static func saveDeviceToken(_ token:String){
        guard (gettingDeviceToken() ?? "").isEmpty else {
            return
        }
        defaults.removeObject(forKey: "deviceToken")
        defaults.set(token, forKey: "deviceToken")
        defaults.synchronize()
    }

    //MARK: - Getting Token here
    static func gettingDeviceToken()->String?{
        let token = defaults.object(forKey: "deviceToken") as? String
        if token == nil{
            return ""
        }
    }
}
```

```

        }else{ return token}
    }

    //MARK: - Setting here User Details
    static func setUserSessionDetails(_ dic :[String : AnyObject]){
        defaults.removeObject(forKey: "LoginSession")
        defaults.set(dic, forKey: "LoginSession")
        defaults.synchronize()
    }

    //MARK:- Removing here all Default Values
    static func userSessionLogout(){
        //Set Activity
        defaults.removeObject(forKey: "LoginSession")
        defaults.synchronize()
    }

    //MARK: - Get value from session here
    static func getUserValues(value: userValues) -> String? {
        let dic = getUserSessionDetails() ?? [:]
        guard let value = dic[value.rawValue] else{
            return ""
        }
        return value as? String
    }
}
}

```

UserDefaults

```

//Saving user Details
Session.setUserSessionDetails(json ?? [:])

//Retriving user Details
let userId = Session.getUserValues(value: .userId) ?? ""

```

NSUserDefaults <https://riptutorial.com/zh-TW/ios/topic/3150/nsuserdefaults>

55: Objective-C

Examples

◦

UIView

Objective-C

```
#import <objc/runtime.h>

static char ActivityIndicatorKey;

@implementation UIView (ActivityIndicator)

- (UIActivityIndicatorView *)activityIndicator {
    return (UIActivityIndicatorView *)objc_getAssociatedObject(self, &ActivityIndicatorKey);
}

- (void)setActivityIndicator: (UIActivityIndicatorView *)activityIndicator {
    objc_setAssociatedObject(self, &ActivityIndicatorKey, activityIndicator,
OBJC_ASSOCIATION_RETAIN_NONATOMIC);
}

- (void)showActivityIndicator {
    UIActivityIndicatorView *activityIndicator = [[UIActivityIndicatorView alloc]
initWithActivityIndicatorStyle: UIActivityIndicatorViewStyleGray];

    [self setActivityIndicator:activityIndicator];

    activityIndicator.center = self.center;
    activityIndicator.autoresizingMask = UIViewAutoresizingFlexibleTopMargin |
UIViewAutoresizingFlexibleLeftMargin | UIViewAutoresizingFlexibleRightMargin |
UIViewAutoresizingFlexibleBottomMargin;

    [activityIndicator startAnimating];

    [self addSubview: activityIndicator];
}

- (void)hideActivityIndicator {
    UIActivityIndicatorView * activityIndicator = [self activityIndicator];

    if (activityIndicator != nil) {
        [[self activityIndicator] removeFromSuperview];
    }
}

@end
```

SwiftObjective-C

SWIFT

```

extension UIView {
    private struct AssociatedKeys {
        static var activityIndicator = "UIView.ActivityIndicatorView"
    }

    private var activityIndicatorView: UIActivityIndicatorView? {
        get {
            return objc_getAssociatedObject(self, &AssociatedKeys.activityIndicator) as?
UIActivityIndicatorView
        }
        set (activityIndicatorView) {
            objc_setAssociatedObject(self, &AssociatedKeys.activityIndicator,
activityIndicatorView, .OBJC_ASSOCIATION_RETAIN_NONATOMIC)
        }
    }

    func showActivityIndicator() {
        activityIndicatorView = UIActivityIndicatorView(activityIndicatorStyle: .gray)
        activityIndicatorView.center = center
        activityIndicatorView.autoresizingMask = [.flexibleLeftMargin, .flexibleRightMargin,
.flexibleTopMargin, .flexibleBottomMargin]

        activityIndicatorView.startAnimating()

        addSubview(activityIndicatorView)
    }

    func hideActivityIndicator() {
        activityIndicatorView.removeFromSuperview()
    }
}

```

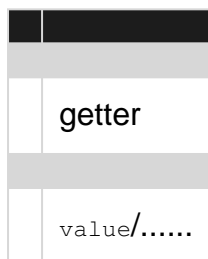
Objective-C <https://riptutorial.com/zh-TW/ios/topic/10120/objective-c>

56: Objective-C

Objective-C iOS 3.1.

◦

- void objc_setAssociatedObjectid objcvoid * keyid valueobjc_AssociationPolicy policy
- id objc_getAssociatedObjectid objcvoid * key
- void objc_removeAssociatedObjectsid object



[NSHipster](#)

[@kostiakoal](#)

[kingscocoa](#)

Examples

NSStringSomeClass ◦

```
#import <objc/runtime.h>

@interface SomeClass (MyCategory)
// This is the property wrapping the associated object. below we implement the setter and
getter which actually utilize the object association
@property (nonatomic, retain) NSString *associated;
@end

@implementation SomeClass (MyCategory)

- (void)setAssociated:(NSString *)object {
    objc_setAssociatedObject(self, @selector(associated), object,
                            OBJC_ASSOCIATION_RETAIN_NONATOMIC);
}

- (NSString *)associated {
    return objc_getAssociatedObject(self, @selector(associated));
}
```

```
SomeClass *instance = [SomeClass alloc] init];
```

```
instance.associated = @"this property is an associated object under the hood";
```

Objective-C <https://riptutorial.com/zh-TW/ios/topic/9102/objective-c>

57: OpenGL

OpenGL ES iOS 3D。

Examples

[Git repo](#) 3D。 OpenGL。 Xcode。

OpenGL <https://riptutorial.com/zh-TW/ios/topic/9324/opengl>

58: PDFiOS

Examples

PDF

```
UIGraphicsBeginPDFContextToFile(fileName, CGRectZero, nil);

UIGraphicsBeginPDFPageWithInfo(CGRectMake(0, 0, 612, 792), nil);

[self drawText];

UIGraphicsEndPDFContext();
```

fileName

```
NSString* temporaryFile = @"firstIOS.PDF";
NSArray *arrayPaths =
    NSSearchPathForDirectoriesInDomains(
        NSDocumentDirectory,
        NSUserDomainMask,
        YES);

NSString *path = [arrayPaths objectAtIndex:0];

NSString* fileName = [path stringByAppendingPathComponent:fileName];
```

drawText

```
(void)drawText
{
    NSString* textToDraw = @"Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when
an unknown printer took a galley of type and scrambled it to make a type specimen book.";

    CFStringRef stringRef = (__bridge CFStringRef)textToDraw;

    CFAttributedStringRef currentText = CFAttributedStringCreate(NULL, stringRef, NULL);

    CTFramesetterRef framesetter = CTFramesetterCreateWithAttributedString(currentText);

    CGRect frameRect = CGRectMake(0, 0, 300, 100);

    CGMutablePathRef framePath = CGPathCreateMutable();

    CGPathAddRect(framePath, NULL, frameRect);

    CFRange currentRange = CFRangeMake(0, 0);

    CTFrameRef frameRef = CTFramesetterCreateFrame(framesetter, currentRange, framePath,
    NULL);
    CGPathRelease(framePath);

    CGContextRef currentContext = UIGraphicsGetCurrentContext();
```

```
CGContextSetTextMatrix(currentContext, CGAffineTransformIdentity);

CGContextTranslateCTM(currentContext, 0, 450);

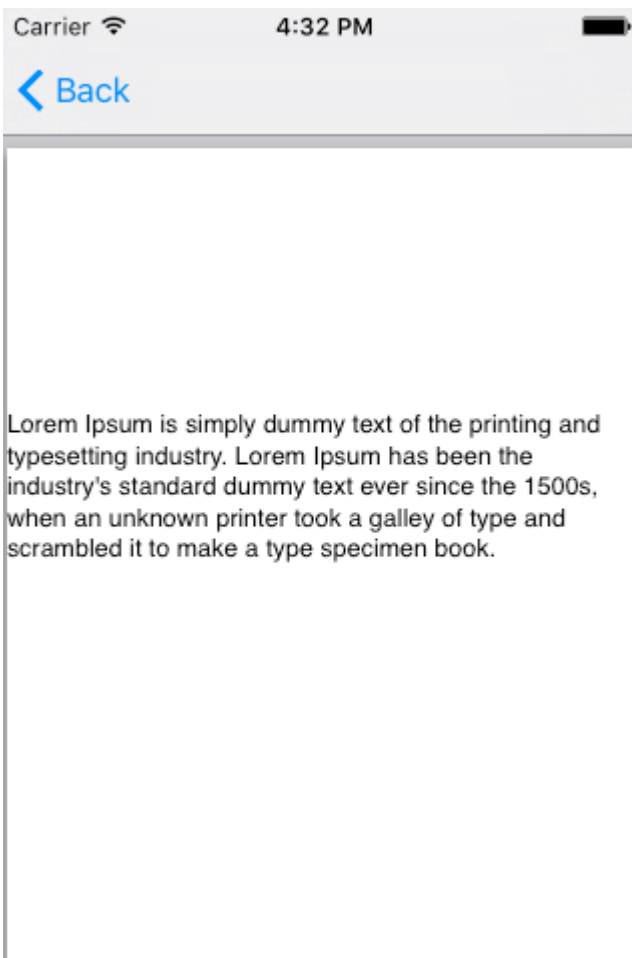
CGContextScaleCTM(currentContext, 2, -2);

CTFrameDraw(frameRef, currentContext);

CFRelease(frameRef);

CFRelease(stringRef);

CFRelease(framesetter);
}
```



PDF

```
NSString* fileName = @"firstIOS.PDF";

NSArray *arrayPaths =
NSSearchPathForDirectoriesInDomains(
    NSDocumentDirectory,
    NSUserDomainMask,
    YES);
```

```

NSString *path = [arrayPaths objectAtIndex:0];

NSString* pdfFileName = [path stringByAppendingPathComponent:fileName];

UIWebView* webView = [[UIWebView alloc] initWithFrame:CGRectMake(0, 0, 320, 480)];

NSURL *url = [NSURL fileURLWithPath:pdfFileName];

NSURLRequest *request = [NSURLRequest requestWithURL:url];

[webView setScalesPageToFit:YES];

[webView loadRequest:request];

[self.view addSubview:webView];

```

PDF

```

UIGraphicsBeginPDFContextToFile(fileName, CGRectZero, nil);

    UIGraphicsBeginPDFPageWithInfo(CGRectMake(0, 0, 600, 792), nil);

    UIGraphicsBeginPDFPageWithInfo(CGRectMake(0, 0, 600, 792), nil);

    UIGraphicsBeginPDFPageWithInfo(CGRectMake(0, 0, 600, 792), nil);

    UIGraphicsEndPDFContext();

```

UIWebViewMicrosoft DocumentPDF

```
#define kPaperSizeA4 CGSizeMake(595.2,841.8)
```

UIPrintPageRenderer

```

@interface UIPrintPageRenderer (PDF)

- (NSData*) printToPDF;

@end

@implementation UIPrintPageRenderer (PDF)

- (NSData*) printToPDF
{
    NSMutableData *pdfData = [NSMutableData data];
    UIGraphicsBeginPDFContextToData(pdfData, self.paperRect, nil);
    [self prepareForDrawingPages:NSMakeRange(0, self.numberOfPages)];
    CGRect bounds = UIGraphicsGetPDFContextBounds();
    for (int i = 0; i < self.numberOfPages; i++)
    {
        UIGraphicsBeginPDFPage();
        [self drawPageAtIndex:i inRect:bounds];
    }
    UIGraphicsEndPDFContext();
    return pdfData;
}

```

```
}  
@end
```

UIWebView

```
-(void)createPDF:(UIWebView *)webView {  
  
    UIPrintPageRenderer *render = [[UIPrintPageRenderer alloc] init];  
    [render addPrintFormatter:webView.viewPrintFormatter startingAtPageAtIndex:0];  
  
    float padding = 10.0f;  
    CGRect paperRect = CGRectMake(0, 0, kPaperSizeA4.width, kPaperSizeA4.height);  
    CGRect printableRect = CGRectMake(padding, padding, kPaperSizeA4.width-(padding * 2),  
    kPaperSizeA4.height-(padding * 2));  
  
    [render setValue:[NSValue valueWithCGRect:paperRect] forKey:@"paperRect"];  
    [render setValue:[NSValue valueWithCGRect:printableRect] forKey:@"printableRect"];  
  
    NSData *pdfData = [render printToPDF];  
  
    dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{  
  
        if (pdfData) {  
            [pdfData writeToFile:directoryPath atomically: YES];  
        }  
        else  
        {  
            NSLog(@"PDF couldnot be created");  
        }  
    });}
```

PDFiOS <https://riptutorial.com/zh-TW/ios/topic/2416/pdfios>

59: plist iOS

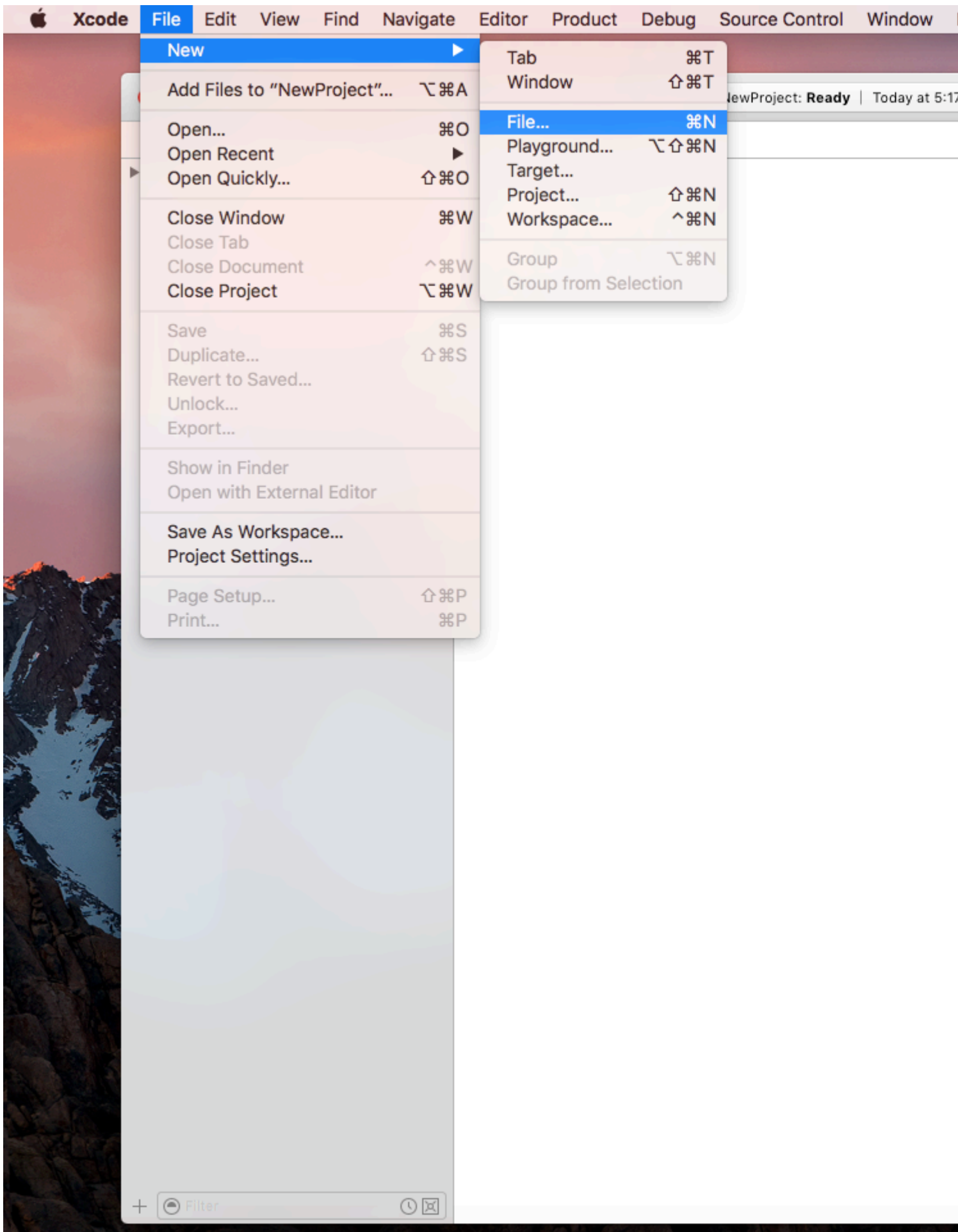
PlistiOS。 PlistArrayDictionaries。 plist1。 app。 2。

Examples

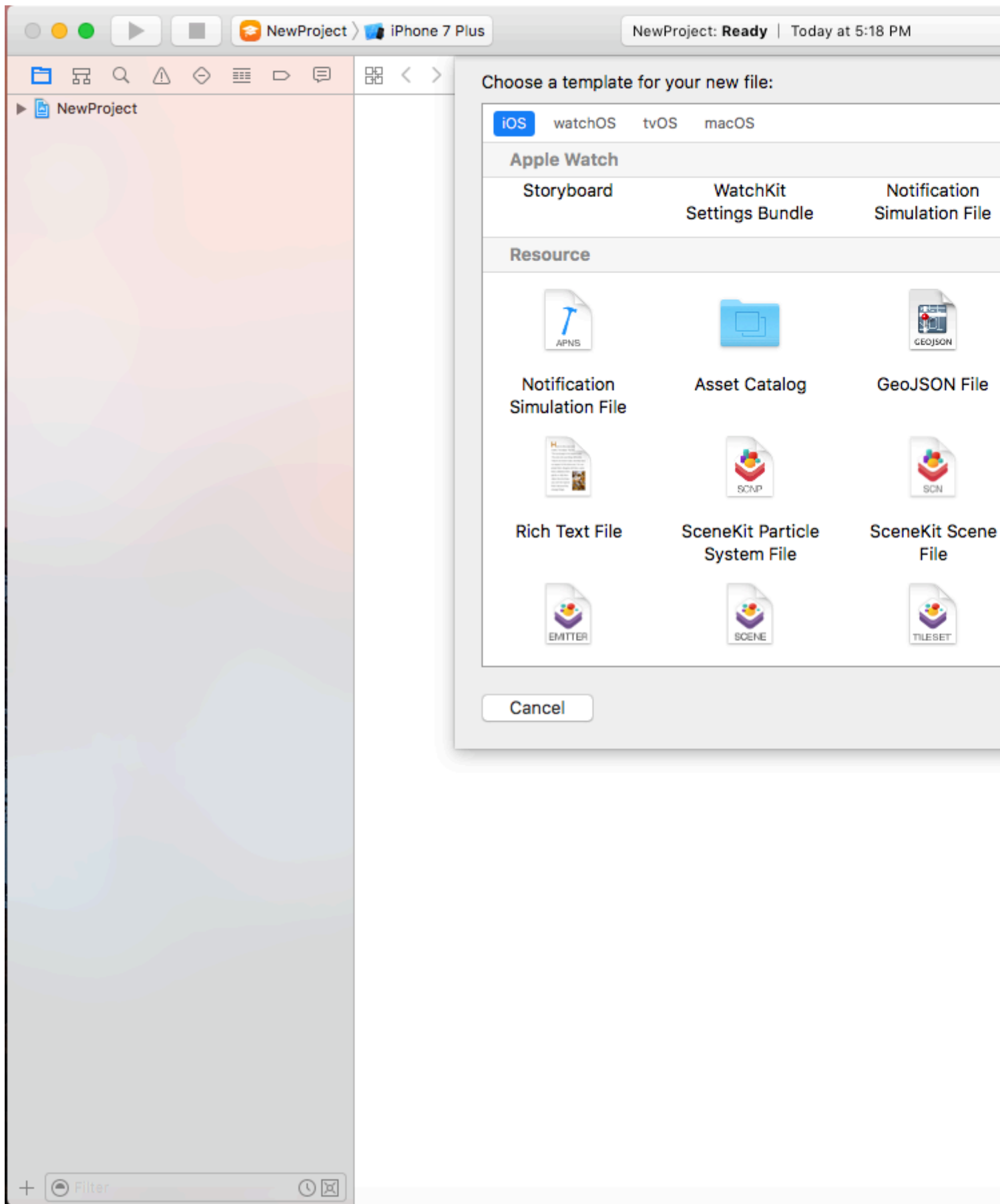
1.app。

plist

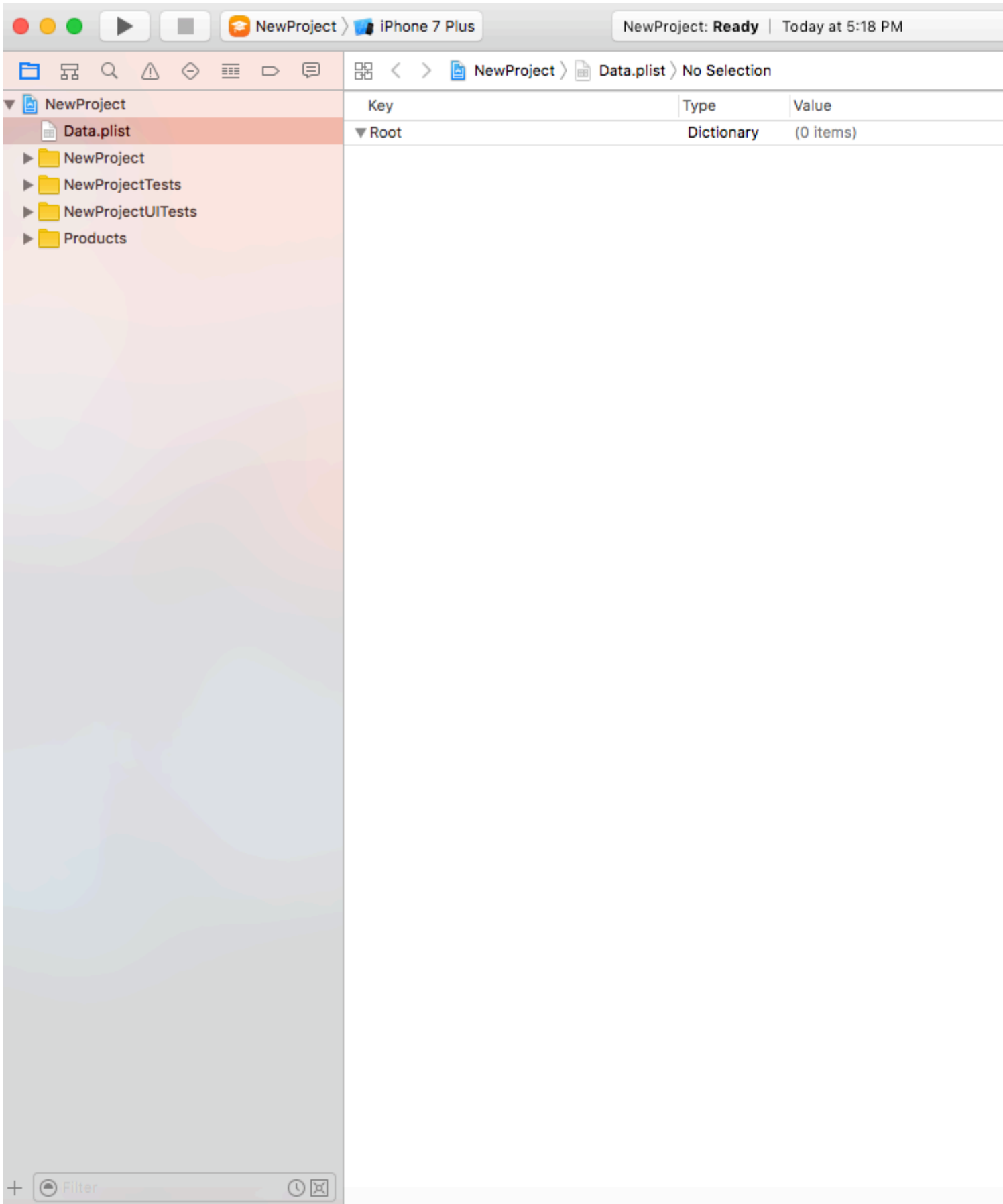
a



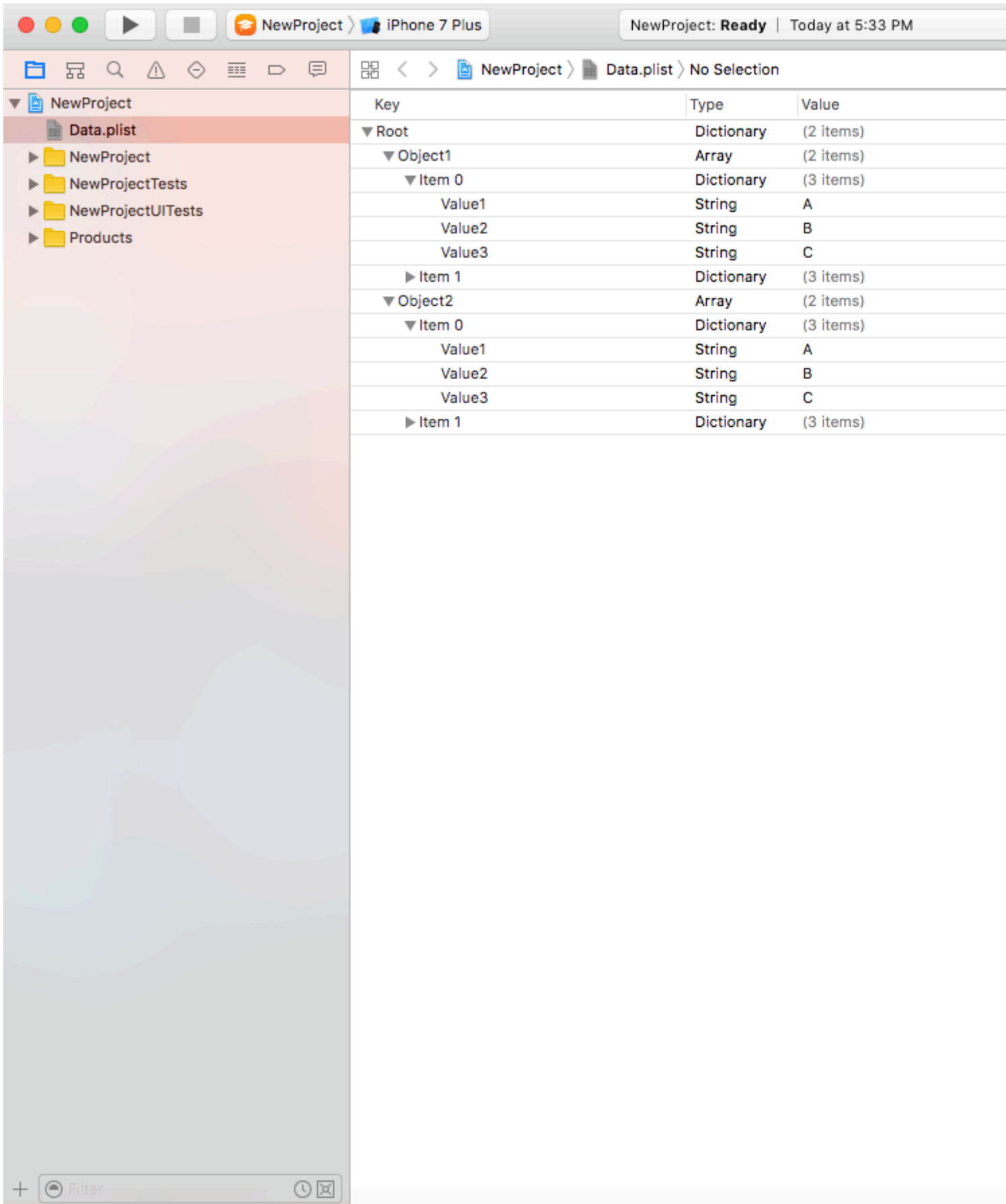
b



cdata.plist here



dplist



//bundleplistRoot Dictionary

```
NSMutableDictionary *dictRoot = [NSMutableDictionary dictionaryWithContentsOfFile:[NSBundle mainBundle]
pathForResource:@"Data" ofType:@"plist"];
```

////。

```
NSArray *arrayList = [NSArray arrayWithArray:[dictRoot objectForKey:@"Object1"]];

for(int i=0; i < [arrayList count]; i++)
{
    NSMutableDictionary *details=[arrayList objectAtIndex:i];
}
```

/Plist

plist。 plist。 plistplist。

plistDocument Directory。 plist。

plist

```
NSString *filePath = [[NSBundle mainBundle] pathForResource:@"Data" ofType:@"plist"];

NSDictionary *dict = [[NSDictionary alloc] initWithContentsOfFile:filePath];

NSDictionary *plistDict = dict;

NSFileManager *fileManager = [NSFileManager defaultManager];

NSString *error = nil;

NSData *plistData = [NSPropertyListSerialization dataFromPropertyList:plistDict
format:NSPropertyListXMLFormat_v1_0 errorDescription:&error];

if (![fileManager fileExistsAtPath: plistPath]) {

    if(plistData)
    {
        [plistData writeToFile:plistPath atomically:YES];
    }
}
else
{
}
```

Plist

```
NSArray *paths = NSSearchPathForDirectoriesInDomains (NSDocumentDirectory,
NSUserDomainMask, YES);
NSString *documentsPath = [paths objectAtIndex:0];
NSString *plistPath = [documentsPath stringByAppendingPathComponent:@"Data.plist"];
NSDictionary *dict = [[NSDictionary alloc] initWithContentsOfFile:plistPath];

NSArray *usersArray = [dict objectForKey:@"Object1"];
```

plist。

plist iOS <https://riptutorial.com/zh-TW/ios/topic/8141/plist-ios>

60: QR

QR。 iOS/iOS 7/AVFoundation/QR。 API/QR。

Examples

UIViewController/QR

```
import AVFoundation
class QRScannerViewController: UIViewController,
    AVCaptureMetadataOutputObjectsDelegate {

    func viewDidLoad() {
        self.initCaptureSession()
    }

    private func initCaptureSession() {
        let captureDevice = AVCaptureDevice
            .defaultDevice(withMediaType: AVMediaTypeVideo)
        do {
            let input = try AVCaptureDeviceInput(device: captureDevice)
            let captureMetadataOutput = AVCaptureMetadataOutput()
            self.captureSession?.addOutput(captureMetadataOutput)
            captureMetadataOutput.setMetadataObjectsDelegate(self,
                queue: DispatchQueue.main)
            captureMetadataOutput
                .metadataObjectTypes = [AVMetadataObjectTypeQRCode]

            self.videoPreviewLayer =
                AVCaptureVideoPreviewLayer(session: self.captureSession)
            self.videoPreviewLayer?
                .videoGravity = AVLayerVideoGravityResizeAspectFill
            self.videoPreviewLayer?.frame =
                self.view.layer.bounds

            self._viewController?.view.layer
                .addSublayer(videoPreviewLayer!)
            self.captureSession?.startRunning()
        } catch {
            //TODO: handle input open error
        }
    }

    private func dismissCaptureSession() {
        if let running = self.captureSession?.isRunning, running {
            self.captureSession?.stopRunning()
        }
        self.captureSession = nil
        self.videoPreviewLayer?.removeFromSuperLayer()
        self.videoPreviewLayer = nil
    }

    func captureOutput(_ captureOutput: AVCaptureOutput,
        didOutputMetadataObjects metadataObjects: [Any]!,
        from connection: AVCaptureConnection) {
        guard metadataObjects != nil && metadataObjects.count != 0 else {
            //Nothing captured
        }
    }
}
```

```

        return
    }

    if let metadataObj =
        metadataObjects[0] as? AVMetadataMachineReadableCodeObject {
        guard metadataObj.type == AVMetadataObjectTypeQRCode else {
            return
        }

        let barCodeObject = videoPreviewLayer?
            .transformedMetadataObject(for:
                metadataObj as AVMetadataMachineReadableCodeObject)
            as! AVMetadataMachineReadableCodeObject

        if let qrValue = metadataObj.stringValue {
            self.handleQRRead(value: qrValue)
        }
    }
}

private handleQRRead(value: String) {
    //TODO: Handle the read qr
}
private captureSession: AVCaptureSession?
private videoPreviewLayer: AVCaptureVideo
}

```

handleQRRead - initCaptureSession - QRdismissCaptureSession -

AVFoundationQR

iOS 7QRzBarzXing ◦ AppleiOS 7AVCaptureMetaDataOutput◦

AVFoundationQR/AVCaptureSessioncaptureOutput:didOutputMetadataObjects:fromConnection: delegate
◦

1

AVFoundationAVCaptureMetadataOutputObjectsDelegate

```

import AVFoundation
class ViewController: UIViewController, AVCaptureMetadataOutputObjectsDelegate

```

2

QR◦ AVCaptureSession◦ viewDidLoad

```

// Create an instance of the AVCaptureDevice and provide the video as the media type
parameter.
let captureDevice = AVCaptureDevice.defaultDevice(withMediaType: AVMediaTypeVideo)

do {

```

```

    // Create an instance of the AVCaptureDeviceInput class using the device object and
    initialise capture session
    let input = try AVCaptureDeviceInput(device: captureDevice)
    captureSession = AVCaptureSession()
    captureSession?.addInput(input)

    // Create a instance of AVCaptureMetadataOutput object and set it as the output device the
    capture session.
    let captureMetadataOutput = AVCaptureMetadataOutput()
    captureSession?.addOutput(captureMetadataOutput)
    // Set delegate with a default dispatch queue
    captureMetadataOutput.setMetadataObjectsDelegate(self, queue: DispatchQueue.main)
    //set meta data object type as QR code, here we can add more then one type as well
    captureMetadataOutput.metadataObjectTypes = [AVMetadataObjectTypeQRCode]

    // Initialize the video preview layer and add it as a sublayer to the viewcontroller
    view's layer.
    videoPreviewLayer = AVCaptureVideoPreviewLayer(session: captureSession)
    videoPreviewLayer?.videoGravity = AVLayerVideoGravityResizeAspectFill
    videoPreviewLayer?.frame = view.layer.bounds
    view.layer.addSublayer(videoPreviewLayer!)

    // Start capture session.
    captureSession?.startRunning()
} catch {
    // If any error occurs, let the user know. For the example purpose just print out the
    error
    print(error)
    return
}

```

3

AVCaptureMetadataOutputObjectsDelegateQR

```

func captureOutput(_ captureOutput: AVCaptureOutput!, didOutputMetadataObjects
metadataObjects: [Any]!, from connection: AVCaptureConnection!) {

    // Check if the metadataObjects array contains at least one object. If not no QR code is
    in our video capture
    if metadataObjects == nil || metadataObjects.count == 0 {
        // NO QR code is being detected.
        return
    }

    // Get the metadata object and cast it to `AVMetadataMachineReadableCodeObject`
    let metadataObj = metadataObjects[0] as! AVMetadataMachineReadableCodeObject

    if metadataObj.type == AVMetadataObjectTypeQRCode {
        // If the found metadata is equal to the QR code metadata then get the string value
        from meta data
        let barCodeObject = videoPreviewLayer?.transformedMetadataObject(for: metadataObj)

        if metadataObj.stringValue != nil {
            // metadataObj.stringValue is our QR code
        }
    }
}

```

```
}
```

QR◦ videoPreviewLayervideoPreviewLayertransformedMetadataObject◦

```
let barCodeObject = videoPreviewLayer?.transformedMetadataObject(for: metadataObj)
    qrCodeFrameView?.frame = barCodeObject!.bounds
```

QR <https://riptutorial.com/zh-TW/ios/topic/7963/qr>

61: Safari

Examples

SFSafariViewControllerDelegate

SFSafariViewControllerDelegate SafariViewControllerDone

◦

```
class MyClass: SFSafariViewControllerDelegate {  
}
```

◦

```
func safariViewControllerDidFinish(controller: SFSafariViewController) {  
    // Dismiss the SafariViewController when done  
    controller.dismissViewControllerAnimated(true, completion: nil)  
}
```

SafariViewController

```
let safariVC = SFSafariViewController(URL: yourURL)  
safariVC.delegate = self
```

```
// Called when the initial URL load is complete.  
safariViewController(_ controller: SFSafariViewController, didCompleteInitialLoad  
didLoadSuccessfully: Bool) { }
```

```
// Called when the user taps an Action button.  
safariViewController(_ controller: SFSafariViewController, activityItemsFor URL: URL, title:  
String?) -> [UIActivity] { }
```

Safari

SSReadingListaddItem Safari

```
let readingList = SSReadingList.default()  
readingList?.addItem(with: yourURL, title: "optional title", previewText: "optional preview  
text")
```

nil

supportsURL URL

```
SSReadingList.default().supportsURL(URL(string: "https://example.com")!)
```


truefalseSafariURL。 URL。

SafariViewControllerURL

。

```
import SafariServices
//Objective-C
@import SafariServices;
```

SafariViewController。

```
let safariVC = SFSafariViewController(URL: URL(string: "your_url")!)
//Objective-C
@import SafariServices;
NSURL *URL = [NSURL URLWithString:[NSString stringWithFormat:@"http://www.google.com"]];
SFSafariViewController *sfvc = [[SFSafariViewController alloc] initWithURL:URL];
```

SafariViewController。

```
let safariVC = SFSafariViewController(URL: URL(string: "your_url")!, entersReaderIfAvailable:
true)
//Objective-C
NSURL *URL = [NSURL URLWithString:[NSString stringWithFormat:@"http://www.google.com"]];
SFSafariViewController *sfvc = [[SFSafariViewController alloc] initWithURL:URL
entersReaderIfAvailable:YES];
```

。

```
present(safariVC, animated: true, completion: nil)
//Objective-C
[self presentViewController:sfvc animated:YES completion:nil];
```

Safari <https://riptutorial.com/zh-TW/ios/topic/1371/safari>

62: SLComposeViewController

Examples

TwitterFacebookSinaWeiboTencentWeiboSLComposeViewController

Objective-C

Social FrameworkXCode

```
#import "Social/Social.h"ViewController
```

Twitter

```
//- - To Share text on twitter - -
if([SLComposeViewController isAvailableForServiceType:SLSERVICE_TYPE_TWITTER])
{
    //Tweet
    SLComposeViewController *twitterVC=[SLComposeViewController
composeViewControllerForServiceType:SLSERVICE_TYPE_TWITTER];
    //To send link together with text
    [twitterVC addURL:[NSURL URLWithString:@"https://twitter.com/IbrahimH_ss_n"]];
    //To add a photo to a link
    [twitterVC addImage:[UIImage imageNamed:@"image"]];
    //Sending link and Image with the tweet
    [twitterVC setInitialText:text];
    /* While adding link and images in a tweet the effective length of a tweet i.e.
the number of characters which can be entered by the user decreases.
The default maximum length of a tweet is 140 characters*/
    [self presentViewController:twitterVC animated:YES completion:nil];
}
else
{//Shows alert if twitter is not signed in
    UIAlertController *alertCont=[UIAlertController alertControllerWithTitle:@"SocialShare"
message:@"You are not signed in to twitter."preferredStyle:UIAlertControllerStyleAlert];
    [self presentViewController:alertCont animated:YES completion:nil];
    UIAlertAction *okay=[UIAlertAction actionWithTitle:@"Okay"
style:UIAlertActionStyleDefault handler:nil];
    [alertCont addAction:okay];
}
}
```

Facebook

```
if([SLComposeViewController isAvailableForServiceType:SLSERVICE_TYPE_FACEBOOK])
{
    SLComposeViewController *fbVC=[SLComposeViewController
composeViewControllerForServiceType:SLSERVICE_TYPE_FACEBOOK];
    [fbVC setInitialText:text];
    //To send link together with text
    [fbVC addURL:[NSURL URLWithString:@"https://twitter.com/IbrahimH_ss_n"]];
    //To add a photo to a link
    [fbVC addImage:[UIImage imageNamed:@"image"]];
    [self presentViewController:fbVC animated:YES completion:nil];
}
```

```

}
else
{//Shows alert if twitter is not signed in
    UIAlertController *alertCont=[UIAlertController alertControllerWithTitle:@"SocialShare"
message:@"You are not signed in to twitter."preferredStyle:UIAlertControllerStyleAlert];
    [self presentViewController:alertCont animated:YES completion:nil];
    UIAlertAction *okay=[UIAlertAction actionWithTitle:@"Okay"
style:UIAlertActionStyleDefault handler:nil];
    [alertCont addAction:okay];
}

```

```

//-- SinaWeibo --
if([SLComposeViewController isAvailableForServiceType:SLServiceTypeSinaWeibo]){

    SLComposeViewController *SinaWeiboVC=[SLComposeViewController
composeViewControllerForServiceType:SLServiceTypeSinaWeibo];
    [SinaWeiboVC setInitialText:text];

    [self presentViewController:SinaWeiboVC animated:YES completion:nil];
}
else
{
    UIAlertController *alertCont=[UIAlertController alertControllerWithTitle:@"SocialShare"
message:@"You are not signed in to SinaWeibo."preferredStyle:UIAlertControllerStyleAlert];
    [self presentViewController:alertCont animated:YES completion:nil];
    UIAlertAction *okay=[UIAlertAction actionWithTitle:@"Okay" style:UIAlertActionStyleDefault
handler:nil];
    [alertCont addAction:okay];
}

```

```

//--TencentWeibo text share
if([SLComposeViewController isAvailableForServiceType:SLServiceTypeTencentWeibo])
{
    SLComposeViewController *tencentWeiboVC=[SLComposeViewController
composeViewControllerForServiceType:SLServiceTypeTencentWeibo];
    [tencentWeibo setInitialText:text];
    [self presentViewController:tencentWeibo animated:YES completion:nil];
}
else
{
    UIAlertController *alertCont=[UIAlertController alertControllerWithTitle:@"SocialShare"
message:@"You are not signed in to SinaWeibo."preferredStyle:UIAlertControllerStyleAlert];
    [self presentViewController:alertCont animated:YES completion:nil];
    UIAlertAction *okay=[UIAlertAction actionWithTitle:@"Okay" style:UIAlertActionStyleDefault
handler:nil];
    [alertCont addAction:okay];
}

```

SLComposeViewController <https://riptutorial.com/zh-TW/ios/topic/7366/slcomposeviewController>

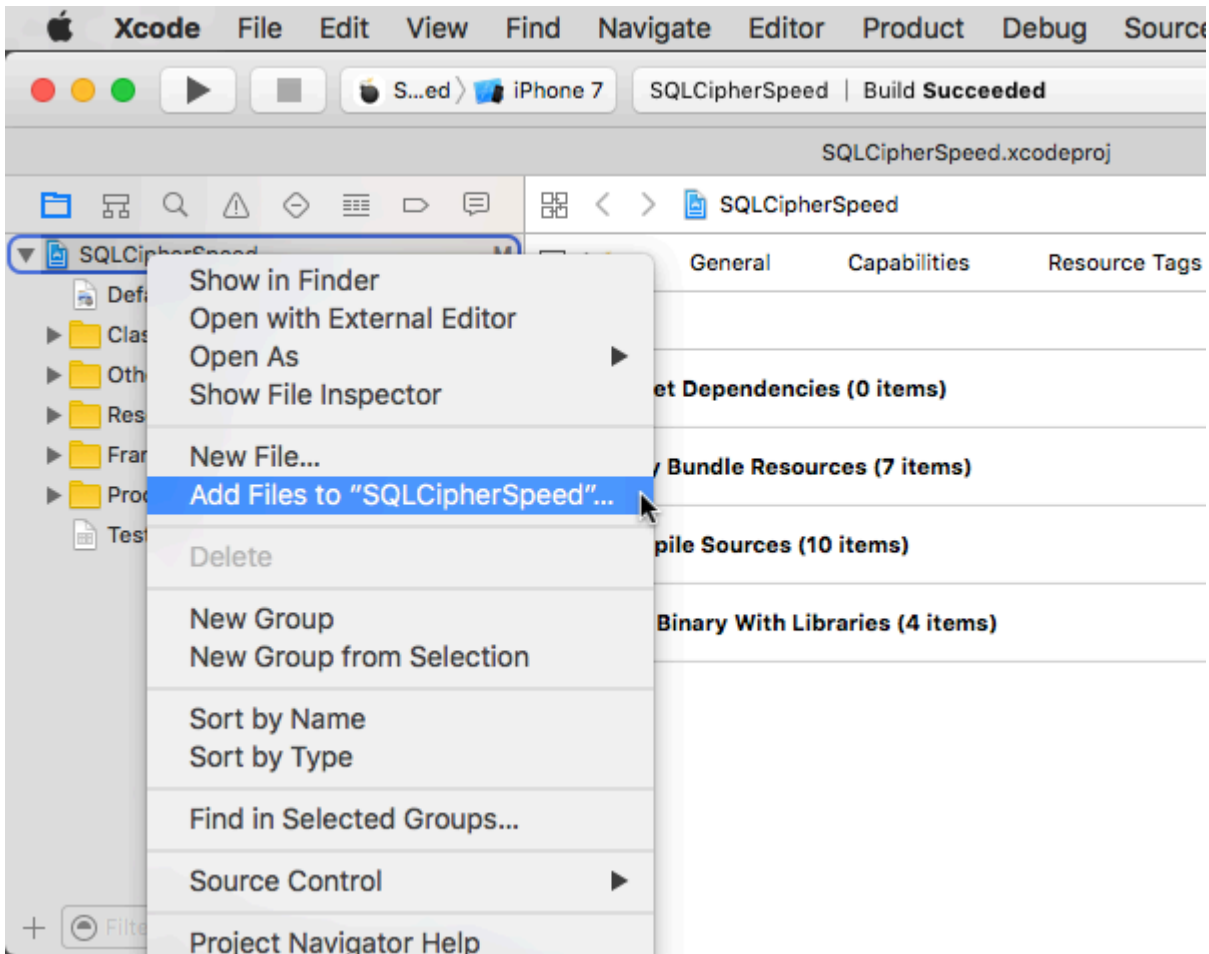
63: SqlCipher

SQLiteiOSAPI。 APIObjective-CFMDB。 。

1. GitSQLCipher

```
$ git clone https://github.com/sqlcipher/sqlcipher.git
```

2. “””。 SQLCipheriOSsqlcipher。 **sqlcipher.xcodeproj**

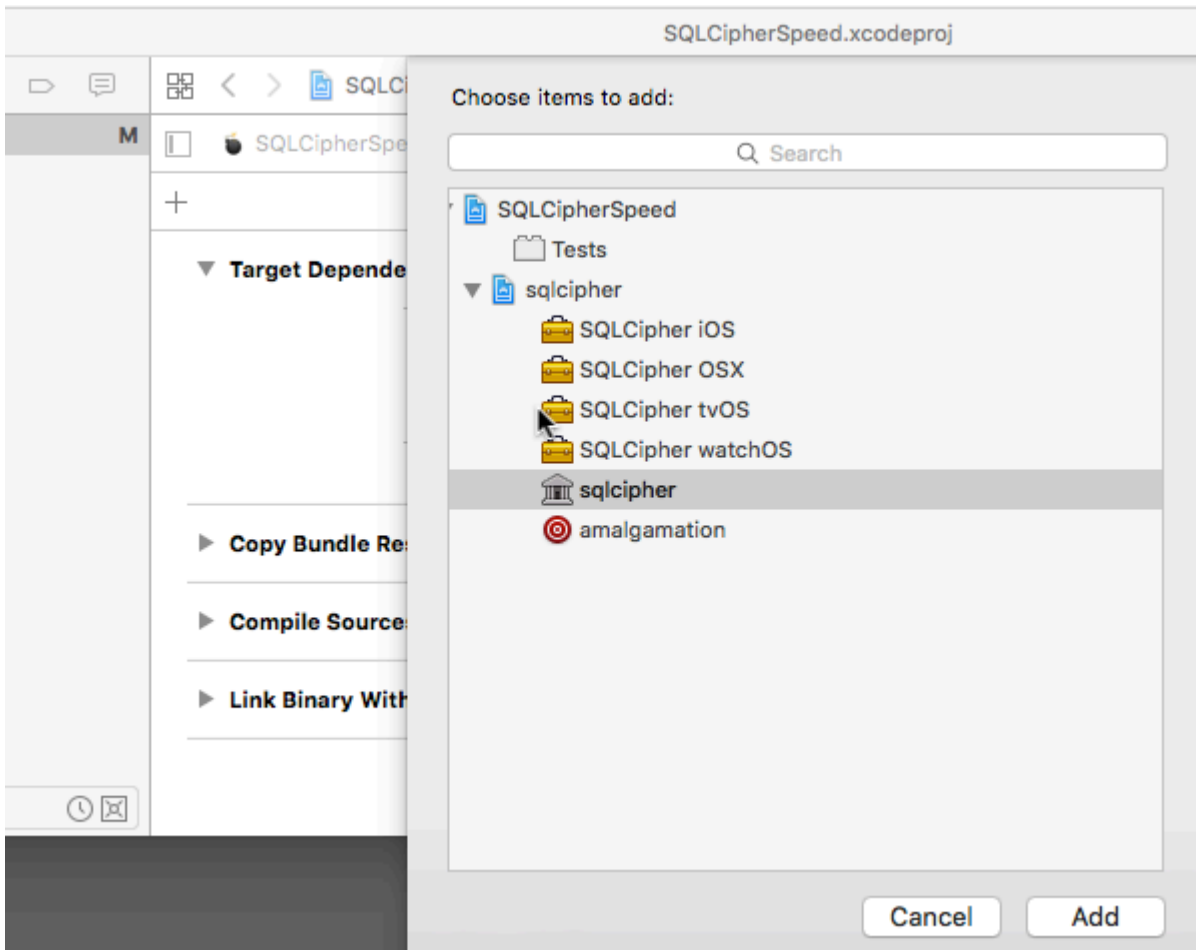


3. “””。 **\$PROJECT_DIR/ sqlcipher / src**

4. “Other Linker Flags” **\$BUILT_PRODUCTS_DIR/libsqlcipher.a**

5. “Other C Flags” **-DSQLITE_HAS_CODEC**

6. Target Dependencies+。 **sqlcipher**



7. Link Binary With Libraries+**libsqlcipher.a**.



8. Link With Libraries `Security.framework` ◦

Examples

◦

```

-(void) checkAndOpenDB{
    sqlite3 *db;
    NSString *strPassword = @"password";

    if (sqlite3_open_v2([[databaseURL path] UTF8String], &db, SQLITE_OPEN_READWRITE |
SQLITE_OPEN_CREATE, NULL) == SQLITE_OK) {
        const char* key = [strPassword UTF8String];
        sqlite3_key(db, key, (int)strlen(key));
        if (sqlite3_exec(db1, (const char*) "SELECT count(*) FROM sqlite_master;", NULL,
NULL, NULL) == SQLITE_OK) {
            NSLog(@"Password is correct, or a new database has been initialized");
        } else {
            NSLog(@"Incorrect password!");
        }
        sqlite3_close(db);
    }
}

- (NSURL *)databaseURL
{
    NSArray *URLs = [[NSFileManager defaultManager] URLsForDirectory:NSDocumentDirectory
inDomains:NSUserDomainMask];

```

```
NSURL *directoryURL = [URLs firstObject];
NSURL *databaseURL = [directoryURL URLByAppendingPathComponent:@"database.sqlite"];
return databaseURL;
}
```

SqlCipher <https://riptutorial.com/zh-TW/ios/topic/9969/sqlcipher>

64: StoreKit

Examples

App Store

SKProductsRequest

```
import StoreKit

let productIdentifierSet = Set(["yellowSubmarine", "pennyLane"])
let productsRequest = SKProductsRequest(productIdentifiers: productIdentifierSet)
```

productsRequest ◦ SKProductsRequestDelegateNSObject FoundationproductsRequest

```
class PaymentManager: NSObject, SKProductsRequestDelegate {

    var products: [SKProduct] = []

    func productsRequest(request: SKProductsRequest,
                        didReceiveResponse response: SKProductsResponse) {

        products = response.products

    }

}
```

productsRequestPaymentManagerproducts-requeststart()

```
let paymentManager = PaymentManager()
productsRequest.delegate = paymentManager
productsRequest.start()
```

paymentManager.products ◦

StoreKit <https://riptutorial.com/zh-TW/ios/topic/6025/storekit>

65: AppDelegate rootViewController main login / onboarding

- NotesPhotos and Music do in iOS11◦

Storyboard◦ Main.storyboard◦ /◦ Onboarding.storyboard

-
- git
-

- AppDelegate

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    let isFirstRun = true // logic to determine goes here
    if isFirstRun {
        showOnboarding()
    }
    return true
}
```

Onboarding◦

1. App
2. Onboarding◦

AppDelegate◦

Examples

1

UIViewAnimationOptions - ◦

UIApplication.shared.keyWindow.rootViewController◦ Onboarding

UIView.transition(with:) UIViewAnimationOptions◦ Cross Dissolve◦ ◦

◦

```
// MARK: - Onboarding

extension AppDelegate {

    func showOnboarding() {
        if let window = UIApplication.shared.keyWindow, let onboardingViewController =
```

```

UIStoryboard(name: "Onboarding", bundle: nil).instantiateInitialViewController() as?
OnboardingViewController {
    onboardingViewController.delegate = self
    window.rootViewController = onboardingViewController
}
}

func hideOnboarding() {
    if let window = UIApplication.shared.keyWindow, let mainViewController =
UIStoryboard(name: "Main", bundle: nil).instantiateInitialViewController() {
        mainViewController.view.frame = window.bounds
        UIView.transition(with: window, duration: 0.5, options: .transitionCrossDissolve,
animations: {
            window.rootViewController = mainViewController
        }, completion: nil)
    }
}
}
}

```

2

Onboarding

[Apple -] [1]

◦

◦

ViewController.present() **API**

```

// MARK: - Onboarding

extension AppDelegate {

    func showOnboarding() {
        if let window = window, let onboardingViewController = UIStoryboard(name:
"Onboarding", bundle: nil).instantiateInitialViewController() as? OnboardingViewController {
            onboardingViewController.delegate = self
            window.makeKeyAndVisible()
            window.rootViewController?.present(onboardingViewController, animated: false,
completion: nil)
        }
    }

    func hideOnboarding() {
        if let window = UIApplication.shared.keyWindow {
            window.rootViewController?.dismiss(animated: true, completion: nil)
        }
    }
}
}

```

SwiftAppDelegaterootViewControllermainlogin / onboarding <https://riptutorial.com/zh-TW/ios/topic/10880/swift-appdelegaterootviewcontrollermainlogin---onboarding>

66: SwiftFCM

FCM <https://firebase.google.com/docs/cloud-messaging/ios/client>

Examples

SwiftFCM

swift ProjectFCM

1-Xcode. Podfile

```
$ cd
$ pod init
```

2-Pod. PodfilePod

```
pod'Firebase / Core'
pod'Firebase / Messaging'
```

3-pod.xcworkspaceXcode.

```
$ pod
$ open your-project.xcworkspace
```

4-[plist](#)GoogleService-Info.plist.

5-APNsFirebase. [APN](#)

6-appDelegate“import Firebase”

7-“applicationdidFinishLaunchingWithOptions”“FIRApp.configure”

8-

```
if #available(iOS 10.0, *) {
    let authOptions : UNAuthorizationOptions = [.Alert, .Badge, .Sound]
    UNUserNotificationCenter.currentNotificationCenter().requestAuthorizationWithOptions(
        authOptions,
        completionHandler: {_,_ in })

    // For iOS 10 display notification (sent via APNS)
    UNUserNotificationCenter.currentNotificationCenter().delegate = self
    // For iOS 10 data message (sent via FCM)
    FIRMessaging.messaging().remoteMessageDelegate = self
} else {
    let settings: UIUserNotificationSettings =
    UIUserNotificationSettings(forTypes: [.Alert, .Badge, .Sound], categories: nil)
    application.registerUserNotificationSettings(settings)
}
```

```
application.registerForRemoteNotifications()
```

9-

```
let token = FIRInstanceID.instanceID().token()!
```

10-appDelegate

```
func tokenRefreshNotification(notification: NSNotification) {
    if let refreshedToken = FIRInstanceID.instanceID().token() {
        print("InstanceID token: \(refreshedToken)")
    }

    // Connect to FCM since connection may have failed when attempted before having a token.
    connectToFcm()
}
```

11-fcmAppDelegate

```
func connectToFcm() {
    FIRMessaging.messaging().connectWithCompletion { (error) in
        if (error != nil) {
            print("Unable to connect with FCM. \(error)")
        } else {
            print("Connected to FCM.")
        }
    }
}
```

12-

```
func applicationDidEnterBackground(application: UIApplication) {
    FIRMessaging.messaging().disconnect()
    print("Disconnected from FCM.")
}
```

AppDelegate

fcm

SwiftFCM <https://riptutorial.com/zh-TW/ios/topic/7326/swiftfcm>

67: SwiftObjective-C

Examples

SwiftObjective-C

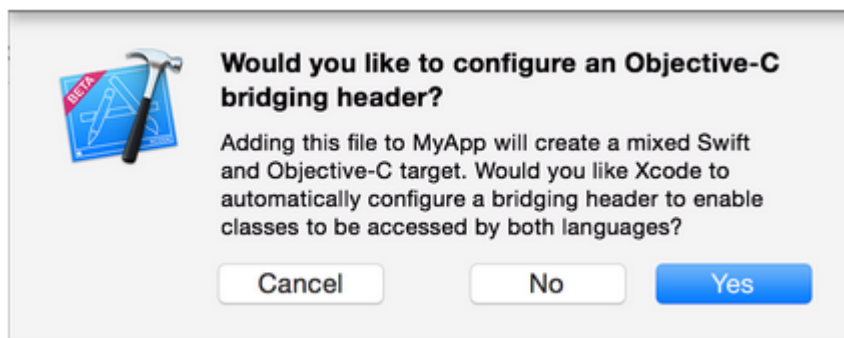
```
25 ◦ ObjC#import <Foundation/Foundation.h
```

1Objective-C - .m

```
.mCustomObject.m
```

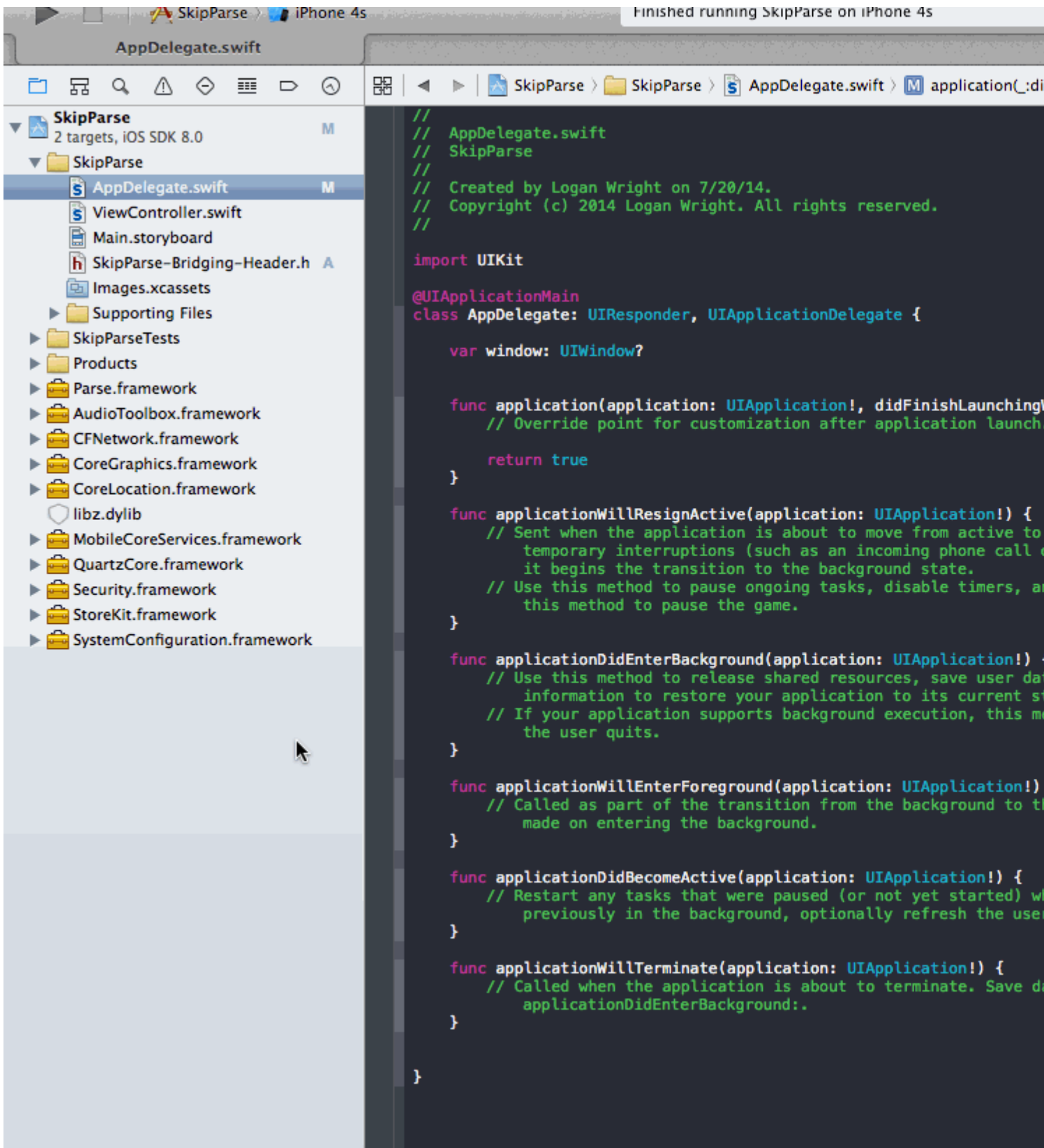
2

```
.m
```



```
.h<#YourProjectName#>-Bridging-Header.h
```

```
ObjCObjective-CXcode◦ .h
```



```
$ (SRCROOT) ◦ $ (SRCROOT) .xcodeproj ◦
```

```
$ (SRCROOT) /Folder/Folder/<#YourProjectName#>-Bridging-Header.h
```

3Objective-C - .h

```
.hCustomObject.h
```

4Objective-C

CustomObject.h

```
#import <Foundation/Foundation.h>

@interface CustomObject : NSObject

@property (strong, nonatomic) id someProperty;

- (void) someMethod;

@end
```

CustomObject.m

```
#import "CustomObject.h"

@implementation CustomObject

- (void) someMethod {
    NSLog(@"SomeMethod Ran");
}

@end
```

5Bridging-Header

YourProject-Bridging-Header.h

```
#import "CustomObject.h"
```

6

SomeSwiftFile.swift

```
var instanceOfCustomObject: CustomObject = CustomObject()
instanceOfCustomObject.someProperty = "Hello World"
println(instanceOfCustomObject.someProperty)
instanceOfCustomObject.someMethod()
```

o

Objective-CSwift

1Swift

.swiftMySwiftObject.swift

MySwiftObject.swift

```
import Foundation
```

```

class MySwiftObject : NSObject {

    var someProperty: AnyObject = "Some Initializer Val"

    init() {}

    func someFunction(someArg:AnyObject) -> String {
        var returnVal = "You sent me \(someArg)"
        return returnVal
    }

}

```

2SwiftObjC

SomeRandomClass.m

```
#import "<#YourProjectName#>-Swift.h"
```

```
<#YourProjectName#>-Swift.h
```

3

```

MySwiftObject * myOb = [MySwiftObject new];
NSLog(@"MyOb.someProperty: %@", myOb.someProperty);
myOb.someProperty = @"Hello World";
NSLog(@"MyOb.someProperty: %@", myOb.someProperty);
NSString * retString = [myOb someFunction:@"Arg"];
NSLog(@"RetString: %@", retString);

```

1. CodeCompletion。 w /“cmd + r”SwiftObjc。

2..swift dyld: Library not loaded: @rpath/libswift_stdlib_core.dylib [Xcode](#)。

3.@objcObjective-CSwiftSwift 2.0。 Swift。

[SwiftObjective-C](https://riptutorial.com/zh-TW/ios/topic/1497/swiftobjective-c) <https://riptutorial.com/zh-TW/ios/topic/1497/swiftobjective-c>

68: SWRevealViewController

SWRevealViewController. 55.

/o /o

AppleiOS .

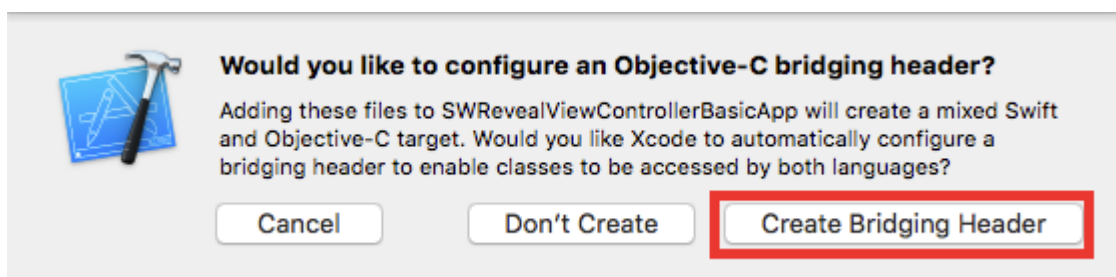
Examples

SWRevealViewController

swift

SWRevealViewController.hSWRevealViewController.m

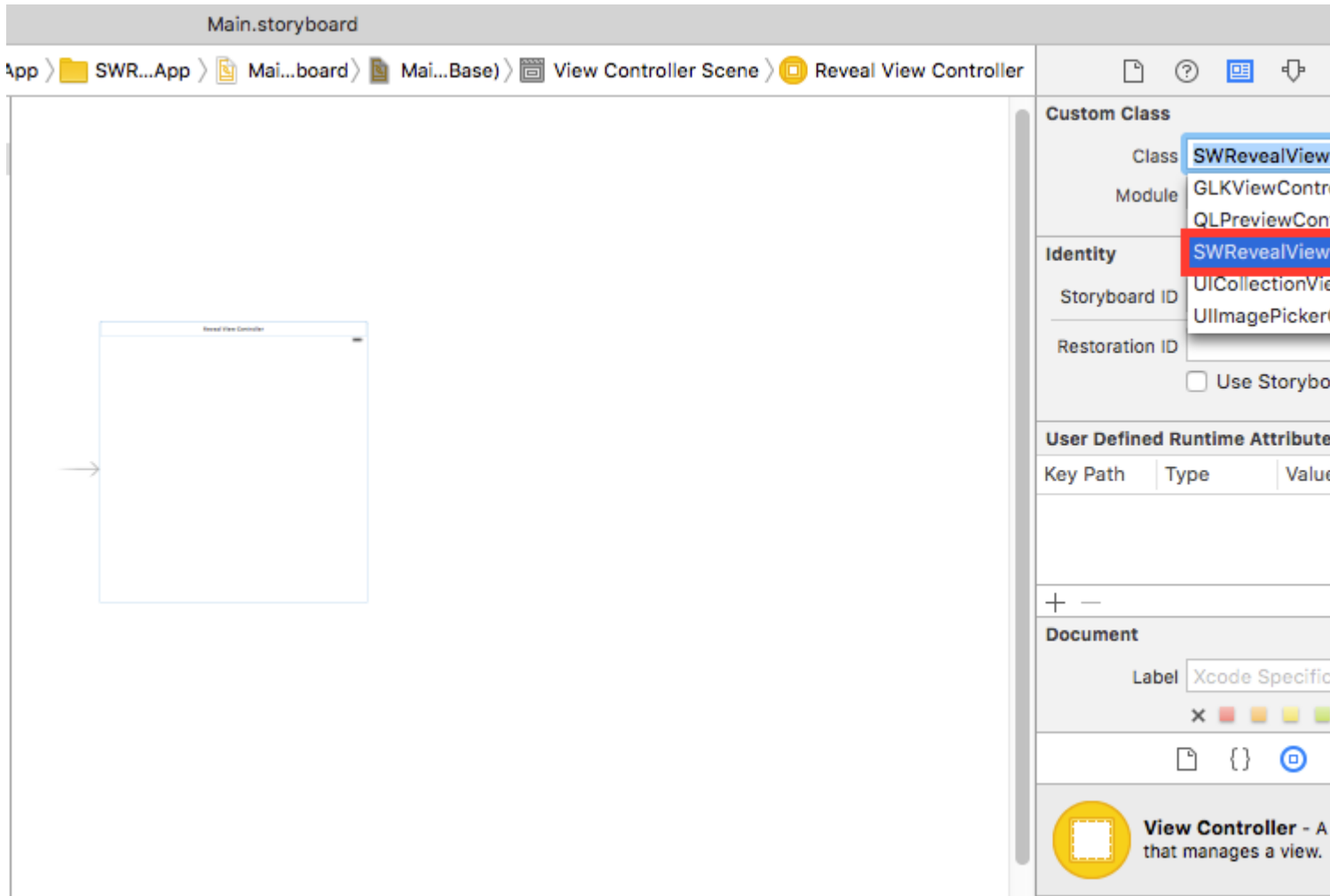
Create Bridging Header



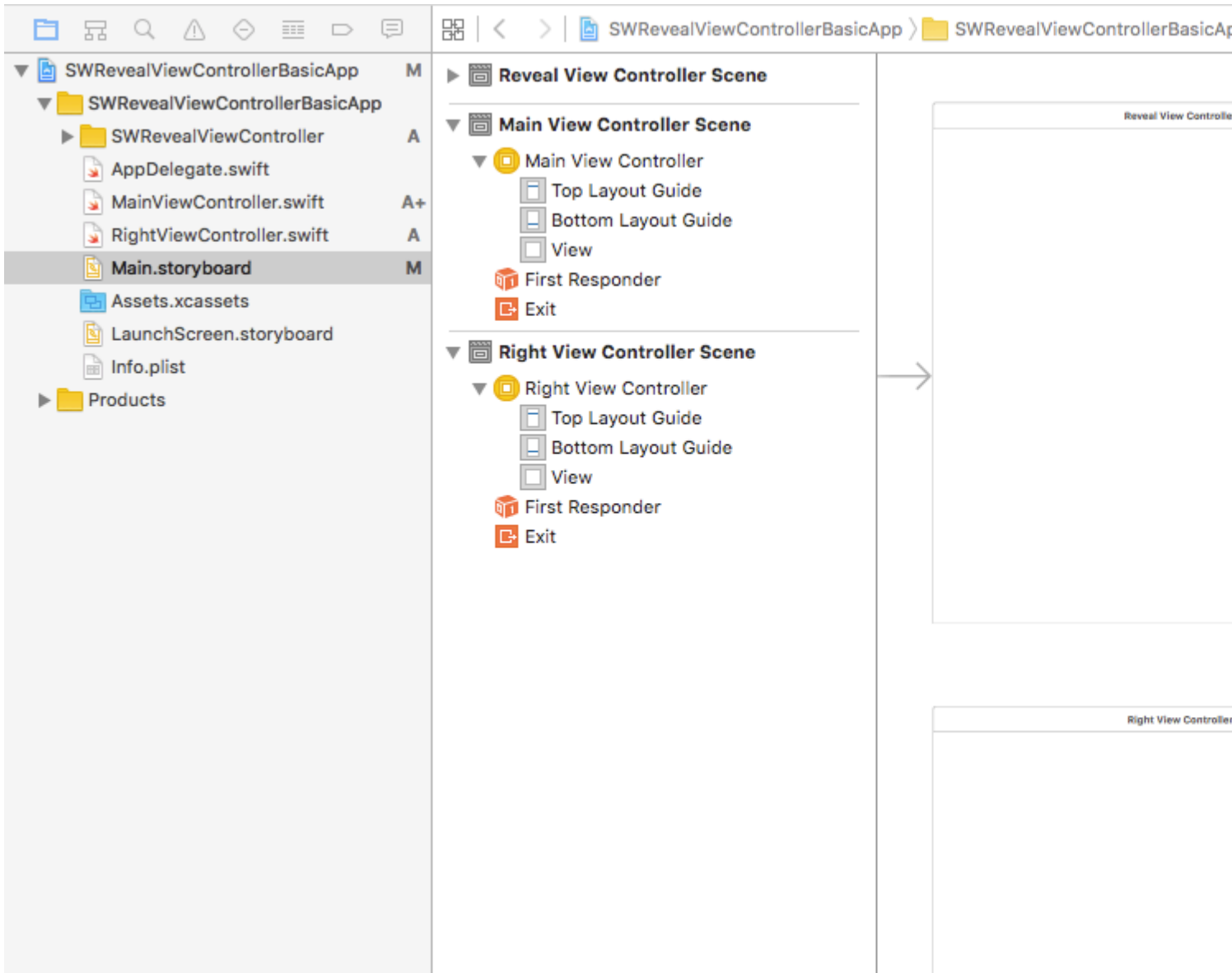
```
#import "SWRevealViewController.h"
```

Bridging

storyboardviewControllerSWRevealViewController

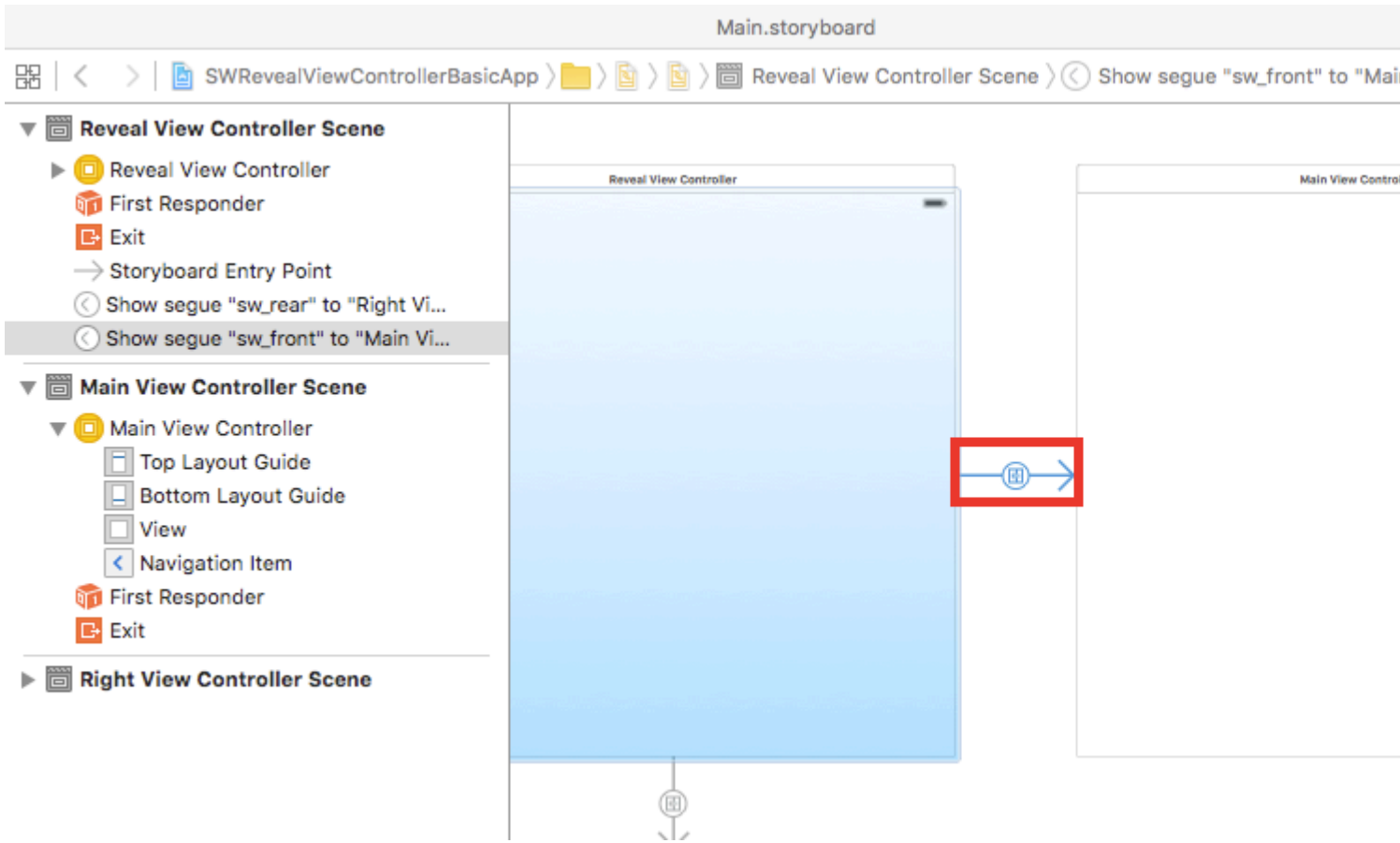


viewControllerMainViewControllerRightViewControllerViewController



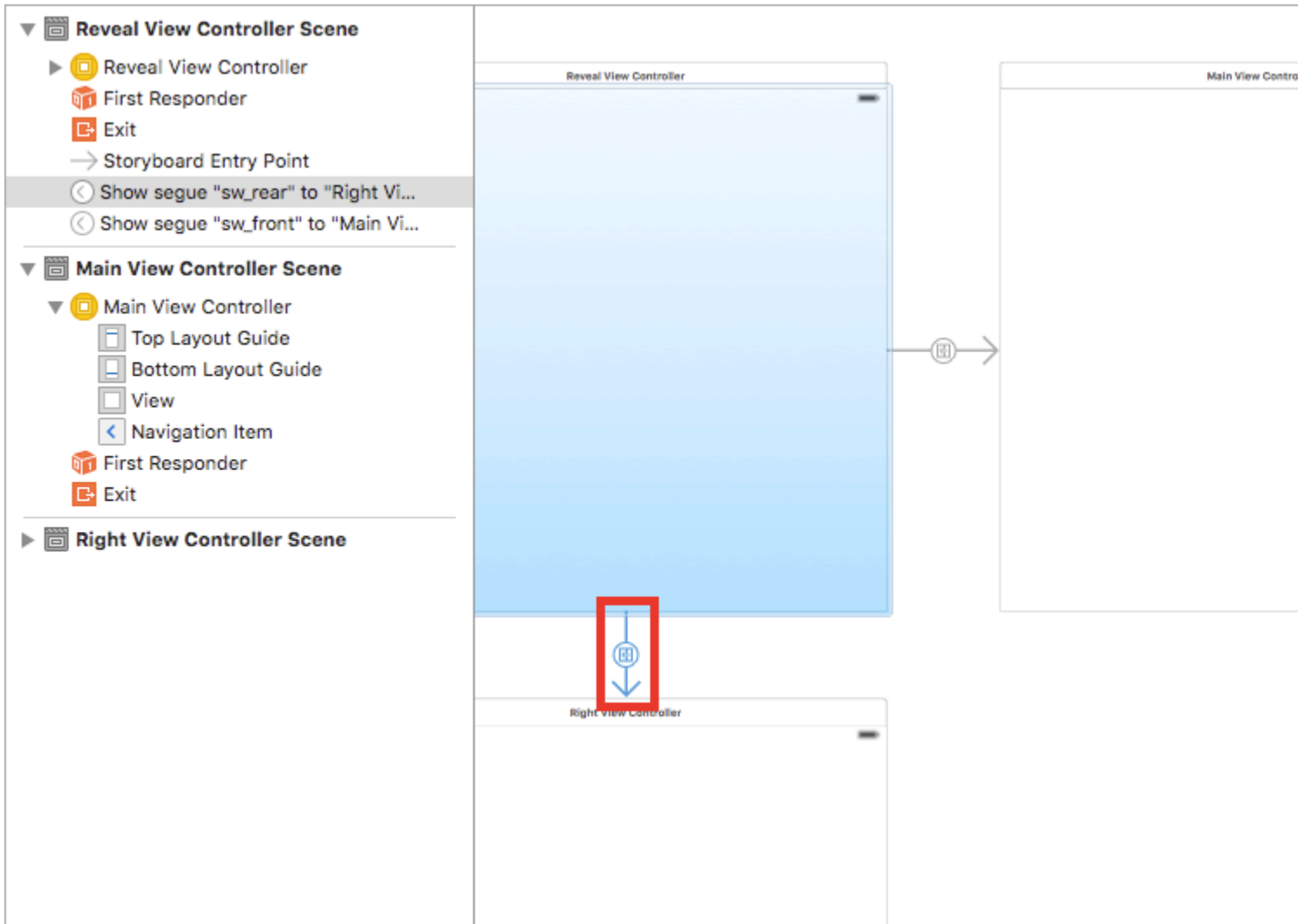
segueSWRevealViewControllerMainViewControllerSWRevealViewControllerRightViewController
 SWRevealViewControllerMainViewController

sw_frontSWRevealViewControllerSegueSetController



segueSWRevealViewControllerRightViewController

sw_rearSWRevealViewControllerSegueSetController



MainViewController.viewDidLoad

```
self.view.addGestureRecognizer(self.revealViewController().panGestureRecognizer());
```

SWRevealViewController.RightViewController

SWRevealViewController <https://riptutorial.com/zh-TW/ios/topic/4614/swrevealviewController>

69: UIActivityViewController

activityItems	◦ nil◦
applicationActivities	UIActivity◦ ◦

Examples

Objective-C

```
NSString *textToShare = @"StackOverflow Documentation!! Together, we can do for Documentation what we did for Q&A.";
NSURL *documentationURL = [NSURL URLWithString:@"http://stackoverflow.com/tour/documentation"];

NSArray *objectsToShare = @[textToShare, documentationURL];

UIActivityViewController *activityVC = [[UIActivityViewController alloc] initWithActivityItems:objectsToShare applicationActivities:nil];

[self presentViewController:activityVC animated:YES completion:nil];
```

```
let textToShare = "StackOverflow Documentation!! Together, we can do for Documentation what we did for Q&A."
let documentationURL = NSURL(string:"http://stackoverflow.com/tour/documentation")

let objToShare : [AnyObject] = [textToShare, documentationURL!]

let activityVC = UIActivityViewController(activityItems: objToShare, applicationActivities: nil)
self.presentViewController(activityVC, animated: true, completion: nil)
```

[UIActivityViewController](https://riptutorial.com/zh-TW/ios/topic/2889/uiactivityviewcontroller) <https://riptutorial.com/zh-TW/ios/topic/2889/uiactivityviewcontroller>

70: UIAlertController

```
UIAlertController◦ UIAlertControllerView◦  
presentViewController:animated:completion:◦
```

Apple

SwiftUIAlertController

Examples

UIAlertControllerAlertViews

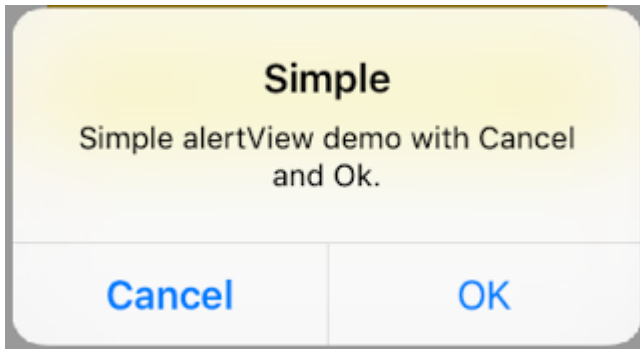
UIAlertViewUIAlertSheetiOS 8iOS 8◦ AppleAlertViewUIAlertControllerpreferredStyle
AlertViewUIAlertSheet◦ ◦

AlertView

```
let alert = UIAlertController(title: "Simple", message: "Simple alertView demo with Cancel and  
OK.", preferredStyle: .alert)  
  
alert.addAction(UIAlertAction(title: "Cancel", style: .cancel) { _ in  
    print("Cancel")  
})  
alert.addAction(UIAlertAction(title: "OK", style: .default) { _ in  
    print("OK")  
})  
  
present(alert, animated: true)
```

Objective-C

```
UIAlertController *alertController = [UIAlertController alertControllerWithTitle:@"Simple"  
message:@"Simple alertView demo with Cancel and OK."  
preferredStyle:UIAlertControllerStyleAlert];  
UIAlertAction *cancelAction = [UIAlertAction actionWithTitle:@"Cancel"  
style:UIAlertActionStyleCancel handler:^(UIAlertAction * action) {  
    NSLog(@"Cancel");  
}];  
UIAlertAction *okAction = [UIAlertAction actionWithTitle:@"OK"  
style:UIAlertActionStyleDefault handler:^(UIAlertAction * action) {  
    NSLog(@"OK");  
}];  
  
[alertController addAction:cancelAction];  
[alertController addAction:okAction];  
[self presentViewController:alertController animated: YES completion: nil];
```



```
let alert = UIAlertController(title: "Simple", message: "Simple alertView demo with Cancel and
OK.", preferredStyle: .alert)

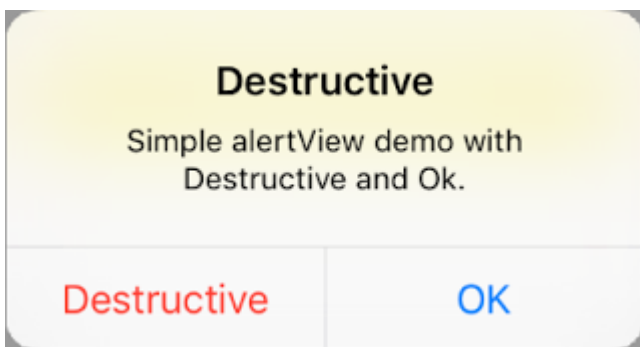
alert.addAction(UIAlertAction(title: "Destructive", style: .destructive) { _ in
    print("Destructive")
})
alert.addAction(UIAlertAction(title: "OK", style: .default) { _ in
    print("OK")
})

present(alert, animated: true)
```

Objective-C

```
UIAlertController *alertController = [UIAlertController
alertControllerWithTitle:@"Destructive" message:@"Simple alertView demo with Destructive and
OK." preferredStyle:UIAlertControllerStyleAlert];
UIAlertAction *destructiveAction = [UIAlertAction actionWithTitle:@"Cancel"
style:UIAlertActionStyleDestructive handler:^(UIAlertAction * action) {
    NSLog(@"Destructive");
}];
UIAlertAction *okAction = [UIAlertAction actionWithTitle:@"OK"
style:UIAlertActionStyleDefault handler:^(UIAlertAction * action) {
    NSLog(@"OK");
}];

[alertController addAction:destructiveAction];
[alertController addAction:okAction];
[self presentViewController:alertController animated: YES completion: nil];
```



```
let alert = UIAlertController(title: "Toast", message: "Hello World", preferredStyle: .Alert)
```



```

presentViewController(alert, animated: true) {
    let delay_s:Double = 2
    let delayTime = dispatch_time(DISPATCH_TIME_NOW, Int64(delay_s * Double(NSEC_PER_SEC)))
    dispatch_after(delayTime, dispatch_get_main_queue()) {
        alert.dismissViewControllerAnimated(true, completion: nil)
    }
}

```

UIAlertController

```

let alert = UIAlertController(title: "Hello",
                             message: "Welcome to the world of iOS",
                             preferredStyle: UIAlertControllerStyle.alert)
let defaultAction = UIAlertAction(title: "OK", style: UIAlertActionStyle.default) { (action)
in
}
defaultAction.isEnabled = false
alert.addAction(defaultAction)

alert.addTextFieldWithConfigurationHandler { (textField) in
    textField.delegate = self
}

present(alert, animated: true, completion: nil)

```

Objective-C

```

UIAlertController* alert = [UIAlertController alertControllerWithTitle:@"Hello"
                                                                message:@"Welcome to the world
of iOS"
                                                                preferredStyle:UIAlertControllerStyleAlert];

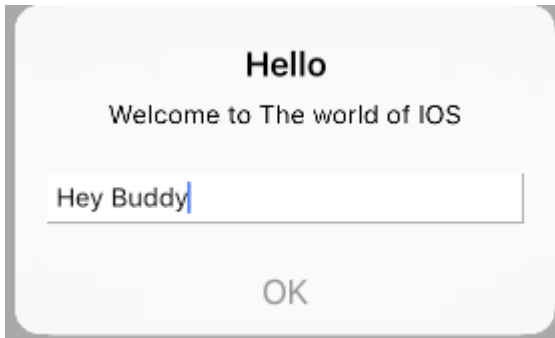
UIAlertAction* defaultAction = [UIAlertAction actionWithTitle:@"OK"
                                                            style:UIAlertActionStyleDefault
                                                            handler:^(UIAlertAction * action) {}];

defaultAction.enabled = NO;
[alert addAction:defaultAction];

[alert addTextFieldWithConfigurationHandler:^(UITextField *textField) {
    textField.delegate = self;
}];

[self presentViewController:alert animated:YES completion:nil];

```



UIAlertController

`UIAlertController` `AlertViewsAPI` `UIActionSheet`

```
let alertController = UIAlertController(title: "Demo", message: "A demo with two buttons",
preferredStyle: UIAlertControllerStyle.actionSheet)
```

Objective-C

```
UIAlertController *alertController = [UIAlertController alertControllerWithTitle:@"Demo"
message:@"A demo with two buttons" preferredStyle:UIAlertControllerStyleActionSheet];
```

“”

```
let cancelAction = UIAlertAction(title: "Cancel", style: .cancel) { (result : UIAlertAction) -
> Void in
    //action when pressed button
}
let okAction = UIAlertAction(title: "Okay", style: .default) { (result : UIAlertAction) ->
Void in
    //action when pressed button
}
```

Objective-C

```
UIAlertAction *cancelAction = [UIAlertAction actionWithTitle:@"Cancel"
style:UIAlertActionStyleCancel handler:^(UIAlertAction * action) {
    //action when pressed button
}];
```

```
UIAlertAction * okAction = [UIAlertAction actionWithTitle:@"Okay"
style:UIAlertActionStyleDefault handler:^(UIAlertAction * action) {
    //action when pressed button
}];
```

```
alertController.addAction(cancelAction)
alertController.addAction(okAction)
```

Objective-C

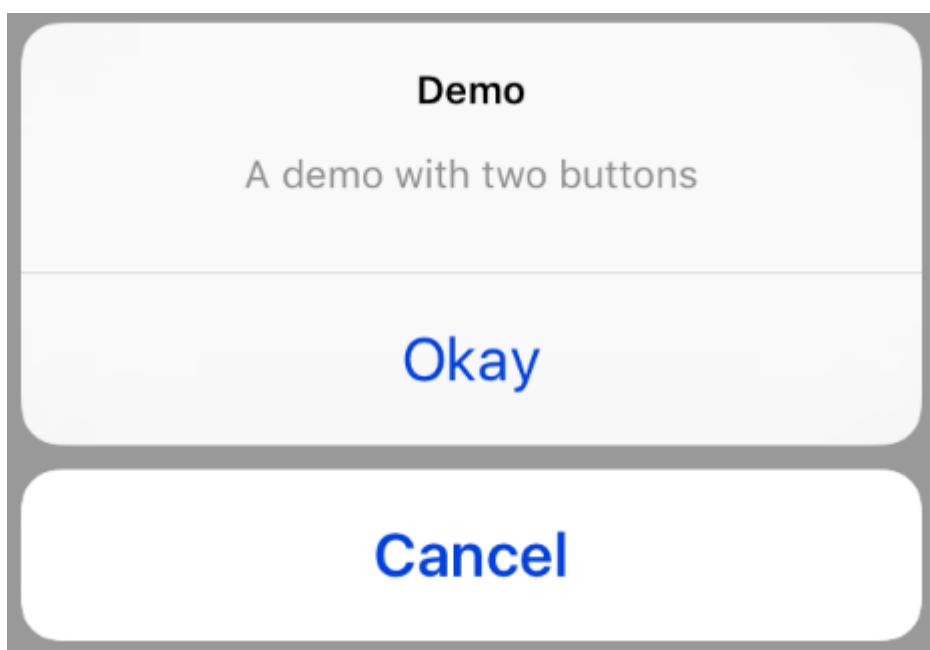
```
[alertController addAction:cancelAction];  
[alertController addAction:okAction];
```

UIAlertController

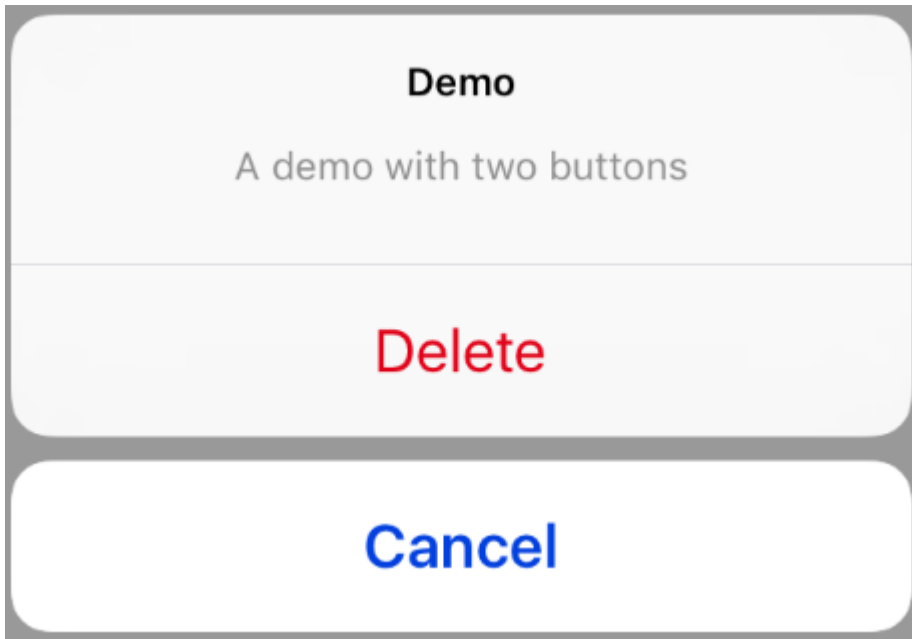
```
self.present(alertController, animated: true, completion: nil)
```

Objective-C

```
[self presentViewController:alertController animated: YES completion: nil];
```



```
UIAlertActionStyle .destructiveUIAlertAction
```

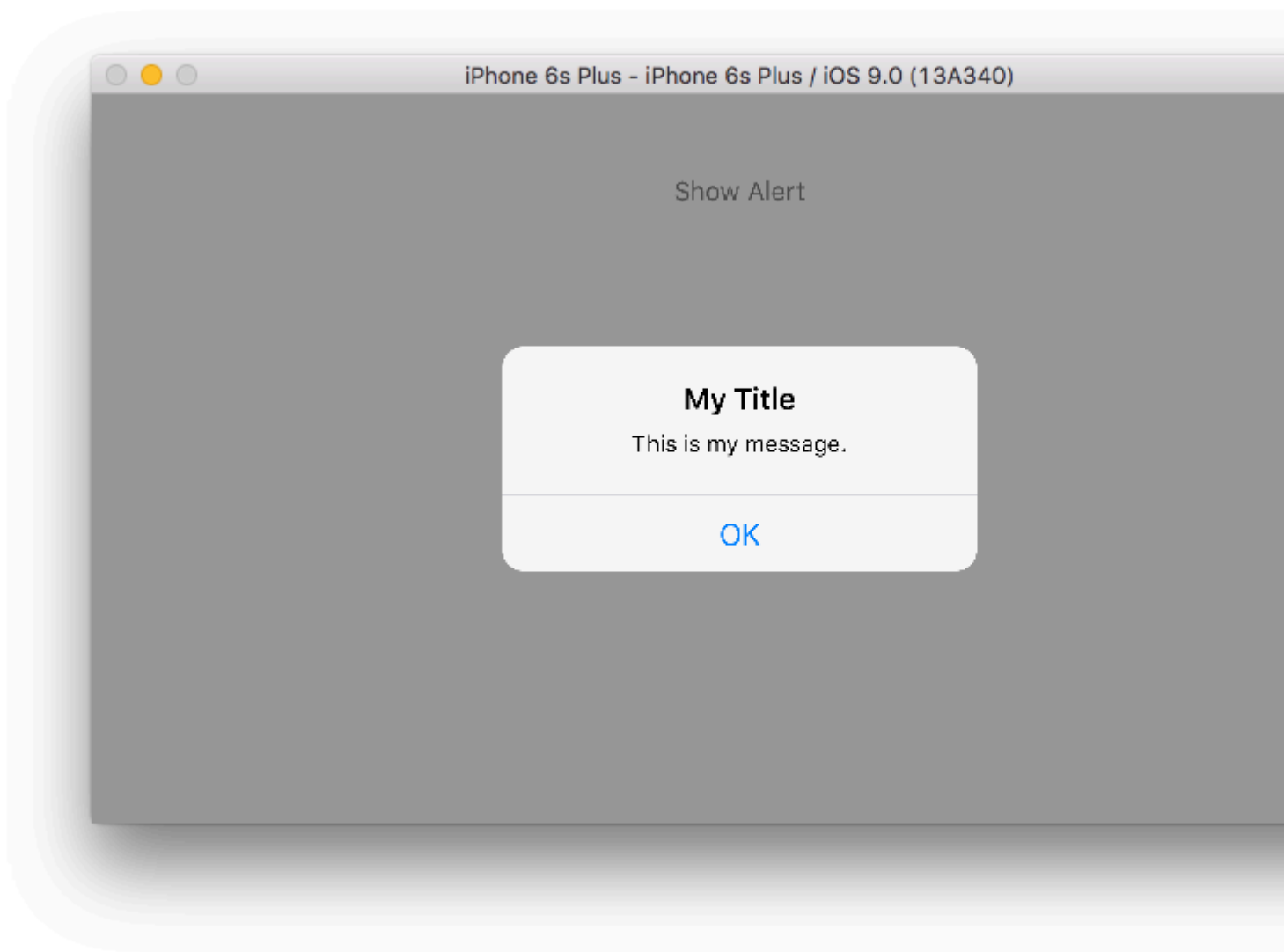


okActionUIAlertAction

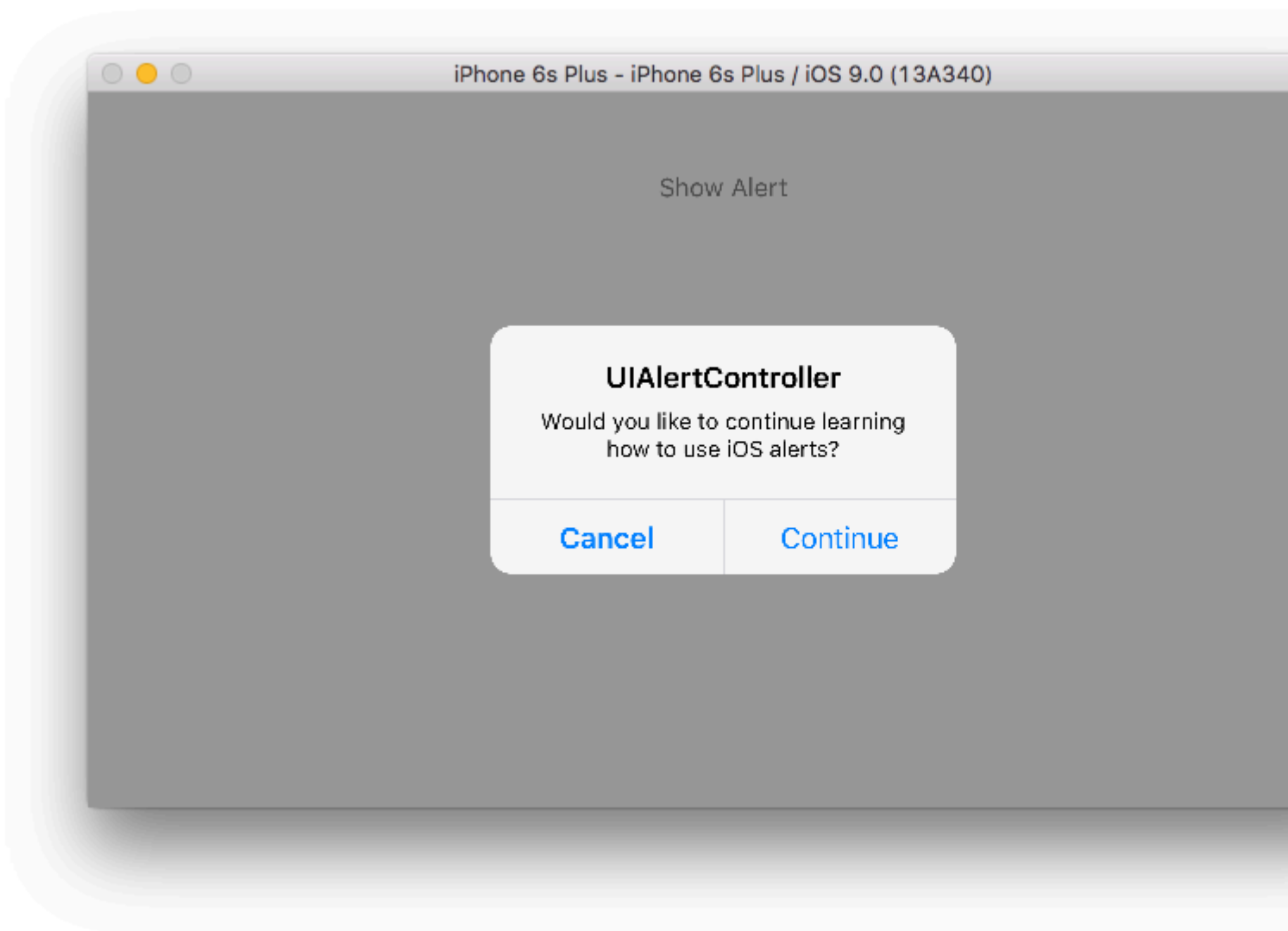
```
let destructiveAction = UIAlertAction(title: "Delete", style: .destructive) { (result :  
UIAlertAction) -> Void in  
    //action when pressed button  
}
```

Objective-C

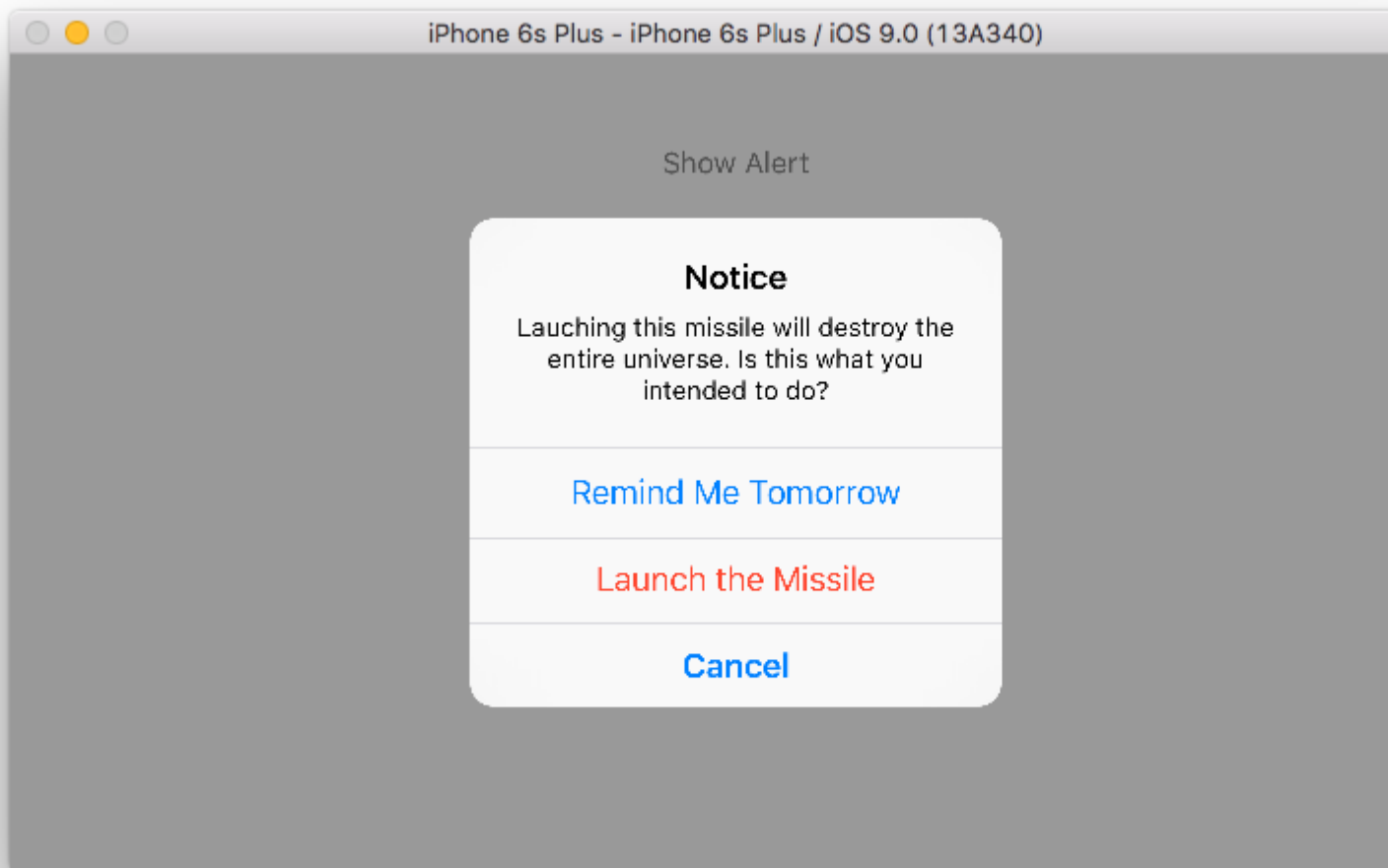
```
UIAlertAction * destructiveAction = [UIAlertAction initWithTitle:@"Delete"  
style:UIAlertActionStyleDestructive handler:^(UIAlertAction * action) {  
    //action when pressed button  
}];
```



```
class ViewController: UIViewController {  
  
    @IBAction func showAlertButtonTapped(sender: UIButton) {  
  
        // create the alert  
        let alert = UIAlertController(title: "My Title", message: "This is my message.",  
preferredStyle: UIAlertControllerStyle.Alert)  
  
        // add an action (button)  
        alert.addAction(UIAlertAction(title: "OK", style: UIAlertActionStyle.Default, handler:  
nil))  
  
        // show the alert  
        self.presentViewController(alert, animated: true, completion: nil)  
    }  
}
```



```
class ViewController: UIViewController {  
  
    @IBAction func showAlertButtonTapped(sender: UIButton) {  
  
        // create the alert  
        let alert = UIAlertController(title: "UIAlertController", message: "Would you like to  
continue learning how to use iOS alerts?", preferredStyle: UIAlertControllerStyle.Alert)  
  
        // add the actions (buttons)  
        alert.addAction(UIAlertAction(title: "Continue", style: UIAlertActionStyle.Default,  
handler: nil))  
        alert.addAction(UIAlertAction(title: "Cancel", style: UIAlertActionStyle.Cancel,  
handler: nil))  
  
        // show the alert  
        self.presentViewController(alert, animated: true, completion: nil)  
    }  
}
```



```
class ViewController: UIViewController {  
  
    @IBAction func showAlertButtonTapped(sender: UIButton) {  
  
        // create the alert  
        let alert = UIAlertController(title: "Notice", message: "Lauching this missile will  
destroy the entire universe. Is this what you intended to do?", preferredStyle:  
UIAlertControllerStyle.Alert)  
  
        // add the actions (buttons)  
        alert.addAction(UIAlertAction(title: "Remind Me Tomorrow", style:  
UIAlertActionStyle.Default, handler: nil))  
        alert.addAction(UIAlertAction(title: "Cancel", style: UIAlertActionStyle.Cancel,  
handler: nil))  
        alert.addAction(UIAlertAction(title: "Launch the Missile", style:  
UIAlertActionStyle.Destructive, handler: nil))  
  
        // show the alert  
        self.presentViewController(alert, animated: true, completion: nil)  
    }  
}
```

handlernil ◦ nil

```

alert.addAction(UIAlertAction(title: "Launch the Missile", style:
UIAlertActionStyle.Destructive, handler: { action in

    // do something like...
    self.launchMissile()

}))

```

- UIAlertControllerStyle.Default

- UIAlertControllerStyle.Alert

- UIAlertControllerStyle.ActionSheet

```

@property(nonatomic, strong) UIAlertAction *preferredAction

```

- UIAlertControllerStyle.Alert

UIAlertControllerStyle.Alert

- UIAlertControllerStyle.Alert

```

UIAlertController *alertController = [UIAlertController alertControllerWithTitle:@"Cancel
edit" message:@"Are you really want to cancel your edit?"
preferredStyle:UIAlertControllerStyle.Alert];

```

```

UIAlertAction *cancel = [UIAlertAction actionWithTitle:@"Cancel"
style:UIAlertActionStyle.Cancel handler:^(UIAlertAction * action) {
    NSLog(@"Cancel");
}];

```

```

UIAlertAction *no = [UIAlertAction actionWithTitle:@"NO" style:UIAlertActionStyle.Default
handler:^(UIAlertAction * action) {
    NSLog(@"Highlighted button is pressed.");
}];

```

```

[alertController addAction:cancel];
[alertController addAction:no];

```

```

//add no action to preferred action.
//Note
//the action should already be added to alert controller
alertController.preferredAction = no;

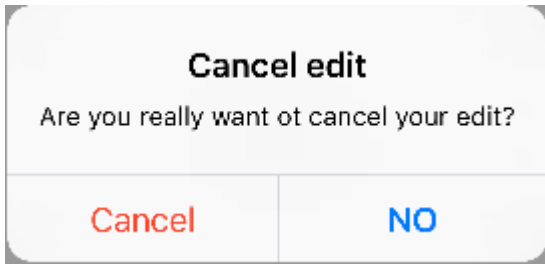
```

```

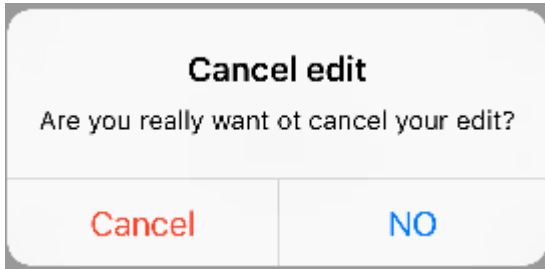
[self presentViewController:alertController animated: YES completion: nil];

```

.The NO



。 “ ”。



[UIAlertController](https://riptutorial.com/zh-TW/ios/topic/874/uialertcontroller) <https://riptutorial.com/zh-TW/ios/topic/874/uialertcontroller>

71: UIAppearance

Examples

◦

UIButton

```
UIButton.appearance().tintColor = UIColor.greenColor()
```

Objective-C

```
[UIButton appearance].tintColor = [UIColor greenColor];
```

UIButton

```
UIButton.appearance().backgroundColor = UIColor.blueColor()
```

Objective-C

```
[UIButton appearance].backgroundColor = [UIColor blueColor];
```

UILabel

```
UILabel.appearance().textColor = UIColor.redColor()
```

Objective-C

```
[UILabel appearance].textColor = [UIColor redColor];
```

UILabel

```
UILabel.appearance().backgroundColor = UIColor.greenColor()
```

Objective-C

```
[UILabel appearance].backgroundColor = [UIColor greenColor];
```

UINavigationController

```
UINavigationController.appearance().tintColor = UIColor.cyanColor()
```

Objective-C

```
[UINavigationController appearance].tintColor = [UIColor cyanColor];
```

UINavigationController

```
UINavigationController.appearance().backgroundColor = UIColor.redColor()
```

Objective-C

```
[UINavigationController appearance].backgroundColor = [UIColor redColor];
```

```
appearanceWhenContainedInInstancesOfClasses:@(UIViewController UILabel) textColor backgroundColor
```

UILabel

```
UILabel.appearanceWhenContainedInInstancesOfClasses([UIViewController.self]).textColor =  
UIColor.whiteColor()
```

Objective-C

```
[UILabel appearanceWhenContainedInInstancesOfClasses:@([UIViewController class]).textColor =  
[UIColor whiteColor];
```

UILabel

```
UILabel.appearanceWhenContainedInInstancesOfClasses([UIViewController.self]).backgroundColor =  
UIColor.blueColor()
```

Objective-C

```
[UILabel appearanceWhenContainedInInstancesOfClasses:@([UIViewController  
class]).backgroundColor = [UIColor blueColor];
```

UIAppearance <https://riptutorial.com/zh-TW/ios/topic/3422/uiappearance>

72: UIBezierPath

Examples

UIBezierPath

4



```
UIBezierPath* rectanglePath = [UIBezierPath bezierPathWithRoundedRect:
CGRectMake(x,y,width,height) cornerRadius: 11];
[UIColor.grayColor setFill];
[rectanglePath fill];
```



```
UIBezierPath* rectanglePath = [UIBezierPath bezierPathWithRoundedRect:
CGRectMake(x,y,width,height) byRoundingCorners: UIRectCornerTopLeft cornerRadii:
CGSizeMake(11, 11)];
[rectanglePath closePath];
[UIColor.grayColor setFill];
[rectanglePath fill];
```



```
UIBezierPath* rectanglePath = [UIBezierPath bezierPathWithRoundedRect:
CGRectMake(x,y,width,height) byRoundingCorners: UIRectCornerTopRight cornerRadii:
CGSizeMake(11, 11)];
[rectanglePath closePath];
[UIColor.grayColor setFill];
[rectanglePath fill];
```



```
UIBezierPath* rectanglePath = [UIBezierPath bezierPathWithRoundedRect:
CGRectMake(x,y,width,height) byRoundingCorners: UIRectCornerBottomLeft cornerRadii:
CGSizeMake(11, 11)];
[rectanglePath closePath];
[UIColor.grayColor setFill];
[rectanglePath fill];
```



```
UIBezierPath* rectanglePath = [UIBezierPath bezierPathWithRoundedRect:
CGRectMake(x,y,width,height) byRoundingCorners: UIRectCornerBottomRight cornerRadii:
CGSizeMake(11, 11)];
[rectanglePath closePath];
[UIColor.grayColor setFill];
[rectanglePath fill];
```



```
UIBezierPath* rectanglePath = [UIBezierPath bezierPathWithRoundedRect:
CGRectMake(x,y,width,height) byRoundingCorners: UIRectCornerBottomLeft |
UIRectCornerBottomRight cornerRadii: CGSizeMake(11, 11)];
[rectanglePath closePath];
[UIColor.grayColor setFill];
[rectanglePath fill];
```



```
UIBezierPath* rectanglePath = [UIBezierPath bezierPathWithRoundedRect:
```

```
CGRectMake(x,y,width,height) byRoundingCorners: UIRectCornerTopLeft | UIRectCornerTopRight
cornerRadii: CGSizeMake(11, 11)];
[rectanglePath closePath];
[UIColor.grayColor setFill];
[rectanglePath fill];
```

UIBezierPath



```
UIBezierPath* ovalPath = [UIBezierPath bezierPathWithOvalInRect: CGRectMake(0,0,50,50)];
[UIColor.grayColor setFill];
[ovalPath fill];
```

```
let ovalPath = UIBezierPath(ovalInRect: CGRect(x: 0, y: 0, width: 50, height: 50))
UIColor.grayColor().setFill()
ovalPath.fill()
```

Rectangle



```
UIBezierPath* rectanglePath = [UIBezierPath bezierPathWithRect: CGRectMake(0,0,50,50)];
[UIColor.grayColor setFill];
[rectanglePath fill];
```

```
let rectanglePath = UIBezierPath(rect: CGRect(x: 0, y: 0, width: 50, height: 50))
UIColor.grayColor().setFill()
rectanglePath.fill()
```



```
UIBezierPath* bezierPath = [UIBezierPath bezierPath];
[bezierPath moveToPoint: CGPointMake(x1,y1)];
[bezierPath addLineToPoint: CGPointMake(x2,y2)];
[UIColor.blackColor setStroke];
bezierPath.lineWidth = 1;
```

```
[bezierPath stroke];
```

```
let bezierPath = UIBezierPath()  
bezierPath.moveToPoint(CGPoint(x: x1, y: y1))  
bezierPath.addLineToPoint(CGPoint(x: x2, y: y2))  
UIColor.blackColor().setStroke()  
bezierPath.lineWidth = 1  
bezierPath.stroke()
```



```
CGRect ovalRect = CGRectMake(x,y,width,height);  
UIBezierPath* ovalPath = [UIBezierPath bezierPath];  
[ovalPath addArcWithCenter: CGPointMake(0, 0) radius: CGRectGetWidth(ovalRect) / 2 startAngle:  
180 * M_PI/180 endAngle: 0 * M_PI/180 clockwise: YES];  
[ovalPath addLineToPoint: CGPointMake(0, 0)];  
[ovalPath closePath];  
  
CGAffineTransform ovalTransform = CGAffineTransformMakeTranslation(CGRectGetMidX(ovalRect),  
CGRectGetMidY(ovalRect));  
ovalTransform = CGAffineTransformScale(ovalTransform, 1, CGRectGetHeight(ovalRect) /  
CGRectGetWidth(ovalRect));  
[ovalPath applyTransform: ovalTransform];  
  
[UIColor.grayColor setFill];  
[ovalPath fill];
```

```
let ovalRect = CGRect(x: 0, y: 0, width: 50, height: 50)  
let ovalPath = UIBezierPath()  
ovalPath.addArcWithCenter(CGPoint.zero, radius: ovalRect.width / 2, startAngle: 180 *  
CGFloat(M_PI)/180, endAngle: 0 * CGFloat(M_PI)/180, clockwise: true)  
ovalPath.addLineToPoint(CGPoint.zero)  
ovalPath.closePath()  
  
var ovalTransform = CGAffineTransformMakeTranslation(CGRectGetMidX(ovalRect),  
CGRectGetMidY(ovalRect))  
ovalTransform = CGAffineTransformScale(ovalTransform, 1, ovalRect.height / ovalRect.width)  
ovalPath.applyTransform(ovalTransform)  
  
UIColor.grayColor().setFill()  
ovalPath.fill()
```



```
UIBezierPath* polygonPath = [UIBezierPath bezierPath];  
[polygonPath moveToPoint: CGPointMake(x1, y1)];  
[polygonPath addLineToPoint: CGPointMake(x2, y2)];  
[polygonPath addLineToPoint: CGPointMake(x3, y2)];
```

```
[polygonPath closePath];
[UIColor.grayColor setFill];
[polygonPath fill];
```

```
let polygonPath = UIBezierPath()
polygonPath.moveToPoint(CGPoint(x: x1, y: y1))
polygonPath.addLineToPoint(CGPoint(x: x2, y: y2))
polygonPath.addLineToPoint(CGPoint(x: x3, y: y3))
polygonPath.closePath()
UIColor.grayColor().setFill()
polygonPath.fill()
```

UIBezierPath + AutoLayout

bezierdrawRect

```
- (void)drawRect:(CGRect) frame
{
    UIBezierPath* rectanglePath = [UIBezierPath bezierPathWithRect:
    CGRectMake(CGRectGetMinX(frame), CGRectGetMinY(frame), CGRectGetWidth(frame),
    CGRectGetHeight(frame))];
    [UIColor.grayColor setFill];
    [rectanglePath fill];
}
```

UIBezierPath

o



```
UIBezierPath* rectanglePath = [UIBezierPath bezierPathWithRect:
CGRectMake(x, y, width, height)];
[UIColor.grayColor setFill];
[rectanglePath fill];
```




```

CGContextRef context = UIGraphicsGetCurrentContext();

NSShadow* shadow = [[NSShadow alloc] init];
[shadow setShadowColor: UIColor.blackColor];
[shadow setShadowOffset: CGSizeMake(7.1, 5.1)];
[shadow setShadowBlurRadius: 5];

UIBezierPath* rectanglePath = [UIBezierPath bezierPathWithRect: CGRectMake(x,y,width,height)];
CGContextSaveGState(context);
CGContextSetShadowWithColor(context, shadow.shadowOffset, shadow.shadowBlurRadius,
[shadow.shadowColor CGColor]);
[UIColor.grayColor setFill];
[rectanglePath fill];
CGContextRestoreGState(context);

```



```

CGContextRef context = UIGraphicsGetCurrentContext();

NSShadow* shadow = [[NSShadow alloc] init];
[shadow setShadowColor: UIColor.blackColor];
[shadow setShadowOffset: CGSizeMake(9.1, -7.1)];
[shadow setShadowBlurRadius: 6];

UIBezierPath* rectanglePath = [UIBezierPath bezierPathWithRect: CGRectMake(x,y,width,height)];
[UIColor.grayColor setFill];
[rectanglePath fill];

CGContextSaveGState(context);
UIRectClip(rectanglePath.bounds);
CGContextSetShadowWithColor(context, CGSizeZero, 0, NULL);

CGContextSetAlpha(context, CGColorGetAlpha([shadow.shadowColor CGColor]));
CGContextBeginTransparencyLayer(context, NULL);
{
    UIColor* opaqueShadow = [shadow.shadowColor colorWithAlphaComponent: 1];
    CGContextSetShadowWithColor(context, shadow.shadowOffset, shadow.shadowBlurRadius,
[opaqueShadow CGColor]);
    CGContextSetBlendMode(context, kCGBlendModeSourceOut);
    CGContextBeginTransparencyLayer(context, NULL);

    [opaqueShadow setFill];
    [rectanglePath fill];

    CGContextEndTransparencyLayer(context);
}
CGContextEndTransparencyLayer(context);
CGContextRestoreGState(context);

```

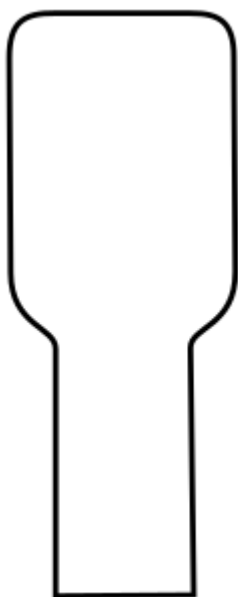
◦ shap◦

Bézier

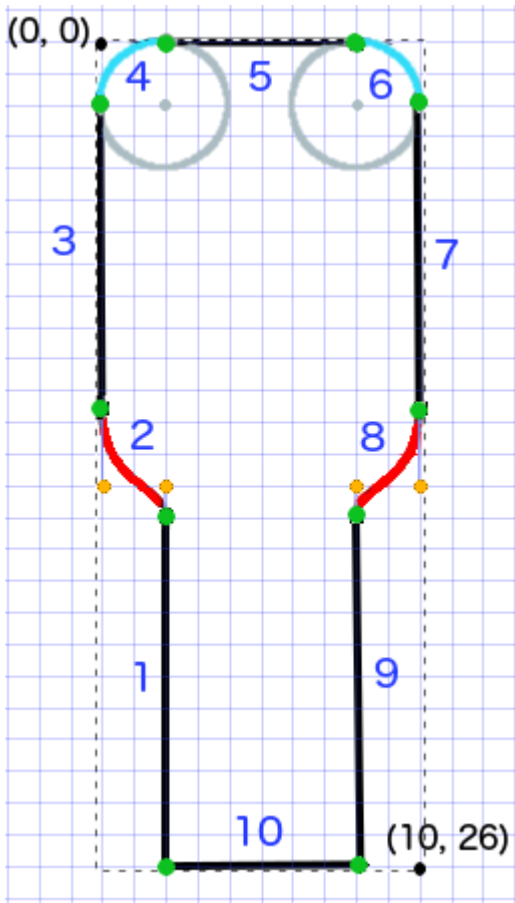
1. ◦
2. ◦
3. ◦
4. `drawRectCAShapeLayer`◦



◦ ◦



◦



-
-
-
-
-
-
-

◦ xy ◦ ◦

1. UIBezierPath
2. moveToPointmoveToPoint
3.
 - **line** addLineToPoint
 - **arc** addArcWithCenter
 - **curve** addCurveToPoint
4. closePath

◦

```
func createBezierPath() -> UIBezierPath {
    // create a new path
    let path = UIBezierPath()
```

```

// starting point for the path (bottom left)
path.moveToPoint(CGPoint(x: 2, y: 26))

// *****
// ***** Left side *****
// *****

// segment 1: line
path.addLineToPoint(CGPoint(x: 2, y: 15))

// segment 2: curve
path.addCurveToPoint(CGPoint(x: 0, y: 12), // ending point
    controlPoint1: CGPoint(x: 2, y: 14),
    controlPoint2: CGPoint(x: 0, y: 14))

// segment 3: line
path.addLineToPoint(CGPoint(x: 0, y: 2))

// *****
// ***** Top side *****
// *****

// segment 4: arc
path.addArcWithCenter(CGPoint(x: 2, y: 2), // center point of circle
    radius: 2, // this will make it meet our path line
    startAngle: CGFloat(M_PI), //  $\pi$  radians = 180 degrees = straight left
    endAngle: CGFloat(3*M_PI_2), //  $3\pi/2$  radians = 270 degrees = straight up
    clockwise: true) // startAngle to endAngle goes in a clockwise direction

// segment 5: line
path.addLineToPoint(CGPoint(x: 8, y: 0))

// segment 6: arc
path.addArcWithCenter(CGPoint(x: 8, y: 2),
    radius: 2,
    startAngle: CGFloat(3*M_PI_2), // straight up
    endAngle: CGFloat(0), // 0 radians = straight right
    clockwise: true)

// *****
// ***** Right side *****
// *****

// segment 7: line
path.addLineToPoint(CGPoint(x: 10, y: 12))

// segment 8: curve
path.addCurveToPoint(CGPoint(x: 8, y: 15), // ending point
    controlPoint1: CGPoint(x: 10, y: 14),
    controlPoint2: CGPoint(x: 8, y: 14))

// segment 9: line
path.addLineToPoint(CGPoint(x: 8, y: 26))

// *****
// ***** Bottom side *****
// *****

// segment 10: line
path.closePath() // draws the final line to close the path

```

```
    return path
}
```

◦

drawRect◦

1

◦ **Bezier**CAShapeLayer ◦

```
import UIKit
class MyCustomView: UIView {

    override init(frame: CGRect) {
        super.init(frame: frame)
        setup()
    }

    required init?(coder aDecoder: NSCoder) {
        super.init(coder: aDecoder)
        setup()
    }

    func setup() {

        // Create a CAShapeLayer
        let shapeLayer = CAShapeLayer()

        // The Bezier path that we made needs to be converted to
        // a CGPath before it can be used on a layer.
        shapeLayer.path = createBezierPath().CGPath

        // apply other properties related to the path
        shapeLayer.strokeColor = UIColor.blueColor().CGColor
        shapeLayer.fillColor = UIColor.whiteColor().CGColor
        shapeLayer.lineWidth = 1.0
        shapeLayer.position = CGPoint(x: 10, y: 10)

        // add the new layer to our custom view
        self.layer.addSublayer(shapeLayer)
    }

    func createBezierPath() -> UIBezierPath {

        // see previous code for creating the Bezier path
    }
}
```

View Controller

```
override func viewDidLoad() {
    super.viewDidLoad()

    // create a new UIView and add it to the view controller
    let myView = MyCustomView()
```

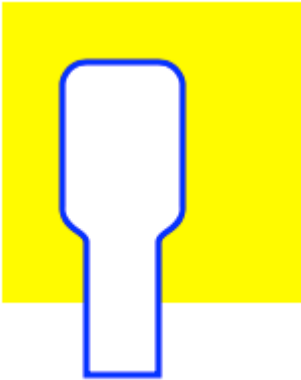
```
myView.frame = CGRect(x: 100, y: 100, width: 50, height: 50)
myView.backgroundColor = UIColor.yellowColor()
view.addSubview(myView)
}
```

.....



o

```
let path = createBezierPath()
let scale = CGAffineTransformMakeScale(2, 2)
path.applyTransform(scale)
shapeLayer.path = path.CGPath
```



2drawRect

drawRect o

```
import UIKit
class MyCustomView: UIView {

    override func drawRect(rect: CGRect) {

        // create path (see previous code)
        let path = createBezierPath()

        // fill
        let fillColor = UIColor.whiteColor()
        fillColor.setFill()
    }
}
```

```

// stroke
path.lineWidth = 1.0
let strokeColor = UIColor.blueColor()
strokeColor.setStroke()

// Move the path to a new location
path.applyTransform(CGAffineTransformMakeTranslation(10, 10))

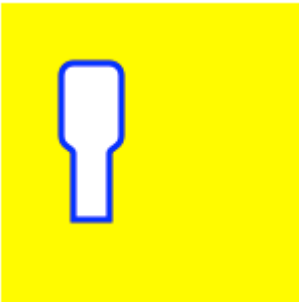
// fill and stroke the path (always do these last)
path.fill()
path.stroke()
}

func createBezierPath() -> UIBezierPath {

    // see previous code for creating the Bezier path
}
}

```

.....



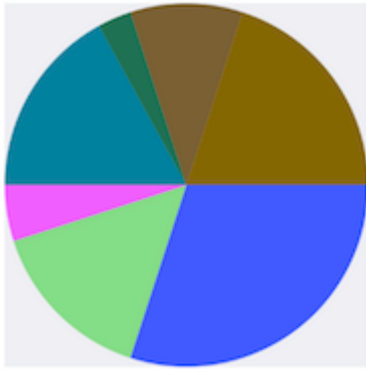
Bezier

- [Bézier](#)
- [19 -](#)
- [Bezier](#)
- [Bezier](#)

-
- [Stack Overflow](#)
 - [.](#)

UIBezierPath

- [.](#)



```

- (void)drawRect:(CGRect)rect {

    NSArray *data = @[30, 15, 5, 17, 3, 10, 20];

    // 1. context
    CGContextRef cxtRef = UIGraphicsGetCurrentContext();

    CGPoint center = CGPointMake(150, 150);
    CGFloat radius = 150;
    CGFloat startAngle = 0;
    [data enumerateObjectsUsingBlock:^(NSNumber * _Nonnull obj, NSUInteger idx, BOOL *
    _Nonnull stop) {

        // 2. create path
        CGFloat endAngle = obj.floatValue / 100 * M_PI * 2 + startAngle;
        UIBezierPath *circlePath = [UIBezierPath bezierPathWithArcCenter:center radius:radius
startAngle:startAngle endAngle:endAngle clockwise:YES];
        [circlePath addLineToPoint:center];

        // 3. add path
        CGContextAddPath(cxtRef, circlePath.CGPath);

        // set color
        [[UIColor colorWithRed:((float)arc4random_uniform(256) / 255.0)
green:((float)arc4random_uniform(256) / 255.0) blue:((float)arc4random_uniform(256) / 255.0)
alpha:1.0] setFill];

        // 4. render
        CGContextDrawPath(cxtRef, kCGPathFill);

        // reset angle
        startAngle = endAngle;
    }];
}

```

```

override func draw(_ rect: CGRect) {
    // define data to create pie chart
    let data: [Int] = [30, 15, 5, 17, 3, 10, 20]

    // 1. find center of draw rect
    let center: CGPoint = CGPoint(x: rect.midX, y: rect.midY)

    // 2. calculate radius of pie
    let radius = min(rect.width, rect.height) / 2.0

    var startAngle: CGFloat = 0.0
    for value in data {

```



```

// 3. calculate end angle for slice
let endAngle = CGFloat(value) / 100.0 * CGFloat.pi * 2.0 + startAngle

// 4. create UIBezierPath for slice
let circlePath = UIBezierPath(arcCenter: center, radius: radius, startAngle: startAngle,
endAngle: endAngle, clockwise: true)

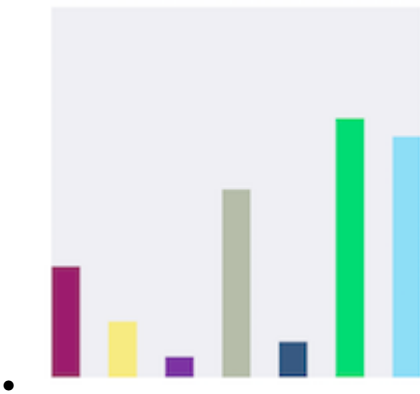
// 5. add line to center to close path
circlePath.addLine(to: center)

// 6. set fill color for current slice
UIColor(red: (CGFloat(arc4random_uniform(256)) / 255.0), green:
(CGFloat(arc4random_uniform(256)) / 255.0), blue: (CGFloat(arc4random_uniform(256)) / 255.0),
alpha: 1.0).setFill()

// 7. fill slice path
circlePath.fill()

// 8. set end angle as start angle for next slice
startAngle = endAngle
}
}

```



```

- (void)drawRect:(CGRect)rect {

    NSArray *data = @[300, 150.65, 55.3, 507.7, 95.8, 700, 650.65];

    // 1.
    CGContextRef cxtRef = UIGraphicsGetCurrentContext();

    NSInteger columnCount = 7;
    CGFloat width = self.bounds.size.width / (columnCount + columnCount - 1);
    for (NSInteger i = 0; i < columnCount; i++) {

        // 2.
        CGFloat height = [data[i] floatValue] / 1000 * self.bounds.size.height; // floatValue
        CGFloat x = 0 + width * (2 * i);
        CGFloat y = self.bounds.size.height - height;
        UIBezierPath *rectPath = [UIBezierPath bezierPathWithRect:CGRectMake(x, y, width,
height)];
        CGContextAddPath(cxtRef, rectPath.CGPath);

        // 3.
        [[UIColor colorWithRed:((float)arc4random_uniform(256) / 255.0)
green:((float)arc4random_uniform(256) / 255.0) blue:((float)arc4random_uniform(256) / 255.0)
alpha:1.0] setFill];
    }
}

```

```
        CGContextDrawPath(cxtRef, kCGPathFill);
    }
}
```

```
override func draw(_ rect: CGRect) {
    // define data for chart
    let data: [CGFloat] = [300, 150.65, 55.3, 507.7, 95.8, 700, 650.65]

    // 1. calculate number of columns
    let columnCount = data.count

    // 2. calculate column width
    let columnWidth = rect.width / CGFloat(columnCount + columnCount - 1)

    for (columnIndex, value) in data.enumerated() {
        // 3. calculate column height
        let columnHeight = value / 1000.0 * rect.height

        // 4. calculate column origin
        let columnOrigin = CGPoint(x: (columnWidth * 2.0 * CGFloat(columnIndex)), y:
(rect.height - columnHeight))

        // 5. create path for column
        let columnPath = UIBezierPath(rect: CGRect(origin: columnOrigin, size: CGSize(width:
columnWidth, height: columnHeight)))

        // 6. set fill color for current column
        UIColor(red: (CGFloat(arc4random_uniform(256)) / 255.0), green:
(CGFloat(arc4random_uniform(256)) / 255.0), blue: (CGFloat(arc4random_uniform(256)) / 255.0),
alpha: 1.0).setFill()

        // 7. fill column path
        columnPath.fill()
    }
}
```

UIBezierPath <https://riptutorial.com/zh-TW/ios/topic/3186/uiBezierPath>

73: UICollectionView

Examples

```
func createCollectionView() {
    let layout: UICollectionViewFlowLayout = UICollectionViewFlowLayout()
    let collectionView = UICollectionView(frame: CGRect(x: 0, y: 0, width: view.frame.width,
height: view.frame.height), collectionViewLayout: layout)
    collectionView.dataSource = self
    collectionView.delegate = self
    view.addSubview(collectionView)
}
```

Objective-C

```
- (void)createCollectionView {
    UICollectionViewFlowLayout *layout = [[UICollectionViewFlowLayout alloc] init];
    UICollectionView *collectionView = [[UICollectionView alloc] initWithFrame:CGRectMake(0,
0, self.view.frame.size.width, self.view.frame.size.height) collectionViewLayout:layout];
    [collectionView setDataSource:self];
    [collectionView setDelegate:self];
    [self.view addSubview:collectionView];
}
```

Swift - UICollectionViewDelegateFlowLayout

```
// MARK: - UICollectionViewDelegateFlowLayout
extension ViewController: UICollectionViewDelegateFlowLayout {
    func collectionView(collectionView: UICollectionView, layout collectionViewLayout:
UICollectionViewLayout, sizeForItemAtIndexPath indexPath: NSIndexPath) -> CGSize {
        return CGSize(width: 50, height: 50)
    }
    func collectionView(collectionView: UICollectionView, layout collectionViewLayout:
UICollectionViewLayout, insetForSectionAtIndex section: Int) -> UIEdgeInsets {
        return UIEdgeInsets(top: 5, left: 5, bottom: 5, right: 5)
    }
    func collectionView(collectionView: UICollectionView, layout collectionViewLayout:
UICollectionViewLayout, minimumLineSpacingForSectionAtIndex section: Int) -> CGFloat {
        return 5.0
    }
    func collectionView(collectionView: UICollectionView, layout collectionViewLayout:
UICollectionViewLayout, minimumInteritemSpacingForSectionAtIndex section: Int) -> CGFloat {
        return 5.0
    }
}
```

UICollectionView

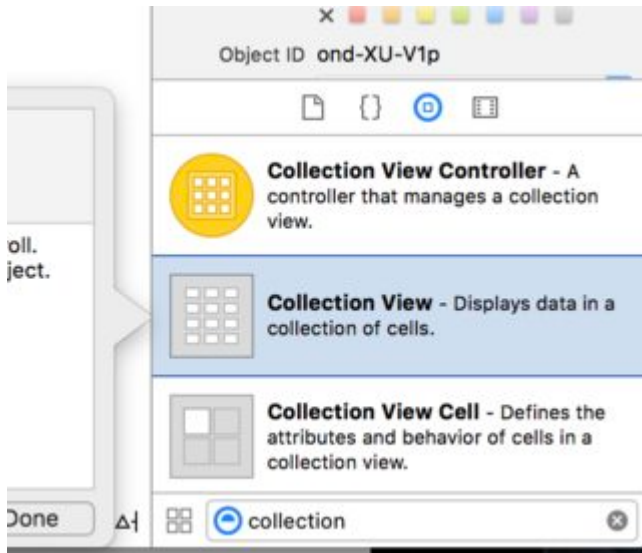
UICollectionViewCGRect

```
let collection = UICollectionView(frame: CGRect(x: 0, y: 0, width: 200, height: 21))
```

C

```
UICollectionView *collection = [[UICollectionView alloc] initWithFrame:CGRectMake(0, 0, 200, 21)];
```

Interface Builder `UICollectionView`



`UICollectionView` -

`Datasource` ◦ `Datasource` `UICollectionView` `UICollectionView` `UICollectionView` ◦ `Datasource`
`collectionView:numberOfItemsInSection:collectionView:cellForItemAtIndexPath:` ◦

```
func collectionView(collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
    // Return how many items in section
    let dataArray = _data[section]
    return dataArray.count
}

func collectionView(collectionView: UICollectionView, cellForItemAtIndexPath indexPath: NSIndexPath) -> UICollectionViewCell {

    let cell = collectionView.dequeueReusableCellWithReuseIdentifier(MyCellID)
    // If you use a custom cell class then cast the cell returned, like:
    // as! MyCollectionViewCellClass
    // or you will have errors when you try to use features of that class.

    //Customize your cell here, default UICollectionViewCells do not contain any inherent
    //text or image views (like UITableView), but some could be added,
    //or a custom UICollectionViewCell sub-class could be used
    return cell
}
```

C.

```
- (NSInteger)collectionView:(UICollectionView*)collectionView
numberOfItemsInSection:(NSInteger)section {
    // Return how many items in section
```

```

    NSArray *sectionArray = [_data objectAtIndex:section];
    return [sectionArray count];
}

- (UICollectionViewCell *)collectionView:(UICollectionView *)collectionView
  cellForItemAtIndexPath:(NSIndexPath *)indexPath {
    // Return a cell
    UICollectionViewCell *newCell = [self.collectionView
                                     dequeueReusableCellWithReuseIdentifier:MyCellID

                                     forIndexPath:indexPath];
    //Customize your cell here, default UICollectionViewCells do not contain any inherent
    //text or image views (like UITableView), but some could be added,
    //or a custom UICollectionViewCell sub-class could be used
    return newCell;
}

```

Swift

◦

Cocoa Touch>>...> iOS> Cocoa Touch◦ MyCollectionViewCell◦◦

```

import UIKit
class MyCollectionViewCell: UICollectionViewCell {

    @IBOutlet weak var myLabel: UILabel!
}

```

◦

ViewController.swift

```

import UIKit
class ViewController: UIViewController, UICollectionViewDataSource, UICollectionViewDelegate {

    let reuseIdentifier = "cell" // also enter this string as the cell identifier in the
    storyboard
    var items = ["1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14",
"15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29",
"30", "31", "32", "33", "34", "35", "36", "37", "38", "39", "40", "41", "42", "43", "44",
"45", "46", "47", "48"]

    // MARK: - UICollectionViewDataSource protocol

    // tell the collection view how many cells to make
    func collectionView(collectionView: UICollectionView, numberOfItemsInSection section: Int)
-> Int {
        return self.items.count
    }

    // make a cell for each cell index path

```

```

func collectionView(collectionView: UICollectionView, cellForItemAtIndexPath indexPath:
NSIndexPath) -> UICollectionViewCell {

    // get a reference to our storyboard cell
    let cell = collectionView.dequeueReusableCellWithReuseIdentifier(reuseIdentifier,
forIndexPath: indexPath) as! MyCollectionViewCell

    // Use the outlet in our custom class to get a reference to the UILabel in the cell
    cell.myLabel.text = self.items[indexPath.item]
    cell.backgroundColor = UIColor.yellowColor() // make cell more visible in our example
project

    return cell
}

// MARK: - UICollectionViewDelegate protocol

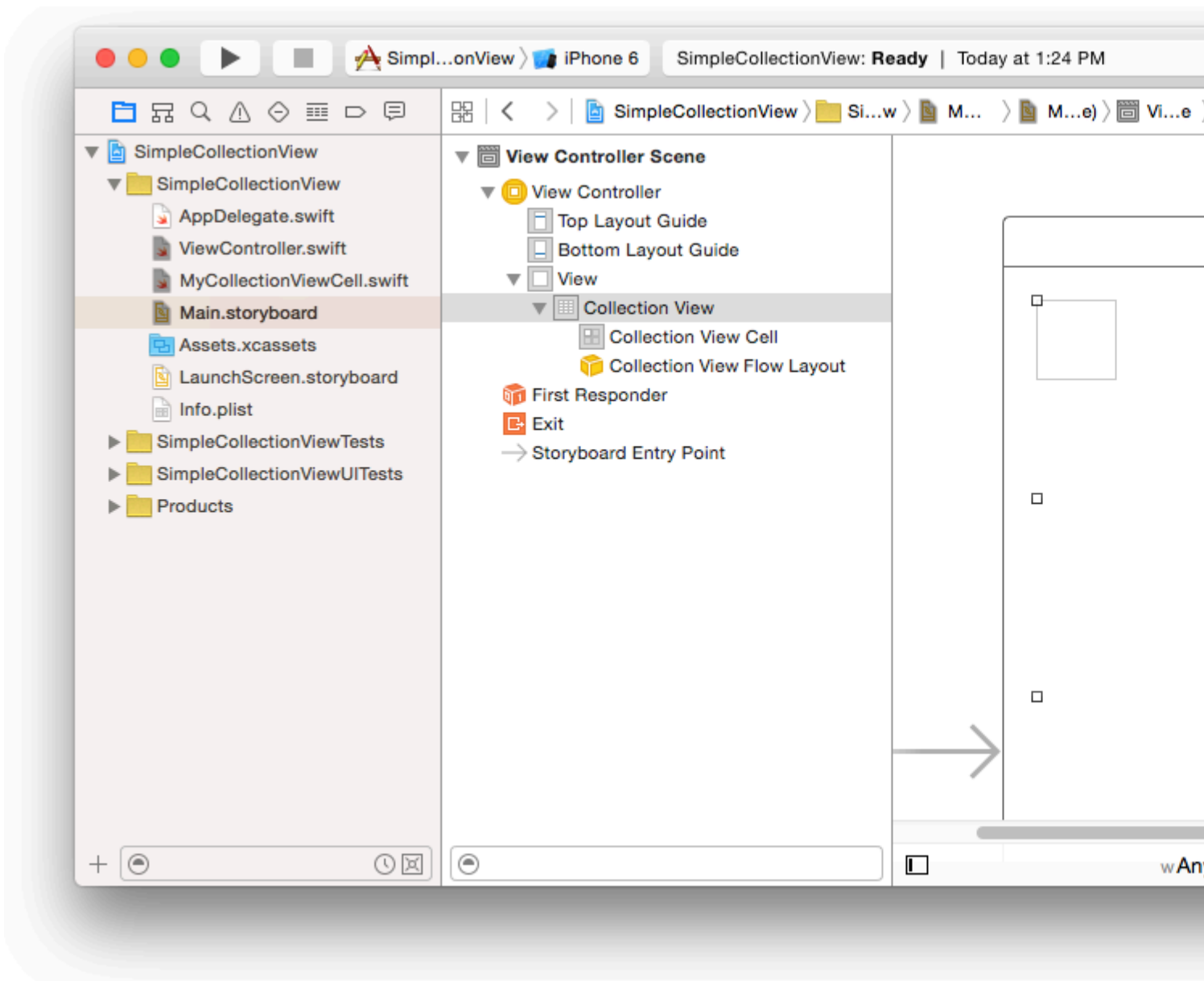
func collectionView(collectionView: UICollectionView, didSelectItemAtIndexPath indexPath:
NSIndexPath) {
    // handle tap events
    print("You selected cell #\(indexPath.item)!")
}
}

```

- UICollectionViewDataSourceUICollectionViewDelegate° UICollectionViewDelegateFlowLayout°
- °

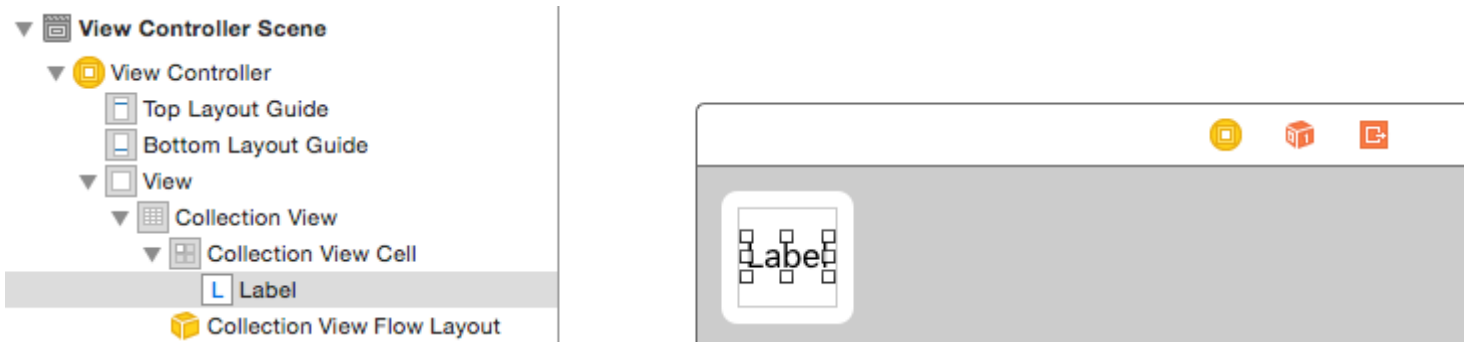


° °



- 1
-

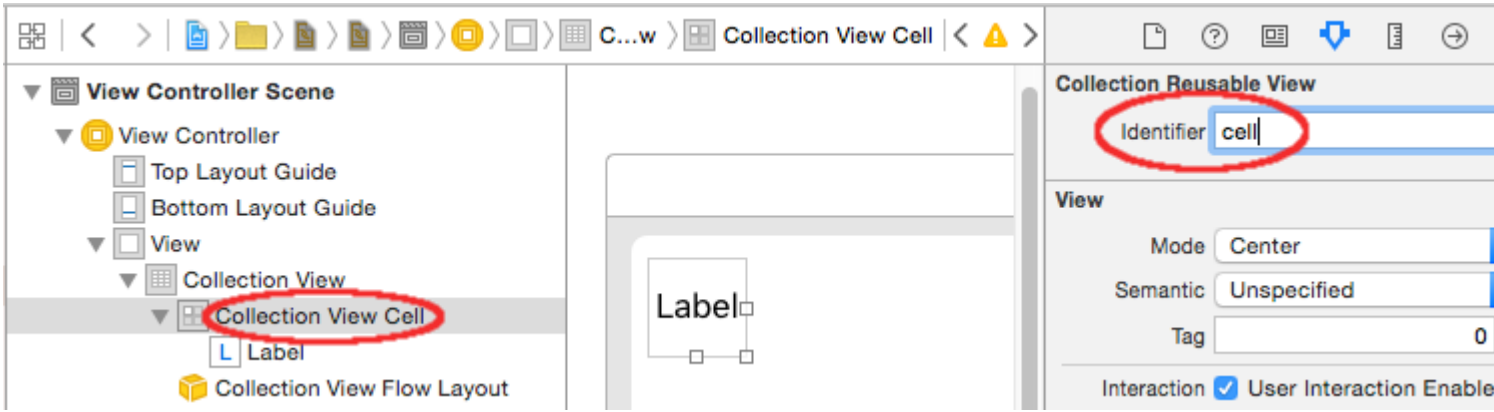
Collection ViewCollection View Cell。 。 。 Label。



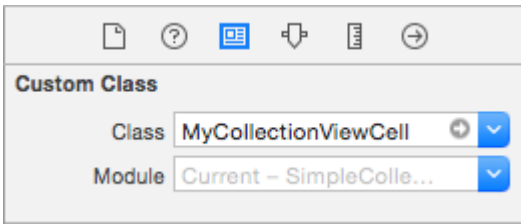
```

"""
"""。 ViewController.swiftlet reuseIdentifier = "cell"

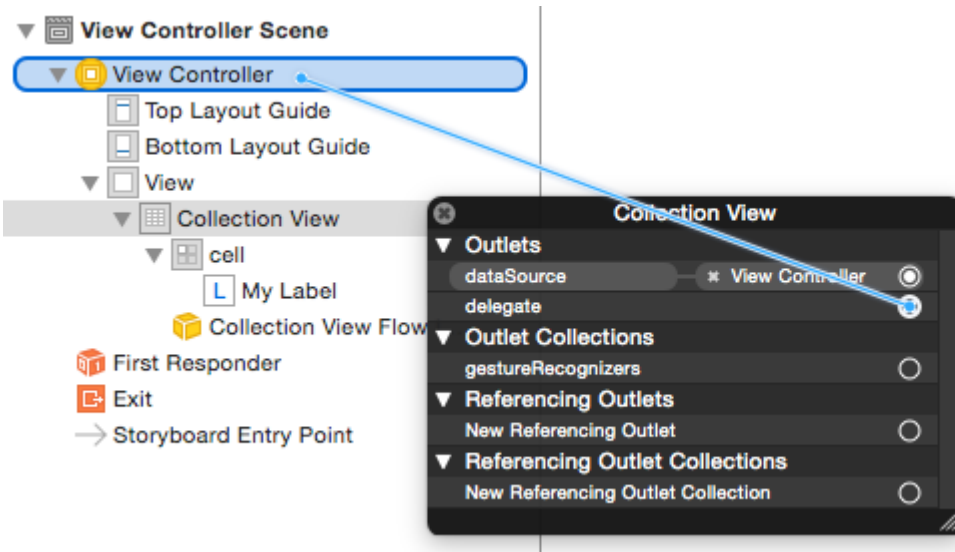
```



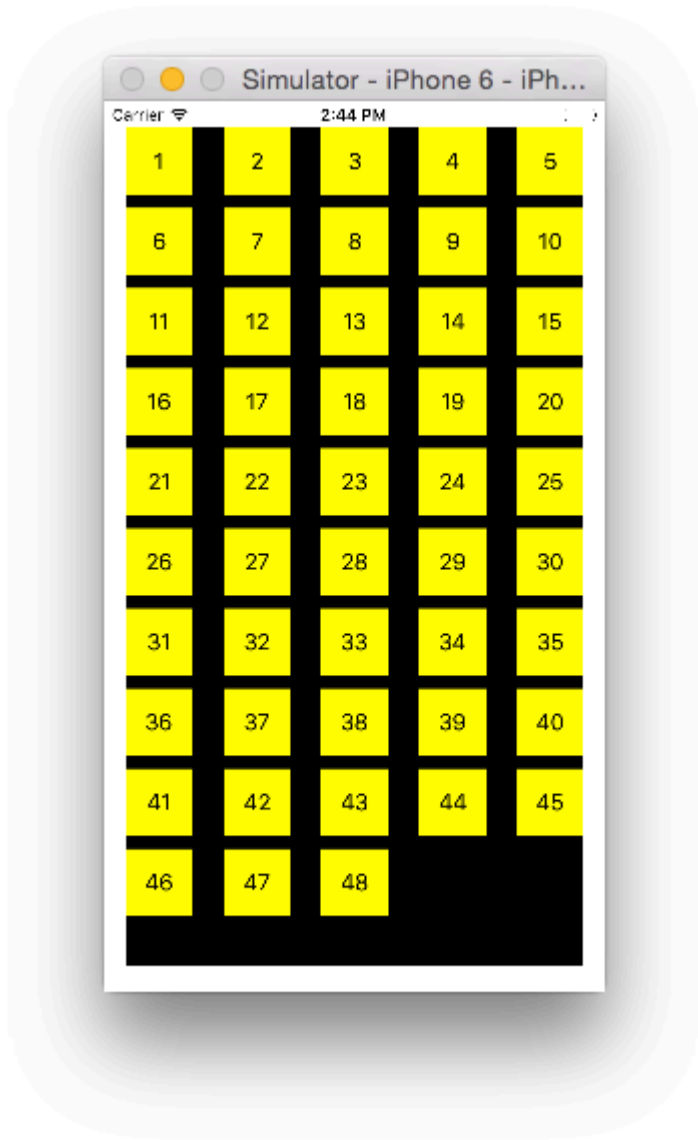
Identity Inspector `MyCollectionViewCell` ◦



- `LabelMyCollectionViewCellmyLabel` ◦ - ◦
- `CollectionView` `delegatedataSourceView Controller` ◦ "" "" ◦ ◦

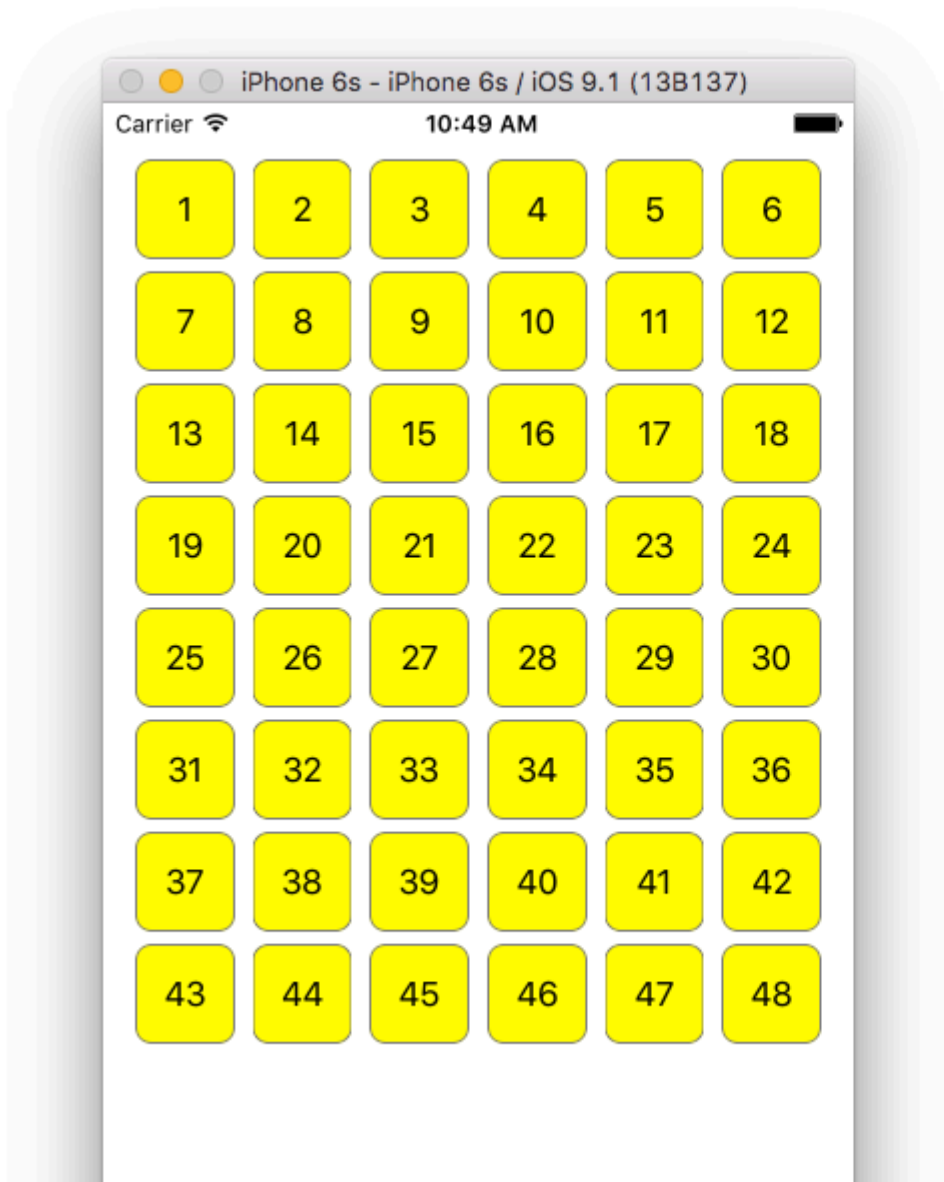


`LabelCollectionView` ◦



—

o



-
- [UICollectionView](#)
 - [UICollectionView1](#)
 - [UICollectionView2](#)

performBatchUpdates◦ ◦

```
collectionView.performBatchUpdates({
    // Perform updates
}, nil)
```

◦ indexPath

NSIndexPath

from old array to new array

reload ◦ ◦

deletion move-from reload `indexPath` insertion move-to reload `indexPath` ◦

```
let from = [1, 2, 3, 4, 5]
let to = [1, 3, 6, 4, 5]

collectionView.performBatchUpdates({
    collectionView.insertItemsAtIndexPaths([NSIndexPath(forItem: 2, inSection: 0)])
    collectionView.deleteItemsAtIndexPaths([NSIndexPath(forItem: 1, inSection: 0)])
    collectionView.moveItemAtIndexPath(NSIndexPath(forItem: 2, inSection: 0),
                                        toIndexPath: NSIndexPath(forItem: 1, inSection: 0))
}, nil)
```

UICollectionViewDelegate

UICollectionViewDelegate ◦

UIViewController MyViewController ◦

Objective-C

MyViewController.h UICollectionViewDelegate

```
@interface MyViewController : UIViewController <UICollectionViewDelegate, .../* previous existing delegate, as UICollectionViewDataSource */>
```

MyViewController.swift

```
class MyViewController : UICollectionViewDelegate {
}
```

Objective-C

```
-(void)collectionView:(UICollectionView *)collectionView didSelectItemAtIndexPath:(NSIndexPath *)indexPath
{
}
```

```
func collectionView(collectionView: UICollectionView, didSelectItemAtIndexPath indexPath: NSIndexPath)
{
}
```

◦

Objective-C

```
-(void)collectionView:(UICollectionView *)collectionView didSelectItemAtIndexPath:(NSIndexPath *)indexPath
{
    UICollectionViewCell* cell = [collectionView cellForItemAtIndexPath:indexPath];
    cell.backgroundColor = [UIColor greenColor];
}
```

```
class MyViewController : UICollectionViewDelegate {
    func collectionView(collectionView: UICollectionView, didSelectItemAtIndexPath indexPath: NSIndexPath)
    {
        var cell : UICollectionViewCell = collectionView.cellForItemAtIndexPath(indexPath)!
        cell.backgroundColor = UIColor.greenColor()
    }
}
```

DataSourceFlowLayout

didselect.

```
extension ProductsVC: UICollectionViewDelegate, UICollectionViewDataSource{

    // MARK: - UICollectionViewDataSource
    func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection
section: Int) -> Int {
        guard collectionView == collectionCategory else {
            return arrOfProducts.count
        }
        return arrOfCategory.count
    }

    func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath:
IndexPath) -> UICollectionViewCell {

        guard collectionView == collectionProduct else {
            let cell = collectionView.dequeueReusableCell(withReuseIdentifier:
"ProductCategoryCell", for: indexPath) as! ProductCategoryCell
            cell.viewBackground.layer.borderWidth = 0.5
            //Do some thing as per use
            return cell
        }

        let cell = collectionView.dequeueReusableCell(withReuseIdentifier: cellIdentifier,
for: indexPath) as! ProductCell
        cell.contentView.layer.borderWidth = 0.5
        cell.contentView.layer.borderColor = UIColor.black.cgColor
        let json = arrOfProducts[indexPath.row]
        //Do something as per use

        return cell
    }

    func collectionView(_ collectionView: UICollectionView, didSelectItemAt indexPath:
IndexPath) {
        guard collectionView == collectionCategory else {
```

```

        let json = arrOfProducts[indexPath.row]
        // Do something for collectionProduct here
        return
    }
    let json = arrOfCategory[indexPath.row] as [String: AnyObject]
    let id = json["cId"] as? String ?? ""
    // Do something
}

extension ProductsVC: UICollectionViewDelegateFlowLayout{

    // MARK: - UICollectionViewDelegateFlowLayout
    func collectionView(_ collectionView: UICollectionView, layout collectionViewLayout:
UICollectionViewLayout, sizeForItemAt indexPath: IndexPath) -> CGSize {

        let collectionWidth = collectionView.bounds.width
        guard collectionView == collectionProduct else {
            var itemWidth = collectionWidth / 4 - 1;

            if(UI_USER_INTERFACE_IDIOM() == .pad) {
                itemWidth = collectionWidth / 4 - 1;
            }
            return CGSize(width: itemWidth, height: 50)
        }

        var itemWidth = collectionWidth / 2 - 1;
        if(UI_USER_INTERFACE_IDIOM() == .pad) {
            itemWidth = collectionWidth / 4 - 1;
        }
        return CGSize(width: itemWidth, height: 250);
    }

    func collectionView(_ collectionView: UICollectionView, layout collectionViewLayout:
UICollectionViewLayout, minimumInteritemSpacingForSectionAt section: Int) -> CGFloat {
        return 1
    }

    func collectionView(_ collectionView: UICollectionView, layout collectionViewLayout:
UICollectionViewLayout, minimumLineSpacingForSectionAt section: Int) -> CGFloat {
        return 1
    }
}

```

23-01-2017

24-01-2017

25-01-2017

11:00

11:15

11:30

11:45

12:00

12:15

12:30

12:45

13:00

13:15

13:30

13:45

14:00

14:15

14:30

14:45

15:00

15:15

15:30

15:45

16:00

16:15

16:30

16:45

UICollectionView <https://riptutorial.com/zh-TW/ios/topic/2399/uicollectionview>

74: UIControl -

Examples

UIButton selector◦

```
import UIKit

class ViewController: UIViewController {
    @IBOutlet weak var button: UIButton!

    override func viewDidLoad() {
        super.viewDidLoad()

        let button = UIButton(frame: CGRect(x: 0, y: 0, width: 100, height: 44))
        button.addTarget(self, action: #selector(self.onButtonPress(_:)), for: .touchUpInside)
        self.view.addSubview(button)
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
    }

    func onButtonPress(_ button: UIButton!) {
        print("PRESSED")
    }
}
```

selector ◦ protocol◦

```
import UIKit

@objc
protocol ButtonEvent {
    @objc optional func onButtonPress(_ button: UIButton)
}

class ViewController: UIViewController, ButtonEvent {
    @IBOutlet weak var button: UIButton!

    override func viewDidLoad() {
        super.viewDidLoad()

        let button = UIButton(frame: CGRect(x: 0, y: 0, width: 100, height: 44))
        button.addTarget(self, action: #selector(ButtonEvent.onButtonPress(_:)), for:
        .touchUpInside)
        self.view.addSubview(button)
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
    }
}
```

onButtonPress◦

```

import Foundation
import UIKit

protocol RemovableTarget {
    func enable();
    func disable();
}

extension UIControl {
    func addEventHandler(event: UIControlEvents, runnable: (control: UIControl) -> Void) -> RemovableTarget {

        class Target : RemovableTarget {
            private var event: UIControlEvents
            private weak var control: UIControl?
            private var runnable: (control: UIControl) -> Void

            private init(event: UIControlEvents, control: UIControl, runnable: (control:
UIControl) -> Void) {
                self.event = event
                self.control = control
                self.runnable = runnable
            }

            @objc
            private func run(_ control: UIControl) {
                runnable(control: control)
            }

            private func enable() {
                control?.addTarget(self, action: #selector(Target.run(_:)), for: event)
                objc_setAssociatedObject(self, unsafeAddress(of: self), self,
.OBJC_ASSOCIATION_RETAIN)
            }

            private func disable() {
                control?.removeTarget(self, action: #selector(Target.run(_:)), for:
self.event)
                objc_setAssociatedObject(self, unsafeAddress(of: self), nil,
.OBJC_ASSOCIATION_ASSIGN)
            }

            let target = Target(event: event, control: self, runnable: runnable)
            target.enable()
            return target
        }
    }
}

```

UIControl ◦ func run(_ control: UIControl) ◦

object association UIControl ◦

Protocol Targetenableisable ◦

```
import Foundation
```



```

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        //Create a button.
        let button = UIButton(frame: CGRect(x: 0, y: 0, width: 100, height: 44))

        //Add an event action block/listener -- Handles Button Press.
        let target = button.addTarget(event: .touchUpInside) { (control) in
            print("Pressed")
        }

        self.view.addSubview(button)

        //Example of enabling/disabling the listener/event-action-block.
        DispatchQueue.main.after(when: DispatchTime.now() + 5) {
            target.disable() //Disable the listener.

            DispatchQueue.main.after(when: DispatchTime.now() + 5) {
                target.enable() //Enable the listener.
            }
        }
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
    }
}

```

UIControl - <https://riptutorial.com/zh-TW/ios/topic/3180/uicontrol---->

75: UIFeedbackGenerator

UIFeedbackGenerator iPhone 7 iOS Taptic Engine Haptics Taptics
UIFeedbackGenerator
UIFeedbackGenerator

Examples

UIImpactFeedbackGenerator

```
class ViewController: UIViewController
{
    lazy var button: UIButton =
    {
        let button = UIButton()
        button.translatesAutoresizingMaskIntoConstraints = false
        self.view.addSubview(button)
        button.centerXAnchor.constraint(equalTo: self.view.centerXAnchor).isActive = true
        button.centerYAnchor.constraint(equalTo: self.view.centerYAnchor).isActive = true
        button.setTitle("Impact", for: .normal)
        button.setTitleColor(UIColor.gray, for: .normal)
        return button
    }()

    // Choose between heavy, medium, and light for style
    let impactFeedbackGenerator = UIImpactFeedbackGenerator(style: .heavy)

    override func viewDidLoad()
    {
        super.viewDidLoad()
        button.addTarget(self, action: #selector(self.didPressButton(sender:)), for:
        .touchUpInside)

        // Primes feedback generator for upcoming events and reduces latency
        impactFeedbackGenerator.prepare()
    }

    func didPressButton(sender: UIButton)
    {
        // Triggers haptic
        impactFeedbackGenerator.impactOccurred()
    }
}
```

Objective-C

```
@interface ViewController ()
@property (nonatomic, strong) UIImpactFeedbackGenerator *impactFeedbackGenerator;
@property (nonatomic, strong) UIButton *button;
@end

@implementation ViewController

- (void)viewDidLoad
```

```

{
    [super viewDidLoad];
    [self.button addTarget:self action:@selector(didPressButton:)
forControlEvents:UIControlEventTouchUpInside];

    // Choose between heavy, medium, and light for style
    self.impactFeedbackGenerator = [[UIImpactFeedbackGenerator alloc]
initWithStyle:UIImpactFeedbackStyleHeavy];

    // Primes feedback generator for upcoming events and reduces latency
    [self.impactFeedbackGenerator prepare];
}

- (void)didPressButton:(UIButton *)sender
{
    // Triggers haptic
    [self.impactFeedbackGenerator impactOccurred];
}

#pragma mark - Lazy Init
- (UIButton *)button
{
    if (!_button)
    {
        _button = [[UIButton alloc]init];
        _button.translatesAutoresizingMaskIntoConstraints = NO;
        [self.view addSubview:_button];
        [_button.centerXAnchor constraintEqualToAnchor:self.view.centerXAnchor].active = YES;
        [_button.centerYAnchor constraintEqualToAnchor:self.view.centerYAnchor].active = YES;
        [_button setTitle:@"Impact" forState:UIControlStateNormal];
        [_button setTitleColor:[UIColor grayColor] forState:UIControlStateNormal];
    }
    return _button;
}

@end

```

UIFeedbackGenerator <https://riptutorial.com/zh-TW/ios/topic/10048/uifeedbackgenerator>

76: UIFont

UIFont ◦ NSObject Hashable Equatable CVarArgNSCopying ◦

Examples

UIFont

UIFont

```
var font: UIFont!
```

UIFontinit()

- UIFont.init(descriptor: UIFontDescriptor, size: CGFloat)
- UIFont.init(name: String, size: CGFloat)

UIFont

```
let font = UIFont(name: "Helvetica Neue", size: 15)
```

System 17 ◦

font

```
label.font = UIFont(name:"Helvetica Neue", size: 15)
```

Helvetica Neue 15 ◦ **nil**

Optionalnil

UIFont <https://riptutorial.com/zh-TW/ios/topic/9792/uifont>

77: UITapGestureRecognizer

Examples

UITapGestureRecognizer

```
self.action = UITapGestureRecognizer(target: self, action: #selector(handleTap(_:)))
```

```
o
```

```
override func viewDidLoad() {
    super.viewDidLoad()
    let recognizer = UITapGestureRecognizer(target: self,
                                          action: #selector(handleTap(_:)))
    view.addGestureRecognizer(recognizer)
}

func handleTap(recognizer: UITapGestureRecognizer) {
}

```

Objective-C

```
- (void)viewDidLoad {
    [super viewDidLoad];
    UITapGestureRecognizer *recognizer =
        [[UITapGestureRecognizer alloc] initWithTarget:self
                                             action:@selector(handleTap:)];
    [self.view addGestureRecognizer:recognizer];
}

- (void)handleTap:(UITapGestureRecognizer *)recognizer {
}

```

UITapGestureRecognizer

```
func dismissKeyboard() {
    view.endEditing(true)
}

```

```
let tap: UITapGestureRecognizer = UITapGestureRecognizer(target: self, action:
"dismissKeyboard")
view.addGestureRecognizer(tap)

```

UITapGestureRecognizerSwift 3

```
func handleTap(gestureRecognizer: UITapGestureRecognizer) {
    print("tap working")
}

```

```

if gestureRecognizer.state == UIGestureRecognizerState.recognized
{
    print(gestureRecognizer.location(in: gestureRecognizer.view))
}
}

```

UIPanGestureRecognizer

◦ ◦

Objective-C

```

- (void)viewDidLoad {
    [super viewDidLoad];

    UIImageView *imageView = [[UIImageView alloc] initWithImage:[UIImage
imageName:@"imageToDrag"]];
    [imageView sizeToFit];
    imageView.userInteractionEnabled = YES;
    [self.view addSubview:imageView];

    UIPanGestureRecognizer *pan = [[UIPanGestureRecognizer alloc] initWithTarget:self
action:@selector(handlePan:)];
    [imageView addGestureRecognizer:pan];
}

- (void)handlePan:(UIPanGestureRecognizer *)recognizer {
    CGPoint translation = [recognizer translationInView:self.view];
    recognizer.view.center = CGPointMake(recognizer.view.center.x + translation.x,
                                           recognizer.view.center.y + translation.y);
    [recognizer setTranslation:CGPointZero inView:self.view];
}

```

```

override func viewDidLoad() {
    super.viewDidLoad()

    let imageView = UIImageView.init(image: UIImage.init(named: "imageToDrag"))
    imageView.sizeToFit()
    imageView.isUserInteractionEnabled = true
    self.view.addSubview(imageView)

    let pan = UIPanGestureRecognizer.init(target: self, action:
#selector(handlePan(recognizer:)))
    imageView.addGestureRecognizer(pan)
}

func handlePan(recognizer: UIPanGestureRecognizer) {
    let translation = recognizer.translation(in: self.view)
    if let view = recognizer.view {
        view.center = CGPoint(x: view.center.x + translation.x, y: view.center.y +
translation.y)
    }
    recognizer.setTranslation(CGPoint.zero, in: self.view)
}

```

UIPanGestureRecognizer//UISwipeGestureRecognizer ◦ translationInView:velocityInView:

UIPanGestureRecognizer°

UITapGestureRecognizer

UITapGestureRecognizer ° numberOfTapsRequired2 °

```
override func viewDidLoad() {
    super.viewDidLoad()

    // Double Tap
    let doubleTapGesture = UITapGestureRecognizer(target: self, action:
#selector(handleDoubleTap))
    doubleTapGesture.numberOfTapsRequired = 2
    doubleTapView.addGestureRecognizer(doubleTapGesture)
}

// Double tap action
func handleDoubleTap() {
    label.text = "Double tap recognized"
}
```

-
- °
 - numberOfTapsRequired3°

UILongPressGestureRecognizer

UILongPressGestureRecognizer° °

```
override func viewDidLoad() {
    super.viewDidLoad()

    // Long Press
    let longPressGesture = UILongPressGestureRecognizer(target: self, action:
#selector(handleLongPress(_:)))
    longPressView.addGestureRecognizer(longPressGesture)
}

// Long press action
func handleLongPress(gesture: UILongPressGestureRecognizer) {
    if gesture.state == UIGestureRecognizerState.Began {
        label.text = "Long press recognized"
    }
}
```

-
- °
 - minimumPressDuration°

UISwipeGestureRecognizer

```

override func viewDidLoad() {
    super.viewDidLoad()

    // Swipe (right and left)
    let swipeRightGesture = UISwipeGestureRecognizer(target: self, action:
#selector(handleSwipe(_:)))
    let swipeLeftGesture = UISwipeGestureRecognizer(target: self, action:
#selector(handleSwipe(_:)))
    swipeRightGesture.direction = UISwipeGestureRecognizerDirection.Right
    swipeLeftGesture.direction = UISwipeGestureRecognizerDirection.Left
    swipeView.addGestureRecognizer(swipeRightGesture)
    swipeView.addGestureRecognizer(swipeLeftGesture)
}

// Swipe action
func handleSwipe(gesture: UISwipeGestureRecognizer) {
    label.text = "Swipe recognized"

    // example task: animate view off screen
    let originalLocation = swipeView.center
    if gesture.direction == UISwipeGestureRecognizerDirection.Right {
        label.text = "Swipe right"
    } else if gesture.direction == UISwipeGestureRecognizerDirection.Left {
        label.text = "Swipe left"
    }
}
}

```

Objective-C

```

- (void)viewDidLoad
{
    [super viewDidLoad];
    UISwipeGestureRecognizer *swipeLeft = [[UISwipeGestureRecognizer alloc] initWithTarget:self
    action:@selector(handleSwipe:)];
    UISwipeGestureRecognizer *swipeRight = [[UISwipeGestureRecognizer alloc] initWithTarget:self
    action:@selector(handleSwipe:)];

    // Setting the swipe direction.
    [swipeLeft setDirection:UISwipeGestureRecognizerDirectionLeft];
    [swipeRight setDirection:UISwipeGestureRecognizerDirectionRight];

    // Adding the swipe gesture on image view
    [self.view addGestureRecognizer:swipeLeft];
    [self.view addGestureRecognizer:swipeRight];
}

//Handling Swipe Gesture Events

- (void)handleSwipe:(UISwipeGestureRecognizer *)swipe {

    if (swipe.direction == UISwipeGestureRecognizerDirectionLeft) {
        NSLog(@"Left Swipe");
    }

    if (swipe.direction == UISwipeGestureRecognizerDirectionRight) {
        NSLog(@"Right Swipe");
    }
}

```



```
}
```

- ◦

UIPinchGestureRecognizer

◦ ◦

```
override func viewDidLoad() {
    super.viewDidLoad()

    // Pinch
    let pinchGesture = UIPinchGestureRecognizer(target: self, action:
#selector(handlePinch(_:)))
    pinchView.addGestureRecognizer(pinchGesture)
}

// Pinch action
func handlePinch(gesture: UIPinchGestureRecognizer) {
    label.text = "Pinch recognized"

    if gesture.state == UIGestureRecognizerState.Changed {
        let transform = CGAffineTransformMakeScale(gesture.scale, gesture.scale)
        pinchView.transform = transform
    }
}
```

- ◦

UIRotationGestureRecognizer

UIRotationGestureRecognizer ◦ ◦

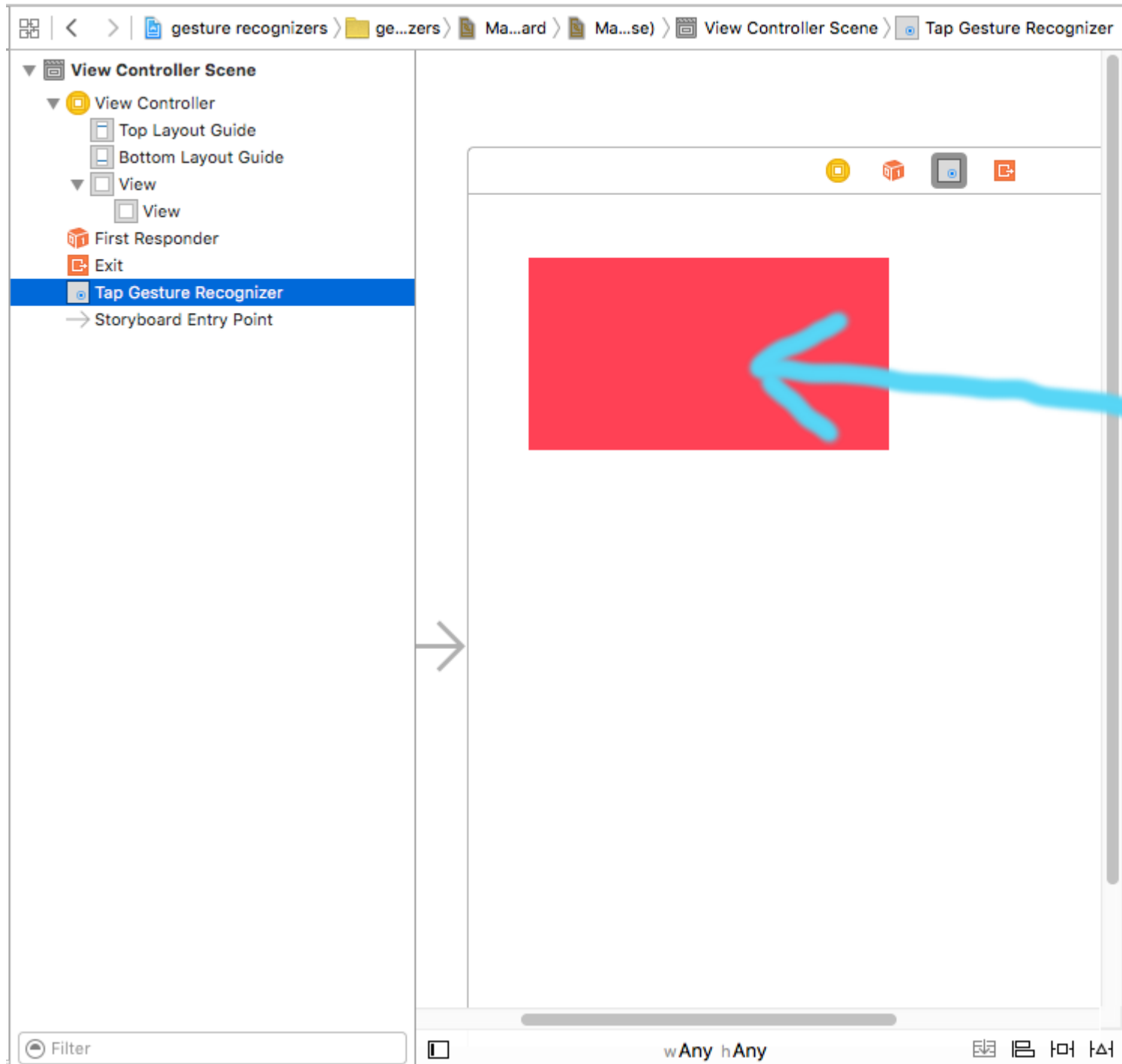
```
override func viewDidLoad() {
    super.viewDidLoad()

    // Rotate
    let rotateGesture = UIRotationGestureRecognizer(target: self, action:
#selector(handleRotate(_:)))
    rotateView.addGestureRecognizer(rotateGesture)
}

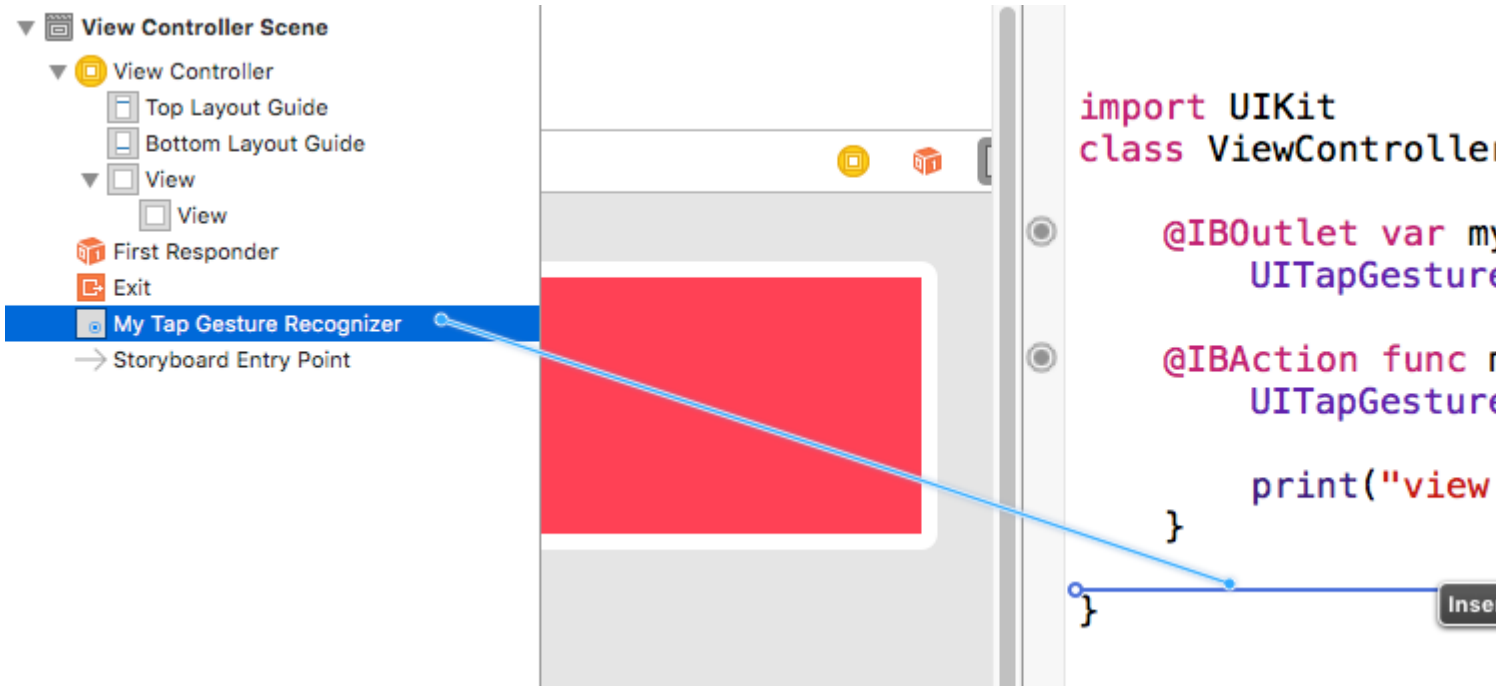
// Rotate action
func handleRotate(gesture: UIRotationGestureRecognizer) {
    label.text = "Rotate recognized"

    if gesture.state == UIGestureRecognizerState.Changed {
        let transform = CGAffineTransformMakeRotation(gesture.rotation)
        rotateView.transform = transform
    }
}
```

Interface Builder



View ControllerOutletAction.



[UIGestureRecognizer](https://riptutorial.com/zh-TW/ios/topic/1289/uigesturerecognizer) <https://riptutorial.com/zh-TW/ios/topic/1289/uigesturerecognizer>

78: UIKit

UIKit DynamicsUIKit ◦ ◦

UIKit DynamicsiOS◦

UIKit Dynamics◦ UIDynamicBehavior◦ ◦ UIDynamicBehavior◦

◦ UILabel labelUIGravityBehavior

```
label.frame = CGRect(x: 0.0, y: 0.0, width: label.intrinsicContentSize.width, height:
label.intrinsicContentSize.height)
dynamicAnimator.updateItem(usingCurrentState: label)
```

Objective-C

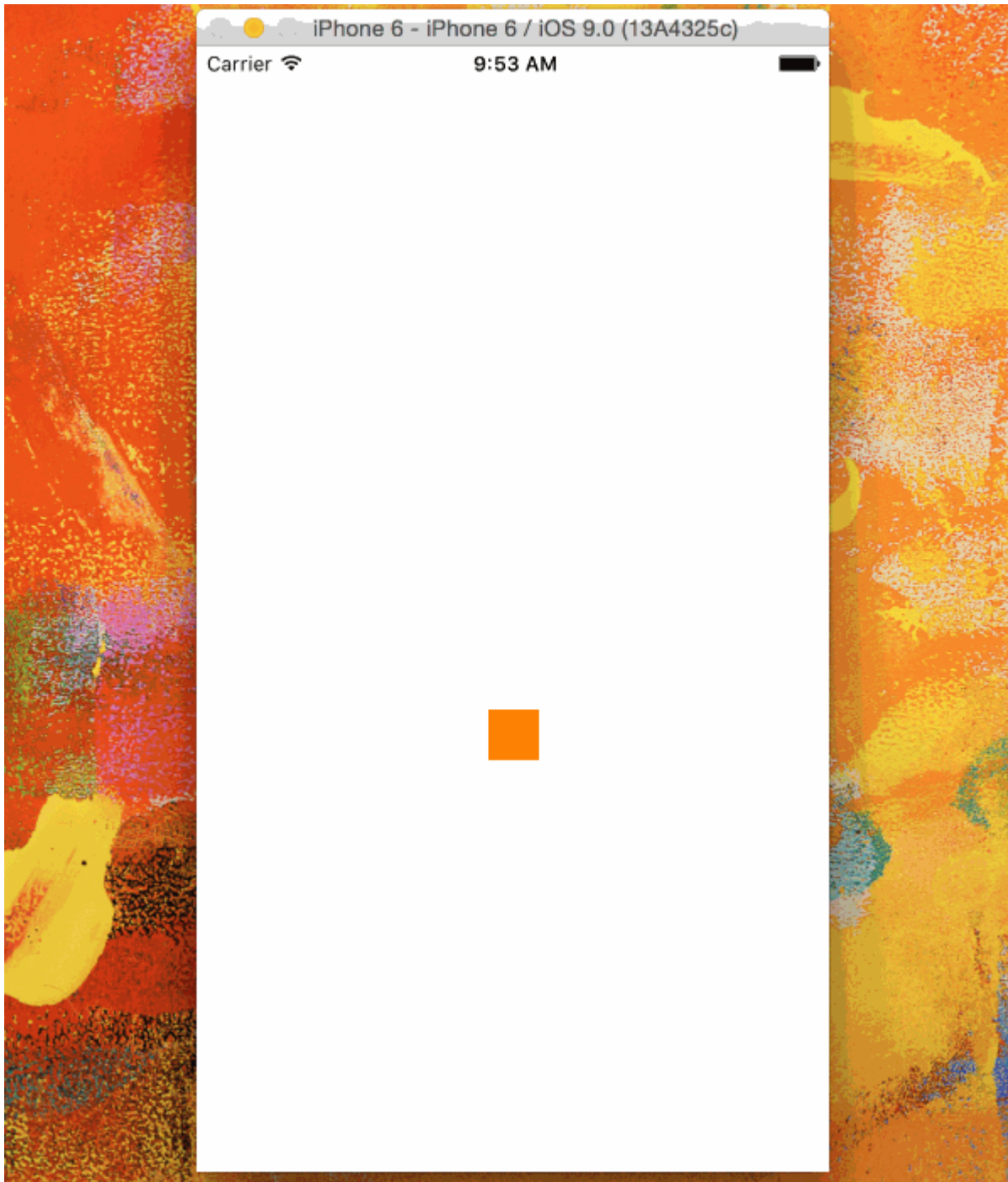
```
self.label.frame = CGRectMake(0.0, 0.0, self.label.intrinsicContentSize.width,
self.label.intrinsicContentSize.height);
[self.dynamicAnimator updateItemUsingCurrentState: self.label];
```

◦

UIDynamicBehaviors◦ UIAttachmentBehaviortouchesMoved UIGestureRecognizeranchorPoint◦

Examples

◦



```
@IBOutlet var animationView: UIView!  
var squareView: UIView!  
var collision: UICollisionBehavior!  
var animator: UIDynamicAnimator!  
var gravity: UIGravityBehavior!  
  
override func viewDidLoad() {  
    super.viewDidLoad()  
    let squareSize = CGSize(width: 30.0, height: 30.0)  
    let centerPoint = CGPoint(x: self.animationView.bounds.midX - (squareSize.width/2), y:  
self.animationView.bounds.midY - (squareSize.height/2))  
    let frame = CGRect(origin: centerPoint, size: squareSize)  
    squareView = UIView(frame: frame)  
    squareView.backgroundColor = UIColor.orangeColor()  
    animationView.addSubview(squareView)  
    animator = UIDynamicAnimator(referenceView: view)
```

```

gravity = UIGravityBehavior(items: [squareView])
animator.addBehavior(gravity)
collision = UICollisionBehavior(items: [square])
collision.translatesReferenceBoundsIntoBoundary = true
animator.addBehavior(collision)
}

```

o

Carrier ▼

11:34 AM



```

class ViewController: UIViewController
{
    // Adjust to change speed of view from flick
    let magnitudeMultiplier: CGFloat = 0.0008

    lazy var dynamicAnimator: UIDynamicAnimator =
    {
        let dynamicAnimator = UIDynamicAnimator(referenceView: self.view)
        return dynamicAnimator
    }()

    lazy var gravity: UIGravityBehavior =
    {
        let gravity = UIGravityBehavior(items: [self.orangeView])
        return gravity
    }()

    lazy var collision: UICollisionBehavior =
    {
        let collision = UICollisionBehavior(items: [self.orangeView])
        collision.translatesReferenceBoundsIntoBoundary = true
        return collision
    }()

    lazy var orangeView: UIView =
    {
        let widthHeight: CGFloat = 40.0
        let orangeView = UIView(frame: CGRect(x: 0.0, y: 0.0, width: widthHeight, height:
widthHeight))
        orangeView.backgroundColor = UIColor.orange
        self.view.addSubview(orangeView)
        return orangeView
    }()
}

```

```

}()

lazy var panGesture: UIPanGestureRecognizer =
{
    let panGesture = UIPanGestureRecognizer(target: self, action:
#selector(self.handlePan(sender:)))
    return panGesture
}()

lazy var attachment: UIAttachmentBehavior =
{
    let attachment = UIAttachmentBehavior(item: self.orangeView, attachedToAnchor: .zero)
    return attachment
}()

override func viewDidLoad()
{
    super.viewDidLoad()
    dynamicAnimator.addBehavior(gravity)
    dynamicAnimator.addBehavior(collision)
    orangeView.addGestureRecognizer(panGesture)
}

override func viewDidLoadSubviews()
{
    super.viewDidLoadSubviews()
    orangeView.center = view.center
    dynamicAnimator.updateItem(usingCurrentState: orangeView)
}

func handlePan(sender: UIPanGestureRecognizer)
{
    let location = sender.location(in: view)
    let velocity = sender.velocity(in: view)
    let magnitude = sqrt((velocity.x * velocity.x) + (velocity.y * velocity.y))
    switch sender.state
    {
    case .began:
        attachment.anchorPoint = location
        dynamicAnimator.addBehavior(attachment)
    case .changed:
        attachment.anchorPoint = location
    case .cancelled, .ended, .failed, .possible:
        let push = UIPushBehavior(items: [self.orangeView], mode: .instantaneous)
        push.pushDirection = CGVector(dx: velocity.x, dy: velocity.y)
        push.magnitude = magnitude * magnitudeMultiplier
        dynamicAnimator.removeBehavior(attachment)
        dynamicAnimator.addBehavior(push)
    }
}
}

```

Objective-C

```

@interface ViewController ()

@property (nonatomic, assign) CGFloat magnitudeMultiplier;
@property (nonatomic, strong) UIDynamicAnimator *dynamicAnimator;
@property (nonatomic, strong) UIGravityBehavior *gravity;

```

```

@property (nonatomic, strong) UICollisionBehavior *collision;
@property (nonatomic, strong) UIView *orangeView;
@property (nonatomic, strong) UIPanGestureRecognizer *panGesture;
@property (nonatomic, strong) UIAttachmentBehavior *attachment;

@end

@implementation ViewController

- (void) viewDidLoad
{
    [super viewDidLoad];
    [self.dynamicAnimator addBehavior:self.gravity];
    [self.dynamicAnimator addBehavior:self.collision];
    [self.orangeView addGestureRecognizer:self.panGesture];
    // Adjust to change speed of view from flick
    self.magnitudeMultiplier = 0.0008f;
}

- (void) viewDidLoadSubviews
{
    [super viewDidLoadSubviews];
    self.orangeView.center = self.view.center;
    [self.dynamicAnimator updateItemUsingCurrentState:self.orangeView];
}

- (void) handlePan:(UIPanGestureRecognizer *) sender
{
    CGPoint location = [sender locationInView:self.view];
    CGPoint velocity = [sender velocityInView:self.view];
    CGFloat magnitude = sqrt((velocity.x * velocity.x) + (velocity.y * velocity.y));
    if (sender.state == UIGestureRecognizerStateBegan)
    {
        self.attachment.anchorPoint = location;
        [self.dynamicAnimator addBehavior:self.attachment];
    }
    else if (sender.state == UIGestureRecognizerStateChanged)
    {
        self.attachment.anchorPoint = location;
    }
    else if (sender.state == UIGestureRecognizerStateCancelled ||
             sender.state == UIGestureRecognizerStateEnded ||
             sender.state == UIGestureRecognizerStateFailed ||
             sender.state == UIGestureRecognizerStatePossible)
    {
        UIPushBehavior *push = [[UIPushBehavior alloc] initWithItems:@[self.orangeView]
mode:UIPushBehaviorModeInstantaneous];
        push.pushDirection = CGVectorMake(velocity.x, velocity.y);
        push.magnitude = magnitude * self.magnitudeMultiplier;
        [self.dynamicAnimator removeBehavior:self.attachment];
        [self.dynamicAnimator addBehavior:push];
    }
}

#pragma mark - Lazy Init
- (UIDynamicAnimator *) dynamicAnimator
{
    if (!_dynamicAnimator)
    {
        _dynamicAnimator = [[UIDynamicAnimator alloc] initWithReferenceView:self.view];
    }
}

```



```

    return _dynamicAnimator;
}

- (UIGravityBehavior *)gravity
{
    if (!_gravity)
    {
        _gravity = [[UIGravityBehavior alloc] initWithItems:@[self.orangeView]];
    }
    return _gravity;
}

- (UICollisionBehavior *)collision
{
    if (!_collision)
    {
        _collision = [[UICollisionBehavior alloc] initWithItems:@[self.orangeView]];
        _collision.translatesReferenceBoundsIntoBoundary = YES;
    }
    return _collision;
}

- (UIView *)orangeView
{
    if (!_orangeView)
    {
        CGFloat widthHeight = 40.0f;
        _orangeView = [[UIView alloc] initWithFrame:CGRectMake(0.0, 0.0, widthHeight,
widthHeight)];
        _orangeView.backgroundColor = [UIColor orangeColor];
        [self.view addSubview:_orangeView];
    }
    return _orangeView;
}

- (UIPanGestureRecognizer *)panGesture
{
    if (!_panGesture)
    {
        _panGesture = [[UIPanGestureRecognizer alloc] initWithTarget:self
action:@selector(handlePan:)];
    }
    return _panGesture;
}

- (UIAttachmentBehavior *)attachment
{
    if (!_attachment)
    {
        _attachment = [[UIAttachmentBehavior alloc] initWithItem:self.orangeView
attachedToAnchor:CGPointZero];
    }
    return _attachment;
}

@end

```

UIFieldBehaviors“”

FaceTime。



```
class ViewController: UIViewController
{
    lazy var dynamicAnimator: UIDynamicAnimator =
    {
        let dynamicAnimator = UIDynamicAnimator(referenceView: self.view)
        return dynamicAnimator
    }()

    lazy var collision: UICollisionBehavior =
    {
        let collision = UICollisionBehavior(items: [self.orangeView])
        collision.translatesReferenceBoundsIntoBoundary = true
        return collision
    }()

    lazy var fieldBehaviors: [UIFieldBehavior] =
    {
        var fieldBehaviors = [UIFieldBehavior]()
        for _ in 0 ..< 2
        {
            let field = UIFieldBehavior.springField()
            field.addItem(self.orangeView)
            fieldBehaviors.append(field)
        }
        return fieldBehaviors
    }()

    lazy var itemBehavior: UIDynamicItemBehavior =
    {
        let itemBehavior = UIDynamicItemBehavior(items: [self.orangeView])
        // Adjust these values to change the "stickiness" of the view
        itemBehavior.density = 0.01
        itemBehavior.resistance = 10
        itemBehavior.friction = 0.0
        itemBehavior.allowsRotation = false
        return itemBehavior
    }()

    lazy var orangeView: UIView =
    {
        let widthHeight: CGFloat = 40.0
        let orangeView = UIView(frame: CGRect(x: 0.0, y: 0.0, width: widthHeight, height:
```

```

widthHeight))
    orangeView.backgroundColor = UIColor.orange
    self.view.addSubview(orangeView)
    return orangeView
}()

lazy var panGesture: UIPanGestureRecognizer =
{
    let panGesture = UIPanGestureRecognizer(target: self, action:
#selector(self.handlePan(sender:)))
    return panGesture
}()

lazy var attachment: UIAttachmentBehavior =
{
    let attachment = UIAttachmentBehavior(item: self.orangeView, attachedToAnchor: .zero)
    return attachment
}()

override func viewDidLoad()
{
    super.viewDidLoad()
    dynamicAnimator.addBehavior(collision)
    dynamicAnimator.addBehavior(itemBehavior)
    for field in fieldBehaviors
    {
        dynamicAnimator.addBehavior(field)
    }

    orangeView.addGestureRecognizer(panGesture)
}

override func viewDidLoadSubviews()
{
    super.viewDidLoadSubviews()

    orangeView.center = view.center
    dynamicAnimator.updateItem(usingCurrentState: orangeView)

    for (index, field) in fieldBehaviors.enumerated()
    {
        field.position = CGPoint(x: view.bounds
            .midX, y: view.bounds.height * (0.25 + 0.5 * CGFloat(index)))
        field.region = UIRegion(size: CGSize(width: view.bounds.width, height:
view.bounds.height * 0.5))
    }
}

func handlePan(sender: UIPanGestureRecognizer)
{
    let location = sender.location(in: view)
    let velocity = sender.velocity(in: view)
    switch sender.state
    {
    case .began:
        attachment.anchorPoint = location
        dynamicAnimator.addBehavior(attachment)
    case .changed:
        attachment.anchorPoint = location
    case .cancelled, .ended, .failed, .possible:
        itemBehavior.addLinearVelocity(velocity, for: self.orangeView)
    }
}

```

```

        dynamicAnimator.removeBehavior(attachment)
    }
}
}

```

Objective-C

```

@interface ViewController ()

@property (nonatomic, strong) UIDynamicAnimator *dynamicAnimator;
@property (nonatomic, strong) UICollisionBehavior *collision;
@property (nonatomic, strong) UIAttachmentBehavior *attachment;
@property (nonatomic, strong) UIDynamicItemBehavior *itemBehavior;
@property (nonatomic, strong) NSArray <UIFieldBehavior *> *fieldBehaviors;
@property (nonatomic, strong) UIView *orangeView;
@property (nonatomic, strong) UIPanGestureRecognizer *panGesture;

@end

@implementation ViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    [self.dynamicAnimator addBehavior:self.collision];
    [self.dynamicAnimator addBehavior:self.itemBehavior];
    for (UIFieldBehavior *field in self.fieldBehaviors)
    {
        [self.dynamicAnimator addBehavior:field];
    }

    [self.orangeView addGestureRecognizer:self.panGesture];
}

- (void)viewDidLayoutSubviews
{
    [super viewDidLayoutSubviews];
    self.orangeView.center = self.view.center;
    [self.dynamicAnimator updateItemUsingCurrentState:self.orangeView];

    for (NSInteger i = 0; i < self.fieldBehaviors.count; i++)
    {
        UIFieldBehavior *field = self.fieldBehaviors[i];
        field.position = CGPointMake(CGRectGetMidX(self.view.bounds),
        CGRectGetHeight(self.view.bounds) * (0.25f + 0.5f * i));
        field.region = [[UIRegion
        alloc] initWithSize:CGSizeMake(CGRectGetWidth(self.view.bounds),
        CGRectGetHeight(self.view.bounds) * 0.5)];
    }
}

- (void)handlePan:(UIPanGestureRecognizer *)sender
{
    CGPoint location = [sender locationInView:self.view];
    CGPoint velocity = [sender velocityInView:self.view];
    if (sender.state == UIGestureRecognizerStateBegan)
    {
        self.attachment.anchorPoint = location;
        [self.dynamicAnimator addBehavior:self.attachment];
    }
}

```

```

    }
    else if (sender.state == UIGestureRecognizerStateChanged)
    {
        self.attachment.anchorPoint = location;
    }
    else if (sender.state == UIGestureRecognizerStateCancelled ||
            sender.state == UIGestureRecognizerStateEnded ||
            sender.state == UIGestureRecognizerStateFailed ||
            sender.state == UIGestureRecognizerStatePossible)
    {
        [self.itemBehavior addLinearVelocity:velocity forItem:self.orangeView];
        [self.dynamicAnimator removeBehavior:self.attachment];
    }
}

#pragma mark - Lazy Init
- (UIDynamicAnimator *)dynamicAnimator
{
    if (!_dynamicAnimator)
    {
        _dynamicAnimator = [[UIDynamicAnimator alloc] initWithReferenceView:self.view];
    }
    return _dynamicAnimator;
}

- (UICollisionBehavior *)collision
{
    if (!_collision)
    {
        _collision = [[UICollisionBehavior alloc] initWithItems:@[self.orangeView]];
        _collision.translatesReferenceBoundsIntoBoundary = YES;
    }
    return _collision;
}

- (NSArray <UIFieldBehavior *> *)fieldBehaviors
{
    if (!_fieldBehaviors)
    {
        NSMutableArray *fields = [[NSMutableArray alloc] init];
        for (NSInteger i = 0; i < 2; i++)
        {
            UIFieldBehavior *field = [UIFieldBehavior springField];
            [field addItem:self.orangeView];
            [fields addObject:field];
        }
        _fieldBehaviors = fields;
    }
    return _fieldBehaviors;
}

- (UIDynamicItemBehavior *)itemBehavior
{
    if (!_itemBehavior)
    {
        _itemBehavior = [[UIDynamicItemBehavior alloc] initWithItems:@[self.orangeView]];
        // Adjust these values to change the "stickiness" of the view
        _itemBehavior.density = 0.01;
        _itemBehavior.resistance = 10;
        _itemBehavior.friction = 0.0;
        _itemBehavior.allowsRotation = NO;
    }
}

```

```

    }
    return _itemBehavior;
}

- (UIView *)orangeView
{
    if (!_orangeView)
    {
        CGFloat widthHeight = 40.0f;
        _orangeView = [[UIView alloc] initWithFrame:CGRectMake(0.0, 0.0, widthHeight,
widthHeight)];
        _orangeView.backgroundColor = [UIColor orangeColor];
        [self.view addSubview:_orangeView];
    }
    return _orangeView;
}

- (UIPanGestureRecognizer *)panGesture
{
    if (!_panGesture)
    {
        _panGesture = [[UIPanGestureRecognizer alloc] initWithTarget:self
action:@selector(handlePan:)];
    }
    return _panGesture;
}

- (UIAttachmentBehavior *)attachment
{
    if (!_attachment)
    {
        _attachment = [[UIAttachmentBehavior alloc] initWithItem:self.orangeView
attachedToAnchor:CGPointZero];
    }
    return _attachment;
}

@end

```

UIFieldBehaviors [2015 WWDC“UIKit”](#) ◦

UIDynamicBehavior



UIDynamicBehavior ◦ ◦

```
class PresentingViewController: UIViewController
{
    lazy var button: UIButton =
    {
        let button = UIButton()
        button.translatesAutoresizingMaskIntoConstraints = false
        self.view.addSubview(button)
        button.centerXAnchor.constraint(equalTo: self.view.centerXAnchor).isActive
            = true
        button.centerYAnchor.constraint(equalTo: self.view.centerYAnchor).isActive = true
        button.setTitle("Present", for: .normal)
        button.setTextColor(UIColor.blue, for: .normal)

        return button
    }()

    override func viewDidLoad()
    {
        super.viewDidLoad()
        button.addTarget(self, action: #selector(self.didPressPresent), for: .touchUpInside)
    }

    func didPressPresent()
    {
        let modal = ModalViewController()
        modal.view.frame = CGRect(x: 0.0, y: 0.0, width: 200.0, height: 200.0)
        modal.modalPresentationStyle = .custom
        modal.transitioningDelegate = modal
        self.present(modal, animated: true)
    }
}
```

Objective-C

```
@interface PresentingViewController ()
@property (nonatomic, strong) UIButton *button;
@end
```

```

@implementation PresentingViewController

- (void) viewDidLoad
{
    [super viewDidLoad];
    [self.button addTarget:self action:@selector(didPressPresent)
    forControlEvents:UIControlEventTouchUpInside];
}

- (void) didPressPresent
{
    ModalViewController *modal = [[ModalViewController alloc] init];
    modal.view.frame = CGRectMake(0.0, 0.0, 200.0, 200.0);
    modal.modalPresentationStyle = UIModalPresentationCustom;
    modal.transitioningDelegate = modal;
    [self presentViewController:modal animated:YES completion:nil];
}

- (UIButton *) button
{
    if (!_button)
    {
        _button = [[UIButton alloc] init];
        _button.translatesAutoresizingMaskIntoConstraints = NO;
        [self.view addSubview:_button];
        [_button.centerXAnchor constraintEqualToAnchor:self.view.centerXAnchor].active = YES;
        [_button.centerYAnchor constraintEqualToAnchor:self.view.centerYAnchor].active = YES;
        [_button setTitle:@"Present" forState:UIControlStateNormal];
        [_button setTitleColor:[UIColor blueColor] forState:UIControlStateNormal];
    }
    return _button;
}

@end

```

ModalViewController.customTransitionDelegate ◦ ◦ modal ◦

ModalViewController

```

class ModalViewController: UIViewController
{
    lazy var button: UIButton =
    {
        let button = UIButton()
        button.translatesAutoresizingMaskIntoConstraints = false
        self.view.addSubview(button)
        button.centerXAnchor.constraint(equalTo: self.view.centerXAnchor).isActive = true
        button.centerYAnchor.constraint(equalTo: self.view.centerYAnchor).isActive = true
        button.setTitle("Dismiss", for: .normal)
        button.setTitleColor(.white, for: .normal)

        return button
    }()

    override func viewDidLoad()
    {
        super.viewDidLoad()
    }
}

```



```

        button.addTarget(self, action: #selector(self.didPressDismiss), for: .touchUpInside)
        view.backgroundColor = .red
        view.layer.cornerRadius = 15.0
    }

    func didPressDismiss()
    {
        dismiss(animated: true)
    }
}

extension ModalViewController: UIViewControllerTransitioningDelegate
{
    func animationController(forPresented presented: UIViewController, presenting:
UIViewController, source: UIViewController) -> UIViewControllerAnimatedTransitioning?
    {
        return DropOutAnimator(duration: 1.5, isAppearing: true)
    }

    func animationController(forDismissed dismissed: UIViewController) ->
UIViewControllerAnimatedTransitioning?
    {
        return DropOutAnimator(duration: 4.0, isAppearing: false)
    }
}

```

Objective-C

```

@interface ModalViewController () <UIViewControllerTransitioningDelegate>
@property (nonatomic, strong) UIButton *button;
@end

@implementation ModalViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    [self.button addTarget:self action:@selector(didPressPresent)
forControlEvents:UIControlEventTouchUpInside];
    self.view.backgroundColor = [UIColor redColor];
    self.view.layer.cornerRadius = 15.0f;
}

- (void)didPressPresent
{
    [self dismissViewControllerAnimated:YES completion:nil];
}

- (UIButton *)button
{
    if (!_button)
    {
        _button = [[UIButton alloc] init];
        _button.translatesAutoresizingMaskIntoConstraintsIntoConstraints = NO;
        [self.view addSubview:_button];
        [_button.centerXAnchor constraintEqualToAnchor:self.view.centerXAnchor].active = YES;
        [_button.centerYAnchor constraintEqualToAnchor:self.view.centerYAnchor].active = YES;
        [_button setTitle:@"Dismiss" forState:UIControlStateNormal];
        [_button setTitleColor:[UIColor blueColor] forState:UIControlStateNormal];
    }
}

```

```

    }
    return _button;
}

-
(id<UIViewControllerAnimatedTransitioning>)animationControllerForPresentedController:(UIViewController
*)presented presentingController:(UIViewController *)presenting
sourceController:(UIViewController *)source
{
    return [[DropOutAnimator alloc] initWithDuration: 1.5 appearing:YES];
}

-
(id<UIViewControllerAnimatedTransitioning>)animationControllerForDismissedController:(UIViewController
*)dismissed
{
    return [[DropOutAnimator alloc] initWithDuration:4.0 appearing:NO];
}

@end

```

◦ ModalViewControllerAnimatedTransitioningDelegate ◦ UIDynamicBehavior◦

◦ ◦ ◦ DropOutAnimatorViewControllerAnimatedTransitioning func animateTransition(using transitionContext: UIViewControllerContextTransitioning)func animateTransition(using transitionContext: UIViewControllerContextTransitioning) ◦

```

class DropOutAnimator: UIDynamicBehavior
{
    let duration: TimeInterval
    let isAppearing: Bool

    var transitionContext: UIViewControllerContextTransitioning?
    var hasElapsedTimeExceededDuration = false
    var finishTime: TimeInterval = 0.0
    var collisionBehavior: UICollisionBehavior?
    var attachmentBehavior: UIAttachmentBehavior?
    var animator: UIDynamicAnimator?

    init(duration: TimeInterval = 1.0, isAppearing: Bool)
    {
        self.duration = duration
        self.isAppearing = isAppearing
        super.init()
    }
}

extension DropOutAnimator: UIViewControllerAnimatedTransitioning
{
    func animateTransition(using transitionContext: UIViewControllerContextTransitioning)
    {
        // Get relevant views and view controllers from transitionContext
        guard let fromVC = transitionContext.viewController(forKey: .from),
            let toVC = transitionContext.viewController(forKey: .to),
            let fromView = fromVC.view,
            let toView = toVC.view else { return }

        let containerView = transitionContext.containerView
        let duration = self.transitionDuration(using: transitionContext)
    }
}

```

```

// Hold reference to transitionContext to notify it of completion
self.transitionContext = transitionContext

// Create dynamic animator
let animator = UIDynamicAnimator(referenceView: containerView)
animator.delegate = self
self.animator = animator

// Presenting Animation
if self.isAppearing
{
    fromView.isUserInteractionEnabled = false

    // Position toView just off-screen
    let fromViewInitialFrame = transitionContext.initialFrame(for: fromVC)
    var toViewInitialFrame = toView.frame
    toViewInitialFrame.origin.y -= toViewInitialFrame.height
    toViewInitialFrame.origin.x = fromViewInitialFrame.width * 0.5 -
toViewInitialFrame.width * 0.5
    toView.frame = toViewInitialFrame

    containerView.addSubview(toView)

    // Prevent rotation and adjust bounce
    let bodyBehavior = UIDynamicItemBehavior(items: [toView])
    bodyBehavior.elasticity = 0.7
    bodyBehavior.allowsRotation = false

    // Add gravity at exaggerated magnitude so animation doesn't seem slow
    let gravityBehavior = UIGravityBehavior(items: [toView])
    gravityBehavior.magnitude = 10.0

    // Set collision bounds to include off-screen view and have collision in center
    // where our final view should come to rest
    let collisionBehavior = UICollisionBehavior(items: [toView])
    let insets = UIEdgeInsets(top: toViewInitialFrame.minY, left: 0.0, bottom:
fromViewInitialFrame.height * 0.5 - toViewInitialFrame.height * 0.5, right: 0.0)
    collisionBehavior.setTranslatesReferenceBoundsIntoBoundary(with: insets)
    self.collisionBehavior = collisionBehavior

    // Keep track of finish time in case we need to end the animator before the
animator pauses
    self.finishTime = duration + (self.animator?.elapsedTime ?? 0.0)

    // Closure that is called after every "tick" of the animator
    // Check if we exceed duration
    self.action =
    { [weak self] in
        guard let strongSelf = self,
            (strongSelf.animator?.elapsedTime ?? 0.0) >= strongSelf.finishTime else {
return }

        strongSelf.hasElapsedTimeExceededDuration = true
        strongSelf.animator?.removeBehavior(strongSelf)
    }

    // `DropOutAnimator` is a composit behavior, so add child behaviors to self
    self.addChildBehavior(collisionBehavior)
    self.addChildBehavior(bodyBehavior)
    self.addChildBehavior(gravityBehavior)

```

```

        // Add self to dynamic animator
        self.animator?.addBehavior(self)
    }
    // Dismissing Animation
    else
    {
        // Create allow rotation and have a elastic item
        let bodyBehavior = UIDynamicItemBehavior(items: [fromView])
        bodyBehavior.elasticity = 0.8
        bodyBehavior.angularResistance = 5.0
        bodyBehavior.allowsRotation = true

        // Create gravity with exaggerated magnitude
        let gravityBehavior = UIGravityBehavior(items: [fromView])
        gravityBehavior.magnitude = 10.0

        // Collision boundary is set to have a floor just below the bottom of the screen
        let collisionBehavior = UICollisionBehavior(items: [fromView])
        let insets = UIEdgeInsets(top: 0.0, left: -1000, bottom: -225, right: -1000)
        collisionBehavior.setTranslatesReferenceBoundsIntoBoundary(with: insets)
        self.collisionBehavior = collisionBehavior

        // Attachment behavior so view will have effect of hanging from a rope
        let offset = UIOffset(horizontal: 70.0, vertical: fromView.bounds.height * 0.5)
        var anchorPoint = CGPoint(x: fromView.bounds.maxX - 40.0, y: fromView.bounds.minY)
        anchorPoint = containerView.convert(anchorPoint, from: fromView)
        let attachmentBehavior = UIAttachmentBehavior(item: fromView, offsetFromCenter:
offset, attachedToAnchor: anchorPoint)
        attachmentBehavior.frequency = 3.0
        attachmentBehavior.damping = 3.0
        self.attachmentBehavior = attachmentBehavior

        // `DropOutAnimator` is a composit behavior, so add child behaviors to self
        self.addChildBehavior(collisionBehavior)
        self.addChildBehavior(bodyBehavior)
        self.addChildBehavior(gravityBehavior)
        self.addChildBehavior(attachmentBehavior)

        // Add self to dynamic animator
        self.animator?.addBehavior(self)

        // Animation has two parts part one is hanging from rope.
        // Part two is bouncing off-screen
        // Divide duration in two
        self.finishTime = (2.0 / 3.0) * duration + (self.animator?.elapsedTime ?? 0.0)

        // After every "tick" of animator check if past time limit
        self.action =
        { [weak self] in
            guard let strongSelf = self,
                (strongSelf.animator?.elapsedTime ?? 0.0) >= strongSelf.finishTime else {
return }

            strongSelf.hasElapsedTimeExceededDuration = true
            strongSelf.animator?.removeBehavior(strongSelf)
        }
    }
}

func transitionDuration(using transitionContext: UIViewControllerContextTransitioning?) ->
TimeInterval

```

```

    {
        // Return the duration of the animation
        return self.duration
    }
}

extension DropOutAnimator: UIDynamicAnimatorDelegate
{
    func dynamicAnimatorDidPause(_ animator: UIDynamicAnimator)
    {
        // Animator has reached stasis
        if self.isAppearing
        {
            // Check if we are out of time
            if self.hasElapsedTimeExceededDuration
            {
                // Move to final positions
                let toView = self.transitionContext?.viewController(forKey: .to)?.view
                let containerView = self.transitionContext?.containerView
                toView?.center = containerView?.center ?? .zero
                self.hasElapsedTimeExceededDuration = false
            }

            // Clean up and call completion

self.transitionContext?.completeTransition(! (self.transitionContext?.transitionWasCancelled ??
false))

                self.childBehaviors.forEach { self.removeChildBehavior($0) }
                animator.removeAllBehaviors()
                self.transitionContext = nil
            }
        else
        {
            if let attachmentBehavior = self.attachmentBehavior
            {
                // If we have an attachment, we are at the end of part one and start part two.
                self.removeChildBehavior(attachmentBehavior)
                self.attachmentBehavior = nil
                animator.addBehavior(self)
                let duration = self.transitionDuration(using: self.transitionContext)
                self.finishTime = 1.0 / 3.0 * duration + animator.elapsedTime
            }
            else
            {
                // Clean up and call completion
                let fromView = self.transitionContext?.viewController(forKey: .from)?.view
                let toView = self.transitionContext?.viewController(forKey: .to)?.view
                fromView?.removeFromSuperview()
                toView?.isUserInteractionEnabled = true

self.transitionContext?.completeTransition(! (self.transitionContext?.transitionWasCancelled ??
false))

                self.childBehaviors.forEach { self.removeChildBehavior($0) }
                animator.removeAllBehaviors()
                self.transitionContext = nil
            }
        }
    }
}
}
}

```

Objective-C

```
@interface ObjcDropOutAnimator() <UIDynamicAnimatorDelegate,
UIViewControlllerAnimatedTransitioning>
@property (nonatomic, strong) id<UIViewControlllerContextTransitioning> transitionContext;
@property (nonatomic, strong) UIDynamicAnimator *animator;
@property (nonatomic, assign) NSTimeInterval finishTime;
@property (nonatomic, assign) BOOL elapsedTimeExceededDuration;
@property (nonatomic, assign, getter=isAppearing) BOOL appearing;
@property (nonatomic, assign) NSTimeInterval duration;
@property (nonatomic, strong) UIAttachmentBehavior *attachBehavior;
@property (nonatomic, strong) UICollisionBehavior * collisionBehavior;

@end

@implementation ObjcDropOutAnimator

- (instancetype) initWithDuration:(NSTimeInterval)duration appearing:(BOOL)appearing
{
    self = [super init];
    if (self)
    {
        _duration = duration;
        _appearing = appearing;
    }
    return self;
}

- (void) animateTransition:(id<UIViewControlllerContextTransitioning>)transitionContext
{
    // Get relevant views and view controllers from transitionContext
    UIViewController *fromVC = [transitionContext
viewControllerForKey:UITransitionContextFromViewControllerKey];
    UIViewController *toVC = [transitionContext
viewControllerForKey:UITransitionContextToViewControllerKey];
    UIView *fromView = fromVC.view;
    UIView *toView = toVC.view;

    UIView *containerView = transitionContext.containerView;
    NSTimeInterval duration = [self transitionDuration:transitionContext];

    // Hold refrence to transitionContext to notify it of completion
    self.transitionContext = transitionContext;

    // Create dynamic animator
    UIDynamicAnimator *animator = [[UIDynamicAnimator
alloc] initWithReferenceView:containerView];
    animator.delegate = self;
    self.animator = animator;

    // Presenting Animation
    if (self.isAppearing)
    {
        fromView.userInteractionEnabled = NO;

        // Position toView just above screen
        CGRect fromViewInitialFrame = [transitionContext
initialFrameForViewController:fromVC];
        CGRect toViewInitialFrame = toView.frame;
        toViewInitialFrame.origin.y -= CGRectGetHeight(toViewInitialFrame);
    }
}
```

```

        toViewInitialFrame.origin.x = CGRectGetWidth(fromViewInitialFrame) * 0.5 -
CGRectGetWidth(toViewInitialFrame) * 0.5;
        toView.frame = toViewInitialFrame;

        [containerView addSubview:toView];

        // Prevent rotation and adjust bounce
        UIDynamicItemBehavior *bodyBehavior = [[UIDynamicItemBehavior
alloc] initWithItems:@[toView]];
        bodyBehavior.elasticity = 0.7;
        bodyBehavior.allowsRotation = NO;

        // Add gravity at exaggerated magnitude so animation doesn't seem slow
        UIGravityBehavior *gravityBehavior = [[UIGravityBehavior
alloc] initWithItems:@[toView]];
        gravityBehavior.magnitude = 10.0f;

        // Set collision bounds to include off-screen view and have collision floor in center
        // where our final view should come to rest
        UICollisionBehavior *collisionBehavior = [[UICollisionBehavior
alloc] initWithItems:@[toView]];
        UIEdgeInsets insets = UIEdgeInsetsMake(CGRectGetMinY(toViewInitialFrame), 0.0,
CGRectGetHeight(fromViewInitialFrame) * 0.5 - CGRectGetHeight(toViewInitialFrame) * 0.5, 0.0);
        [collisionBehavior setTranslatesReferenceBoundsIntoBoundaryWithInsets:insets];
        self.collisionBehavior = collisionBehavior;

        // Keep track of finish time in case we need to end the animator before the animator
pauses
        self.finishTime = duration + self.animator.elapsedTime;

        // Closure that is called after every "tick" of the animator
        // Check if we exceed duration
        __weak ObjcDropOutAnimator *weakSelf = self;
        self.action = ^{
            __strong ObjcDropOutAnimator *strongSelf = weakSelf;
            if (strongSelf)
            {
                if (strongSelf.animator.elapsedTime >= strongSelf.finishTime)
                {
                    strongSelf.elapsedTimeExceededDuration = YES;
                    [strongSelf.animator removeBehavior:strongSelf];
                }
            }
        };

        // `DropOutAnimator` is a composit behavior, so add child behaviors to self
        [self addChildBehavior:collisionBehavior];
        [self addChildBehavior:bodyBehavior];
        [self addChildBehavior:gravityBehavior];

        // Add self to dynamic animator
        [self.animator addBehavior:self];
    }
    // Dismissing Animation
    else
    {
        // Allow rotation and have a elastic item
        UIDynamicItemBehavior *bodyBehavior = [[UIDynamicItemBehavior alloc]
initWithItems:@[fromView]];
        bodyBehavior.elasticity = 0.8;
        bodyBehavior.angularResistance = 5.0;

```

```

bodyBehavior.allowsRotation = YES;

// Create gravity with exaggerated magnitude
UIGravityBehavior *gravityBehavior = [[UIGravityBehavior alloc]
initWithItems:@[fromView]];
gravityBehavior.magnitude = 10.0f;

// Collision boundary is set to have a floor just below the bottom of the screen
UICollisionBehavior *collisionBehavior = [[UICollisionBehavior alloc]
initWithItems:@[fromView]];
UIEdgeInsets insets = UIEdgeInsetsMake(0, -1000, -225, -1000);
[collisionBehavior setTranslatesReferenceBoundsIntoBoundaryWithInsets:insets];
self.collisionBehavior = collisionBehavior;

// Attachment behavior so view will have effect of hanging from a rope
UIOffset offset = UIOffsetMake(70, -(CGRectGetHeight(fromView.bounds) / 2.0));

CGPoint anchorPoint = CGPointMake(CGRectGetMaxX(fromView.bounds) - 40,
                                   CGRectGetMinY(fromView.bounds));
anchorPoint = [containerView convertPoint:anchorPoint fromView:fromView];
UIAttachmentBehavior *attachBehavior = [[UIAttachmentBehavior alloc]
initWithItem:fromView offsetFromCenter:offset attachedToAnchor:anchorPoint];
attachBehavior.frequency = 3.0;
attachBehavior.damping = 0.3;
attachBehavior.length = 40;
self.attachBehavior = attachBehavior;

// `DropOutAnimator` is a composit behavior, so add child behaviors to self
[self addChildBehavior:collisionBehavior];
[self addChildBehavior:bodyBehavior];
[self addChildBehavior:gravityBehavior];
[self addChildBehavior:attachBehavior];

// Add self to dynamic animator
[self.Animator addBehavior:self];

// Animation has two parts part one is hanging from rope.
// Part two is bouncing off-screen
// Divide duration in two
self.finishTime = (2./3.) * duration + [self.Animator elapsedTime];

// After every "tick" of animator check if past time limit
__weak ObjcDropOutAnimator *weakSelf = self;
self.action = ^{
    __strong ObjcDropOutAnimator *strongSelf = weakSelf;
    if (strongSelf)
    {
        if ([strongSelf.Animator elapsedTime] >= strongSelf.finishTime)
        {
            strongSelf.elapsedTimeExceededDuration = YES;
            [strongSelf.Animator removeBehavior:strongSelf];
        }
    }
};
}
}
-
(NSTimeInterval)transitionDuration:(id<UIViewControllerContextTransitioning>)transitionContext
{
    return self.duration;
}

```



```

}

- (void)dynamicAnimatorDidPause:(UIDynamicAnimator *)animator
{
    // Animator has reached stasis
    if (self.isAppearing)
    {
        // Check if we are out of time
        if (self.elapsedTimeExceededDuration)
        {
            // Move to final positions
            UIView *toView = [self.transitionContext
viewControllerForKey:UITransitionContextToViewControllerKey].view;
            UIView *containerView = [self.transitionContext containerView];
            toView.center = containerView.center;
            self.elapsedTimeExceededDuration = NO;
        }

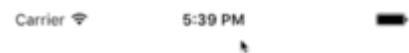
        // Clean up and call completion
        [self.transitionContext completeTransition:[self.transitionContext
transitionWasCancelled]];
        for (UIDynamicBehavior *behavior in self.childBehaviors)
        {
            [self removeChildBehavior:behavior];
        }
        [animator removeAllBehaviors];
        self.transitionContext = nil;
    }
    // Dismissing
    else
    {
        if (self.attachBehavior)
        {
            // If we have an attachment, we are at the end of part one and start part two.
            [self removeChildBehavior:self.attachBehavior];
            self.attachBehavior = nil;
            [animator addBehavior:self];
            NSTimeInterval duration = [self transitionDuration:self.transitionContext];
            self.finishTime = 1./3. * duration + [animator elapsedTime];
        }
        else
        {
            // Clean up and call completion
            UIView *fromView = [self.transitionContext
viewControllerForKey:UITransitionContextFromViewControllerKey].view;
            UIView *toView = [self.transitionContext
viewControllerForKey:UITransitionContextToViewControllerKey].view;
            [fromView removeFromSuperview];
            toView.userInteractionEnabled = YES;

            [self.transitionContext completeTransition:[self.transitionContext
transitionWasCancelled]];
            for (UIDynamicBehavior *behavior in self.childBehaviors)
            {
                [self removeChildBehavior:behavior];
            }
            [animator removeAllBehaviors];
            self.transitionContext = nil;
        }
    }
}

```

UIDynamicBehaviors

iOS“”。



Swipe Down From Top

◦ UINavigationControllerTransitioningDelegate ◦ ◦ ◦ ◦

```
class ViewController: UIViewController
{
    var presentingAnimator: ShadeAnimator!
    var dismissingAnimator: ShadeAnimator!
    let shadeVC = ShadeViewController()

    lazy var label: UILabel =
    {
        let label = UILabel()
        label.textColor = .blue
        label.translatesAutoresizingMaskIntoConstraints = false
        self.view.addSubview(label)
        label.centerXAnchor.constraint(equalTo: self.view.centerXAnchor).isActive = true
        label.centerYAnchor.constraint(equalTo: self.view.centerYAnchor).isActive = true
        return label
    }()

    override func viewDidLoad()
    {
        super.viewDidLoad()
        label.text = "Swipe Down From Top"
        presentingAnimator = ShadeAnimator(isAppearing: true, presentingVC: self, presentedVC:
        shadeVC, transitionDelegate: self)
        dismissingAnimator = ShadeAnimator(isAppearing: false, presentingVC: self,
        presentedVC: shadeVC, transitionDelegate: self)
    }
}
```

```

    }
}
extension ViewController: UIViewControllerTransitioningDelegate
{
    func animationController(forPresented presented: UIViewController, presenting:
UIViewController, source: UIViewController) -> UIViewControllerAnimatedTransitioning?
    {
        return EmptyAnimator()
    }

    func animationController(forDismissed dismissed: UIViewController) ->
UIViewControllerAnimatedTransitioning?
    {
        return EmptyAnimator()
    }

    func interactionControllerForPresentation(using animator:
UIViewControllerAnimatedTransitioning) -> UIViewControllerInteractiveTransitioning?
    {
        return presentingAnimator
    }

    func interactionControllerForDismissal(using animator:
UIViewControllerAnimatedTransitioning) -> UIViewControllerInteractiveTransitioning?
    {
        return dismissingAnimator
    }
}
}

```

Objective-C

```

@interface ObjCViewController () <UIViewControllerTransitioningDelegate>
@property (nonatomic, strong) ShadeAnimator *presentingAnimator;
@property (nonatomic, strong) ShadeAnimator *dismissingAnimator;
@property (nonatomic, strong) UILabel *label;
@property (nonatomic, strong) ShadeViewController *shadeVC;
@end

@implementation ObjCViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    self.label.text = @"Swipe Down From Top";
    self.shadeVC = [[ShadeViewController alloc] init];
    self.presentingAnimator = [[ShadeAnimator alloc] initWithIsAppearing:YES presentingVC:self
presentedVC:self.shadeVC transitionDelegate:self];
    self.dismissingAnimator = [[ShadeAnimator alloc] initWithIsAppearing:NO presentingVC:self
presentedVC:self.shadeVC transitionDelegate:self];
}

- (UILabel *)label
{
    if (!_label)
    {
        _label = [[UILabel alloc] init];
        _label.textColor = [UIColor blueColor];
        _label.translatesAutoresizingMaskIntoConstraints = NO;
        [self.view addSubview:_label];
    }
}

```

```

        [_label.centerXAnchor constraintEqualToAnchor:self.view.centerXAnchor].active = YES;
        [_label.centerYAnchor constraintEqualToAnchor:self.view.centerYAnchor].active = YES;
    }
    return _label;
}

#pragma mark - UINavigationControllerTransitioningDelegate

-
(id<UINavigationControllerAnimatedTransitioning>)animationControllerForPresentedController:(UINavigationController
*)presented presentingController:(UINavigationController *)presenting
sourceController:(UINavigationController *)source
{
    return [[EmptyAnimator alloc] init];
}

-
(id<UINavigationControllerAnimatedTransitioning>)animationControllerForDismissedController:(UINavigationController
*)dismissed
{
    return [[EmptyAnimator alloc] init];
}

-
(id<UINavigationControllerInteractiveTransitioning>)interactionControllerForPresentation:(id<UINavigationController
*)presentation
{
    return self.presentingAnimator;
}

-
(id<UINavigationControllerInteractiveTransitioning>)interactionControllerForDismissal:(id<UINavigationController
*)dismissal
{
    return self.dismissingAnimator;
}

@end

```

UINavigationControllerTransitioningDelegate° UINavigationControllerAnimatedTransitioningEmptyAnimator°

```

class EmptyAnimator: NSObject
{
}

extension EmptyAnimator: UINavigationControllerAnimatedTransitioning
{
    func animateTransition(using transitionContext: UINavigationControllerContextTransitioning)
    {
    }

    func transitionDuration(using transitionContext: UINavigationControllerContextTransitioning?) ->
TimeInterval
    {
        return 0.0
    }
}

```

Objective-C

```
@implementation EmptyAnimator

- (void)animateTransition:(id<UIViewControllerContextTransitioning>)transitionContext
{
}

-
(NSTimeInterval)transitionDuration:(id<UIViewControllerContextTransitioning>)transitionContext
{
    return 0.0;
}

@end
```

ShadeAnimatorUIDynamicBehaviorUIViewControllerInteractiveTransitioning ◦

```
class ShadeAnimator: UIDynamicBehavior
{
    // Whether we are presenting or dismissing
    let isAppearing: Bool

    // The view controller that is not the shade
    weak var presentingVC: UIViewController?

    // The view controller that is the shade
    weak var presentedVC: UIViewController?

    // The delegate will vend the animator
    weak var transitionDelegate: UIViewControllerTransitioningDelegate?

    // Feedback generator for haptics on collisions
    let impactFeedbackGenerator = UIImpactFeedbackGenerator(style: .light)

    // The context given to the animator at the start of the transition
    var transitionContext: UIViewControllerContextTransitioning?

    // Time limit of the dynamic part of the animation
    var finishTime: TimeInterval = 4.0

    // The Pan Gesture that drives the transition. Not using EdgePan because triggers
    Notifications screen
    lazy var pan: UIPanGestureRecognizer =
    {
        let pan = UIPanGestureRecognizer(target: self, action:
#selector(self.handlePan(sender:)))
        return pan
    }()

    // The dynamic animator that we add `ShadeAnimator` to
    lazy var animator: UIDynamicAnimator! =
    {
        let animator = UIDynamicAnimator(referenceView: self.transitionContext!.containerView)
        return animator
    }()
```

```

// init with all of our dependencies
init(isAppearing: Bool, presentingVC: UIViewController, presentedVC: UIViewController,
transitionDelegate: UIViewControllerTransitioningDelegate)
{
    self.isAppearing = isAppearing
    self.presentingVC = presentingVC
    self.presentedVC = presentedVC
    self.transitionDelegate = transitionDelegate
    super.init()
    self.impactFeedbackGenerator.prepare()

    if isAppearing
    {
        self.presentingVC?.view.addGestureRecognizer(pan)
    }
    else
    {
        self.presentedVC?.view.addGestureRecognizer(pan)
    }
}

// Setup and moves shade view controller to just above screen if appearing
func setupViewsForTransition(with transitionContext: UIViewControllerContextTransitioning)
{
    // Get relevant views and view controllers from transitionContext
    guard let fromVC = transitionContext.viewController(forKey: .from),
        let toVC = transitionContext.viewController(forKey: .to),
        let toView = toVC.view else { return }

    let containerView = transitionContext.containerView

    // Hold reference to transitionContext to notify it of completion
    self.transitionContext = transitionContext
    if isAppearing
    {
        // Position toView just off-screen
        let fromViewInitialFrame = transitionContext.initialFrame(for: fromVC)
        var toViewInitialFrame = toView.frame
        toViewInitialFrame.origin.y -= toViewInitialFrame.height
        toViewInitialFrame.origin.x = fromViewInitialFrame.width * 0.5 -
toViewInitialFrame.width * 0.5
        toView.frame = toViewInitialFrame

        containerView.addSubview(toView)
    }
    else
    {
        fromVC.view.addGestureRecognizer(pan)
    }
}

// Handles the entire interaction from presenting/dismissing to completion
func handlePan(sender: UIPanGestureRecognizer)
{
    let location = sender.location(in: transitionContext?.containerView)
    let velocity = sender.velocity(in: transitionContext?.containerView)
    let fromVC = transitionContext?.viewController(forKey: .from)
    let toVC = transitionContext?.viewController(forKey: .to)

    let touchStartHeight: CGFloat = 90.0

```

```

let touchLocationFromBottom: CGFloat = 20.0

switch sender.state
{
case .began:
    let beginLocation = sender.location(in: sender.view)
    if isAppearing
    {
        guard beginLocation.y <= touchStartHeight,
            let presentedVC = self.presentedVC else { break }
        presentedVC.modalPresentationStyle = .custom
        presentedVC.transitioningDelegate = transitionDelegate
        presentingVC?.present(presentedVC, animated: true)
    }
    else
    {
        guard beginLocation.y >= (sender.view?.frame.height ?? 0.0) - touchStartHeight
else { break }
        presentedVC?.dismiss(animated: true)
    }
case .changed:
    guard let view = isAppearing ? toVC?.view : fromVC?.view else { return }
    UIView.animate(withDuration: 0.2)
    {
        view.frame.origin.y = location.y - view.bounds.height +
touchLocationFromBottom
    }

    transitionContext?.updateInteractiveTransition(view.frame.maxY / view.frame.height
)
case .ended, .cancelled:
    guard let view = isAppearing ? toVC?.view : fromVC?.view else { return }
    let isCancelled = isAppearing ? (velocity.y < 0.5 || view.center.y < 0.0) :
(velocity.y > 0.5 || view.center.y > 0.0)
    addAttachmentBehavior(with: view, isCancelled: isCancelled)
    addCollisionBehavior(with: view)
    addItemBehavior(with: view)

    animator.addBehavior(self)
    animator.delegate = self

    self.action =
    { [weak self] in
        guard let strongSelf = self else { return }
        if strongSelf.animator.elapsedTime > strongSelf.finishTime
        {
            strongSelf.animator.removeAllBehaviors()
        }
        else
        {
            strongSelf.transitionContext?.updateInteractiveTransition(view.frame.maxY
/ view.frame.height
)
        }
    }
    default:
        break
}
}

// Add collision behavior that causes bounce when finished

```

```

func addCollisionBehavior(with view: UIView)
{
    let collisionBehavior = UICollisionBehavior(items: [view])
    let insets = UIEdgeInsets(top: -view.bounds.height, left: 0.0, bottom: 0.0, right:
0.0)
    collisionBehavior.setTranslatesReferenceBoundsIntoBoundary(with: insets)
    collisionBehavior.collisionDelegate = self
    self.addChildBehavior(collisionBehavior)
}

// Add attachment behavior that pulls shade either to top or bottom
func addAttachmentBehavior(with view: UIView, isCancelled: Bool)
{
    let anchor: CGPoint
    switch (isAppearing, isCancelled)
    {
    case (true, true), (false, false):
        anchor = CGPoint(x: view.center.x, y: -view.frame.height)
    case (true, false), (false, true):
        anchor = CGPoint(x: view.center.x, y: view.frame.height)
    }
    let attachmentBehavior = UIAttachmentBehavior(item: view, attachedToAnchor: anchor)
    attachmentBehavior.damping = 0.1
    attachmentBehavior.frequency = 3.0
    attachmentBehavior.length = 0.5 * view.frame.height
    self.addChildBehavior(attachmentBehavior)
}

// Makes view more bouncy
func addItemBehavior(with view: UIView)
{
    let itemBehavior = UIDynamicItemBehavior(items: [view])
    itemBehavior.allowsRotation = false
    itemBehavior.elasticity = 0.6
    self.addChildBehavior(itemBehavior)
}
}
extension ShadeAnimator: UIDynamicAnimatorDelegate
{
    // Determines transition has ended
    func dynamicAnimatorDidPause(_ animator: UIDynamicAnimator)
    {
        guard let transitionContext = self.transitionContext else { return }
        let fromVC = transitionContext.viewController(forKey: .from)
        let toVC = transitionContext.viewController(forKey: .to)
        guard let view = isAppearing ? toVC?.view : fromVC?.view else { return }
        switch (view.center.y < 0.0, isAppearing)
        {
        case (true, true), (true, false):
            view.removeFromSuperview()
            transitionContext.finishInteractiveTransition()
            transitionContext.completeTransition(!isAppearing)
        case (false, true):
            toVC?.view.frame = transitionContext.finalFrame(for: toVC!)
            transitionContext.finishInteractiveTransition()
            transitionContext.completeTransition(true)
        case (false, false):
            fromVC?.view.frame = transitionContext.initialFrame(for: fromVC!)
            transitionContext.cancelInteractiveTransition()
            transitionContext.completeTransition(false)
        }
    }
}

```



```

    }
    childBehaviors.forEach { removeChildBehavior($0) }
    animator.removeAllBehaviors()
    self.animator = nil
    self.transitionContext = nil
}
}
extension ShadeAnimator: UICollisionBehaviorDelegate
{
    // Triggers haptics
    func collisionBehavior(_ behavior: UICollisionBehavior, beganContactFor item:
UIDynamicItem, withBoundaryIdentifier identifier: NSCopying?, at p: CGPoint)
    {
        guard p.y > 0.0 else { return }
        impactFeedbackGenerator.impactOccurred()
    }
}
extension ShadeAnimator: UIViewControllerInteractiveTransitioning
{
    // Starts transition
    func startInteractiveTransition(_ transitionContext: UIViewControllerContextTransitioning)
    {
        setupViewsForTransition(with: transitionContext)
    }
}
}

```

Objective-C

```

@interface ShadeAnimator() <UIDynamicAnimatorDelegate, UICollisionBehaviorDelegate>
@property (nonatomic, assign) BOOL isAppearing;
@property (nonatomic, weak) UIViewController *presentingVC;
@property (nonatomic, weak) UIViewController *presentedVC;
@property (nonatomic, weak) NSObject<UIViewControllerTransitioningDelegate>
*transitionDelegate;
@property (nonatomic, strong) UIImpactFeedbackGenerator *impactFeedbackGenerator;
@property (nonatomic, strong) id<UIViewControllerContextTransitioning> transitionContext;
@property (nonatomic, assign) NSTimeInterval finishTime;
@property (nonatomic, strong) UIPanGestureRecognizer *pan;
@property (nonatomic, strong) UIDynamicAnimator *animator;
@end

@implementation ShadeAnimator

- (instancetype) initWithIsAppearing: (BOOL) isAppearing presentingVC: (UIViewController
*) presentingVC presentedVC: (UIViewController *) presentedVC
transitionDelegate: (id<UIViewControllerTransitioningDelegate>) transitionDelegate
{
    self = [super init];
    if (self)
    {
        _isAppearing = isAppearing;
        _presentingVC = presentingVC;
        _presentedVC = presentedVC;
        _transitionDelegate = transitionDelegate;
        _impactFeedbackGenerator = [[UIImpactFeedbackGenerator
alloc] initWithStyle: UIImpactFeedbackStyleLight];
        [_impactFeedbackGenerator prepare];
        if (_isAppearing)
        {

```

```

        [_presentingVC.view addGestureRecognizer:self.pan];
    }
    else
    {
        [_presentedVC.view addGestureRecognizer:self.pan];
    }
}
return self;
}

#pragma mark - Lazy Init
- (UIPanGestureRecognizer *)pan
{
    if (!_pan)
    {
        _pan = [[UIPanGestureRecognizer alloc] initWithTarget:self
action:@selector(handlePan:)];
    }
    return _pan;
}

- (UIDynamicAnimator *)animator
{
    if (!_animator)
    {
        _animator = [[UIDynamicAnimator
alloc] initWithReferenceView:self.transitionContext.containerView];
    }
    return _animator;
}

#pragma mark - Setup
-
(void)setupViewForTransitionWithContext:(id<UIViewControllerContextTransitioning>)transitionContext
{
    UIViewController *fromVC = [transitionContext
viewControllerForKey:UITransitionContextFromViewControllerKey];
    UIViewController *toVC = [transitionContext
viewControllerForKey:UITransitionContextToViewControllerKey];
    UIView *toView = toVC.view;
    UIView *containerView = transitionContext.containerView;
    self.transitionContext = transitionContext;
    if (self.isAppearing)
    {
        CGRect fromViewInitialFrame = [transitionContext
initialFrameForViewController:fromVC];
        CGRect toViewInitialFrame = toView.frame;
        toViewInitialFrame.origin.y -= CGRectGetHeight(toViewInitialFrame);
        toViewInitialFrame.origin.x = CGRectGetWidth(fromViewInitialFrame) * 0.5 -
CGRectGetWidth(toViewInitialFrame) * 0.5;

        [containerView addSubview:toView];
    }
    else
    {
        [fromVC.view addGestureRecognizer:self.pan];
    }
}

#pragma mark - Gesture

```

```

- (void)handlePan:(UIPanGestureRecognizer *)sender
{
    CGPoint location = [sender locationInView:self.transitionContext.containerView];
    CGPoint velocity = [sender velocityInView:self.transitionContext.containerView];
    UIViewController *fromVC = [self.transitionContext
viewControllerForKey:UITransitionContextFromViewControllerKey];
    UIViewController *toVC = [self.transitionContext
viewControllerForKey:UITransitionContextToViewControllerKey];

    CGFloat touchStartHeight = 90.0;
    CGFloat touchLocationFromBottom = 20.0;

    if (sender.state == UIGestureRecognizerStateBegan)
    {
        CGPoint beginLocation = [sender locationInView:sender.view];
        if (self.isAppearing)
        {
            if (beginLocation.y <= touchStartHeight)
            {
                self.presentedVC.modalPresentationStyle = UIModalPresentationCustom;
                self.presentedVC.transitioningDelegate = self.transitionDelegate;
                [self.presentingVC presentViewController:self.presentedVC animated:YES
completion:nil];
            }
        }
        else
        {
            if (beginLocation.y >= [sender locationInView:sender.view].y - touchStartHeight)
            {
                [self.presentedVC dismissViewControllerAnimated:true completion:nil];
            }
        }
    }
    else if (sender.state == UIGestureRecognizerStateChanged)
    {
        UIView *view = self.isAppearing ? toVC.view : fromVC.view;
        [UIView animateWithDuration:0.2 animations:^(
            CGRect frame = view.frame;
            frame.origin.y = location.y - CGRectGetHeight(view.bounds) +
touchLocationFromBottom;
            view.frame = frame;
        )];
        [self.transitionContext updateInteractiveTransition:CGRectGetMaxY(view.frame) /
CGRectGetHeight(view.frame)];
    }
    else if (sender.state == UIGestureRecognizerStateEnded || sender.state ==
UIGestureRecognizerStateCancelled)
    {
        UIView *view = self.isAppearing ? toVC.view : fromVC.view;
        BOOL isCancelled = self.isAppearing ? (velocity.y < 0.5 || view.center.y < 0.0) :
(velocity.y > 0.5 || view.center.y > 0.0);
        [self addAttachmentBehaviorWithView:view isCancelled:isCancelled];
        [self addCollisionBehaviorWithView:view];
        [self addItemBehaviorWithView:view];

        [self.animator addBehavior:self];
        self.animator.delegate = self;

        __weak ShadeAnimator *weakSelf = self;
        self.action =
        ^{

```

```

        if (weakSelf.Animator.elapsedTime > weakSelf.finishTime)
        {
            [weakSelf.Animator removeAllBehaviors];
        }
        else
        {
            [weakSelf.transitionContext
updateInteractiveTransition:CGRectGetMaxY(view.frame) / CGRectGetHeight(view.frame)];
        }
    };
}

#pragma mark - UIViewControllerInteractiveTransitioning
- (void)startInteractiveTransition:(id<UIViewControllerContextTransitioning>)transitionContext
{
    [self setupViewForTransitionWithContext:transitionContext];
}

#pragma mark - Behaviors
- (void)addCollisionBehaviorWithView:(UIView *)view
{
    UICollisionBehavior *collisionBehavior = [[UICollisionBehavior
alloc] initWithItems:@[view]];
    UIEdgeInsets insets = UIEdgeInsetsMake(-CGRectGetHeight(view.bounds), 0.0, 0.0, 0.0);
    [collisionBehavior setTranslatesReferenceBoundsIntoBoundaryWithInsets:insets];
    collisionBehavior.collisionDelegate = self;
    [self addChildBehavior:collisionBehavior];
}

- (void)addItemBehaviorWithView:(UIView *)view
{
    UIDynamicItemBehavior *itemBehavior = [[UIDynamicItemBehavior
alloc] initWithItems:@[view]];
    itemBehavior.allowsRotation = NO;
    itemBehavior.elasticity = 0.6;
    [self addChildBehavior:itemBehavior];
}

- (void)addAttachmentBehaviorWithView:(UIView *)view isCancelled:(BOOL)isCancelled
{
    CGPoint anchor;
    if ((self.isAppearing && isCancelled) || (!self.isAppearing && isCancelled))
    {
        anchor = CGPointMake(view.center.x, -CGRectGetHeight(view.frame));
    }
    else
    {
        anchor = CGPointMake(view.center.x, -CGRectGetHeight(view.frame));
    }
    UIAttachmentBehavior *attachmentBehavior = [[UIAttachmentBehavior alloc] initWithItem:view
attachedToAnchor:anchor];
    attachmentBehavior.damping = 0.1;
    attachmentBehavior.frequency = 3.0;
    attachmentBehavior.length = 0.5 * CGRectGetHeight(view.frame);
    [self addChildBehavior:attachmentBehavior];
}

#pragma mark - UICollisionBehaviorDelegate
- (void)collisionBehavior:(UICollisionBehavior *)behavior
beganContactForItem:(id<UIDynamicItem>)item withBoundaryIdentifier:(id<NSCopying>)identifier

```

```

atPoint:(CGPoint)p
{
    if (p.y > 0.0)
    {
        [self.impactFeedbackGenerator impactOccurred];
    }
}

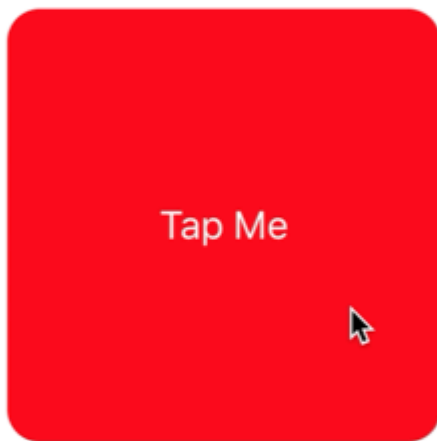
#pragma mark - UIDynamicAnimatorDelegate
- (void)dynamicAnimatorDidPause:(UIDynamicAnimator *)animator
{
    UIViewController *fromVC = [self.transitionContext
viewControllerForKey:UITransitionContextFromViewControllerKey];
    UIViewController *toVC = [self.transitionContext
viewControllerForKey:UITransitionContextToViewControllerKey];
    UIView *view = self.isAppearing ? toVC.view : fromVC.view;
    if (view.center.y < 0.0 && (self.isAppearing || !self.isAppearing))
    {
        [view removeFromSuperview];
        [self.transitionContext finishInteractiveTransition];
        [self.transitionContext completeTransition:!self.isAppearing];
    }
    else if (view.center.y >= 0.0 && self.isAppearing)
    {
        toVC.view.frame = [self.transitionContext finalFrameForViewController:toVC];
        [self.transitionContext finishInteractiveTransition];
        [self.transitionContext completeTransition:YES];
    }
    else
    {
        fromVC.view.frame = [self.transitionContext initialFrameForViewController:fromVC];
        [self.transitionContext cancelInteractiveTransition];
        [self.transitionContext completeTransition:NO];
    }
    for (UIDynamicBehavior *behavior in self.childBehaviors)
    {
        [self removeChildBehavior:behavior];
    }
    [animator removeAllBehaviors];
    self.animator = nil;
    self.transitionContext = nil;
}

@end

```

◦ ◦ UIDynamicBehaviors ◦ [2013 WWDC“UIKit Dynamics”](#) ◦

UIDynamicItemUIButton ◦



UIDynamicItembounds°

```
protocol ResizableDynamicItem: UIDynamicItem
{
    var bounds: CGRect { set get }
}
extension UIView: ResizableDynamicItem {}
```

Objective-C

```
@protocol ResizableDynamicItem <UIDynamicItem>
@property (nonatomic, readwrite) CGRect bounds;
```

```
@end
```

UIDynamicItem° boundstransform° **xy**°

```
final class PositionToBoundsMapping: NSObject, UIDynamicItem
{
    var target: ResizableDynamicItem

    init(target: ResizableDynamicItem)
    {
        self.target = target
        super.init()
    }

    var bounds: CGRect
    {
        get
        {
            return self.target.bounds
        }
    }

    var center: CGPoint
    {
        get
        {
            return CGPoint(x: self.target.bounds.width, y: self.target.bounds.height)
        }

        set
        {
            self.target.bounds = CGRect(x: 0.0, y: 0.0, width: newValue.x, height: newValue.y)
        }
    }

    var transform: CGAffineTransform
    {
        get
        {
            return self.target.transform
        }

        set
        {
            self.target.transform = newValue
        }
    }
}
```

Objective-C

```
@interface PositionToBoundsMapping ()
@property (nonatomic, strong) id<ResizableDynamicItem> target;
@end

@implementation PositionToBoundsMapping
```

```

- (instancetype)initWithTarget:(id<ResizableDynamicItem>)target
{
    self = [super init];
    if (self)
    {
        _target = target;
    }
    return self;
}

- (CGRect)bounds
{
    return self.target.bounds;
}

- (CGPoint)center
{
    return CGPointMake(self.target.bounds.size.width, self.target.bounds.size.height);
}

- (void)setCenter:(CGPoint)center
{
    self.target.bounds = CGRectMake(0, 0, center.x, center.y);
}

- (CGAffineTransform)transform
{
    return self.target.transform;
}

- (void)setTransform:(CGAffineTransform)transform
{
    self.target.transform = transform;
}

@end

```

UIViewController ◦ PositionToBoundsMapping ◦ UIAttachmentBehavior UIPushBehavior ◦ ◦

```

final class ViewController: UIViewController
{
    lazy var button: UIButton =
    {
        let button = UIButton(frame: CGRect(x: 0.0, y: 0.0, width: 300.0, height: 200.0))
        button.backgroundColor = .red
        button.layer.cornerRadius = 15.0
        button.setTitle("Tap Me", for: .normal)
        self.view.addSubview(button)
        return button
    }()

    var buttonBounds = CGRect.zero
    var animator: UIDynamicAnimator?

    override func viewDidLoad()
    {
        super.viewDidLoad()
        view.backgroundColor = .white
        button.addTarget(self, action: #selector(self.didPressButton(sender:)), for:
        .touchUpInside)
    }
}

```



```

        buttonBounds = button.bounds
    }

    override func viewDidLoadSubviews()
    {
        super.viewDidLoadSubviews()
        button.center = view.center
    }

    func didPressButton(sender: UIButton)
    {
        // Reset bounds so if button is press twice in a row, previous changes don't propogate
        button.bounds = buttonBounds
        let animator = UIDynamicAnimator(referenceView: view)

        // Create mapping
        let buttonBoundsDynamicItem = PositionToBoundsMapping(target: button)

        // Add Attachment behavior
        let attachmentBehavior = UIAttachmentBehavior(item: buttonBoundsDynamicItem,
        attachedToAnchor: buttonBoundsDynamicItem.center)

        // Higher frequency faster oscillation
        attachmentBehavior.frequency = 2.0

        // Lower damping longer oscillation lasts
        attachmentBehavior.damping = 0.1
        animator.addBehavior(attachmentBehavior)

        let pushBehavior = UIPushBehavior(items: [buttonBoundsDynamicItem], mode:
        .instantaneous)

        // Change angle to determine how much height/ width should change 45° means
        heigh:width is 1:1
        pushBehavior.angle = .pi / 4.0

        // Larger magnitude means bigger change
        pushBehavior.magnitude = 30.0
        animator.addBehavior(pushBehavior)
        pushBehavior.active = true

        // Hold refrence so animator is not released
        self.animator = animator
    }
}

```

Objective-C

```

@interface ViewController ()
@property (nonatomic, strong) UIButton *button;
@property (nonatomic, assign) CGRect buttonBounds;
@property (nonatomic, strong) UIDynamicAnimator *animator;
@end

@implementation ViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
}

```

```

    self.view.backgroundColor = [UIColor whiteColor];
    [self.button addTarget:self action:@selector(didTapButton:)
forControlEvents:UIControlEventTouchUpInside];
    self.buttonBounds = self.button.bounds;
}

- (void)viewDidLoadSubviews
{
    [super viewDidLoadSubviews];
    self.button.center = self.view.center;
}

- (UIButton *)button
{
    if (!_button)
    {
        _button = [[UIButton alloc] initWithFrame:CGRectMake(0.0, 0.0, 200.0, 200.0)];
        _button.backgroundColor = [UIColor redColor];
        _button.layer.cornerRadius = 15.0;
        [_button setTitle:@"Tap Me" forState:UIControlStateNormal];
        [self.view addSubview:_button];
    }
    return _button;
}

- (void)didTapButton:(id) sender
{
    self.button.bounds = self.buttonBounds;
    UIDynamicAnimator *animator = [[UIDynamicAnimator alloc] initWithReferenceView:self.view];
    PositionToBoundsMapping *buttonBoundsDynamicItem = [[PositionToBoundsMapping
alloc] initWithTarget:sender];
    UIAttachmentBehavior *attachmentBehavior = [[UIAttachmentBehavior
alloc] initWithItem:buttonBoundsDynamicItem attachedToAnchor:buttonBoundsDynamicItem.center];
    [attachmentBehavior setFrequency:2.0];
    [attachmentBehavior setDamping:0.3];
    [animator addBehavior:attachmentBehavior];

    UIPushBehavior *pushBehavior = [[UIPushBehavior alloc]
initWithItems:@[buttonBoundsDynamicItem] mode:UIPushBehaviorModeInstantaneous];
    pushBehavior.angle = M_PI_4;
    pushBehavior.magnitude = 2.0;
    [animator addBehavior:pushBehavior];

    [pushBehavior setActive:YES];

    self.animator = animator;
}

@end

```

UIKit Dynamics Catalog

UIKit <https://riptutorial.com/zh-TW/ios/topic/9479/uikit>

79: UILabelattributedString

◦

attributedString UILabel UILabelHTML

Examples

UILabelHTML

```
NSString * htmlString = @"<html><body> <b> Example bold text in HTML </b> </body></html>";
NSAttributedString * attrStr = [[NSAttributedString alloc] initWithData:[htmlString
dataUsingEncoding:NSUTF8StringEncoding] options:@{ NSDocumentTypeDocumentAttribute:
NSHTMLTextDocumentType } documentAttributes:nil error:nil];

UILabel * yourLabel = [[UILabel alloc] init];
yourLabel.attributedString = attrStr;
```

UILabel

NSMutableAttributedString ◦ NSMutableAttributedString NSAttributedString ◦

```
NSString *fullStr = @"Hello World!";
NSMutableAttributedString *attString = [[NSMutableAttributedString
alloc] initWithString:fullStr];

// Finding the range of text.
NSRange rangeHello = [fullStr rangeOfString:@"Hello"];
NSRange rangeWorld = [fullStr rangeOfString:@"World!"];

// Add font style for Hello
[attString addAttribute: NSFontAttributeName
                 value: [UIFont fontWithName:@"Copperplate" size:14]
                 range: rangeHello];
// Add text color for Hello
[attString addAttribute: NSForegroundColorAttributeName
                 value: [UIColor blueColor]
                 range: rangeHello];

// Add font style for World!
[attString addAttribute: NSFontAttributeName
                 value: [UIFont fontWithName:@"Chalkduster" size:20]
                 range: rangeWorld];
// Add text color for World!
[attString addAttribute: NSForegroundColorAttributeName
                 value: [UIColor colorWithRed:(66.0/255.0) green:(244.0/255.0)
blue:(197.0/255.0) alpha:1]
                 range: rangeWorld];

// Set it to UILabel as attributedText
UILabel * yourLabel = [[UILabel alloc] initWithFrame:CGRectMake(10, 150, 200, 100)];
yourLabel.attributedString = attString;
[self.view addSubview:yourLabel];
```

HELLO World!

UILabelattributedString <https://riptutorial.com/zh-TW/ios/topic/10927/UILabelattributedString>

80: UILabel

Examples

Objective UILabel

```
UILabel *label=[[UILabel alloc]initWithFrame:CGRectMake(0, 0, 320, 480)];
label.backgroundColor=[UIColor lightGrayColor];
NSMutableAttributedString *attributedString;
attributedString = [[NSMutableAttributedString alloc] initWithString:@"Apply Underlining"];
[attributedString addAttribute:NSUnderlineStyleAttributeName value:@1 range:NSMakeRange(0,
[attributedString length])];
[label setAttributedText:attributedString];
```

SwiftUILabel

```
let label = UILabel.init(frame: CGRect(x: 0, y:0, width: 100, height: 40))
label.backgroundColor = .lightGray
let attributedString = NSMutableAttributedString.init(string: "Apply UnderLining")
attributedString.addAttribute(NSUnderlineStyleAttributeName, value: 1, range:
NSRange.init(location: 0, length: attributedString.length))
label.attributedText = attributedString
```

UILabel <https://riptutorial.com/zh-TW/ios/topic/7219/ui-label>

81: UILocalNotification

-
- ◦

UILocalNotificationiOS 10◦ UserNotifications◦

UILocalNotification◦ UILocalNotification◦

- [UILocalNotification](#)
- [Stack OverflowUILocalNotification](#)

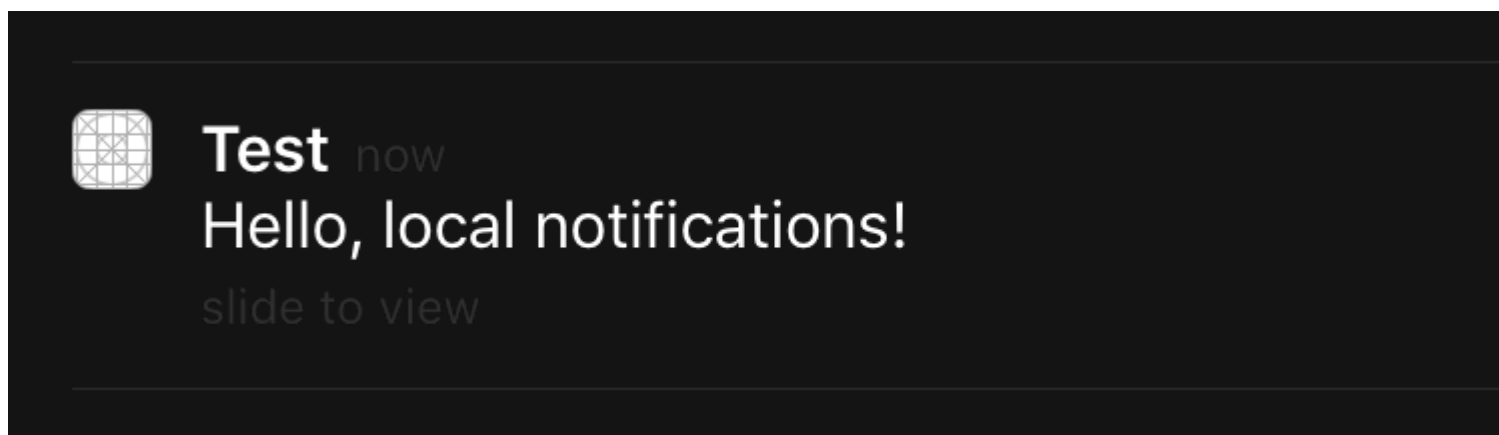
Examples

```
let notification = UILocalNotification()
notification.alertBody = "Hello, local notifications!"
notification.fireDate = NSDate().dateByAddingTimeInterval(10) // 10 seconds after now
UIApplication.sharedApplication().scheduleLocalNotification(notification)
```

Objective-C

```
UILocalNotification *notification = [[UILocalNotification alloc] init];
notification.alertBody = @"Hello, local notifications!";
notification.fireDate = [NSDate dateWithTimeIntervalSinceNow:10]; // 10 seconds after now
[[UIApplication sharedApplication] scheduleLocalNotification:notification];
```

iOS [^]⌘H control-command-H _{⌘L} command-L. ""



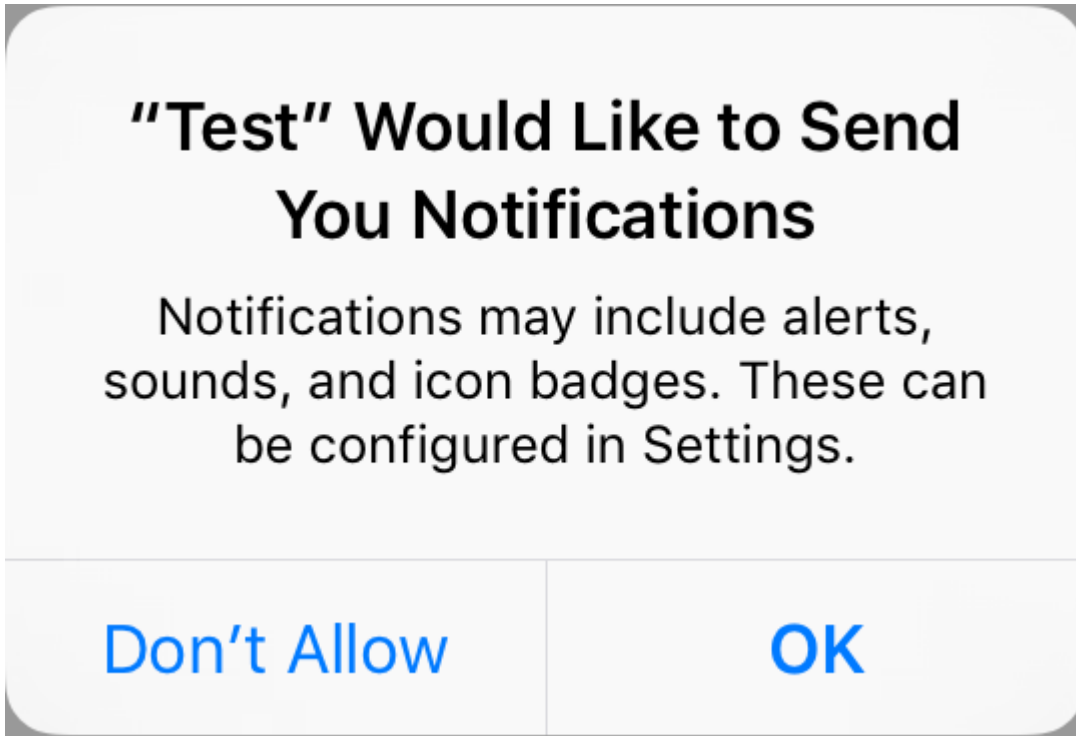
viewDidLoad viewWillAppear viewDidAppear◦

iOS 8

```
let settings = UIUserNotificationSettings(forTypes: [.Badge, .Sound, .Alert], categories: nil)
UIApplication.sharedApplication().registerUserNotificationSettings(settings)
```

Objective-C

```
UIUserNotificationSettings *settings = [UIUserNotificationSettings
settingsForTypes:(UIUserNotificationTypeBadge | UIUserNotificationTypeSound |
UIUserNotificationTypeAlert) categories:nil];
[[UIApplication sharedApplication] registerUserNotificationSettings:settings];
```



“”

°

```
func application(application: UIApplication, didReceiveLocalNotification notification:
UILocalNotification) {
}
}
```

Objective-C

```
- (void)application:(UIApplication *)application
didReceiveLocalNotification:(UILocalNotification *)notification {
}
```

AppDelegateUIApplicationDelegate°

UUID

°

UUID

```

let notification = UILocalNotification()
let uuid = NSUUID().uuidString
notification.userInfo = ["UUID": uuid]
UIApplication.shared.scheduleLocalNotification(notification)

```

Objective-C

```

UILocalNotification *notification = [[UILocalNotification alloc] init];
NSString *uuid = [[NSUUID UUID] UUIDString];
notification.userInfo = @{@"UUID": uuid };
[[UIApplication sharedApplication] scheduleLocalNotification:notification];

```

UUID。。

```

let scheduledNotifications = UIApplication.shared.scheduledLocalNotifications

guard let scheduledNotifications = scheduledNotifications else {
    return
}

for notification in scheduledNotifications where "\(notification.userInfo!["UUID"]!)" ==
UUID_TO_CANCEL {
    UIApplication.sharedApplication().cancelLocalNotification(notification)
}

```

Objective-C

```

NSArray *scheduledNotifications = [[UIApplication sharedApplication]
scheduledLocalNotifications];

for (UILocalNotification *notification in scheduledNotifications) {
    if ([[notification.userInfo objectForKey:@"UUID"] compare: UUID_TO_CANCEL]) {
        [[UIApplication sharedApplication] cancelLocalNotification:notification];
        break;
    }
}

```

UUIDCore DataRealm。

3

```

UIApplication.shared.presentLocalNotificationNow(notification)

```

2

```

UIApplication.sharedApplication().presentLocalNotificationNow(notification)

```

Objective-C

```

[[UIApplication sharedApplication] presentLocalNotificationNow:notification];

```

UILocalNotificationfireDatetimeZone。

◦ ◦

nil◦

```
.caf .wav.aiff◦ 30◦ UILocalNotificationDefaultSoundName ◦
```

Objective-C

```
UILocalNotification *notification = [UILocalNotification new];  
notification.soundName = @"nameOfSoundInBundle.wav"; // Use  
UILocalNotificationDefaultSoundName for the default alert sound
```

```
let notification = UILocalNotification()  
notification.soundName = "nameOfSoundInBundle.wav"
```

Swift 3.0 iOS 10

AppDelegate

```
import UserNotifications
```

didFinishLaunchingWithOptions

```
UNUserNotificationCenter.current().requestAuthorization(options: [.alert, .sound, .badge]) {  
    (granted, error) in  
  
    // Here you can check Request is Granted or not.  
  
}
```

◦

```
let content = UNMutableNotificationContent()  
content.title = "10 Second Notification Demo"  
content.subtitle = "From Wolverine"  
content.body = "Notification after 10 seconds - Your pizza is Ready!!"  
content.categoryIdentifier = "myNotificationCategory"  
  
let trigger = UNTimeIntervalNotificationTrigger(  
    timeInterval: 10.0,  
    repeats: false)  
  
let request = UNNotificationRequest(  
    identifier: "10.second.message",  
    content: content,  
    trigger: trigger  
)  
UNUserNotificationCenter.current().add(request, withCompletionHandler: nil)
```

◦

◦

iOS10UILocalNotification

UILocalNotification API iOS10User NotificationsAPI. iOS10 User Notifications.

o

1. iOS 10
2. o
3. 3D.
4. o
5. UIRich Notification.

UILocalNotification API iOS10API.

DemoAPI [iOS10AdaptationTips](#) o

Swift

1. UserNotifications

```
/// Notification become independent from UIKit
import UserNotifications
```

2. localNotification

```
let center = UNUserNotificationCenter.current()
center.requestAuthorization(options: [.alert, .sound]) { (granted, error) in
    // Enable or disable features based on authorization.
}
```

3. localNotification

4.

```
@IBAction func triggerNotification(){
    let content = UNMutableNotificationContent()
    content.title = NSLocalizedString(localizedUserNotificationString(forKey: "Elon said:",
arguments: nil)
    content.body = NSLocalizedString(localizedUserNotificationString(forKey: "Hello Tom Get up,
let's play with Jerry!", arguments: nil)
    content.sound = UNNotificationSound.default()
    content.badge = UIApplication.shared().applicationIconBadgeNumber + 1;
    content.categoryIdentifier = "com.elonchan.localNotification"
    // Deliver the notification in five seconds.
    let trigger = UNTimeIntervalNotificationTrigger.init(timeInterval: 60.0, repeats:
true)
    let request = UNNotificationRequest.init(identifier: "FiveSecond", content: content,
trigger: trigger)

    // Schedule the notification.
    let center = UNUserNotificationCenter.current()
    center.add(request)
}

@IBAction func stopNotification(_ sender: AnyObject) {
```

```

let center = UNUserNotificationCenter.current()
center.removeAllPendingNotificationRequests()
// or you can remove specifical notification:
// center.removePendingNotificationRequests(withIdentifiers: ["FiveSecond"])
}

```

Objective-C

1. UserNotifications

```

// Notifications are independent from UIKit
#import <UserNotifications/UserNotifications.h>

```

2. localNotification

```

UNUserNotificationCenter *center = [UNUserNotificationCenter currentNotificationCenter];
[center requestAuthorizationWithOptions:(UNAuthorizationOptionBadge |
UNAuthorizationOptionSound | UNAuthorizationOptionAlert)
completionHandler:^(BOOL granted, NSError * _Nullable error) {
    if (!error) {
        NSLog(@"request authorization succeeded!");
        [self showAlert];
    }
}];

```

3. localNotification

- ```

UNMutableNotificationContent *content = [[UNMutableNotificationContent alloc] init];
content.title = [NSString localizedUserNotificationStringForKey:@"Elon said:"
arguments:nil];
content.body = [NSString localizedUserNotificationStringForKey:@"Hello Tom Get up, let's
play with Jerry!"
arguments:nil];
content.sound = [UNNotificationSound defaultSound];

// 4. update application icon badge number
content.badge = [NSNumber numberWithInt:([UIApplication
sharedApplication].applicationIconBadgeNumber + 1)];
// Deliver the notification in five seconds.
UNTimeIntervalNotificationTrigger *trigger = [UNTimeIntervalNotificationTrigger
triggerWithTimeInterval:5.f
repeats:NO];
UNNotificationRequest *request = [UNNotificationRequest
requestWithIdentifier:@"FiveSecond"
content:content
trigger:trigger];

/// 3. schedule localNotification
UNUserNotificationCenter *center = [UNUserNotificationCenter currentNotificationCenter];
[center addNotificationRequest:request withCompletionHandler:^(NSError * _Nullable error)
{
 if (!error) {
 NSLog(@"add NotificationRequest succeeded!");
 }
}];

```

'NSInternalInconsistencyException'60'

```
let trigger = UNTimeIntervalNotificationTrigger.init(timeInterval: 60, repeats: true)
```

**UILocalNotification** <https://riptutorial.com/zh-TW/ios/topic/635/uilocalnotification>

# 82: UINavigationController

UINavigationController. . .

## Examples

```
navigationController?.popViewControllerAnimated(true)
```

### Objective-C

```
[self.navigationController popViewControllerAnimated:YES];
```

```
navigationController?.popToRootViewControllerAnimated(true)
```

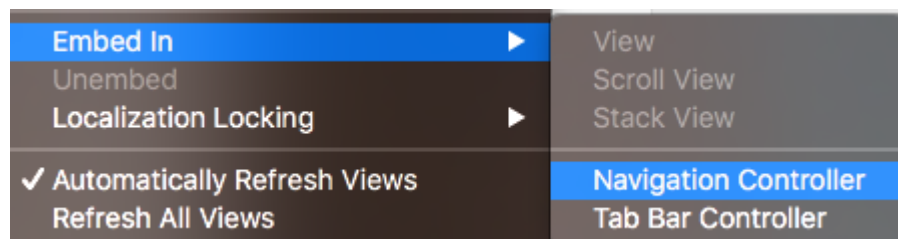
### C.

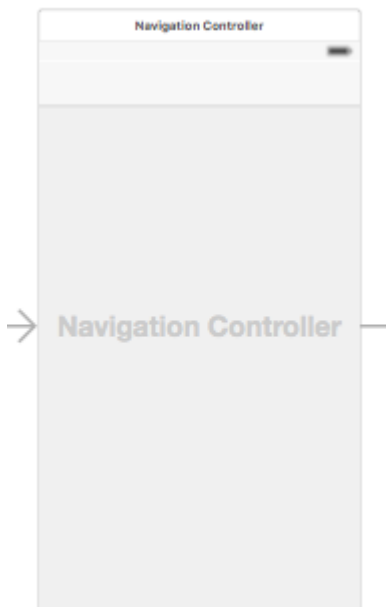
```
[self.navigationController popToRootViewControllerAnimated:YES];
```

## NavigationController

ViewController.

Editor > Embed In > Navigation Controller





```
//Swift
let viewController = UIViewController()
let navigationController = UINavigationController(rootViewController: viewController)

//Objective-C
UIViewController *viewController = [[UIViewController alloc] init];
UINavigationController *navigationController = [[UINavigationController alloc]
initWithRootViewController:viewController];
```

```
//Swift
let fooViewController = UIViewController()
navigationController?.pushViewController(fooViewController, animated: true)

//Objective-C
UIViewController *fooViewController = [[UIViewController alloc] init];
[navigationController pushViewController:fooViewController animated:YES];
```

UINavigationController navigation stack ◦

UI ◦ UINavigationController ◦ UINavigationController navigation stack ◦

UINavigationController ◦

**UINavigationController** <https://riptutorial.com/zh-TW/ios/topic/1079/uINavigationController>

# 83: UIPageViewController

UIPageViewController ◦ UIPageViewControllerUIPageViewControllerDataSource ◦  
UIPageViewControllerUIPageViewControllerpresentationCountpresentationIndex ◦

1. UIPageViewControllerTransitionStyle
2. UIPageViewControllerNavigationOrientation
3. UIPageViewControllerSpineLocation
4. UIPageViewControllerNavigationDirection

Apple Developer

## Examples

### UIPageViewController

1. UIPageViewController ◦ identifier UIPageViewController ◦ ◦

```
UIViewController *firstVC = [[UIViewController alloc] init];
firstVC.identifier = 0
UIViewController *secondVC = [[UIViewController alloc] init];
secondVC.identifier = 1
NSArray *viewControllers = [[NSArray alloc] initWithObjects: firstVC, secondVC, nil];
```

2. UIPageViewController ◦

```
UIPageViewController *pageViewController = [[UIPageViewController alloc]
initWithTransitionStyle:UIPageViewControllerTransitionStyleScroll

navigationOrientation:UIPageViewControllerNavigationOrientationHorizontal

options:nil];
```

3. UIPageViewControllerDataSource ◦

```
pageViewController.dataSource = self;
```

4. setViewControllers

```
if (viewControllers.count) {
 [pageViewController setViewControllers:@[[viewControllers objectAtIndex:0]]
 direction:UIPageViewControllerNavigationDirectionForward
 animated:NO
 completion:nil];
}
```

5. UIPageViewControllerappearancerotation ◦

```
[self addChildViewController:pageViewController];
pageViewController.view.frame = self.view.frame;
[self.view addSubview:pageViewController.view];
[pageViewController didMoveToParentViewController:self];
```

## 6. UIPageViewControllerDataSource

```
- (UIViewController *)pageViewController:(UIPageViewController *)pageViewController
 viewControllerBeforeViewController:(UIViewController *)viewController
{
 index = [(Your View Controller Base Class *)viewController identifier];
 index--;
 return [self childViewControllerAtIndex:index];
}

- (UIViewController *)pageViewController:(UIPageViewController *)pageViewController
 viewControllerAfterViewController:(UIViewController *)viewController
{
 index = [(Your View Controller Base Class *)viewController identifier];
 index++;
 return [self childViewControllerAtIndex:index];
}

- (NSInteger)presentationCountForPageViewController:(UIPageViewController *)pageViewController
{
 return [viewControllers count];
}

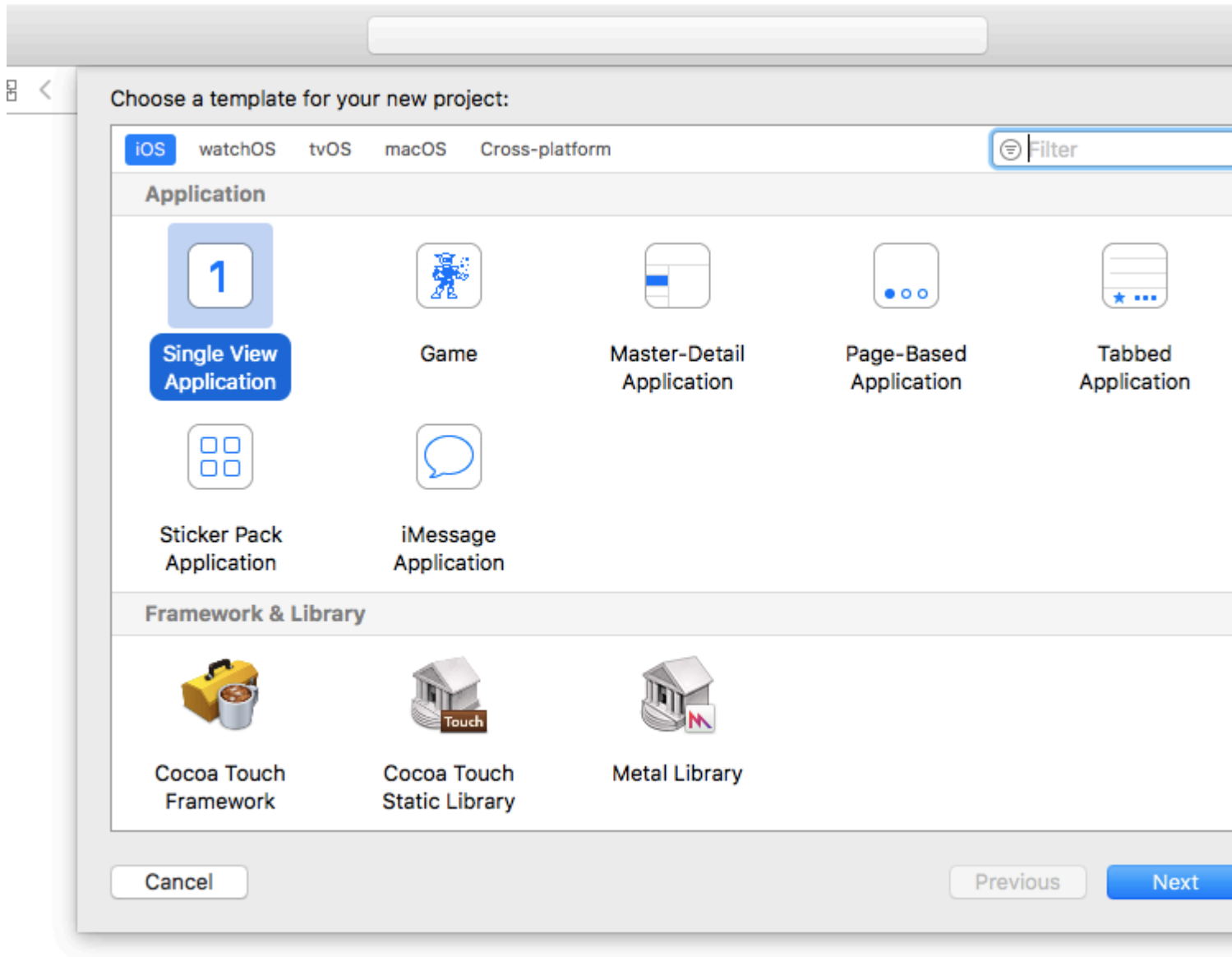
- (NSInteger)presentationIndexForPageViewController:(UIPageViewController *)pageViewController
{
 return index;
}
```

## 7. nil.

```
- (UIViewController *)childViewControllerAtIndex:(NSInteger) index
{
 if (index <= ([viewControllers count] - 1)) {
 return [viewControllers objectAtIndex:index];
 } else {
 return nil;
 }
}
```

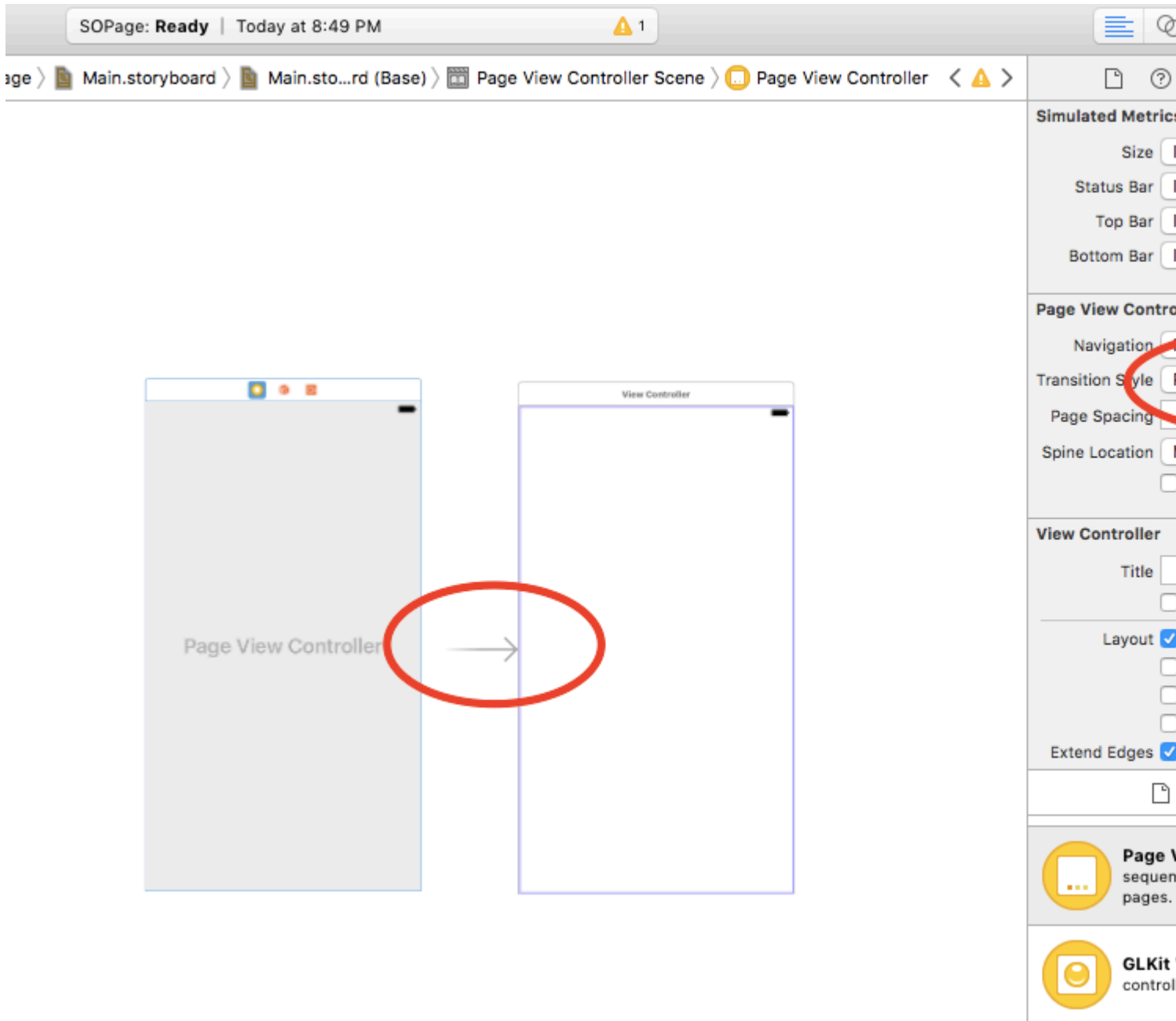
## 1.





2.

- 1.
- 2.



### 3. UIPageViewController

### 4. UIPageViewController:)

```
class PageViewController: UIPageViewController, UIPageViewControllerDataSource {

 override func viewDidLoad() {
 self.dataSource = self
 let controller = createViewController()
 self.setViewControllers([controller], direction: .forward, animated: false,
completion: nil)
 }

 func pageViewController(_ pageViewController: UIPageViewController,
viewControllerBefore viewController: UIViewController) -> UIViewController? {
 let controller = createViewController()
 return controller
 }
}
```

```
func pageViewController(_ pageViewController: UIPageViewController,
viewControllerAfter viewController: UIViewController) -> UIViewController? {
 let controller = createViewController()
 return controller
}

func createViewController() -> UIViewController {
 var randomColor: UIColor {
 return UIColor(hue: CGFloat(arc4random_uniform(360))/360, saturation: 0.5,
brightness: 0.8, alpha: 1)
 }
 let storyboard = UIStoryboard(name: "Main", bundle: nil)
 let controller = storyboard.instantiateViewController(withIdentifier: "View
Controller")
 controller.view.backgroundColor = randomColor
 return controller
}
}
```



---

# 84: UIPheonix - UI

UIPheonixUI+macOSiOSStvOS/。 API。

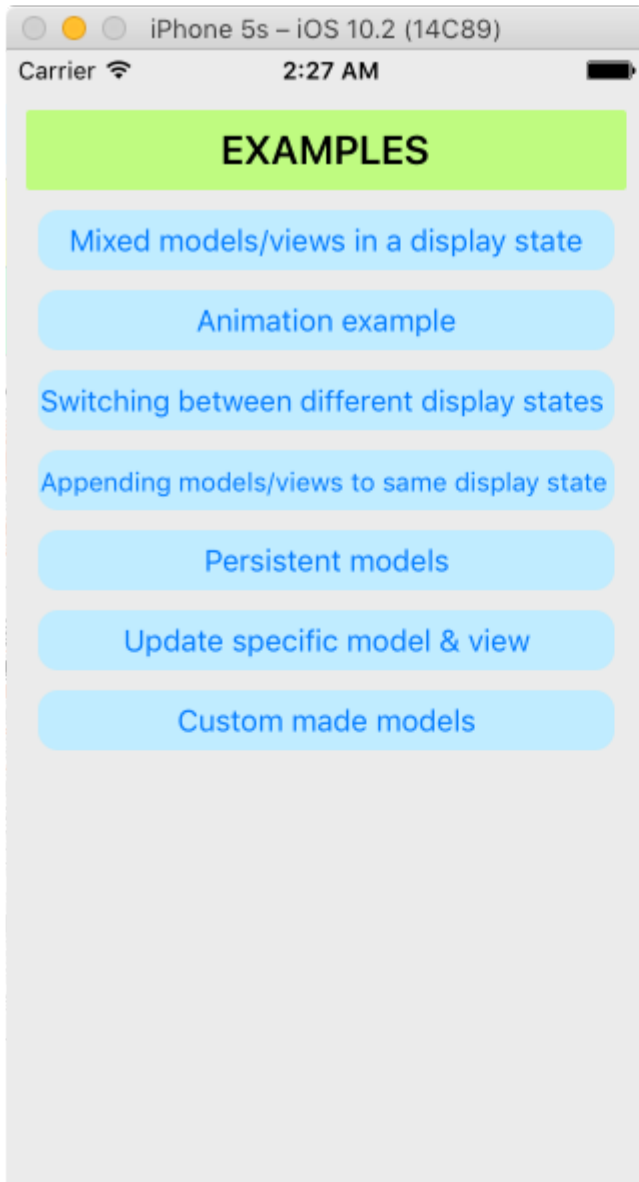
<https://github.com/MKGitHub/UIPheonix>

- ◦
- ◦
- UI◦
- UI◦
- ◦
- 
- ◦
- Apple◦
- macOSiOSStvOSKung Fu

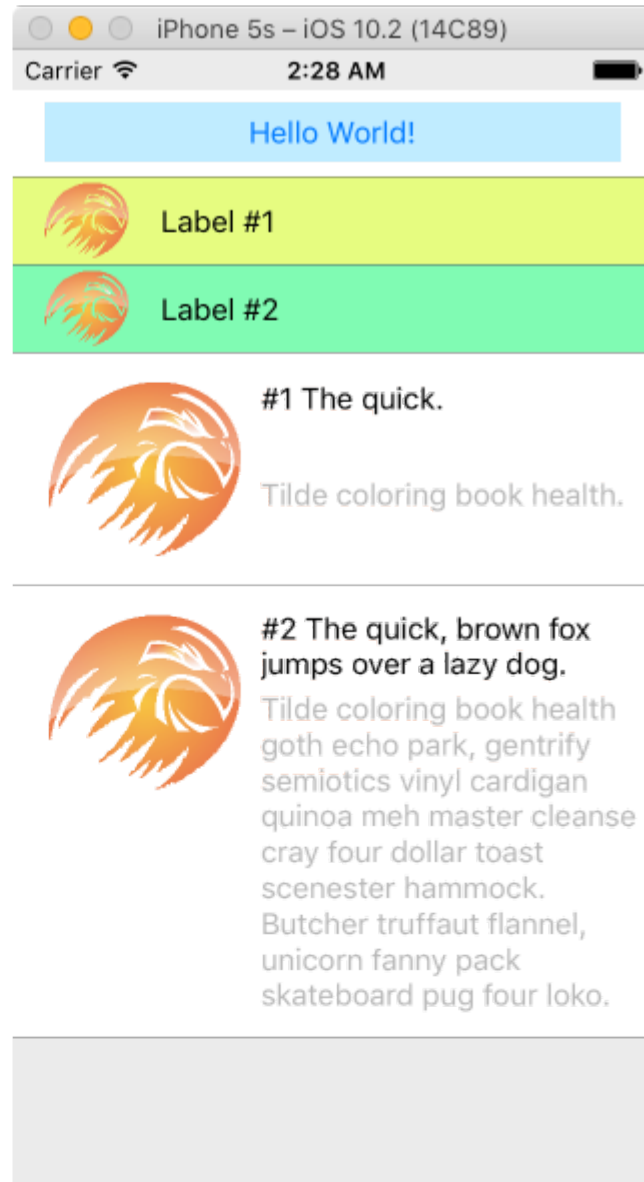
## Examples

UI

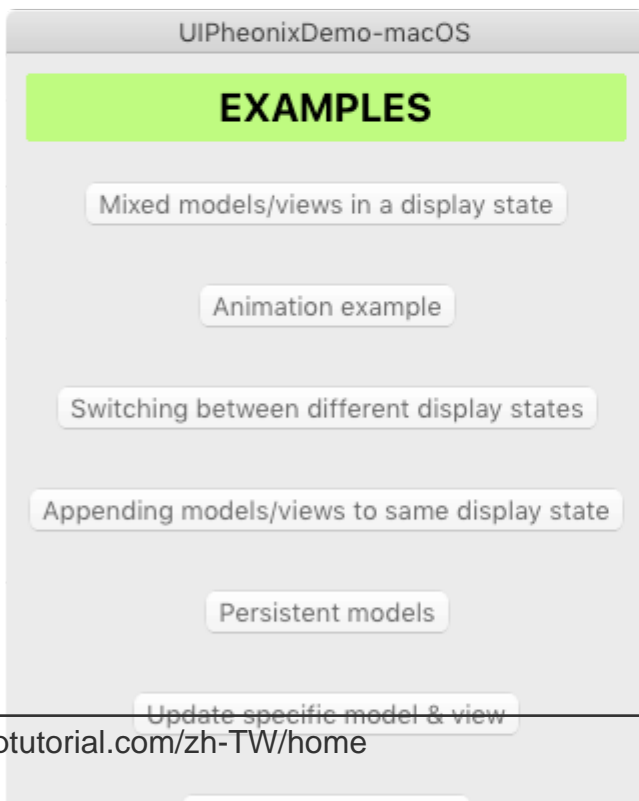
## iOS - Collection View



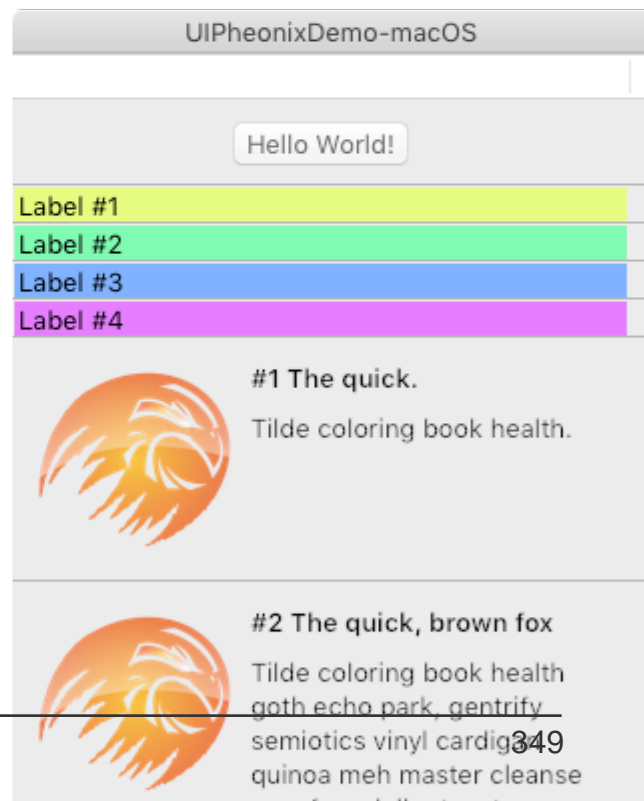
## iOS - Table View



## macOS - Collection View



## macOS - Table View



---

# 85: UIPickerView

## Examples

---

```
class UIPickerViewExampleViewController : UIViewController, UIPickerViewDelegate,
UIPickerViewDataSource {
 @IBOutlet weak var btnFolder: UIButton!
 let pickerView = UIPickerView()
 let pickerViewRows = ["First row,", "Secound row,", "Third row,", "Fourth row"]

 override func viewDidLoad() {
 super.viewDidLoad()
 self.btnFolder.addTarget(self, action: #selector(CreateListVC.btnFolderPress),
forControlEvents: UIControlEvents.TouchUpInside)
 }

 @objc private func btnFolderPress() {
 self.pickerView.delegate = self
 self.pickerView.dataSource = self
 self.view.addSubview(self.pickerView)
 }

 //MARK: UIPickerViewDelegate

 func pickerView(pickerView: UIPickerView, titleForRow row: Int, forComponent component:
Int) -> String? {
 return self.pickerViewRows[row]
 }

 //MARK: UIPickerViewDataSource

 func numberOfComponentsInPickerView(pickerView: UIPickerView) -> Int {
 return 1
 }

 func pickerView(pickerView: UIPickerView, numberOfRowsInComponent component: Int) -> Int {
 return self.pickerViewRows.count
 }
}
```

---

## Objective-C

```
@property (nonatomic, strong) UIPickerView *countryPicker;
@property (nonatomic, strong) NSArray *countryNames;

- (void)viewDidLoad {
 [super viewDidLoad];
 _countryNames = @[@"Australia (AUD)", @"China (CNY)",
 @"France (EUR)", @"Great Britain (GBP)", @"Japan (JPY)", @"INDIA
(IN)", @"AUSTRALIA (AUS)", @"NEW YORK (NW)"];
}
```

```

 [self pickcountry];
}

-(void)pickcountry {
 _countryPicker = [[UIPickerView alloc]init];

 _countryPicker.delegate = self;
 _countryPicker.dataSource = self;

 [[UIPickerView appearance] setBackgroundColor:[UIColor colorWithRed:21/255.0
green:17/255.0 blue:50/255.0 alpha:1.0]];
}

#pragma mark- pickerView Delegates And datasource

- (NSInteger)numberOfComponentsInPickerView:(UIPickerView *)pickerView {
 return 1;
}

- (NSInteger)pickerView:(UIPickerView *)pickerView
numberOfRowsInComponent:(NSInteger)component {
 return _countryNames.count;
}

- (NSString *)pickerView:(UIPickerView *)pickerView
titleForRow:(NSInteger)row
forComponent:(NSInteger)component {
 return _countryNames[row];
}

- (void)pickerView:(UIPickerView *)pickerView didSelectRow:(NSInteger)row
inComponent:(NSInteger)component {
 NSString *pickedCountryName = _countryNames[row];
}

```

## pickerView

### Objective-C

```

//Displays the country pickerView with black background and white text
[self. countryPicker setValue:[UIColor whiteColor] forKey:@"textColor"];
[self. countryPicker setValue:[UIColor blackColor] forKey:@"backgroundColor"];

```

```

let color1 = UIColor(colorLiteralRed: 1, green: 1, blue: 1, alpha: 1)
let color2 = UIColor(colorLiteralRed: 0, green: 0, blue: 0, alpha: 1)
pickerView2.setValue(color1, forKey: "textColor")
pickerView2.setValue(color2, forKey: "backgroundColor")

```

**UIPickerView** <https://riptutorial.com/zh-TW/ios/topic/4242/uipickerview>



# 86: UIRefreshControl TableView

UIRefreshControl . . .

## Examples

### Objective-C

#### ViewController

```
@property (nonatomic) UIRefreshControl *refreshControl;
```

#### viewDidLoad() refreshControl

```
self.refreshControl = [[UIRefreshControl alloc] init];
[self.tableView addSubview:self.refreshControl];
[self.refreshControl addTarget:self action:@selector(refreshTable)
forControlEvents:UIControlEventValueChanged];
//Setting the tint Color of the Activity Animation
self.refreshControl.tintColor = [UIColor redColor];
//Setting the attributed String to the text
NSMutableAttributedString * string = [[NSMutableAttributedString alloc]
initWithString:@"firstsecondthird"];
[string addAttribute:NSForegroundColorAttributeName value:[UIColor redColor]
range:NSMakeRange(0, 5)];
[string addAttribute:NSForegroundColorAttributeName value:[UIColor greenColor]
range:NSMakeRange(5, 6)];
[string addAttribute:NSForegroundColorAttributeName value:[UIColor blueColor]
range:NSMakeRange(11, 5)];
self.refreshControl.attributedString = string;
```

#### refreshTable

```
- (void)refreshTable {
 //TODO: refresh your data
 [self.refreshControl endRefreshing];
 [self.refreshControl beginRefreshing];
 [self.tableView reloadData];
 [self.refreshControl endRefreshing];
}
```



#### tableViewrefreshControl

```
UIRefreshControl *refreshControl = [[UIRefreshControl alloc] init];
```

```
[refreshControl addTarget:self action:@selector(pullToRefresh:)
forControlEvents:UIControlEventValueChanged];
self.scrollView.alwaysBounceVertical = YES;
[self.scrollView addSubview:refreshControl];

- (void)pullToRefresh:(UIRefreshControl*) sender{
//Do work off the main thread
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{
 // Simulate network traffic (sleep for 2 seconds)
 [NSThread sleepForTimeInterval:2];
 //Update data
 //Call complete on the main thread
 dispatch_sync(dispatch_get_main_queue(), ^{
 //Update network activity UI
 NSLog(@"COMPLETE");
 [sender endRefreshing];
 });
});
}
```

**UIRefreshControl TableView** <https://riptutorial.com/zh-TW/ios/topic/8278/uirefreshcontrol-tableview>

# 87: UIScrollView AutoLayout

## Examples

### ScrollableController

#### AutolayoutUIScrollView

UIScrollViewContentViewUIScrollView

```
import Foundation
import UIKit

class ScrollableController : UIViewController {

 private var scrollView: UIScrollView!
 private var contentView: UIView!

 override func viewDidLoad() {
 super.viewDidLoad()

 //Setup
 self.initControls()
 self.setTheme()
 self.layoutScrollView()
 self.layoutContentView()

 //Add child views
 self.addChildViews()
 }

 func initControls() {
 self.scrollView = UIScrollView()
 self.contentView = UIView()
 }

 func setTheme() {
 self.scrollView.backgroundColor = UIColor.blue()
 self.contentView.backgroundColor = UIColor.orange()
 }

 func layoutScrollView() {
 self.view.addSubview(self.scrollView)

 let views: NSDictionary = ["scrollView": self.scrollView]
 var constraints = Array<String>()

 //Constrain the scrollView to our controller's self.view.
 constraints.append("H:|-0-[scrollView]-0-|")
 constraints.append("V:|-0-[scrollView]-0-|")

 for constraint in constraints {
 self.view.addConstraints(NSLayoutConstraint.constraints(withVisualFormat:
 constraint, options: NSLayoutConstraintOptions(rawValue: 0), metrics: nil, views: views as!
 [String : AnyObject]))
 }
 }
}
```

```

 self.scrollView.translatesAutoresizingMaskIntoConstraints = false
 }

 func layoutContentView() {
 self.scrollView.addSubview(self.contentView)

 let views: NSDictionary = ["contentView": self.contentView, "view": self.view]
 var constraints = Array<String>()

 //Constrain the contentView to the scrollView.
 constraints.append("H:|-0-[contentView]-0-|")
 constraints.append("V:|-0-[contentView]-0-|")

 for constraint in constraints {
 self.scrollView.addConstraints(NSLayoutConstraint.constraints(withVisualFormat:
constraint, options: NSLayoutFormatOptions(rawValue: 0), metrics: nil, views: views as!
[String : AnyObject]))
 }

 //Disable Horizontal Scrolling by making the contentView EqualWidth with our
controller's self.view (ScrollView's parentView).
 self.view.addConstraints(NSLayoutConstraint.constraints(withVisualFormat:
"H:[contentView(==view)]", options: NSLayoutFormatOptions(rawValue: 0), metrics: nil, views:
views as! [String : AnyObject]))

 self.contentView.translatesAutoresizingMaskIntoConstraints = false
 }

 func addChildViews() {
 //Init
 let greenView = UIView()
 let whiteView = UIView()

 //Theme
 greenView.backgroundColor = UIColor.green()
 whiteView.backgroundColor = UIColor.orange()

 //Layout -- Child views are added to the 'ContentView'
 self.contentView.addSubview(greenView)
 self.contentView.addSubview(whiteView)

 let views: NSDictionary = ["greenView": greenView, "whiteView": whiteView];
 var constraints = Array<String>()

 //Constrain the greenView to the contentView with a height of 400 and 15 spacing all
around.
 constraints.append("H:|-15-[greenView]-15-|")
 constraints.append("V:|-15-[greenView(400)]")

 //Constrain the whiteView below the greenView with 15 spacing all around and a height
of 500.
 constraints.append("H:|-15-[whiteView]-15-|")
 constraints.append("V:[greenView]-15-[whiteView(500)]-15-|")

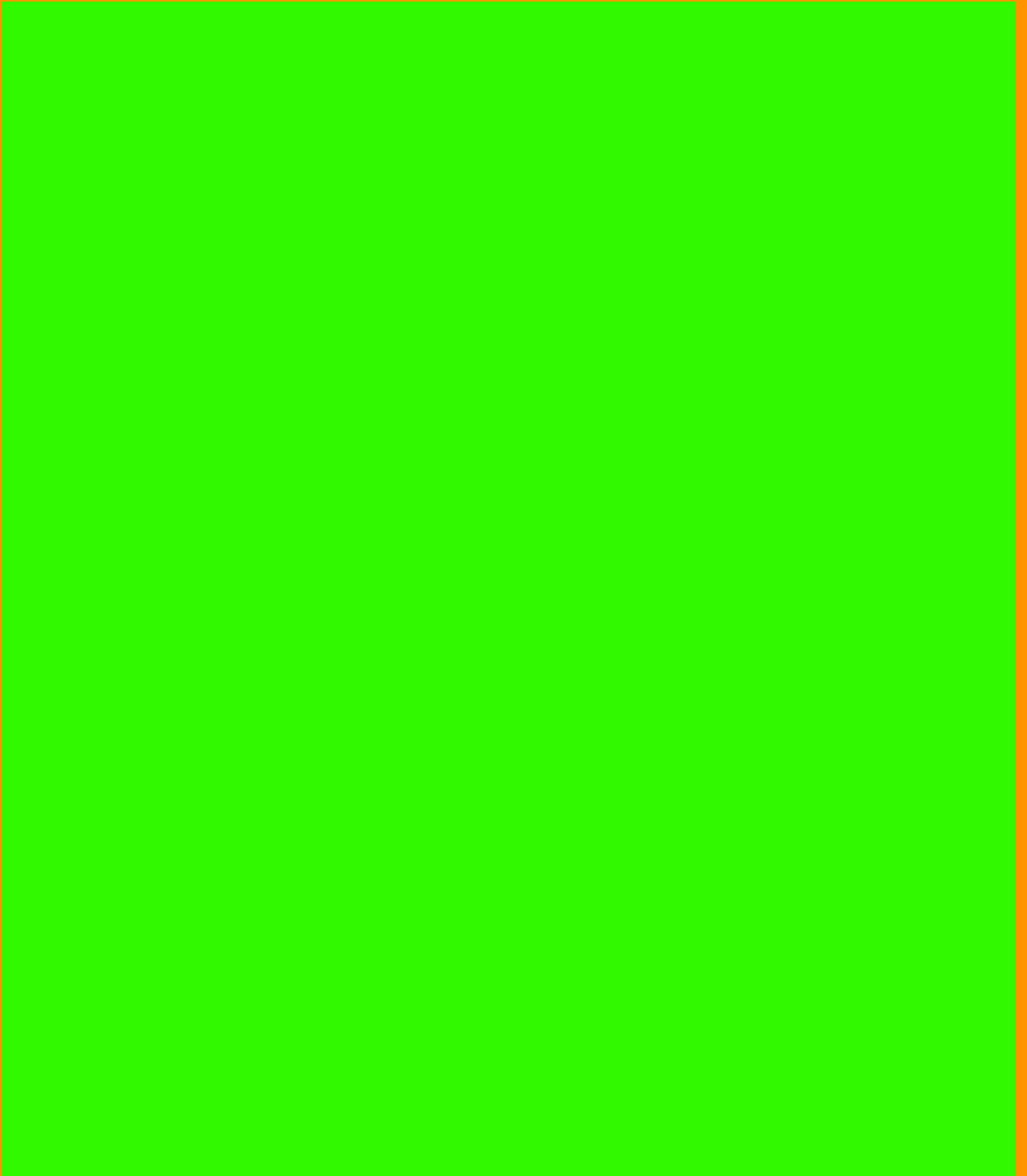
 for constraint in constraints {
 self.contentView.addConstraints(NSLayoutConstraint.constraints(withVisualFormat:
constraint, options: NSLayoutFormatOptions(rawValue: 0), metrics: nil, views: views as!
[String : AnyObject]))
 }
 }

```

```
 greenView.translatesAutoresizingMaskIntoConstraints = false
 whiteView.translatesAutoresizingMaskIntoConstraints = false
 }
}
```

**greenView400+ whiteView500。 ScrollViewcontentSize。**

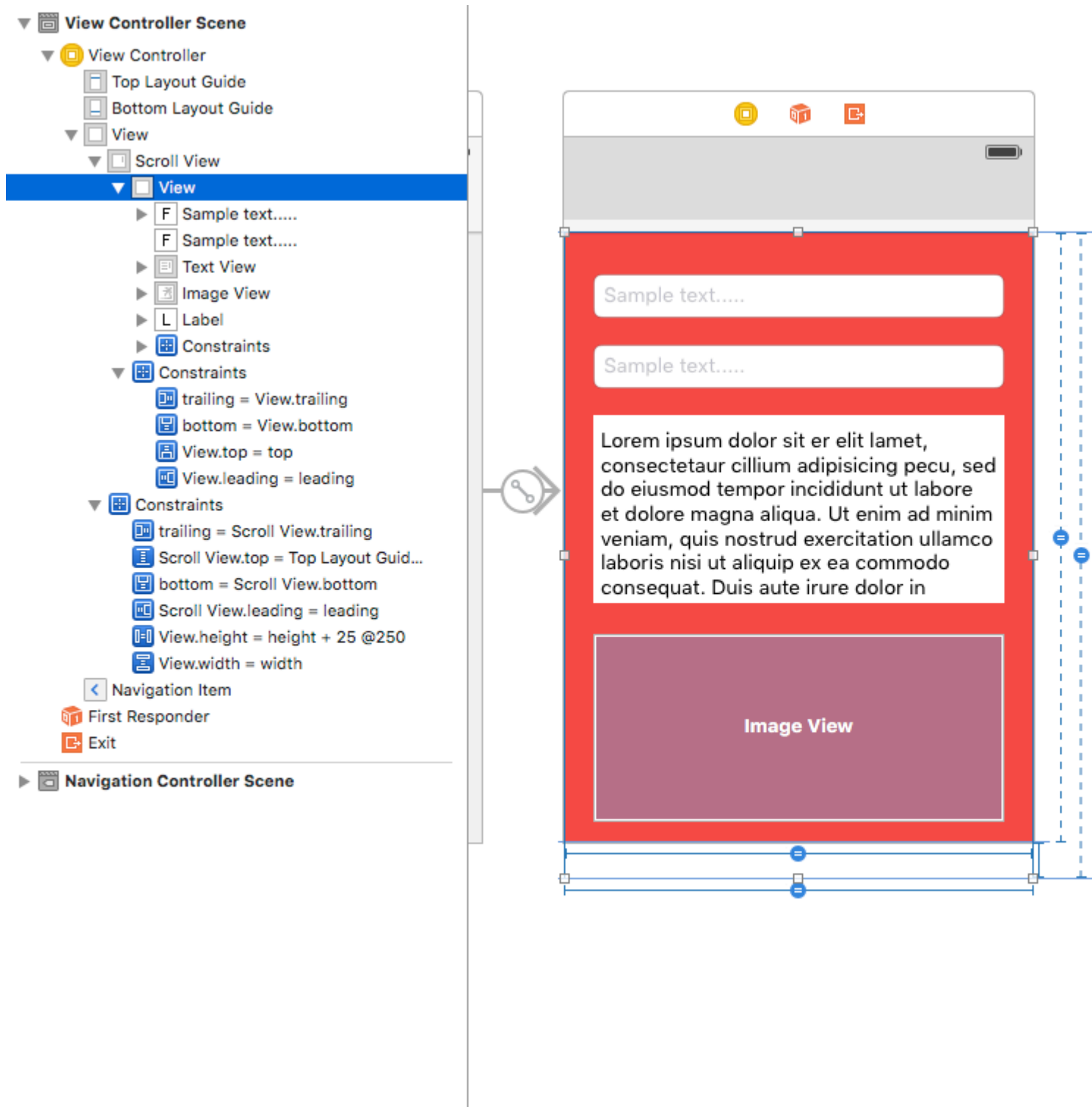
EqualWidthcontentViewself.view

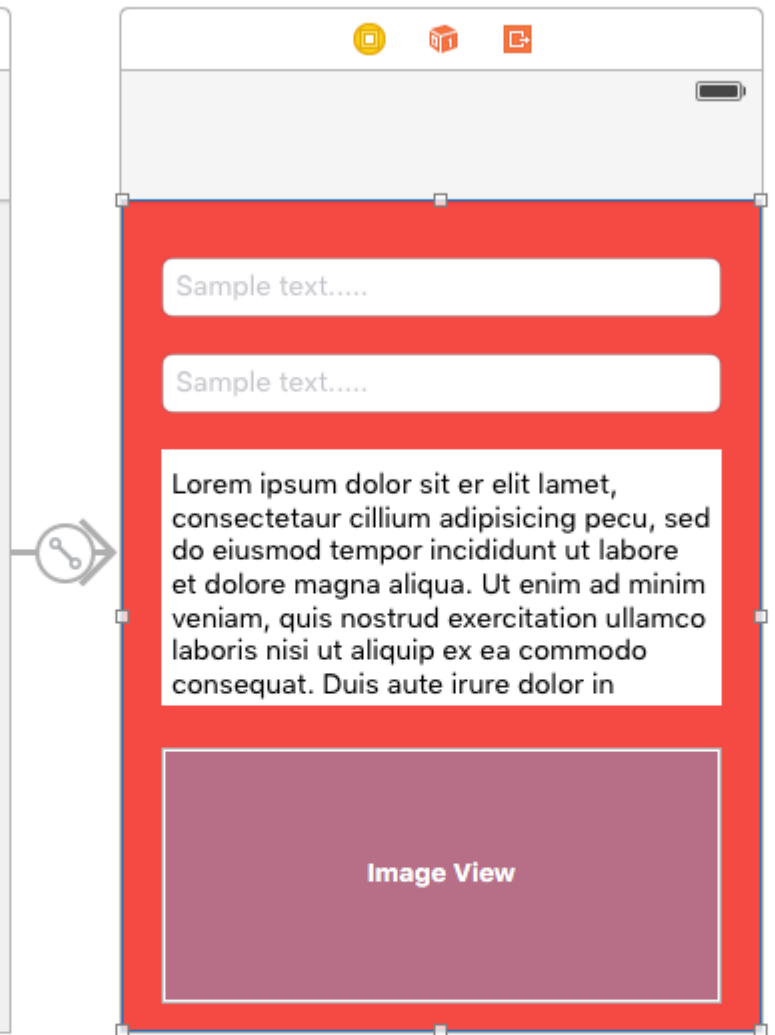
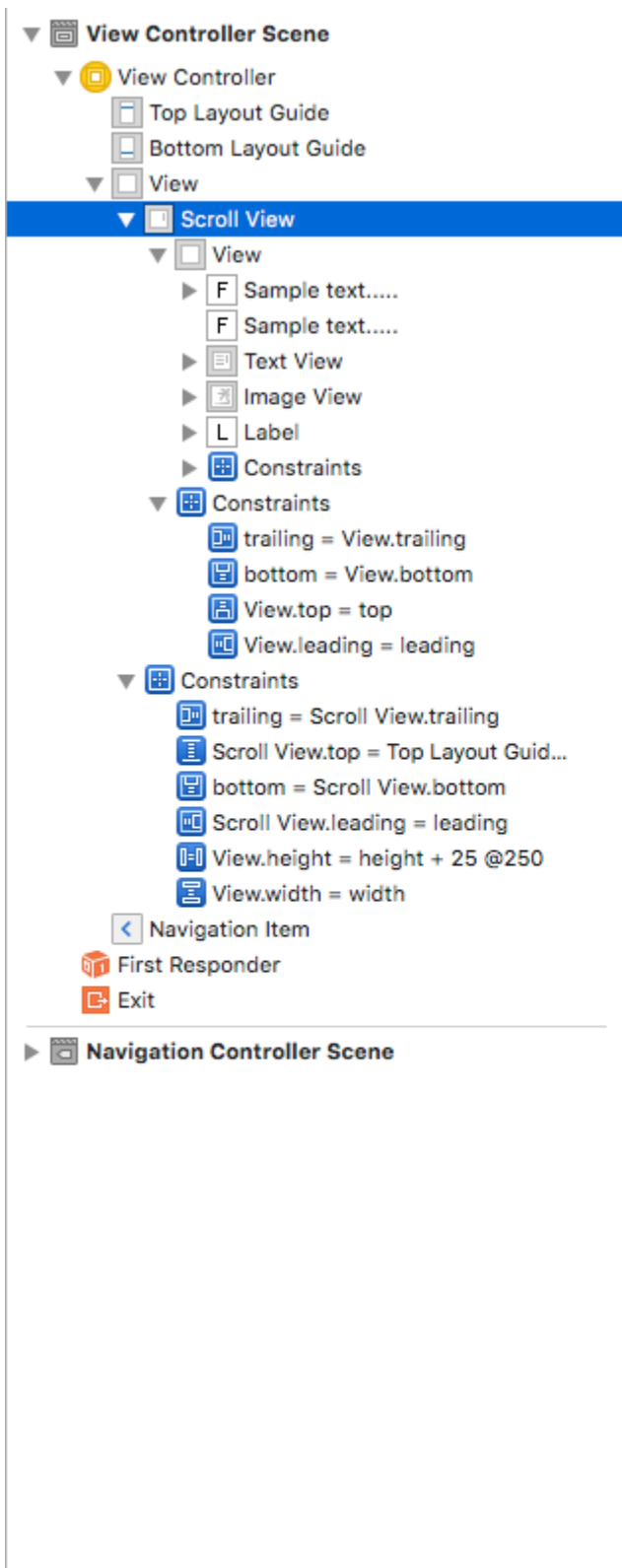


2. scrollView . . .

3. . Y42020440.

3.





UIScrollView AutoLayout <https://riptutorial.com/zh-TW/ios/topic/4671/uiscrollview-autolayout>



---

# 88: UIScrollView

## Examples

### UIScrollView

CGRectUIScrollView°

```
let scrollView = UIScrollView.init(frame: CGRect(x: 0, y: 0, width: 320, height: 400))
```

### Objective-C

```
UIScrollView *scrollView = [[UIScrollView alloc] initWithFrame:CGRectMake(0, 0, 320, 400)];
```

contentSize° ◦ contentSize UIScrollView°

### Autolayout

autolayout4° ◦ contentSize°

```
scrollView.contentSize = CGSize(width: 640, height: 800)
```

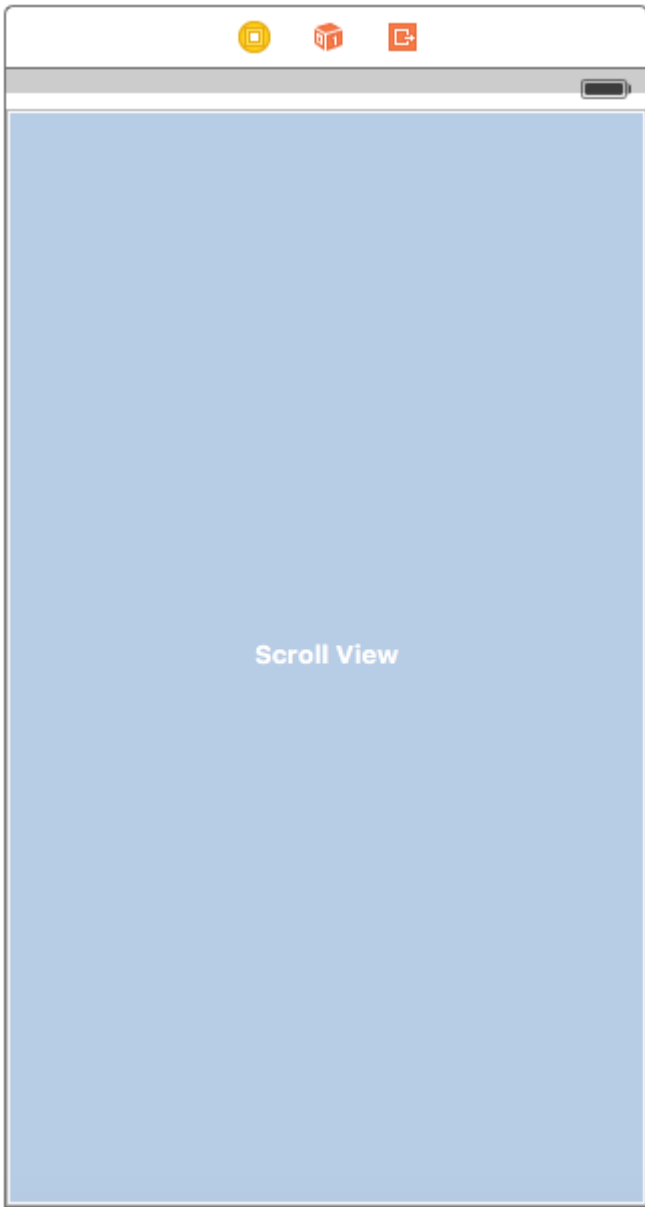
### Objective-C

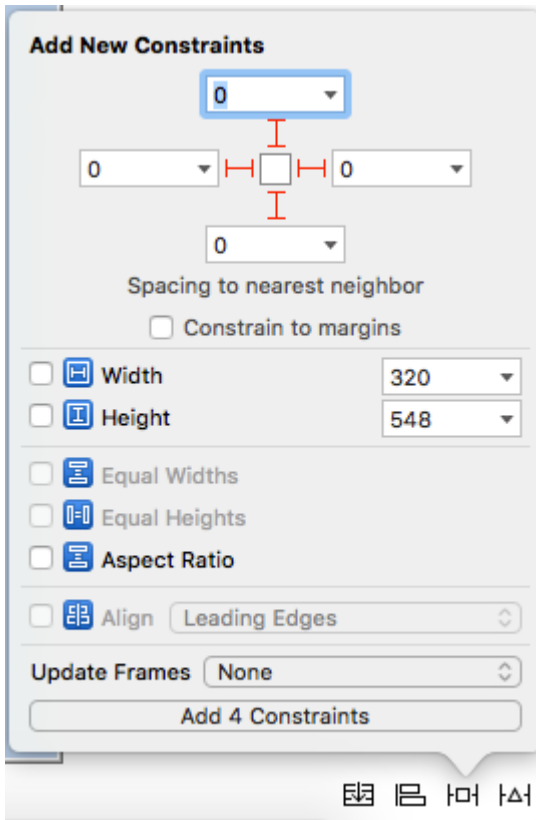
```
scrollView.contentSize = CGSizeMake(640, 800);
```

## ScrollView with AutoLayout

scrollViewautolayout°

- 
- viewControlleriPhone-4inch°
- viewController



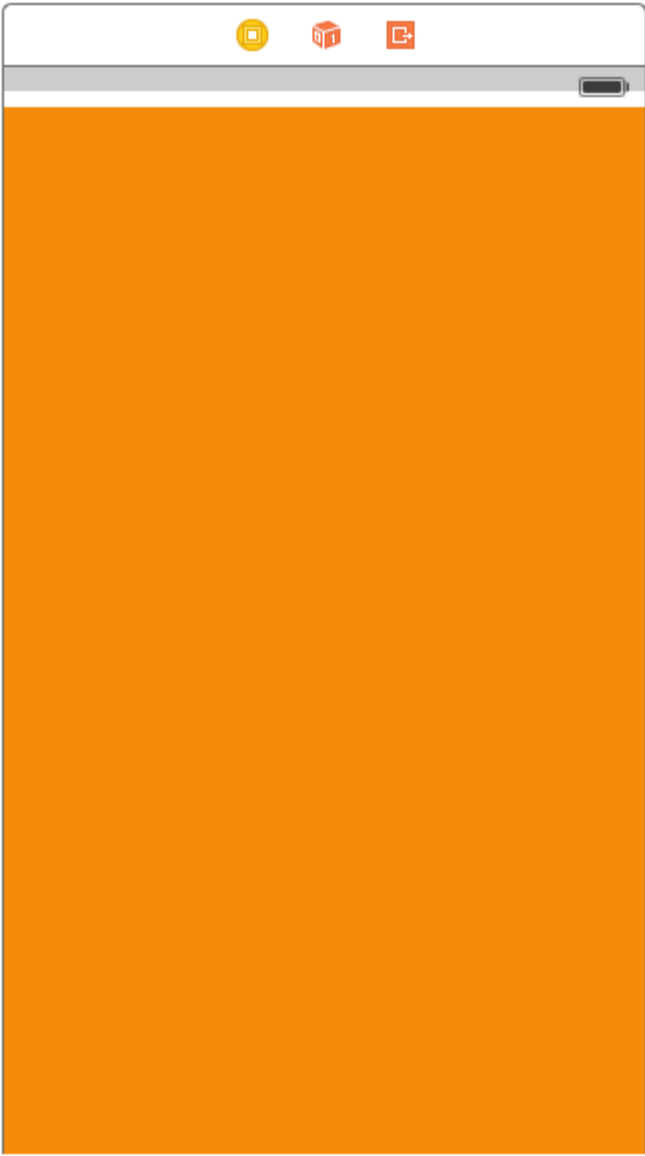


scrollviewviewController

1

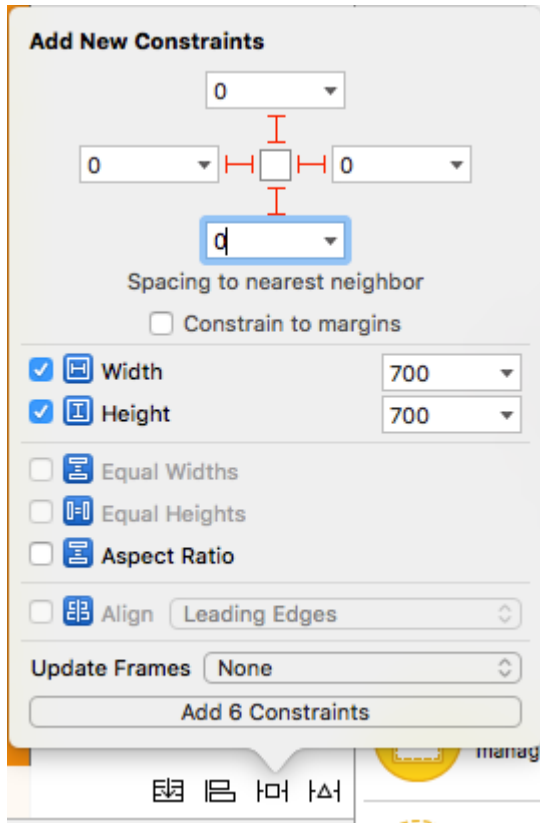
。

- UIView(0,0,700,700)。



o

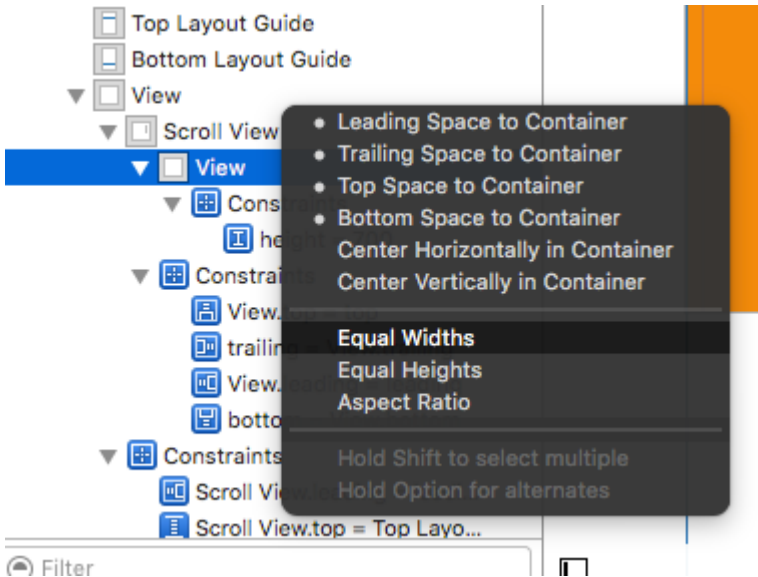
•



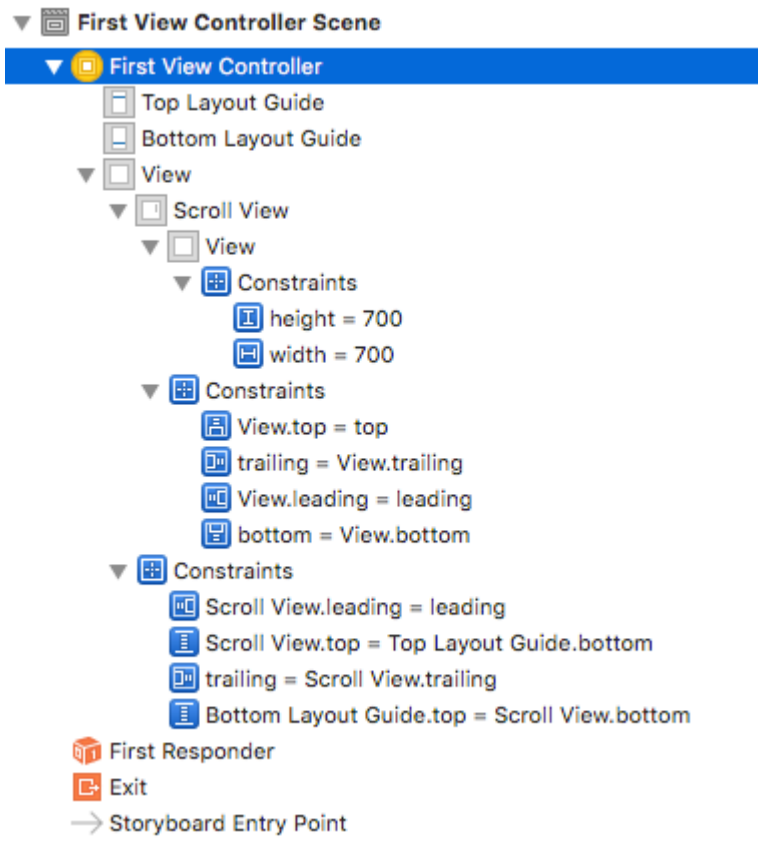
- 
- 700◦
- scrollview = 0◦
- bottom spacescrollview = 0scrollview◦

## 2 scrollview◦

- 
- ◦
- ◦
- Ctrl◦

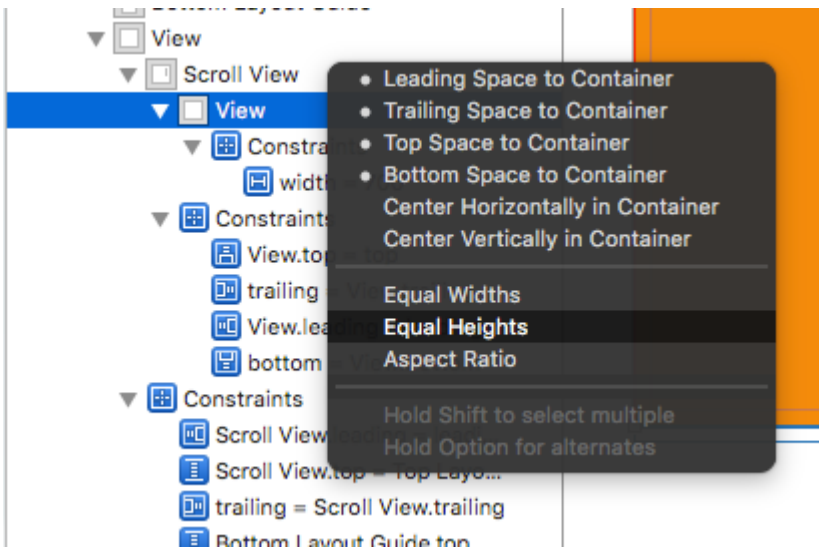


- 
- 3**
- 
- 
- 

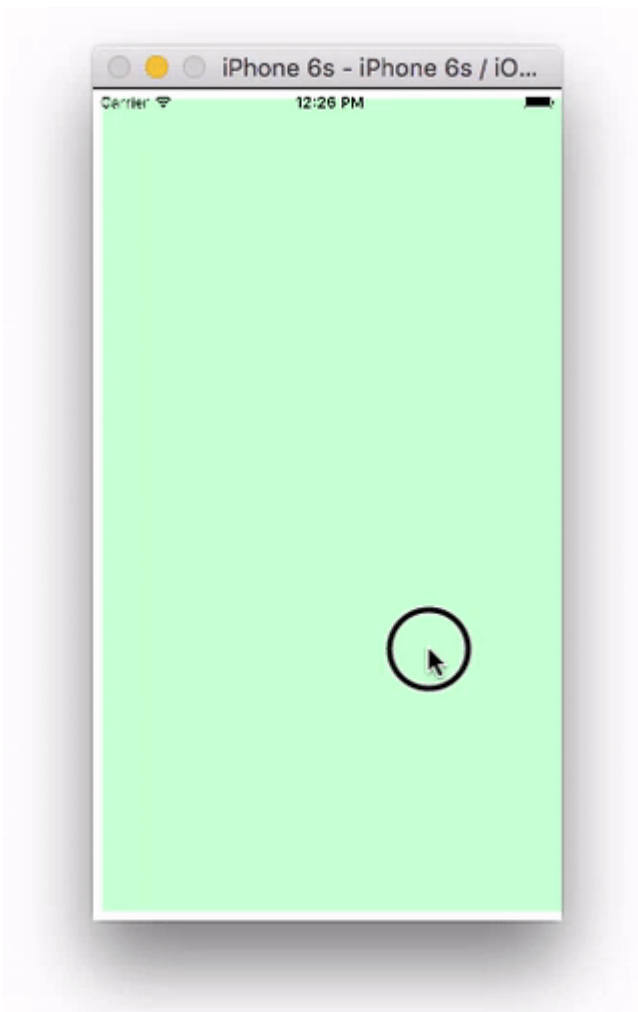


- 0,0,700,700
- ◦
-

- Ctrl.



•  
Interface Builder. 10. ◦



UIView ◦ ◦

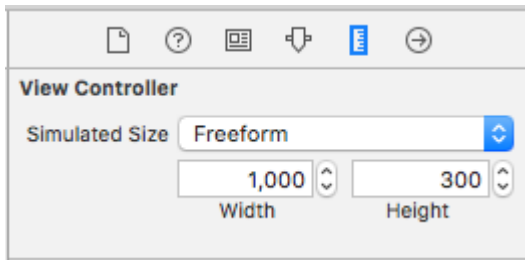
- UIScrollView ◦ 'UIView' ◦
- ◦
- ◦

---

◦

---

- View Controller Size Inspector Freeform ◦ 1,000 300 ◦ ◦



UIScrollView ◦



UIView ◦ ◦ ◦ UIView ◦ ◦ ◦





⌘ ◦ </ kbd> ◦ ◦ ◦

- View Controller Scene
  - View Controller
    - Top Layout Guide
    - Bottom Layout Guide
    - View**
      - Scroll View
        - Content View**
          - Constraints
          - Constraints
        - First Responder
        - Exit
        - Storyboard Entry Point

- Leading Space to View Margin
- Trailing Space to View Margin
- Vertical Spacing to Top Layout Guide
- Vertical Spacing to Bottom Layout Guide
- Center Horizontally in View
- Center Vertically in View

---

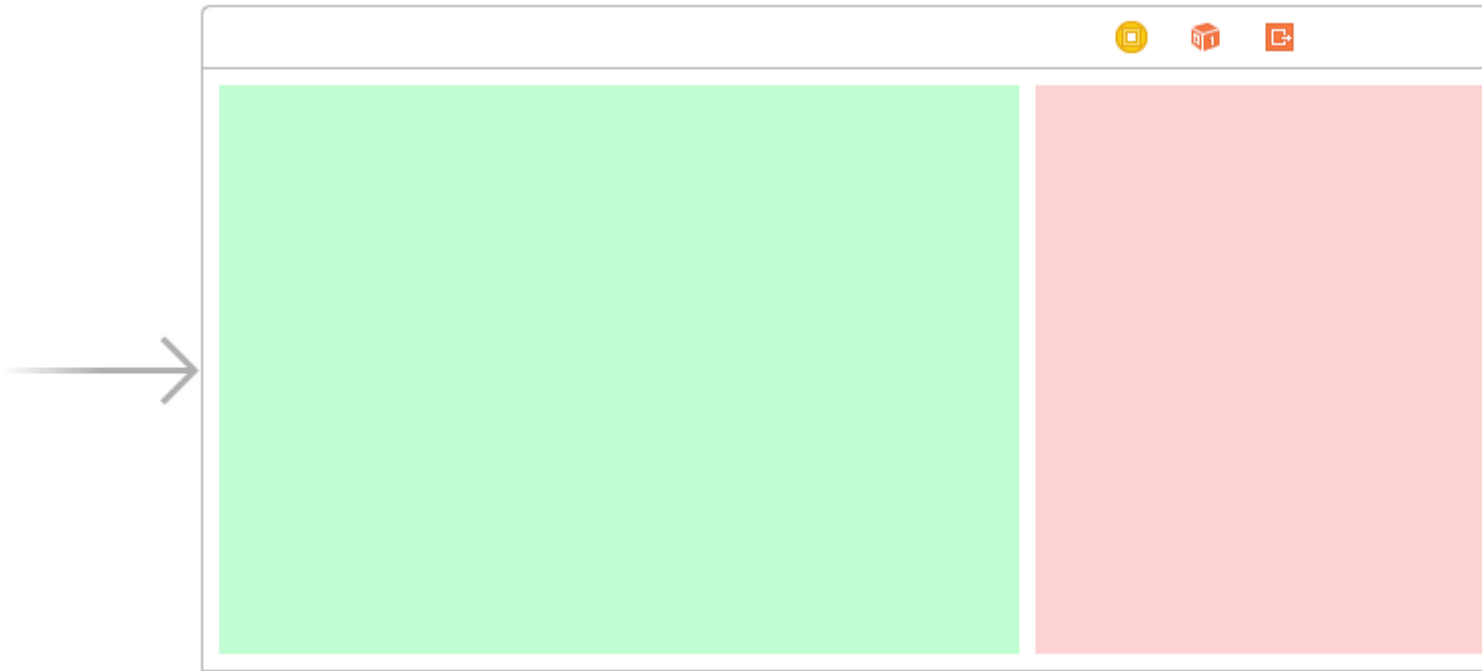
- Equal Widths
- Equal Heights**
- Aspect Ratio

---

- Hold Shift to select multiple
- Hold Option for alternates

• ◦

UIView ◦ 8◦



- ◦ 400◦
- ◦ 300◦
- ◦ 268◦

◦

————

◦ ◦ ◦

————

- [iOSScrollViewAutoLayout](#)
- [Interface BuilderUIScrollView](#)
- [YouTube UIScrollView -](#)

/

scrollEnabledBoolean◦

true / YES◦ true

```
scrollview.isScrollEnabled = true
```

## Objective-C

```
scrollview.scrollEnabled = YES;
```

## /UIImageView

## UIScrollView

```
let scrollView = UIScrollView.init(frame: self.view.bounds)
```

```
scrollView.minimumZoomScale = 0.1
scrollView.maximumZoomScale = 4.0
scrollView.zoomScale = 1.0
scrollView.delegate = self as? UIScrollViewDelegate
```

◦ `minimumZoomScale` `maximumZoomScale` ◦ **1.0**◦

`zoomScale` **1.0**◦

◦ `UIScrollViewDelegate` ◦ `viewForZoomingInScrollView()`◦

## ViewController

```
class ViewController: UIViewController, UIScrollViewDelegate
```

◦

```
func viewForZoomingInScrollView(scrollView: UIScrollView) -> UIView? {
 return imageView
}
```

---

# UIImageView

```
var imageView: UIImageView = UIImageView.init(image: UIImage.init(named: "someImage.jpg"))
```

## scrollView

```
scrollView?.addSubview(imageView)
```

- [iOS](#)
- [UIScrollView](#)

## UIScrollView

### `scrollViewDidEndDecelerating`◦

## C

```
- (void)scrollViewDidEndDecelerating:(UIScrollView *)scrollView {
 [self stoppedScrolling];
}

- (void)scrollViewDidEndDragging:(UIScrollView *)scrollView willDecelerate:(BOOL)decelerate {
```

```

 if (!decelerate) {
 [self stoppedScrolling];
 }
}

- (void)stoppedScrolling {
 // done, do whatever
}

```

```

func scrollViewDidEndDragging(scrollView: UIScrollView, willDecelerate decelerate: Bool) {
 if !decelerate {
 stoppedScrolling()
 }
}

func scrollViewDidEndDecelerating(scrollView: UIScrollView) {
 stoppedScrolling()
}

func stoppedScrolling() {
 // done, do whatever
}

```

```

func scrollViewDidScroll(_ scrollView: UIScrollView) {
 if scrollView.contentOffset.x != 0 {
 scrollView.contentOffset.x = 0
 }
}

```

## xscrollView0.

x Sy S.

scrollViewDidScroll(\_ scrollView: UIScrollView). .

UIScrollViewDelegate

```

class ViewController: UIViewController, UIScrollViewDelegate

```

...scrollViewselfviewDidLoad(\_:)

```

scrollView.delegate = self

```

**UIScrollView** <https://riptutorial.com/zh-TW/ios/topic/1575/uiscrollview>

# 89: UISearchController

- UISearchController.searchResultsController.UIViewController//nil。
- func updateSearchResults(searchController:UISearchController//UISearchResultsUpdating

|                                                         |   |
|---------------------------------------------------------|---|
| UISearchController.searchBar                            | o |
| UISearchController.searchResultsUpdater                 | o |
| UISearchController.isActive                             | o |
| UISearchController.obscuresBackgroundDuringPresentation | o |
| UISearchController.dimsBackgroundDuringPresentation     | o |
| UISearchController.hidesNavigationBarDuringPresentation | o |
| UIViewController.definesPresentationContext             | o |
| UIViewController.navigationItem.titleView               | o |
| UITableViewController.tableView.tableHeaderView         | o |

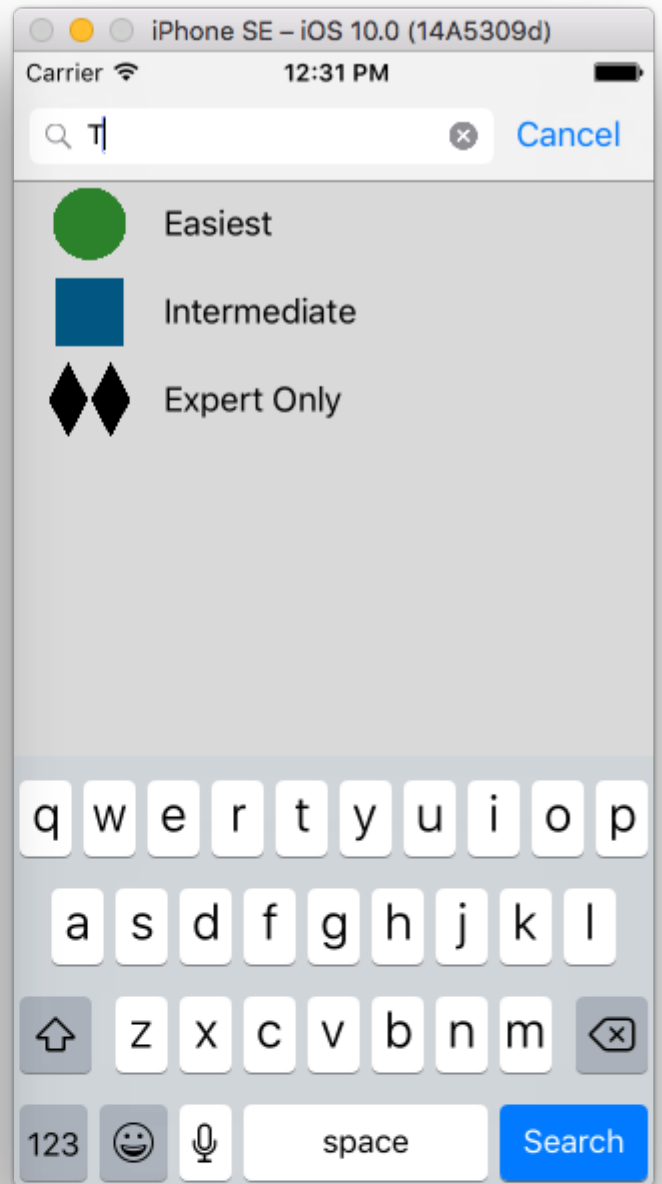
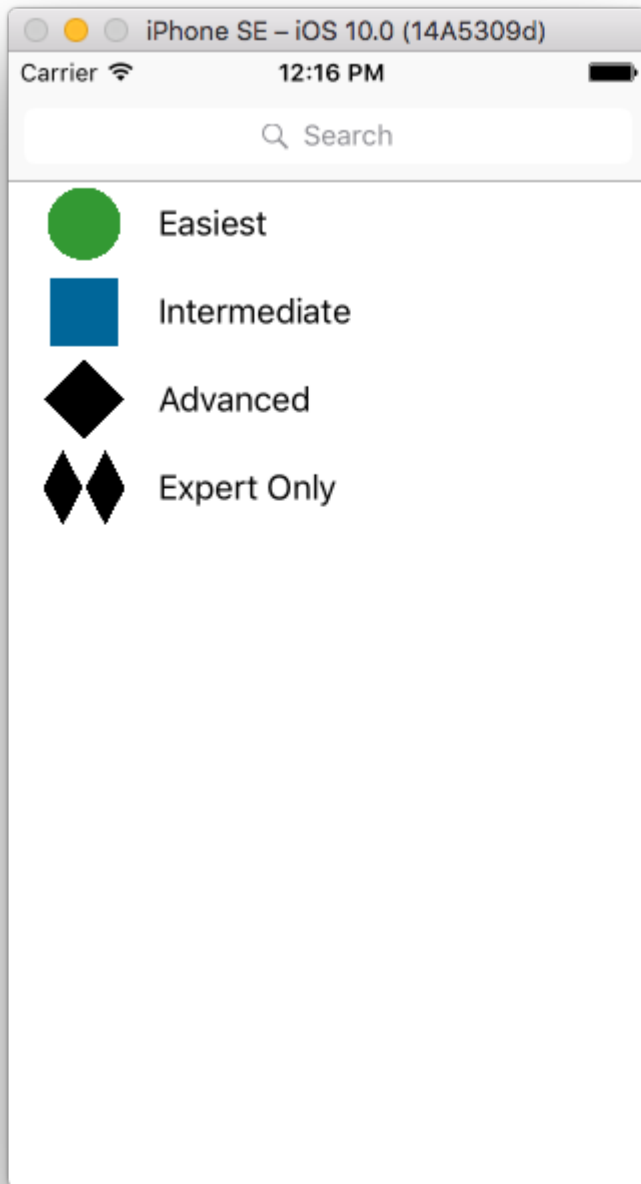
UIKit

[UISearchController](#)

[UISearchResultsUpdating](#)

## Examples

o o



```

class ViewController: UITableViewController, UISearchResultsUpdating {

 let entries = [(title: "Easiest", image: "green_circle"),
 (title: "Intermediate", image: "blue_square"),
 (title: "Advanced", image: "black_diamond"),
 (title: "Expert Only", image: "double_black_diamond")]

 // An empty tuple that will be updated with search results.
 var searchResults : [(title: String, image: String)] = []

 let searchController = UISearchController(searchResultsController: nil)

 override func viewDidLoad() {
 super.viewDidLoad()

 searchController.searchResultsUpdater = self
 self.definesPresentationContext = true

 // Place the search bar in the navigation item's title view.
 self.navigationItem.titleView = searchController.searchBar

 // Don't hide the navigation bar because the search bar is in it.
 searchController.hidesNavigationBarDuringPresentation = false
 }

 func filterContent(for searchText: String) {
 // Update the searchResults array with matches
 // in our entries based on the title value.
 searchResults = entries.filter({ (title: String, image: String) -> Bool in
 let match = title.range(of: searchText, options: .caseInsensitive)
 // Return the tuple if the range contains a match.
 return match != nil
 })
 }

 // MARK: - UISearchResultsUpdating method

 func updateSearchResults(for searchController: UISearchController) {
 // If the search bar contains text, filter our data with the string
 if let searchText = searchController.searchBar.text {
 filterContent(for: searchText)
 // Reload the table view with the search result data.
 tableView.reloadData()
 }
 }

 // MARK: - UITableViewController methods

 override func numberOfSections(in tableView: UITableView) -> Int { return 1 }

 override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) ->
 Int {
 // If the search bar is active, use the searchResults data.
 return searchController.isActive ? searchResults.count : entries.count
 }

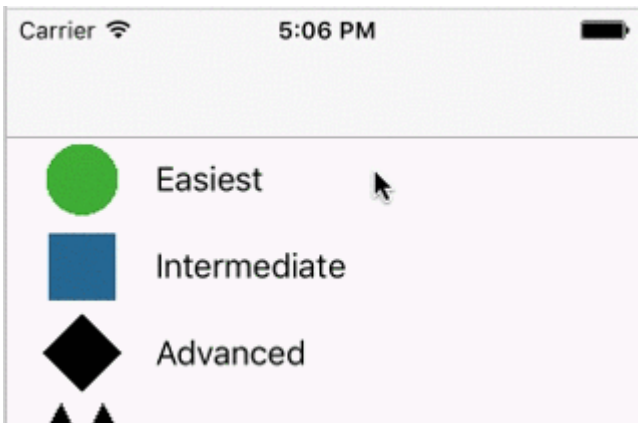
 override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
 UITableViewCell {
 // If the search bar is active, use the searchResults data.

```

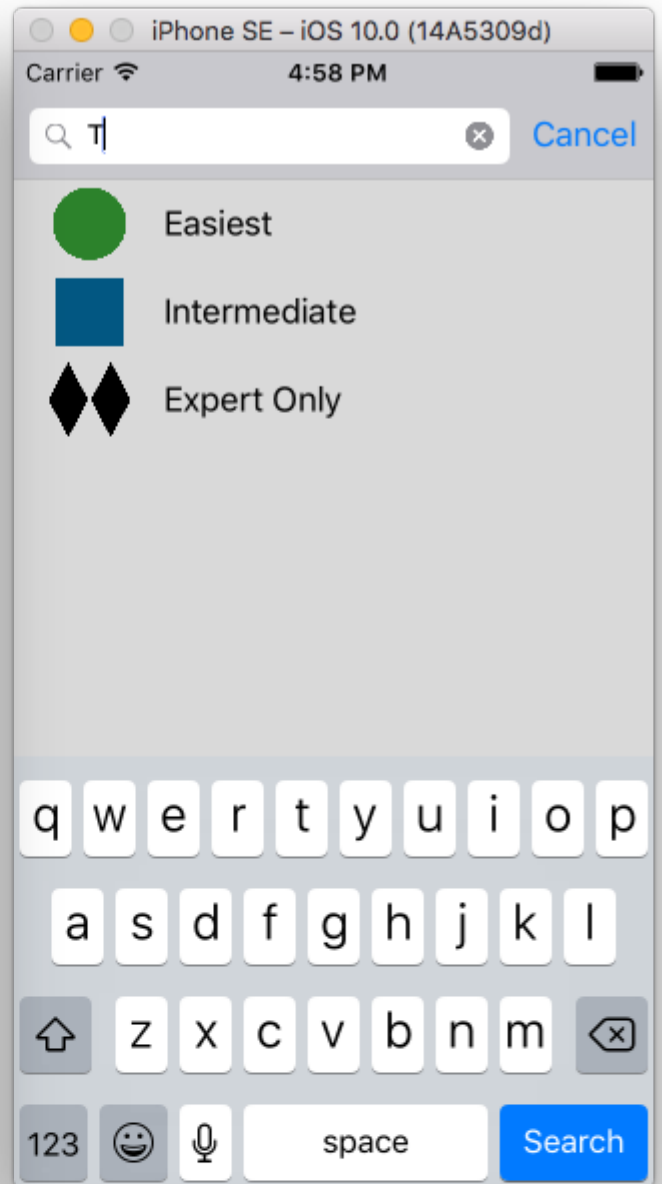
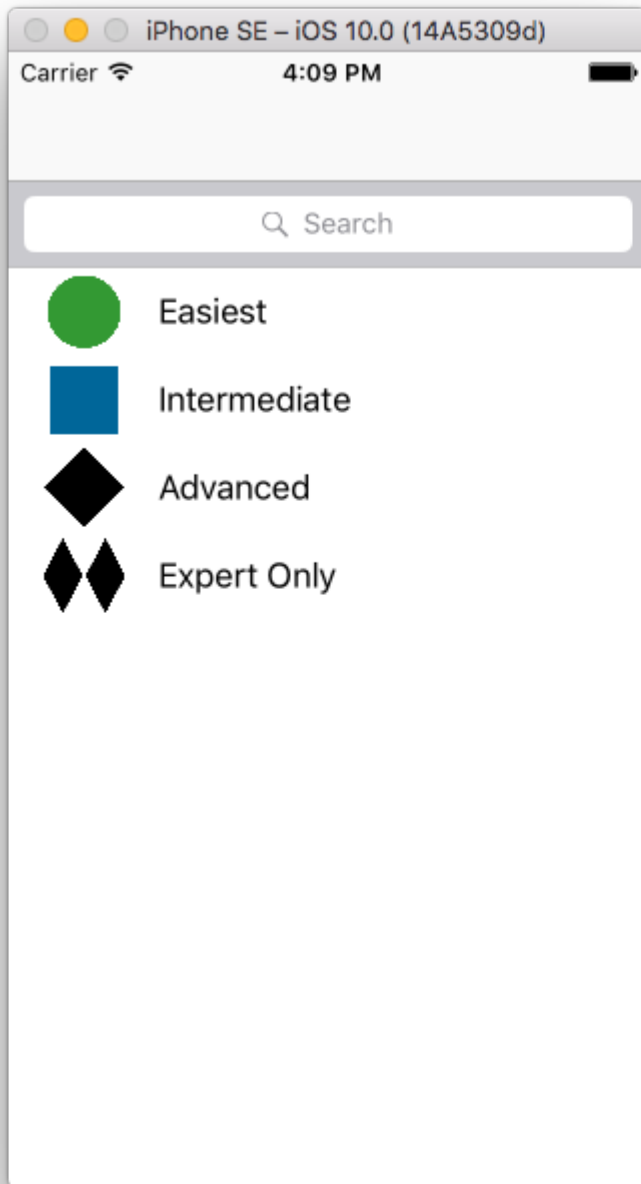
```
let entry = searchController.isActive ?
 searchResults[indexPath.row] : entries[indexPath.row]

let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)
cell.textLabel?.text = entry.title
cell.imageView?.image = UIImage(named: entry.image)
return cell
}
}
```

o o o o o







## UITableViewControllerUISearchResultsUpdating

```
class ViewController: UITableViewController, UISearchResultsUpdating {

 let entries = [(title: "Easiest", image: "green_circle"),
 (title: "Intermediate", image: "blue_square"),
 (title: "Advanced", image: "black_diamond"),
 (title: "Expert Only", image: "double_black_diamond")]

 // An empty tuple that will be updated with search results.
 var searchResults : [(title: String, image: String)] = []

 let searchController = UISearchController(searchResultsController: nil)

 override func viewDidLoad() {
 super.viewDidLoad()

 searchController.searchResultsUpdater = self
 self.definesPresentationContext = true

 // Place the search bar in the table view's header.
 self.tableView.tableHeaderView = searchController.searchBar

 // Set the content offset to the height of the search bar's height
 // to hide it when the view is first presented.
 self.tableView.contentOffset = CGPoint(x: 0, y:
searchController.searchBar.frame.height)
 }

 func filterContent(for searchText: String) {
 // Update the searchResults array with matches
 // in our entries based on the title value.
 searchResults = entries.filter({ (title: String, image: String) -> Bool in
 let match = title.range(of: searchText, options: .caseInsensitive)
 // Return the tuple if the range contains a match.
 return match != nil
 })
 }

 // MARK: - UISearchResultsUpdating method

 func updateSearchResults(for searchController: UISearchController) {
 // If the search bar contains text, filter our data with the string
 if let searchText = searchController.searchBar.text {
 filterContent(for: searchText)
 // Reload the table view with the search result data.
 tableView.reloadData()
 }
 }

 // MARK: - UITableViewController methods

 override func numberOfSections(in tableView: UITableView) -> Int { return 1 }

 override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) ->
Int {
 // If the search bar is active, use the searchResults data.
 return searchController.isActive ? searchResults.count : entries.count
 }

 override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
```

```

UITableViewCell {
 // If the search bar is active, use the searchResults data.
 let entry = searchController.isActive ?
 searchResults[indexPath.row] : entries[indexPath.row]

 let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)
 cell.textLabel?.text = entry.title
 cell.imageView?.image = UIImage(named: entry.image)
 return cell
}
}

```

UISearchResultsUpdating

```

class MyTableViewController: UITableViewController, UISearchResultsUpdating {}

```

```

class MyTableViewController: UITableViewController, UISearchResultsUpdating {
 let searchController = UISearchController(searchResultsController: nil)
}

```

```

override func viewDidLoad() {
 super.viewDidLoad()

 searchController.searchResultsUpdater = self
 searchController.hidesNavigationBarDuringPresentation = false
 searchController.dimsBackgroundDuringPresentation = false
 searchController.searchBar.sizeToFit()
 self.tableView.tableHeaderView = searchController.searchBar
}

```

UISearchResultsUpdatingupdateSearchResultsForSearchController

```

func updateSearchResultsForSearchController(searchController: UISearchController) {
}

```

## Objective-CUISerachController

```

Delegate: UISearchBarDelegate, UISearchControllerDelegate, UISearchBarDelegate

@property (strong, nonatomic) UISearchController *searchController;

- (void)searchBarConfiguration
{
 self.searchController = [[UISearchController alloc] initWithSearchResultsController:nil];
 self.searchController.searchBar.delegate = self;
 self.searchController.hidesNavigationBarDuringPresentation = NO;

 // Hides search bar initially. When the user pulls down on the list, the search bar is
 revealed.
 [self.tableView setContentOffset:CGPointMake(0,
self.searchController.searchBar.frame.size.height)];

 self.searchController.searchBar.backgroundColor = [UIColor DarkBlue];
 self.searchController.searchBar.tintColor = [UIColor DarkBlue];
}

```

```

 self.tableView.contentOffset = CGPointMake(0,
CGRectGetHeight(_searchController.searchBar.frame));
 self.tableView.tableHeaderView = _searchController.searchBar;
 _searchController.searchBar.delegate = self;
 _searchController.searchBar.showsCancelButton = YES;
 self.tapGestureRecognizer = [[UITapGestureRecognizer alloc] initWithTarget:self
action:@selector(resetSearchbarAndTableView)];
 [self.view addGestureRecognizer:self.tapGestureRecognizer];
 }

- (void)resetSearchbarAndTableView{
// Reload your tableview and resign keyboard.
}

- (void)searchBarCancelButtonClicked:(UISearchBar *)searchBar{
// Search cancelled
}
- (void)searchBarSearchButtonClicked:(UISearchBar *)searchBar{
// Implement filtration of your data as per your need using NSPredicate or else.
// then reload your data control like Tableview.
}

```

**UISearchBar** <https://riptutorial.com/zh-TW/ios/topic/2813/uisearchcontroller>

# 90: UISegmentedControl

UISegmentedControl。

## Examples

### UISegmentedControl

#### 1. UISegmentedControl3

```
let mySegmentedControl = UISegmentedControl (items: ["One", "Two", "Three"])
```

2. ;

```
mySegmentedControl.frame = CGRect(x: 0.0, y: 0.0, width: 300, height: 50)
```

3. 0

```
mySegmentedControl.selectedSegmentIndex = 0
```

4.

```
mySegmentedControl.addTarget(self, action: #selector(segmentedValueChanged(_:)), for: .valueChanged)
```

5

```
func segmentedValueChanged(_ sender:UISegmentedControl!) {
 print("Selected Segment Index is : \(sender.selectedSegmentIndex)")
}
```

#### 6. UISegmentedControl

```
yourView.addSubview(mySegmentedControl)
```

[UISegmentedControl](https://riptutorial.com/zh-TW/ios/topic/9963/uisegmentedcontrol) <https://riptutorial.com/zh-TW/ios/topic/9963/uisegmentedcontrol>

# 91: UISlider

## Examples

### UISlider

#### Objective-C

ViewController.hViewController.m

```
@property (strong, nonatomic)UISlider *slider;

//Define frame of slider and add to view
CGRect frame = CGRectMake(0.0, 100.0, 320.0, 10.0);
UISlider *slider = [[UISlider alloc] initWithFrame:frame];
[slider addTarget:self action:@selector(sliderAction:)
forControlEvents:UIControlEventValueChanged];
[self.slider setBackgroundColor:[UIColor clearColor]];
self.slider.minimumValue = 0.0;
self.slider.maximumValue = 50.0;
//sending a NO/False would update the value of slider only when the user is no longer touching
the screen. Hence sending only the final value
self.slider.continuous = YES;
self.slider.value = 25.0;
[self.view addSubview slider];
```

```
- (IBAction)sliderAction:(id)sender {
 NSLog(@"Slider Value %f", sender.value);
}
```

### SWIFT

```
let frame = CGRect(x: 0, y: 100, width: 320, height: 10)
let slider = UISlider(frame: frame)
slider.addTarget(self, action: #selector(sliderAction), for: .valueChanged)
slider.backgroundColor = .clear
slider.minimumValue = 0.0
slider.maximumValue = 50.0
//sending a NO/False would update the value of slider only when the user is no longer
touching the screen. Hence sending only the final value
slider.isContinuous = true
slider.value = 25.0
view.addSubview(slider)
```

```
func sliderAction(sender:UISlider!)
{
 print("value--\(sender.value)")
}
```

### [setThumbImage](#)

## Swift 3.1

```
let slider = UISlider()
let thumbImage = UIImage
slider.setThumbImage(thumbImage, for: .normal)
```

**UISlider** <https://riptutorial.com/zh-TW/ios/topic/7402/uislider>

# 92: UISplitViewController

UISplitViewControllerUITabViewController UINavigationController ◦ masterViewController  
PrimaryViewControllerdetailViewControllerSecondaryViewController ◦ AppleUISplitViewController  
rootviewController ◦ viewcontrollersNSNotificationCenter ◦

## Examples

### Objective CDelegatesMasterDetail View

UISplitViewControllerrootViewController ◦

#### AppDelegate.m

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
 // Override point for customization after application launch.
 self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]]
 self.window.backgroundColor = [UIColor blackColor];
 [self.window makeKeyAndVisible];
 self.window.clipsToBounds = YES;
 SplitViewController *spView = [[SplitViewController alloc]init];
 self.window.rootViewController = spView;
 [self.window makeKeyAndVisible];
 return YES;
}
```

UISplitViewControllerviewControllerrootviewController ◦

#### SplitViewController.h

```
#import <UIKit/UIKit.h>
#import "MasterViewController.h"
#import "DetailViewController.h"
@interface ViewController : UISplitViewController
{
 DetailViewController *detailVC;
 MasterViewController *masterVC;
 NSMutableArray *array;
}
@end
```

MasterViewControllerUISplitViewControllerDetailViewController

#### SplitViewController.m

```
#import "ViewController.h"
#define ANIMATION_LENGTH 0.3
@interface ViewController ()
@end
```



```

@implementation ViewController
- (void)viewDidLoad
{
 [super viewDidLoad];
 masterVC = [[MasterViewController alloc] init];
 detailVC = [[DetailViewController alloc] init];
 [masterVC setDetailDelegate:(id)detailVC];
 NSArray *vcArray = [NSArray arrayWithObjects:masterVC, detailVC, nil];
 self.preferredDisplayMode = UISplitViewControllerDisplayModeAutomatic;
 self.viewControllers = vcArray;
 self.delegate = (id)self;
 self.presentsWithGesture = YES;
}

```

ViewControllers self.viewControllers UISplitViewController ◦ self.preferredDisplayMode  
 DisplayMode masterDetailViewController Apple ◦ self.presentsWithGesture MasterViewController

## MasterViewController.h

```

#import <UIKit/UIKit.h>

@protocol DetailViewDelegate <NSObject>
@required
- (void)sendSelectedNavController:(UIViewController *)viewController;
@end

@interface MasterViewController : UIViewController
{
 UITableView *mainTableView;
 NSMutableArray *viewControllerArray;
}
@property (nonatomic, retain) id<DetailViewDelegate> detailDelegate;
@end

```

sendSelectedNavController:(UIViewController \*)viewController DetailViewDelegate  
 sendSelectedNavController:(UIViewController \*)viewController UIViewController DetailViewController  
 ◦ MasterViewController mainTableView leftside tableView ◦ viewControllerArray DetailViewController  
 UIViewControllers

## MasterViewController.m

```

#import "MasterViewController.h"

@implementation MasterViewController
@synthesize detailDelegate;

- (void)viewDidLoad
{
 [super viewDidLoad];

 UIViewController *dashBoardVC = [[UIViewController alloc] init];
 [dashBoardVC.view setBackgroundColor:[UIColor redColor]];
 UIViewController *inventVC = [[UIViewController alloc] init];
 [inventVC.view setBackgroundColor:[UIColor whiteColor]];
 UIViewController *alarmVC = [[UIViewController alloc] init];
 [alarmVC.view setBackgroundColor:[UIColor purpleColor]];
}

```

```

UIViewController *scanDeviceVC = [[UIViewController alloc] init];
[scanDeviceVC.view setBackgroundColor:[UIColor cyanColor]];
UIViewController *serverDetailVC = [[UIViewController alloc] init];
[serverDetailVC.view setBackgroundColor:[UIColor whiteColor]];
viewControllerArray = [[NSMutableArray
alloc] initWithObjects:dashboardVC, inventVC, alarmVC, scanDeviceVC, serverDetailVC, nil];
mainTableView = [[UITableView alloc] initWithFrame:CGRectMake(0, 50, self.view.frame.size.width,
self.view.frame.size.height-50) style:UITableViewStylePlain];
[mainTableView setDelegate:(id)self];
[mainTableView setDataSource:(id)self];
[mainTableView setSeparatorStyle:UITableViewCellStyleNone];
[mainTableView setScrollsToTop:NO];
[self.view addSubview:mainTableView];
}

- (CGFloat)tableView:(UITableView *)tableView
heightForRowAtIndexPath:(NSIndexPath *)indexPath
{
 return 100;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
 return [viewControllerArray count];
}

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
 return 1; //count of section
}

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
 NSString *cellId = [NSString
stringWithFormat:@"Cell%i%i", (long)indexPath.section, (long)indexPath.row];
UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:cellId];

if (cell == nil)
{
 cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:cellId];
}
[cell.contentView setBackgroundColor:[UIColor redColor]];
cell.textLabel.text = [NSString stringWithFormat:@"My VC at index %ld", (long)indexPath.row];
return cell;
}

- (void)tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
 [detailDelegate sendSelectedNavController:[viewControllerArray
objectAtIndex:indexPath.row]];
}

@end

```

UITableViewController\* Table didSelectRowAtIndexPath detailDelegate UIViewController  
DetailViewController UIViewController

## DetailViewController.h

```
#import <UIKit/UIKit.h>

@interface DetailViewController : UIViewController<UICollectionViewDelegate>
{
 UINavigationController *tempNav;
}
@end
```

## DetailViewController.m

```
#import "DetailViewController.h"

@implementation DetailViewController
-(void)viewDidLoad
{
 [super viewDidLoad];
 [self.view setBackgroundColor:[UIColor whiteColor]];
}
-(void)sendSelectedNavController:(UIViewController *)navController
{
 NSArray *viewsToRemove = [self.view subviews];
 for (UIView *v in viewsToRemove) {
 [v removeFromSuperview];
 }
 tempNav = navController;
 [self.view addSubview:tempNav.view];
}
@end
```

sendSelectedNavController DetailViewControllerMasterViewControllerUIViewController





Carrier 

8:26 PM

My VC at index 0

My VC at index 1

My VC at index 2

My VC at index 3

My VC at index 4



My VC at index 0

My VC at index 1

My VC at index 2

My VC at index 3

My VC at index 4





```

@end

@implementation ViewController
- (void)viewDidLoad
{
 [super viewDidLoad];
 masterVC = [[MasterViewController alloc] init];
 detailVC = [[DetailViewController alloc] init];
 [masterVC setDetailDelegate:(id)detailVC];
 NSArray *vcArray = [NSArray arrayWithObjects:masterVC, detailVC, nil];
 self.preferredDisplayMode = UISplitViewControllerDisplayModeAutomatic;
 self.viewControllers = vcArray;
 self.delegate = (id)self;
 self.presentsWithGesture = YES;
}

```

UINavigationControllerNSArray NSArrayself.viewControllers ◦ self.preferredDisplayMode  
 MasterViewControllerDetailViewController◦ self.presentsWithGestureMasterViewController

## MasterViewController.h

```

#import <UIKit/UIKit.h>

@protocol DetailViewDelegate <NSObject>
@required
- (void)sendSelectedNavController:(UIViewController *)viewController;
@end

@interface MasterViewController : UIViewController
{
 UITableView *mainTableView;
 NSMutableArray *viewControllerArray;
}
@property (nonatomic, retain) id<DetailViewDelegate> detailDelegate;
@end

```

sendSelectedNavControllerDetailViewDelegateUIViewControllerDetailViewController ◦  
 MasterViewControllerUITableView ◦ ViewControllerArrayDetailViewViewControllerUITableViewController

## MasterViewController.m

```

#import "MasterViewController.h"

@implementation MasterViewController
@synthesize detailDelegate;

- (void)viewDidLoad
{
 [super viewDidLoad];

 UIViewController *dashBoardVC = [[UIViewController alloc] init];
 [dashBoardVC.view setBackgroundColor:[UIColor redColor]];
 UIViewController *inventVC = [[UIViewController alloc] init];
 [inventVC.view setBackgroundColor:[UIColor whiteColor]];
 UIViewController *alarmVC = [[UIViewController alloc] init];
 [alarmVC.view setBackgroundColor:[UIColor purpleColor]];
 UIViewController *scanDeviceVC = [[UIViewController alloc] init];

```

```

[scanDeviceVC.view setBackgroundColor:[UIColor cyanColor]];
UIViewController *serverDetailVC = [[UIViewController alloc]init];
[serverDetailVC.view setBackgroundColor: [UIColor whiteColor]];
viewControllerArray = [[NSMutableArray
alloc] initWithObjects:dashboardVC, inventVC, alarmVC, scanDeviceVC, serverDetailVC, nil];
mainTableView = [[UITableView alloc] initWithFrame:CGRectMake(0, 50, self.view.frame.size.width,
self.view.frame.size.height-50) style:UITableViewStylePlain];
[mainTableView setDelegate:(id)self];
[mainTableView setDataSource:(id)self];
[mainTableView setSeparatorStyle:UITableViewCellStyleNone];
[mainTableView setScrollsToTop:NO];
[self.view addSubview:mainTableView];
}

- (CGFloat)tableView:(UITableView *)tableView
heightForRowAtIndexPath:(NSIndexPath *)indexPath
{
 return 100;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection: (NSInteger) section
{
 return [viewControllerArray count];
}

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
 return 1; //count of section
}

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
 NSString *cellId = [NSString
stringWithFormat:@"Cell%i%i", (long)indexPath.section, (long)indexPath.row];
UITableViewCell *cell =[tableView dequeueReusableCellWithIdentifier:cellId];

if (cell == nil)
{
 cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:cellId];
}
[cell.contentView setBackgroundColor:[UIColor redColor]];
cell.textLabel.text =[NSString stringWithFormat:@"My VC at index %ld", (long)indexPath.row];
return cell;
}

- (void)tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
 [detailDelegate sendSelectedNavController:[viewControllerArray
objectAtIndex:indexPath.row]];
}

@end

```

UIViewControllerNSMutableArray ◦ UITableViewdidselectrowatindexpathdetailDelegate  
UIViewControllerDetailViewController NSMutableArrayUIViewController

## DetailViewController.h

```
#import <UIKit/UIKit.h>

@interface DetailViewController : UIViewController<UICollectionViewDelegate>
{
 UINavigationController *tempNav;
}
@end
```

## DetailViewController.m

```
#import "DetailViewController.h"

@implementation DetailViewController
-(void)viewDidLoad
{
 [super viewDidLoad];
 [self.view setBackgroundColor:[UIColor whiteColor]];
}
-(void)sendSelectedNavController:(UIViewController *)navController
{
 NSArray *viewsToRemove = [self.view subviews];
 for (UIView *v in viewsToRemove) {
 [v removeFromSuperview];
 }
 tempNav = navController;
 [self.view addSubview:tempNav.view];
}
@end
```

sendSelectedNavController DetailViewControllerUIView MasterViewControllerUIViewController ◦

**UISplitViewController** <https://riptutorial.com/zh-TW/ios/topic/4844/uisplitviewController>

# 94: UIStackView

## Examples

3

```
let stackView = UIStackView()
stackView.axis = .horizontal
stackView.alignment = .fill // .leading .firstBaseline .center .trailing .lastBaseline
stackView.distribution = .fill // .fillEqually .fillProportionally .equalSpacing
.equalCentering

let label = UILabel()
label.text = "Text"
stackView.addArrangedSubview(label)
// for horizontal stack view, you might want to add width constraint to label or whatever view
you're adding.
```

```
let stackView = UIStackView()
stackView.axis = .Horizontal
stackView.alignment = .Fill // .Leading .FirstBaseline .Center .Trailing .LastBaseline
stackView.distribution = .Fill // .FillEqually .FillProportionally .EqualSpacing
.EqualCentering

let label = UILabel(frame: CGRectZero)
label.text = "Label"
stackView.addArrangedSubview(label)
// for horizontal stack view, you might want to add width constraint to label or whatever view
you're adding.
```

## Objective-C

```
UIStackView *stackView = [[UIStackView alloc] init];
stackView.axis = UILayoutConstraintAxisHorizontal;
stackView.alignment = UIStackViewAlignmentFill; //UIStackViewAlignmentLeading,
UIStackViewAlignmentFirstBaseline, UIStackViewAlignmentCenter, UIStackViewAlignmentTrailing,
UIStackViewAlignmentLastBaseline
stackView.distribution = UIStackViewDistributionFill; //UIStackViewDistributionFillEqually,
UIStackViewDistributionFillProportionally, UIStackViewDistributionEqualSpacing,
UIStackViewDistributionEqualCentering

UILabel *label = [[UILabel alloc] initWithFrame:CGRectZero];
label.text = @"Label";
[stackView addArrangedSubview:label];
//For horizontal stack view, you might want to add a width constraint to your label or
whatever view you are adding.
```

```
let stackView = UIStackView()
stackView.axis = .Vertical
stackView.alignment = .Fill // .Leading .FirstBaseline .Center .Trailing .LastBaseline
stackView.distribution = .Fill // .FillEqually .FillProportionally .EqualSpacing
.EqualCentering

let label = UILabel(frame: CGRectZero)
```

```
label.text = "Label"
stackView.addArrangedSubview(label)
// for vertical stack view, you might want to add height constraint to label or whatever view
you're adding.
```

## Objective-C

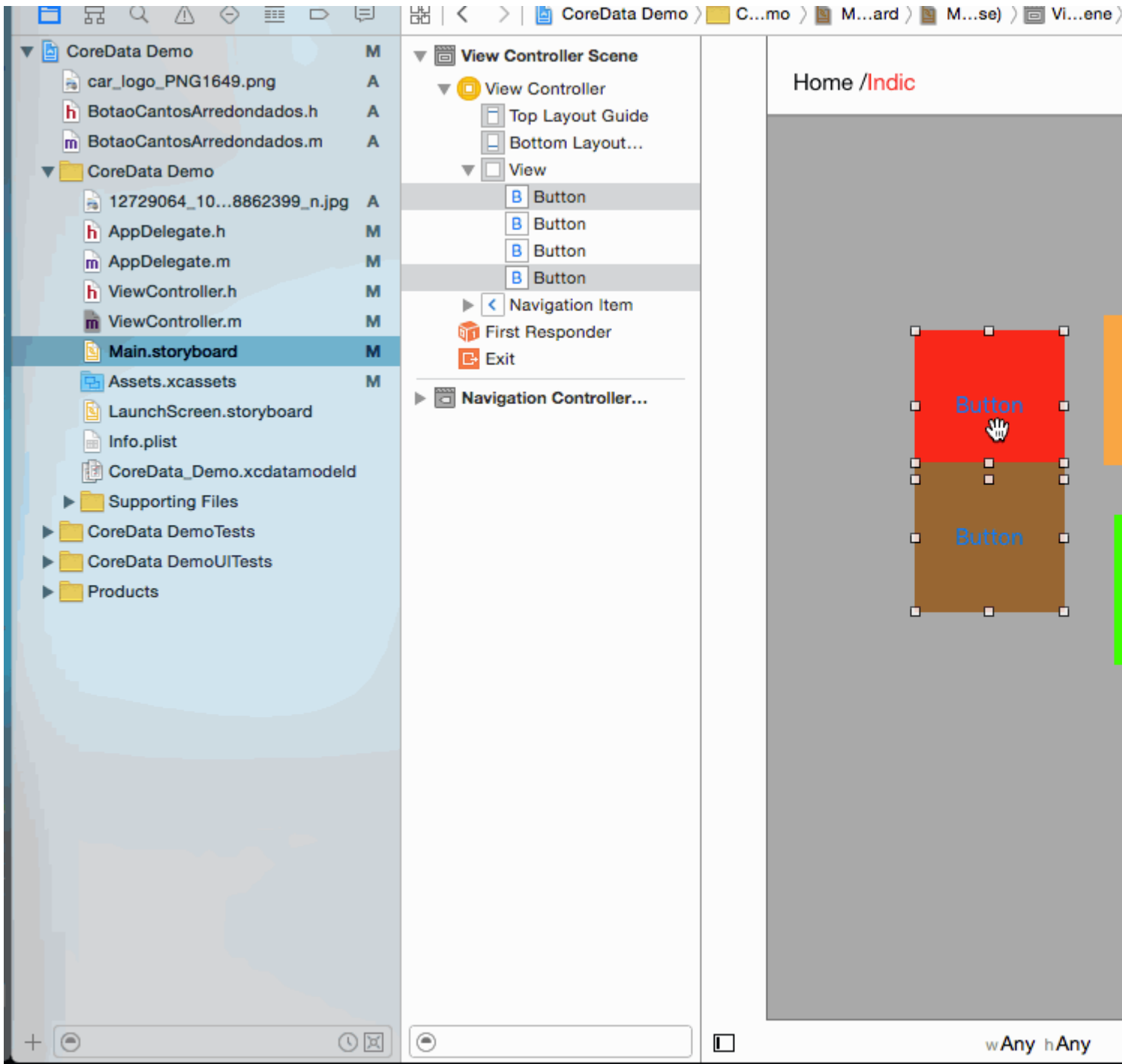
```
UIStackView *stackView = [[UIStackView alloc] init];
stackView.axis = UILayoutConstraintAxisVertical;
stackView.alignment = UIStackViewAlignmentFill; //UIStackViewAlignmentLeading,
UIStackViewAlignmentFirstBaseline, UIStackViewAlignmentCenter, UIStackViewAlignmentTrailing,
UIStackViewAlignmentLastBaseline
stackView.distribution = UIStackViewDistributionFill; //UIStackViewDistributionFillEqually,
UIStackViewDistributionFillProportionally, UIStackViewDistributionEqualSpacing,
UIStackViewDistributionEqualCentering

UILabel *label = [[UILabel alloc] initWithFrame:CGRectZero];
label.text = @"Label";
[stackView addArrangedSubview:label];
//For vertical stack view, you might want to add a height constraint to your label or whatever
view you are adding.
```

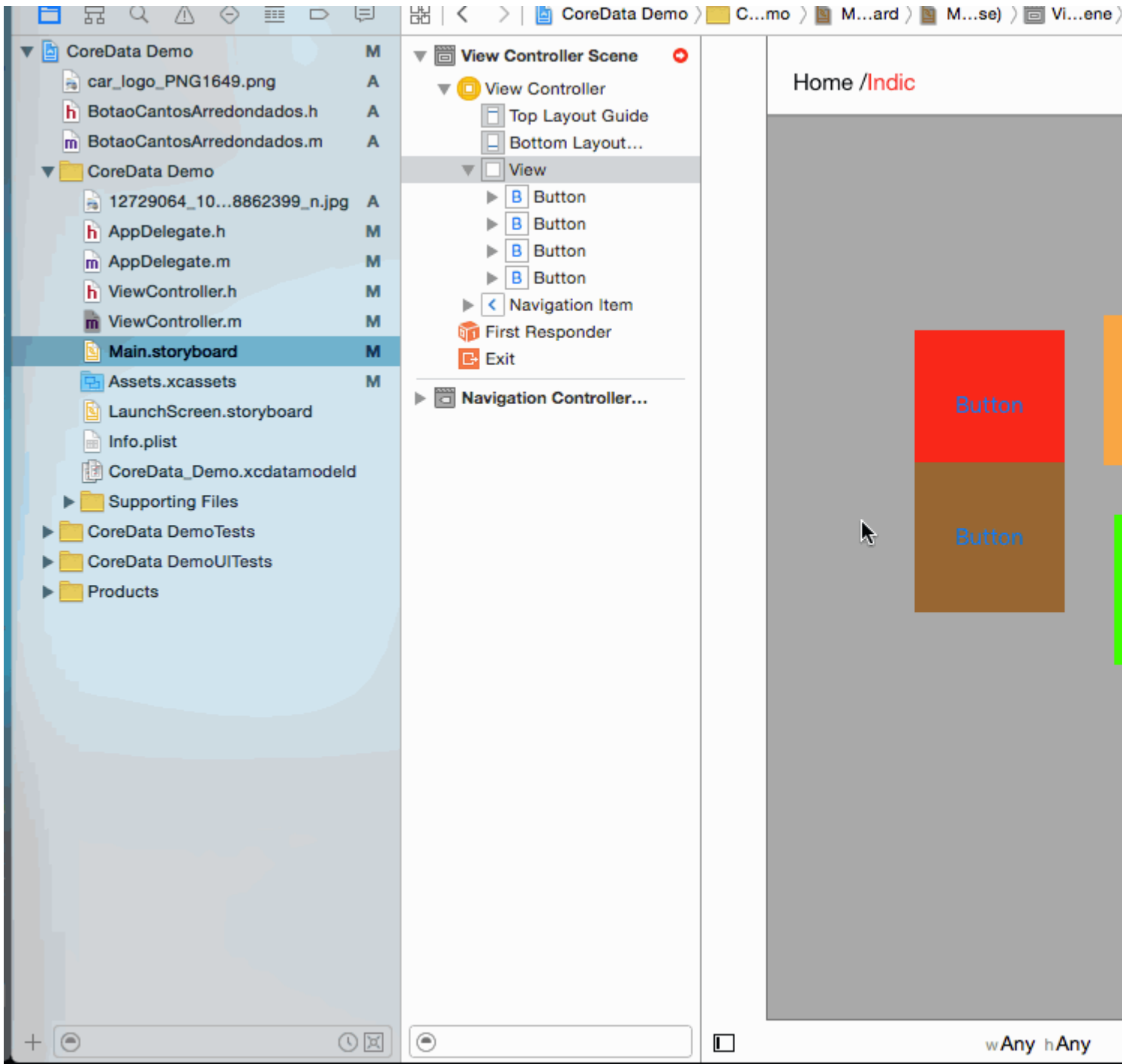
## UIStackview

1 -4。 Button1Button2Button 3Button4

2 -。



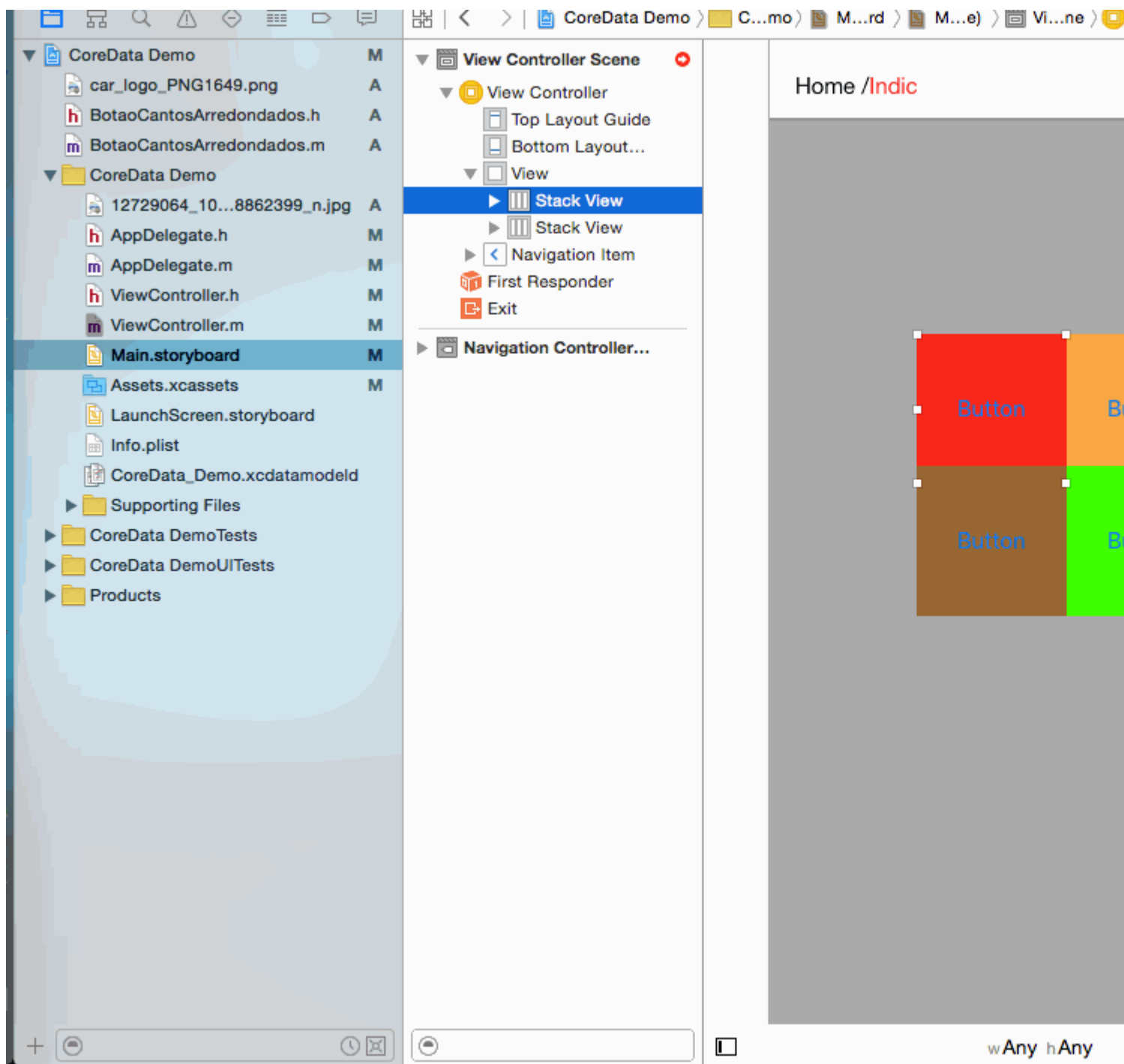
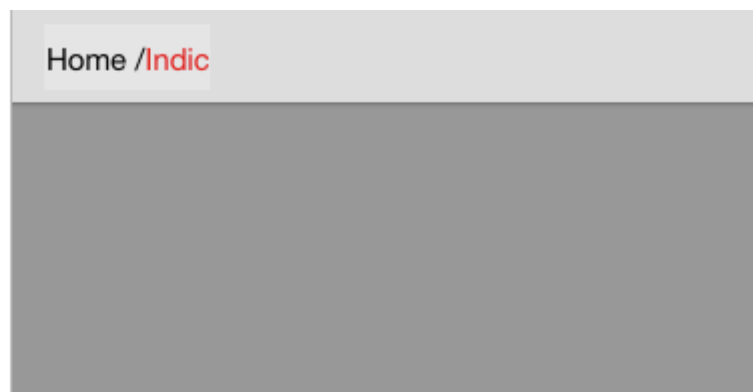
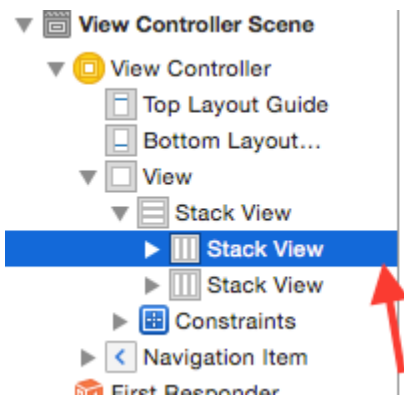
3 - 22 - 2。



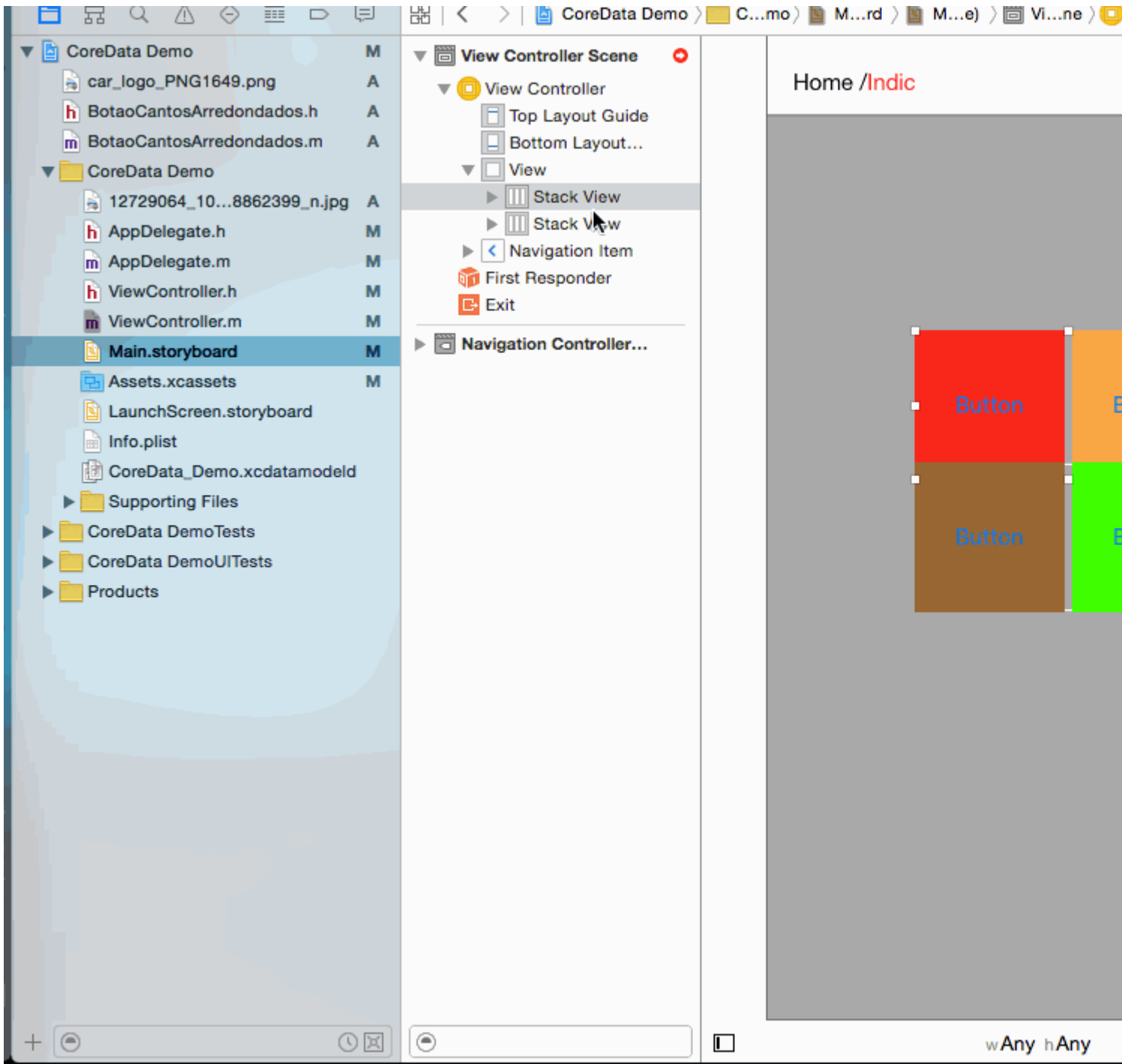
#### 4 -UIStackview◦

```
Distribution -> Fill Equally
Spacing -> 5 (as per your requirement)
```

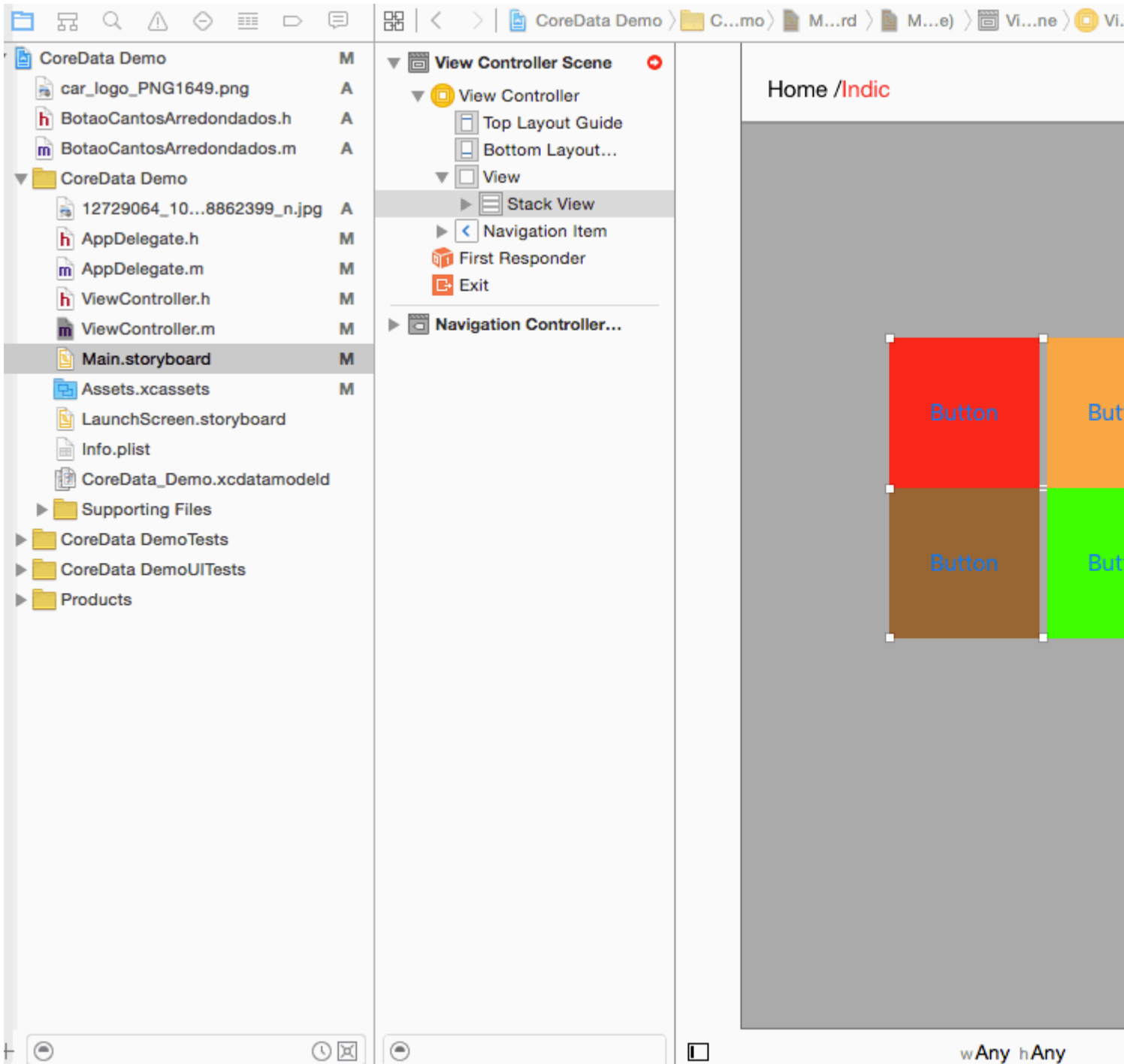




## 5 -StackviewStackview

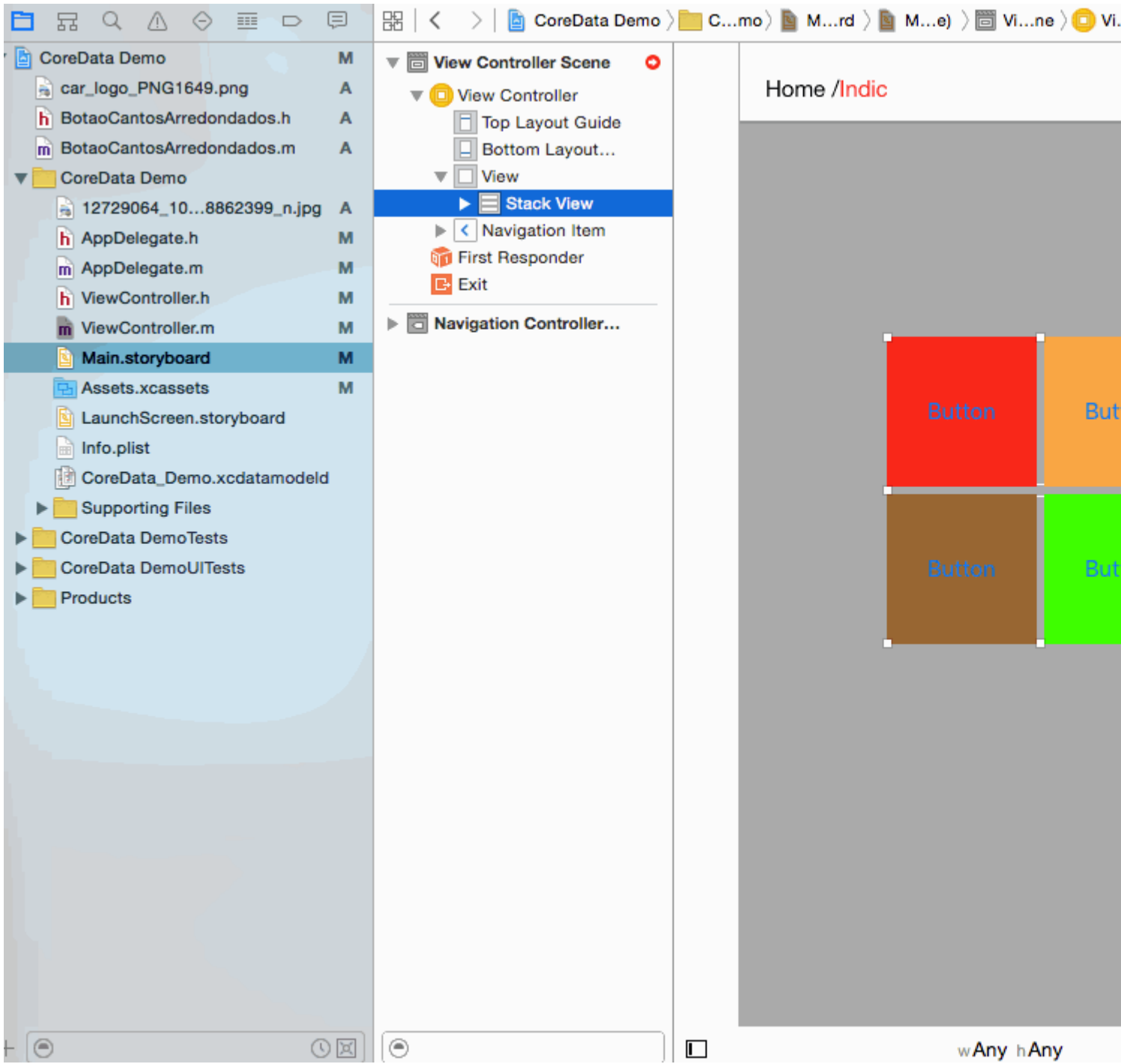


6 -Distribution = Fill equally Spacing =5Distribution = Fill equally Spacing =5

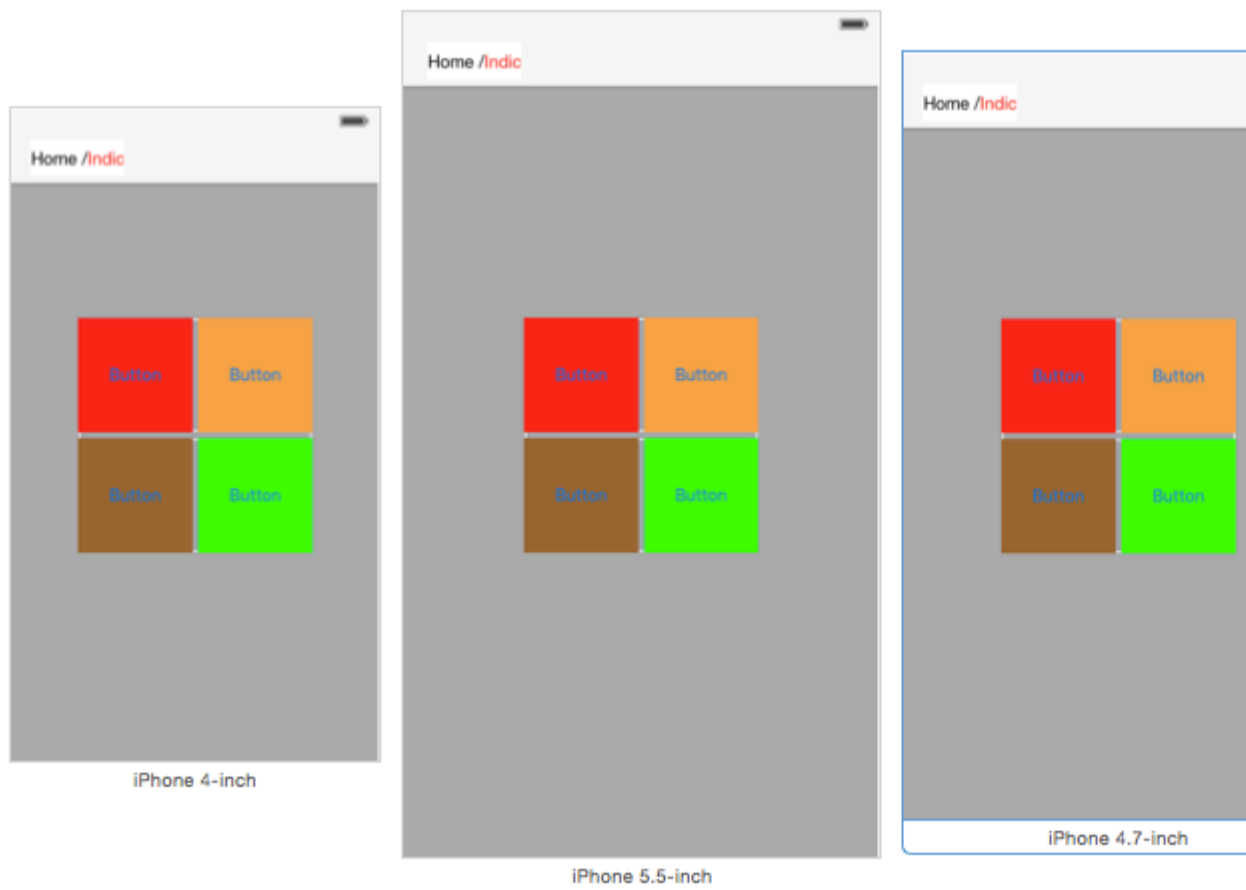


## 7 -Constrain

```
center Horizontally in container
center vertically in container
and select Update Frame.
```



8 -



UIStackView <https://riptutorial.com/zh-TW/ios/topic/1390/uistackview>

# 95: UIStoryboard

UIStoryboardInterface Builder。

## Examples

### UIStoryboard

#### UIStoryboard

```
let storyboard = UIStoryboard(name: "Main", bundle: nil)
```

- **name** =>
- **bundle** => storyboard。 nil。

#### storyboardUIViewController

```
let viewController = storyboard.instantiateViewController(withIdentifier: "yourIdentifier")
```

## Objective-C

### Objective-C UIStoryboard

```
UIStoryboard *storyboard = [UIStoryboard storyboardWithName:@"MainStoryboard" bundle:nil];
```

#### storyboardUIViewController

```
MyViewController *myViewController = [storyboard
instantiateViewControllerWithIdentifier:@"MyViewControllerIdentifier"];
```

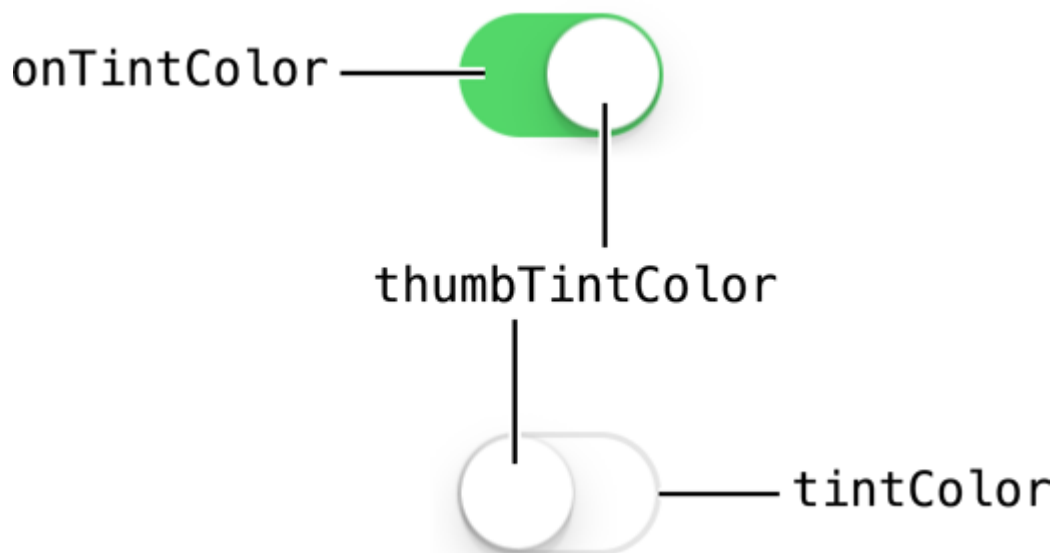
```
let storyboard = UIStoryboard(name: "StoryboardName", bundle: nil)
let vc = storyboard.instantiateViewController(withIdentifier: "ViewControllerID") as
YourViewController
self.present(vc, animated: true, completion: nil)
```

[UIStoryboard](https://riptutorial.com/zh-TW/ios/topic/8795/uistoryboard) <https://riptutorial.com/zh-TW/ios/topic/8795/uistoryboard>

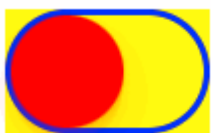
# 96: UISwitch

- `instancetypeinitWithFrameCGRect;`
- `voidsetOnBOOLon animatedBOOLanimated;`
- `initWithCoderNSCoder *aDecoder;`

## 1. UISwitch Apple



## 2. Enoch Huang



## Examples

/

### Objective-C

```
[mySwitch setOn:YES];
//or
[mySwitch setOn:YES animated:YES];
```

```
mySwitch.setOn(false)
//or
mySwitch.setOn(false, animated: false)
```

## Objective-C

```
mySwitch.backgroundColor = [UIColor yellowColor];
[mySwitch setBackgroundColor: [UIColor yellowColor]];
mySwitch.backgroundColor = [UIColor colorWithRed:255/255.0 green:0/255.0 blue:0/255.0
alpha:1.0];
mySwitch.backgroundColor= [UIColor colorWithWhite: 0.5 alpha: 1.0];
mySwitch.backgroundColor=[UIColor colorWithHue: 0.4 saturation: 0.3 brightness:0.7 alpha:
1.0];
```

```
mySwitch.backgroundColor = UIColor.yellow
mySwitch.backgroundColor = UIColor(red: 255.0/255, green: 0.0/255, blue: 0.0/255, alpha: 1.0)
mySwitch.backgroundColor = UIColor(white: 0.5, alpha: 1.0)
mySwitch.backgroundColor = UIColor(hue: 0.4,saturation: 0.3,brightness: 0.7,alpha: 1.0)
```

## Objective-C

```
//for off-state
mySwitch.tintColor = [UIColor blueColor];
[mySwitch setTintColor: [UIColor blueColor]];

//for on-state
mySwitch.onTintColor = [UIColor cyanColor];
[mySwitch setOnTintColor: [UIColor cyanColor]];
```

```
//for off-state
mySwitch.tintColor = UIColor.blueColor()

//for on-state
mySwitch.onTintColor = UIColor.cyanColor()
```

/

## Objective-C

```
//set off-image
mySwitch.offImage = [UIImage imageNamed:@"off_image"];
[mySwitch setOffImage:[UIImage imageNamed:@"off_image"]];

//set on-image
mySwitch.onImage = [UIImage imageNamed:@"on_image"];
[mySwitch setOnImage:[UIImage imageNamed:@"on_image"]];
```

```
//set off-image
mySwitch.offImage = UIImage(named: "off_image")

//set on-image
mySwitch.onImage = UIImage(named: "on_image")
```

**UISwitch** <https://riptutorial.com/zh-TW/ios/topic/2182/uiswitch>



# 97: UITableView

◦ ◦

- - CGFloat UITableView \*tableView heightForRowAtIndexPath NSIndexPath \*indexPath;
- - CGFloat UITableView \*tableView heightForHeaderInSection NSInteger section;
- - CGFloat UITableView \*tableView heightForFooterInSection NSInteger section;
- - UIView \*tableView UITableView \*tableView viewForHeaderInSection NSInteger section;
- - UIView \*tableView UITableView \*tableView viewForFooterInSection NSInteger section;
- - UITableViewCellAccessoryType UITableView \*tableView accessoryTypeForRowAtIndexPath NSIndexPath \*indexPath
- - void UITableView \*tableView accessoryButtonTappedForRowWith NSIndexPath \*indexPath;
- - NSIndexPath \*tableView UITableView \*tableView willSelectRowAtIndexPath NSIndexPath \*indexPath;
- - NSIndexPath \*tableView UITableView \*tableView willDeselectRowAtIndexPath NSIndexPath \*indexPath
- - void UITableView \*tableView didSelectRowAtIndexPath NSIndexPath \*indexPath;
- - void UITableView \*tableView didDeselectRowAtIndexPath NSIndexPath \*indexPath
- - UITableViewCellEditingStyle UITableView \*tableView editingStyleForRowAtIndexPath NSIndexPath \*indexPath;
- - NSString \*tableView UITableView \*tableView titleForDeleteConfirmationButtonForRow NSIndexPath \*indexPath
- - BOOL UITableView \*tableView shouldIndentWhileEditingRowAtIndexPath NSIndexPath \*indexPath;
- - void UITableView \*tableView willBeginEditingRowAtIndexPath NSIndexPath \*indexPath;
- - void UITableView \*tableView didEndEditingRowAtIndexPath NSIndexPath \*indexPath;
- - NSIndexPath \*tableView UITableView \*tableView targetIndexPathForMoveFromRowAtIndexPath NSIndexPath \*sourceIndexPath

toProposedIndexPathNSIndexPath \*proposedDestinationIndexPath;

- - NSIntegertableViewUITableView \*tableView indentationLevelForRowAtIndexPath NSIndexPath \*indexPath;
- - NSIntegertableViewUITableView \*tableView numberOfRowsInSectionNSIntegersection;
- - UITableViewCell \*tableViewUITableView \*tableView cellForRowAtIndexPathNSIndexPath \*indexPath;
- - NSIntegernumberOfSectionsInTableViewUITableView \*tableView;
- - NSString \*tableViewUITableView \*tableView titleForHeaderInSectionNSIntegersection; // UILabel
- - NSString \*tableViewUITableView \*tableView titleForFooterInSectionNSIntegersection;
- - BOOLtableViewUITableView \*tableView canEditRowAtIndexPathNSIndexPath \*indexPath;
- - BOOLtableViewUITableView \*tableView canMoveRowAtIndexPathNSIndexPath \*indexPath;
- - NSArray \*sectionIndexTitlesForTableViewUITableView \*tableView;
- - NSIntegertableViewUITableView \*tableView sectionForSectionIndexTitleNSString \*title atIndexNSIntegerindex;
- - voidtableViewUITableView \*tableView commitEditingStyleUITableViewCellEditingStyle editingStyle forRowAtIndexPathNSIndexPath \*indexPath;
- - voidtableViewUITableView \*tableView moveRowAtIndexPathNSIndexPath \*sourceIndexPath toIndexPathNSIndexPath \*destinationIndexPath;

UITableViewUIScrollView ◦ UITableViewDelegateUIScrollViewDelegate ◦ UITableViewUIScrollView ◦

## Examples

iOS 8Apple ◦ AutolayoutUITableViewCellUITableView ◦ rowHeight  
UITableViewAutomaticDimension ◦

UITableViewestimatedRowHeight ◦

◦

- *AppleUITableView*

```
self.tableView.estimatedRowHeight = 44.0
```

tableViewheightForRowAtIndexPath ◦ ◦

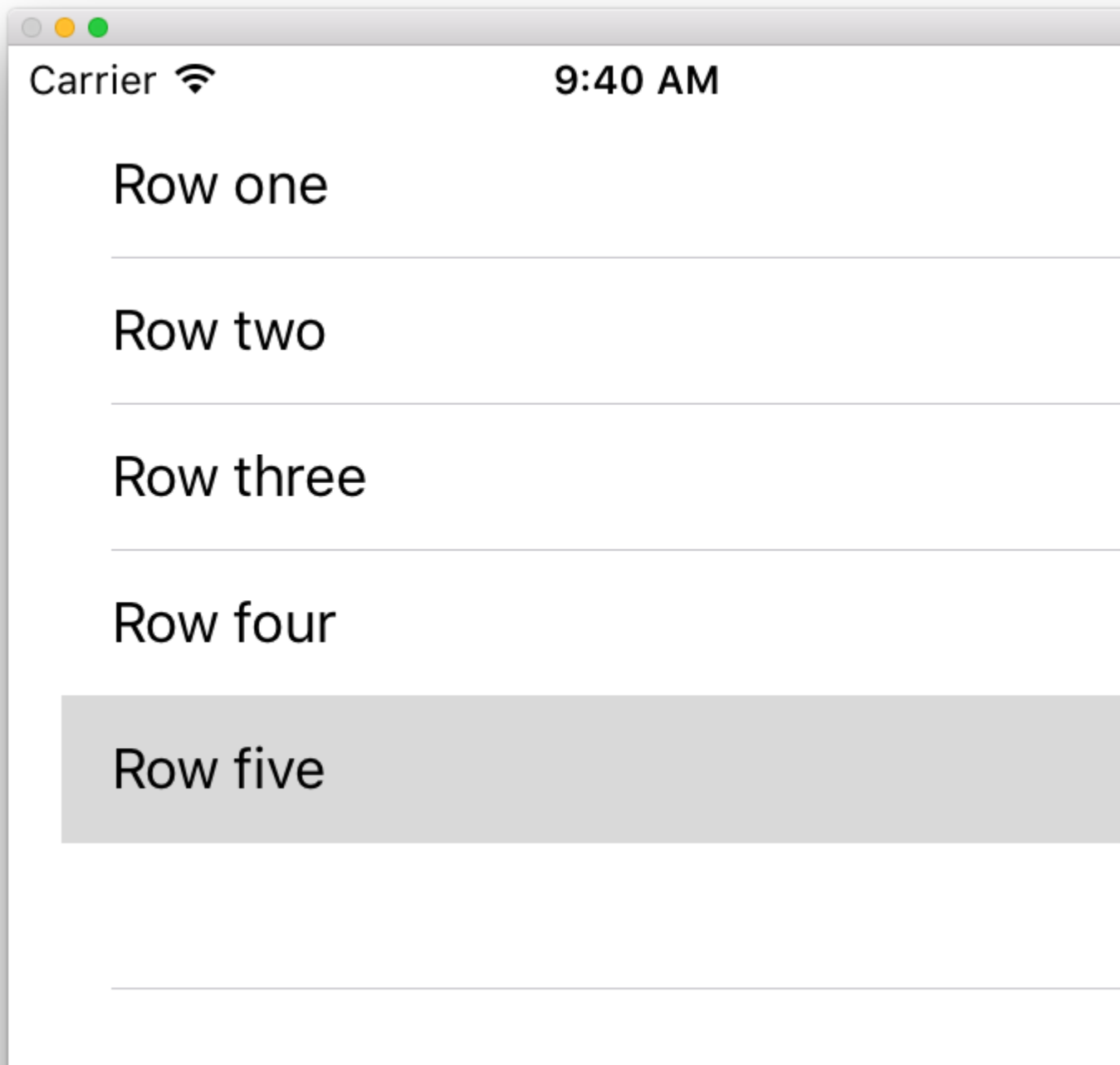
```
override func tableView(tableView: UITableView, heightForRowAtIndexPath indexPath:
NSIndexPath) -> CGFloat {
 switch indexPath.section {
 case 1:
 return 60
 default:
 return UITableViewAutomaticDimension
 }
}
```

## Objective-C

```
- (CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath:(NSIndexPath *)indexPath
{
 switch (indexPath.section) {
 case 1:
 return 60;
 default:
 return UITableViewAutomaticDimension;
 }
}
```

## UITableView

◦ ◦ ◦



## UITableView

UITableView Storyboard ◦ Storyboard UITableViewUIViewController ◦ "" ◦

- 
- ◦ ◦

```
let mydataArray: [String] = ["Row one", "Row two", "Row three", "Row four", "Row five"]
```

### Objective-C

```
// You'll need to define this variable as a global variable (like an @property) so that you
can access it later when needed.
NSArray *mydataArray = @[@"Row one", @"Row two", @"Row three", @"Row four", @"Row five"];
```

## View Controller

UITableViewDataSource◦

```
class ViewController: UIViewController, UITableViewDataSource {
```

### Objective-C

```
@interface ViewController : UIViewController <UITableViewDataSource>
```

UITableViewDataSource

- tableView:numberOfRowsInSection ◦

```
// Swift

func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
 return self.mydataArray.count
}
```

- tableView:cellForRowAtIndexPath tableView:numberOfRowsInSection  
tableView:numberOfRowsInSection ◦ 10◦

```
// Swift

func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) ->
UITableViewCell {
 // Create a new cell here. The reuseIdentifier needs to match the reuse
 identifier from the cell in your Storyboard
 let cell: UITableViewCell =
 tableView.dequeueReusableCellWithIdentifier(reuseIdentifier) as UITableViewCell!

 // Set the label on your cell to the text from your data array
 cell.textLabel?.text = self.mydataArray[indexPath.row]

 return cell
}
```

cellForRowAtIndexPath:nil cellForRowAtIndexPath: .

```
Uncaught exception 'NSInternalInconsistencyException', reason: 'UITableView
dataSource must return a cell from tableView:cellForRowAtIndexPath:'
```

dataSourceself◦ ""dataSource UIViewCONTROLLER UIView UIViewController◦

- . . .

UITableViewDelegate . UITableViewDataSource

```
class ViewController: UIViewController, UITableViewDataSource, UITableViewDelegate {
```

## Objective-C

```
@interface ViewController : UIViewController <UITableViewDataSource, UITableViewDelegate>
```

.

- tableView:didSelectRowAtIndexPath . Xcode .

```
// Swift

func tableView(tableView: UITableView, didSelectRowAtIndexPath indexPath: NSIndexPath) {
 print("You tapped cell number \(indexPath.row).")
}

// Objective-C

- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath
*)indexPath {
 NSLog(@"You tapped cell number %ld.", (long)indexPath.row);
}
```

.

```
import UIKit
class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {

 // Data model: These strings will be the data for the table view cells
 let mydataArray: [String] = ["Row one", "Row two", "Row three", "Row four", "Row five"]

 // cell reuse id (cells that scroll out of view can be reused)
 let reuseIdentifier = "cell"

 // don't forget to hook this up from the storyboard
 @IBOutlet var myTableView: UITableView!

 override func viewDidLoad() {
 super.viewDidLoad()

 // Register the table view cell class and its reuse id
 myTableView.registerClass(UITableViewCell.self, forCellReuseIdentifier:
reuseIdentifier)

 // This view controller itself will provide the delegate methods and row data for the
table view.
 myTableView.delegate = self
 myTableView.dataSource = self
 }
}
```

```

// number of rows in table view
func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
 return self.mydataArray.count
}

// create a cell for each table view row
func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) ->
UITableViewCell {

 // create a new cell if needed or reuse an old one
 let cell:UITableViewCell =
tableView.dequeueReusableCellWithIdentifier(cellReuseIdentifier) as UITableViewCell!

 // set the text from the data model
 cell.textLabel?.text = self.mydataArray[indexPath.row]

 return cell
}

// method to run when table view cell is tapped
func tableView(tableView: UITableView, didSelectRowAtIndexPath indexPath: NSIndexPath) {
 print("You tapped cell number \(indexPath.row).")
}
}

```

## Objective-C

### ViewController.h

```

#import <UIKit/UIKit.h>

@interface ViewController: UIViewController <UITableViewDelegate, UITableViewDataSource> {
 IBOutlet UITableView *myTableView;
 NSArray *mydataArray;
}

@end

```

### ViewController.m

```

#import "ViewController.h"

// cell reuse id (cells that scroll out of view can be reused)
NSString * _Nonnull cellReuseIdentifier = @"cell";

@implementation ViewController

- (void)viewDidLoad {
 [super viewDidLoad];

 // Data model: These strings will be the data for the table view cells
 mydataArray = @[@"Row one", @"Row two", @"Row three", @"Row four", @"Row five"];

 // Register the table view cell class and its reuse id
 [myTableView registerClass:[UITableViewCell class]
forCellReuseIdentifier:cellReuseIdentifier];

 // This view controller itself will provide the delegate methods and row data for the

```

```

table view.
 myTableView.delegate = self;
 myTableView.dataSource = self;
}

// number of rows in table view
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section {
 return mydataArray.count;
}

// create a cell for each table view row
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {
 // create a new cell if needed or reuse an old one
 UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:cellReuseIdentifier];

 // set the text from the data model
 cell.textLabel.text = mydataArray[indexPath.row];

 return cell;
}

// method to run when table view cell is tapped
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath{
 NSLog(@"You tapped cell number %ld.", (long)indexPath.row);
}

@end

```

UITableViewDelegate UITableViewDataSource UITableView° UITableView°

---

## UITableViewDataSource

numberOfRowsInSection:tableView°

### Objective-C

```

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section {
 // Return the number of rows for the table view. Usually populated from an array,
 // or can be statically defined.
 return self.myArray.count;
}

```

### 3

```

func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
 // Return the number of rows for the table view. Usually populated from an array,
 // or can be statically defined.
 return self.myArray.count
}

```

cellForRowAtIndexPath:UITableView° UITableViewCell°

dequeueReusableCellWithIdentifier:forIndexPath:UITableViewregisterClass:forCellReuseIdentifier:registerNib:forCellReuseIdentifier:



methods nib ◦ UIViewController viewDidLoad ◦

## Objective-C

```
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {
 MyCustomCell *cell = [tableView dequeueReusableCellWithIdentifier:@"MyCustomCell"
 forIndexPath:indexPath];

 // All additional customization goes here
 cell.titleLabel.text = [NSString stringWithFormat:@"Title Row %lu", indexPath.row];

 return cell;
}
```

## 3

```
func tableView(_ tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) ->
UITableViewCell {
 let cell = tableView.dequeueReusableCellWithIdentifier("MyCustomCell",
forIndexPath:indexPath)

 // All additional customization goes here
 cell.titleLabel.text = String(format:"Title Row %lu", indexPath.row)

 return cell
}
```

titleForHeaderInSection: ◦ ◦

## Objective-C

```
- (NSString *)tableView:(UITableView *)tableView titleForHeaderInSection:(NSInteger)section {
 switch(section) {
 case 0:
 return @"Title 1";
 break;

 case 1:
 return @"Title 2";
 break;

 default:
 return nil;
 break;
 }
}
```

## 3

```
func tableView(_ tableView: UITableView, titleForHeaderInSection section: Int) -> String? {
 switch section {
 case 0:
 return "Title 1"
 case 1:
 return "Title 2"
 }
}
```

```

 default:
 return nil
 }
}

```

titleForFooterInSection:◦

## Objective-C

```

- (NSString *)tableView:(UITableView *)tableView titleForFooterInSection:(NSInteger)section {
 return @"Footer text";
}

```

## 3

```

func tableView(_ tableView: UITableView, titleForFooterInSection section: Int) -> String? {
 return "Footer text"
}

```

canEditRowAtIndexPath:UI◦ YES ◦

## Objective-C

```

- (BOOL)tableView:(UITableView *)tableView canEditRowAtIndexPath:(NSIndexPath *)indexPath {
 return YES;
}

```

## 3

```

func tableView(_ tableView: UITableView, canEditRowAtIndexPath indexPath: NSIndexPath) -> Bool
{
 return true
}

```

commitEditingStyle:forRowAtIndexPath◦ UITableView◦

## Objective-C

```

- (void)tableView:(UITableView *)tableView
commitEditingStyle:(UITableViewCellEditingStyle)editingStyle forRowAtIndexPath:(NSIndexPath
*)indexPath {
 switch (editingStyle) {
 case UITableViewCellEditingStyleInsert:
 // Insert new data into the backing data model here
 [self insertNewDataIntoDataModel];
 [tableView insertRowsAtIndexPaths:@[indexPath]
withRowAnimation:UITableViewRowAnimationAutomatic];
 break;
 case UITableViewCellEditingStyleDelete:
 [self removeDataFromDataModelAtIndex:indexPath.row];
 [tableView deleteRowsAtIndexPaths:@[indexPath]
withRowAnimation:UITableViewRowAnimationAutomatic];
 break;
 }
}

```

```

 default:
 // Nothing to perform if the editingStyle was neither Insert or Delete
 break;
 }
}

```

### 3

```

func tableView(_ tableView: UITableView, commitEditingStyle editingStyle:
UITableViewCellEditingStyle, forRowAtIndexPath indexPath: NSIndexPath) {
 switch editingStyle {
 case .Insert:
 self.insertNewDataIntoDataModel()
 tableView.insertRowsAtIndexPaths([indexPath], withRowAnimation:.Automatic)
 case .Delete:
 self.removeDataFromDataModelAtIndex(indexPath.row)
 tableView.deleteRowsAtIndexPaths([indexPath], withRowAnimation:.Automatic)
 default:
 // Nothing to perform if the editingStyle was neither Insert or Delete
 }
}

```

editActions:forRowAtUITableviewadditional◦ ◦

### 3

```

override func tableView(_ tableView: UITableView, editActionsForRowAt indexPath: IndexPath) ->
[UITableViewRowAction]? {
 // In the handler you will get passed the action as well as the indexPath for
 // the row that is being edited
 let editAction = UITableViewRowAction(style: .normal, title: "Edit", handler: { [unowned
self] action, indexPath in
 // Do something when edit is tapped
 })

 // Change the color of the edit action
 editAction.backgroundColor = UIColor.blue

 let deleteAction = UITableViewRowAction(style: .destructive, title: "Delete", handler: {
[unowned self] action, indexPath in
 // Handel the delete event
 })

 return [deleteAction, editAction]
}

```

## UITableViewDelegate

UITableViewDelegateUITableView◦

numberOfSectionsInTableView:1◦

## Objective-C

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView {
 return self.numSections;
}
```

### 3

```
func numberOfSectionsInTableView(_ tableView: UITableView) -> Int {
 return self.numSections
}
```

viewForHeaderInSection°

## Objective-C

```
- (UIView *)tableView:(UITableView *)tableView viewForHeaderInSection:(NSInteger)section {

 UIView *view = [[UIView alloc] initWithFrame:CGRectMake(0, 0,
CGRectGetWidth(tableView.frame), 22)];
 view.backgroundColor = [UIColor groupTableViewBackgroundColor];

 UILabel *label = [[UILabel alloc] init];
 label.font = [UIFont systemFontOfSize:12];
 label.textColor = [UIColor darkGrayColor];

 switch (section) {
 case 1: {
 label.text = @"Title";
 label.frame = labelFrame;

 UIButton *more = [[UIButton alloc] initWithFrame:btnFrame];
 [more setTitle:@"See more" forState:UIControlStateNormal];
 [more.titleLabel setFont:[UIFont systemFontOfSize:12]];
 [view addSubview:more];
 } break;

 default:
 label.frame = CGRectMake(0, 0, 0, 0);
 break;
 }

 [view addSubview:label];
 return view;
}
```

### 3

```
func tableView(_ tableView: UITableView, viewForHeaderInSection section: Int) -> UIView? {
 let view = UIView(frame: CGRect(x: 0, y: 0, width: tableView.frame.size.width, height:
22))
 view.backgroundColor = UIColor.groupTableViewBackgroundColor()

 let label = UILabel()
 label.font = UIFont.systemFont(ofSize: 12)
 label.textColor = UIColor.darkGrayColor()
```

```

switch section {
 case 1:
 label.text = "Title"
 label.frame = labelFrame

 let more = UIButton(frame: btnFrame)
 more.setTitle("See more", forState:.Normal)
 view.addSubview(more)

 default:
 label.frame = CGRect.zero
}

view.addSubview(label)
return view;
}

```

heightForRowAtIndexPath:◦

## Objective-C

```

- (CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath:(NSIndexPath *)indexPath
{
 return 44;
}

```

### 3

```

func tableView(_ tableView: UITableView, heightForRowAtIndexPath indexPath: NSIndexPath) ->
CGFloat {
 return 44
}

```

heightForHeaderInSection:heightForFooterInSection

## Objective-C

```

- (CGFloat)tableView:(UITableView *)tableView heightForHeaderInSection:(NSInteger)section {
 return 33;
}

```

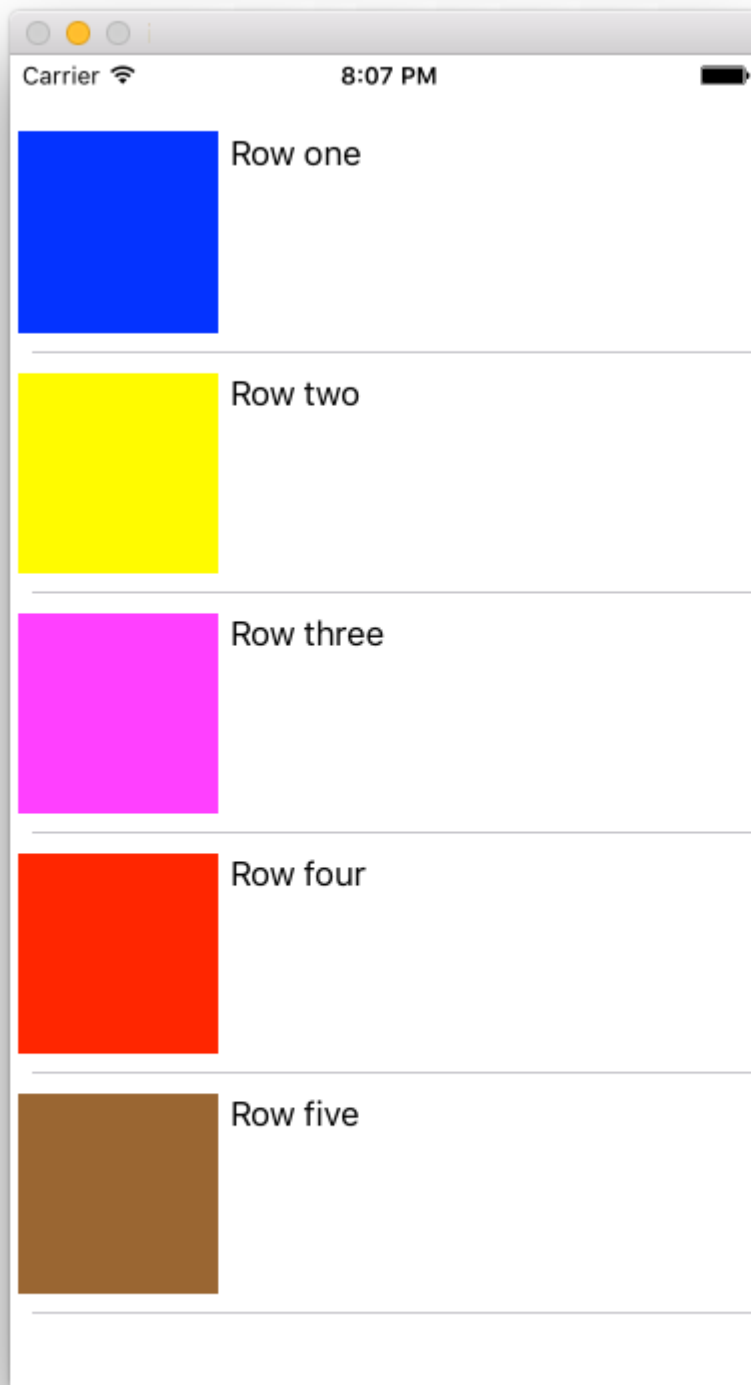
### 3

```

func tableView(_ tableView: UITableView, heightForHeaderInSection section: Int) -> CGFloat {
 return 33
}

```

UITableViewCell◦ ◦ ◦ ◦



◦

UITableViewCell XcodeCocoa Touch UITableViewCell ◦ ◦

```
class CustomTableViewCell: UITableViewCell {
 static var identifier: String {
 return NSStringFromClass(self)
 }

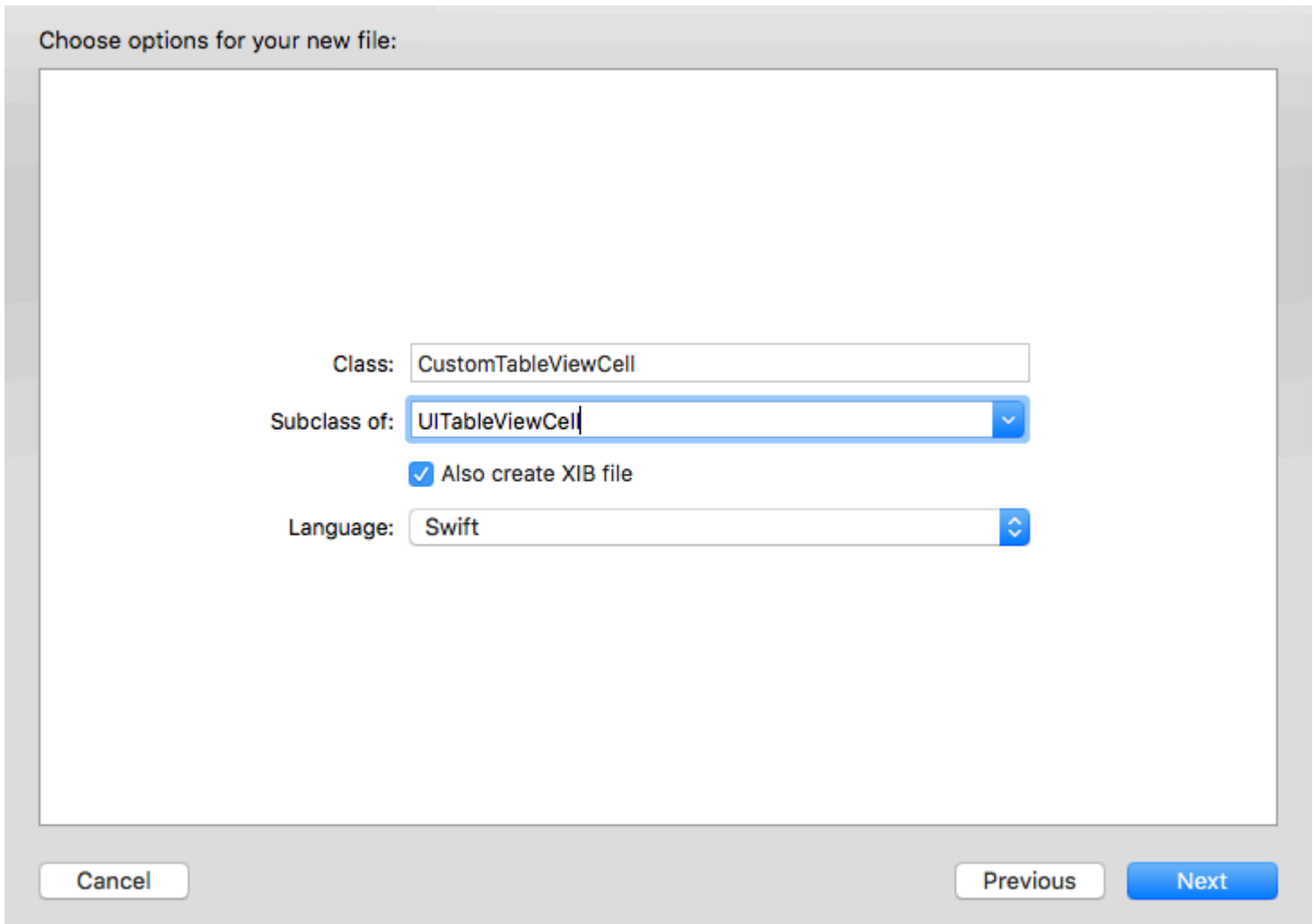
 var customLabel: UILabel!
```

```

override func awakeFromNib() {
 super.awakeFromNib()
 // Initialization code
 customLabel = UILabel(frame: CGRect(x: 0, y: 0, width: contentView.frame.width,
height: contentView.frame.height))
 customLabel.textAlignment = .center
 contentView.addSubview(customLabel)
}
}

```

“XIB”Interface Builder. ◦ customLabel@IBOutlet



tableViewUIViewController. ◦ *Storyboard*. ◦

```

override func viewDidLoad() {
 super.viewDidLoad()

 // Register Cell Class
 tableView.register(CustomTableViewCell.self, forCellReuseIdentifier:
CustomTableViewCell.identifier)
}

```

XIBregisterNib

```
// Register Nib
tableView.register(UINib(nibName: CustomTableViewCell.identifier, bundle: nil),
forCellReuseIdentifier: CustomTableViewCell.identifier)
```

tableViewcellForRowAtIndexPath

```
func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) ->
UITableViewCell {
 // Load the CustomTableViewCell. Make sure the identifier supplied here matches the one
 from your cell
 let cell: CustomTableViewCell =
tableView.dequeueReusableCellWithIdentifier(CustomTableViewCell.identifier) as!
CustomTableViewCell

 // This is where the magic happens - setting a custom property on your very own cell
 cell.customLabel.text = "My Custom Cell"

 return cell
}
```

## UITableViewCells

StoryboardUIViewControllerUITableView° UITableViewDataSourceUITableViewDelegate°

### Objective-C

.h

```
NSMutableArray *arrayForBool;
NSMutableArray *sectionTitleArray;
```

.m

```
- (void)viewDidLoad {
 [super viewDidLoad];

 arrayForBool = [[NSMutableArray alloc] init];
 sectionTitleArray = @[@"Sam",@"Sanju",@"John",@"Staffy"];

 for (int i=0; i<[sectionTitleArray count]; i++) {
 [arrayForBool addObject:[NSNumber numberWithInt:NO]];
 }

 _tableView.dataSource = self;
 _tableView.delegate = self;
}

// Declare number of rows in section
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section {
 if ([[arrayForBool objectAtIndex:section] boolValue]) {
 return section+2;
 } else {
 return 0;
 }
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath
```



```

*)indexPath {

static NSString *cellid=@"hello";
UITableViewCell *cell=[tableView dequeueReusableCellWithIdentifier:cellid];
if (cell==nil) {
 cell=[[UITableViewCell alloc] initWithStyle:UITableViewCellStyleSubtitle
reuseIdentifier:cellid];
}
 BOOL manyCells = [[arrayForBool objectAtIndex:indexPath.section] boolValue];

 /** If the section supposed to be closed***/
 if(!manyCells){
 cell.backgroundColor=[UIColor clearColor];
 cell.textLabel.text=@" ";
 }
 /** If the section supposed to be Opened***/
 else{
 cell.textLabel.text=[NSString stringWithFormat:@"%d", [sectionTitleArray
objectAtIndex:indexPath.section], indexPath.row+1];
 cell.backgroundColor=[UIColor whiteColor];
 cell.selectionStyle=UITableViewCellSelectionStyleNone ;
 }
cell.textLabel.textColor=[UIColor blackColor];

 /** Add a custom Separator with cell*/
 UIView* separatorLineView = [[UIView alloc] initWithFrame:CGRectMake(15, 40,
_expandableTableView.frame.size.width-15, 1)];
separatorLineView.backgroundColor = [UIColor blackColor];
[cell.contentView addSubview:separatorLineView];
return cell;
}

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
return [sectionTitleArray count];
}

- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
 /******* Close the section, once the data is selected
 *****/
 [arrayForBool replaceObjectAtIndex:indexPath.section withObject:[NSNumber numberWithInt:NO]];

 [_expandableTableView reloadSections:[NSIndexPath indexPathWithIndex:indexPath.section]
withRowAnimation:UITableViewRowAnimationAutomatic];
}

- (CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath:(NSIndexPath
*)indexPath
{
if ([[arrayForBool objectAtIndex:indexPath.section] boolValue]) {
 return 40;
}
return 0;
}

- (UIView *)tableView:(UITableView *)tableView viewForHeaderInSection:(NSInteger) section

```

```

{
UIView *sectionView=[[UIView alloc]initWithFrame:CGRectMake(0, 0, 280,40)];
sectionView.tag=section;
UILabel *viewLabel=[[UILabel alloc]initWithFrame:CGRectMake(10, 0,
_expandableTableView.frame.size.width-10, 40)];
viewLabel.backgroundColor=[UIColor clearColor];
viewLabel.textColor=[UIColor blackColor];
viewLabel.font=[UIFont systemFontOfSize:15];
viewLabel.text=[NSString stringWithFormat:@"List of %@",[sectionTitleArray
objectAtIndex:section]];
[sectionView addSubview:viewLabel];
 /***** Add a custom Separator with Section view *****/
UIView* separatorLineView = [[UIView alloc] initWithFrame:CGRectMake(15, 40,
_expandableTableView.frame.size.width-15, 1)];
separatorLineView.backgroundColor = [UIColor blackColor];
[sectionView addSubview:separatorLineView];

/***** Add UITapGestureRecognizer to SectionView *****/

UITapGestureRecognizer *headerTapped = [[UITapGestureRecognizer alloc] initWithTarget:self
action:@selector(sectionHeaderTapped)];
[sectionView addGestureRecognizer:headerTapped];

return sectionView;

}

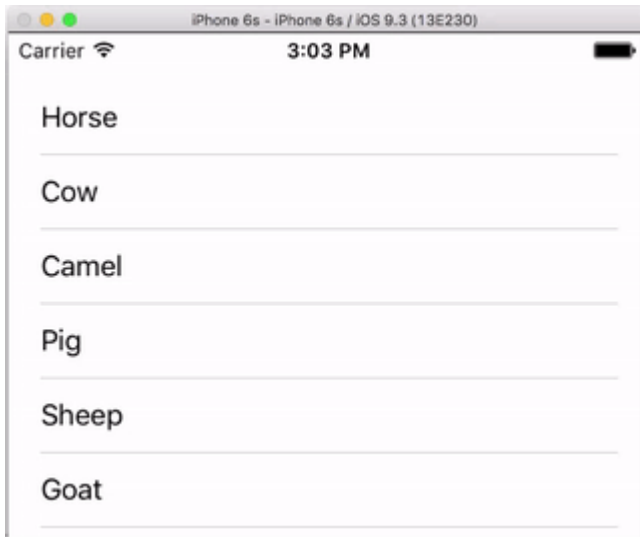
- (void)sectionHeaderTapped:(UITapGestureRecognizer *)gestureRecognizer{

NSIndexPath *indexPath = [NSIndexPath indexPathForRow:0 inSection:gestureRecognizer.view.tag];
if (indexPath.row == 0) {
 BOOL collapsed = [[arrayForBool objectAtIndex:indexPath.section] boolValue];
 for (int i=0; i<[sectionTitleArray count]; i++) {
 if (indexPath.section==i) {
 [arrayForBool replaceObjectAtIndex:i withObject:[NSNumber
numberWithBool:!collapsed]];
 }
 }
 [_expandableTableView reloadData:[NSIndexPath
indexPathSetWithIndex:gestureRecognizer.view.tag]
withRowAnimation:UITableViewRowAnimationAutomatic];

}
}
}

```

- UITableView◦



## SwiftUITableView ◦

### ViewController.swift ◦

```
import UIKit
class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {

 // These strings will be the data for the table view cells
 var animals: [String] = ["Horse", "Cow", "Camel", "Pig", "Sheep", "Goat"]

 let reuseIdentifier = "cell"

 @IBOutlet var tableView: UITableView!

 override func viewDidLoad() {
 super.viewDidLoad()

 // It is possible to do the following three things in the Interface Builder
 // rather than in code if you prefer.
 self.tableView.registerClass(UITableViewCell.self, forCellReuseIdentifier:
cellReuseIdentifier)
 tableView.delegate = self
 tableView.dataSource = self
 }

 // number of rows in table view
 func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
 return self.animals.count
 }

 // create a cell for each table view row
 func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) ->
UITableViewCell {

 let cell:UITableViewCell =
self.tableView.dequeueReusableCellWithIdentifier(reuseIdentifier) as UITableViewCell!

 cell.textLabel?.text = self.animals[indexPath.row]

 return cell
 }
}
```

```

// method to run when table view cell is tapped
func tableView(tableView: UITableView, didSelectRowAtIndexPath indexPath: NSIndexPath) {
 print("You tapped cell number \(indexPath.row).")
}

// this method handles row deletion
func tableView(tableView: UITableView, commitEditingStyle editingStyle:
UITableViewCellEditingStyle, forRowAtIndexPath indexPath: NSIndexPath) {

 if editingStyle == .Delete {

 // remove the item from the data model
 animals.removeAtIndex(indexPath.row)

 // delete the table view row
 tableView.deleteRowsAtIndexPaths([indexPath], withRowAnimation: .Fade)

 } else if editingStyle == .Insert {
 // Not used in our example, but if you were adding a new row, this is where you
 would do it.
 }
}
}

```

◦

```

func tableView(tableView: UITableView, commitEditingStyle editingStyle:
UITableViewCellEditingStyle, forRowAtIndexPath indexPath: NSIndexPath) {

 if editingStyle == .Delete {

 // remove the item from the data model
 animals.removeAtIndex(indexPath.row)

 // delete the table view row
 tableView.deleteRowsAtIndexPaths([indexPath], withRowAnimation: .Fade)

 }
}

```

---

UITableView View Controller ◦ View Controller ◦ @IBOutlet var tableView: UITableView! ◦

---

◦ “” ◦

- 
- iOS 8. ◦
  - ◦

- 
- -
  - [Apple](#)

---

layoutMargins:◦ ◦

cellForRowAtIndexPath: willDisplayCell:layoutMargins:UIEdgeInsetsZero ◦

## Objective-C

```
[cell setLayoutMargins:UIEdgeInsetsZero];

// May also use separatorInset
[cell setSeparatorInset:UIEdgeInsetsZero];
```

```
func tableView(tableView: UITableView, willDisplayCell cell: UITableViewCell,
forRowAtIndexPath indexPath: NSIndexPath) {
 cell.separatorInset = UIEdgeInsetsZero
 cell.layoutMargins = UIEdgeInsetsZero
}

func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) ->
UITableViewCell
{
 cell.separatorInset = UIEdgeInsetsZero
 cell.layoutMargins = UIEdgeInsetsZero
}
```

---

◦ ◦

UIViewController.viewDidLoad:◦ viewDidLoad:◦

```
tableView.separatorStyle = .None
```

## Objective-C

```
tableView.separatorStyle = UITableViewCellSeparatorStyleNone;
```

StoryboardXIBtableViewseparator separatorNone ◦

---

UITableViewCellUITableViewCell

```
tableView.tableFooterView = UIView()
```

## Objective-C

```
tableView.tableFooterView = [[UIView alloc] initWithFrame:CGRectZero];
```

 [Add Player](#)

## Choose Game

Angry Birds

---

Chess

---

Russian Roulette

---

Spin the Bottle

---

Texas Hold'em Poker

---



Tic-Tac-Toe

---

---

---

---

---

---

# 98: UITextField

## Examples

### UITextField -

```
// MARK: - UITextFieldDelegate

let allowedCharacters =
CharacterSet(charactersIn:"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz").inverted

func textField(_ textField: UITextField, shouldChangeCharactersIn range: NSRange,
replacementString string: String) -> Bool {

 let components = string.components(separatedBy: allowedCharacters)
 let filtered = components.joined(separator: "")

 if string == filtered {

 return true

 } else {

 return false

 }
}
```

### Objective-C

```
#define ACCEPTABLE_CHARACTERS @"0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"

- (BOOL)textField:(UITextField *)textField shouldChangeCharactersInRange:(NSRange) range
replacementString:(NSString *)string
{
 NSMutableCharacterSet *cs = [[NSMutableCharacterSet
characterSetWithCharactersInString:ACCEPTABLE_CHARACTERS] invertedSet];

 NSString *filtered = [[string componentsSeparatedByCharactersInSet:cs]
componentsJoinedByString:@""];

 return [string isEqualToString:filtered];
}
```

apple

<https://developer.apple.com/reference/foundation/nscharacterset>

```
let allowedCharacters = CharacterSet.alphanumerics.inverted
let allowedCharacters = CharacterSet.capitalizedLetters.inverted
```

textField\* - **BOOL**textFieldShouldReturnUITextField *textField*

- ◦
- 20textFieldAs50,70,90◦
- 

```

- (BOOL)textFieldShouldReturn:(UITextField *)textField {

 NSInteger nextTag = textField.tag+20;
 // Try to find next responder
 UIResponder *nextResponder = [textField.superview viewWithTag:nextTag];
 if (nextResponder)
 {
 // Found next responder, so set it.
 [nextResponder becomeFirstResponder];
 }
 else
 {
 // Not found, so remove keyboard.
 [textField resignFirstResponder];
 }
 return YES;
}

```

/

## Swift 3.1

- [UITextFieldDelegate](#) [TextField](#)◦

### UITextFieldDelegate

```

class SomeClass: UITextFieldDelegate {

 @IBOutlet var textField: UITextField!

 override func viewDidLoad() {
 super.viewDidLoad()
 textField.delegate = self
 }

}

```

### UITextFieldDelegate◦

#### [textFieldDidBeginEditing\\_ :](#)

```

func textFieldDidBeginEditing(_ textField: UITextField) {
 // now you can perform some action
 // if you have multiple textfields in a class,
 // you can compare them here to handle each one separately
 if textField == emailTextField {
 // e.g. validate email
 }
 else if textField == passwordTextField {

```



```
 // e.g. validate password
 }
}
```

### textFieldDidEndEditing\_ :)

```
func textFieldDidEndEditing(_ textField: UITextField) {
 // now you can perform some action
 // if you have multiple textfields in a class,
 // you can compare them here to handle each one separately
 if textField == emailTextField {
 // e.g. validate email
 }
 else if textField == passwordTextField {
 // e.g. validate password
 }
}
```

TextField/textFieldShouldBeginEditing\_ :)/textFieldShouldEndEditing\_ :)/true / false。

UITextField <https://riptutorial.com/zh-TW/ios/topic/7185/uitextfield>

# 99: UIView

## Examples

```
- (UIImage *)getSnapshot
{
 UIScreen *screen = [UIScreen mainScreen];
 CGRect bounds = [self.view bounds];
 UIGraphicsBeginImageContextWithOptions(bounds.size, false, screen.scale);
 CGContextRef context = UIGraphicsGetCurrentContext();
 CGContextSetInterpolationQuality(context, kCGInterpolationHigh);
 [self.view drawViewHierarchyInRect:bounds afterScreenUpdates:YES];
 UIImage *image = UIGraphicsGetImageFromCurrentImageContext();
 UIGraphicsEndImageContext();
 return image;
}
```

```
var screenshot: UIImage
{
 UIGraphicsBeginImageContext(self.bounds.size);
 let context = UIGraphicsGetCurrentContext();
 self.layer.render(in: context)
 let screenshot = UIGraphicsGetImageFromCurrentImageContext();
 UIGraphicsEndImageContext();
 return screenshot
}
```

- 
- 

```
{
 CGSize fullSize = getImageForEdit.size;
 CGSize sizeInView = AVMakeRectWithAspectRatioInsideRect(imgViewFake.image.size,
imgViewFake.bounds).size;
 CGFloat orgScale = orgScale = fullSize.width/sizeInView.width;
 CGSize newSize = CGSizeMake(orgScale * img.image.size.width, orgScale *
img.image.size.height);
 if(newSize.width <= fullSize.width && newSize.height <= fullSize.height){
 newSize = fullSize;
 }
 CGRect offsetRect;
 if (getImageForEdit.size.height > getImageForEdit.size.width){
 CGFloat scale = newSize.height/fullSize.height;
 CGFloat offset = (newSize.width - fullSize.width*scale)/2;
 offsetRect = CGRectMake(offset, 0, newSize.width-offset*2, newSize.height);
 }
 else{
 CGFloat scale = newSize.width/fullSize.width;
 CGFloat offset = (newSize.height - fullSize.height*scale)/2;
 offsetRect = CGRectMake(0, offset, newSize.width, newSize.height-offset*2);
 }
 UIGraphicsBeginImageContextWithOptions(newSize, NO, getImageForEdit.scale);
 [getImageForEdit drawAtPoint:offsetRect.origin];
 // [img.image drawInRect:CGRectMake(0,0,newSize.width,newSize.height)];
 CGFloat oldScale = img.contentScaleFactor;
```

```
img.contentScaleFactor = getImageForEdit.scale;
[img drawViewHierarchyInRect:CGRectMake(0, 0, newSize.width, newSize.height)
afterScreenUpdates:YES];
img.contentScaleFactor = oldScale;
UIImage *combImage = UIGraphicsGetImageFromCurrentImageContext();
UIGraphicsEndImageContext();
imageData = UIImageJPEGRepresentation(combImage, 1);
}
```

**UIView** <https://riptutorial.com/zh-TW/ios/topic/4622/uiview>

---

# 100: UIWebView

## UIWebView -

### Objective-C Declerations

```
- (BOOL)webView:(UIWebView *)webView
shouldStartLoadWithRequest:(NSURLRequest *)request
navigationType:(UIWebViewNavigationType)navigationType;

- (void)webView:(UIWebView *)webView
didFailLoadWithError:(NSError *)error;

- (void)webViewDidFinishLoad:(UIWebView *)webView;

- (void)webViewDidStartLoad:(UIWebView *)webView;
```

## Examples

### UIWebView

```
let webview = UIWebView(frame: CGRect(x: 0, y: 0, width: 320, height: 480))
```

### Objective-C

```
UIWebView *webview = [[UIWebView alloc] initWithFrame:CGRectMake(0, 0, 320, 480)];

//Alternative way of defining frame for UIWebView
UIWebView *webview = [[UIWebView alloc] init];
CGRect webviewFrame = webview.frame;
webviewFrame.size.width = 320;
webviewFrame.size.height = 480;
webviewFrame.origin.x = 0;
webviewFrame.origin.y = 0;
webview.frame = webviewFrame;
```

## URL

### urlwebview

```
webview.loadRequest (NSURLRequest (URL: NSURL (string: "http://www.google.com"!))
```

### Objective-C

```
[webview loadRequest:[NSURLRequest requestWithURL:[NSURL
URLWithString:@"http://www.google.com"]]];
```

## Web

stopLoading() **webview**◦

```
webview.stopLoading()
```

## Objective-C

```
[webview stopLoading];
```

## Web

```
webview.reload()
```

## Objective-C

```
[webview reload];
```

WebHTML◦ **UIWebViewDelegate**

```
- (void) webViewDidFinishLoad:(UIWebView *) aWebView {
 CGRect frame = aWebView.frame;
 frame.size.height = 1;
 aWebView.frame = frame;
 CGSize fittingSize = [aWebView sizeThatFits:CGSizeZero];
 frame.size = fittingSize;
 aWebView.frame = frame;

 NSLog(@"size: %f, %f", fittingSize.width, fittingSize.height);
}
```

1. . . .

## HTML

WebHTML◦

```
NSString *html = @"<!DOCTYPE html><html><body>Hello World</body></html>";
[webView loadHTMLString:html baseURL:nil];
```

```
let htmlString = "<h1>My First Heading</h1><p>My first paragraph.</p>"
webView.loadHTMLString(htmlString, baseURL: nil)
```

URL◦

```
NSString *html = @"<!DOCTYPE html><html><head><link href='style.css' rel='stylesheet'
type='text/css'></head><body>Hello World</body></html>";
[self loadHTMLString:html baseURL:[NSURL URLWithString:[NSBundle mainBundle]
resourcePath]]];
```

style.css◦ **URL**◦

## JavaScript

`stringByEvaluatingJavaScriptFromString()` [UIWebView JavaScript](#) [JavaScript nil](#)

### String

```
webview.stringByEvaluatingJavaScriptFromString("alert('This is JavaScript!');")
```

```
//Suppose you have javascript file named "JavaScript.js" in project.
let filePath = NSBundle mainBundle().pathForResource("JavaScript", ofType: "js")
do {
 let jsContent = try String.init(contentsOfFile: filePath!, encoding:
NSUTF8StringEncoding)
 webview.stringByEvaluatingJavaScriptFromString(jsContent)
}
catch let error as NSError{
 print(error.debugDescription)
}
```

## Objective-C

### String

```
[webview stringByEvaluatingJavaScriptFromString:@"alert('This is JavaScript!');"];
```

```
//Suppose you have javascript file named "JavaScript.js" in project.
NSString *filePath = [[NSBundle mainBundle] pathForResource:@"JavaScript" ofType:@"js"];
NSString *jsContent = [NSString stringWithContentsOfFile:filePath
encoding:NSUTF8StringEncoding error:nil];
[webview stringByEvaluatingJavaScriptFromString:jsContent];
```

`stringByEvaluatingJavaScriptFromString:` [JavaScript](#) [JavaScriptWeb](#) [WKWebView](#)  
`evaluateJavaScript:completionHandler:` [WKWebView iOS 8.0](#)

[.pdf.txt.doc](#)

[iOS WebView.pdf.txt.doc](#) `loadData:NSData` `webview`

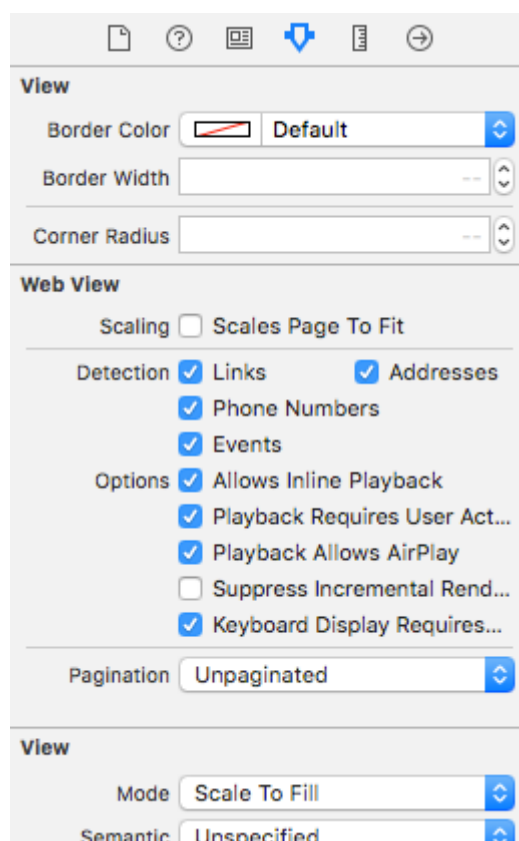
```
//Assuming there is a text file in the project named "home.txt".
let localFilePath = NSBundle mainBundle().pathForResource("home", ofType:"txt");
let data = NSFileManager defaultManager().contentsAtPath(localFilePath!);
webview.loadData(data!, mimeType: "application/txt", textEncodingName:"UTF-8", baseURL:
NSURL())
```

## Objective-C

```
//Assuming there is a text file in the project named "home.txt".
NSString *localFilePath = [[NSBundle mainBundle] pathForResource:@"home" ofType:@"txt"];
NSData *data = [[NSFileManager defaultManager] contentsAtPath:localFilePath];
```

```
[webView loadData:data mimeType:@"application/txt" textEncodingName:@"UTF-8" baseURL:[NSURL new]]];
```

## UIWebView



### vc.h

```
@interface vc : UIViewController<UIWebViewDelegate>
```

### vc.m.

```
- (BOOL)webView:(UIWebView *)webView shouldStartLoadWithRequest:(NSURLRequest *)request
navigationType:(UIWebViewNavigationType)navigationType{

 if (navigationType == UIWebViewNavigationTypeLinkClicked){
 //open it on browser if you want to open it in same web view remove return NO;
 NSURL *url = request.URL;
 if ([[UIApplication sharedApplication] canOpenURL:url]) {
 [[UIApplication sharedApplication] openURL:url];
 }
 return NO;
 }

 return YES;
}
```

## webViewHTML

HTML <sup>'''</sup>

## HTMLwebView

```
webView.loadRequest(NSURLRequest(URL: NSURL(fileURLWithPath:
NSBundle mainBundle().pathForResource("YOUR HTML FILE", ofType: "html")!))
```

- HTMLindex.htmlHTML
- *viewDidLoad**viewDidAppear*

UIWebView <https://riptutorial.com/zh-TW/ios/topic/1452/uiwebview>



# 101: UI

- XCUIApplication//。 Xcode“”。
- XCUIElement//。

## Examples

### Xcode

“UI”。

Language:

Devices:

Use Core Data

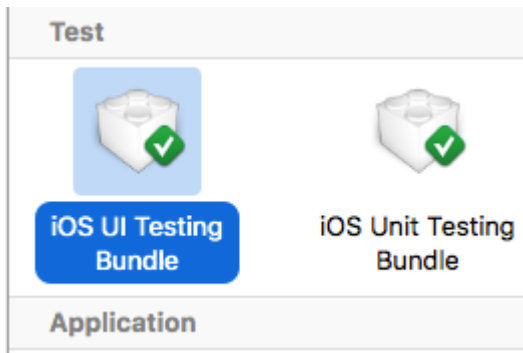
Include Unit Tests

Include UI Tests

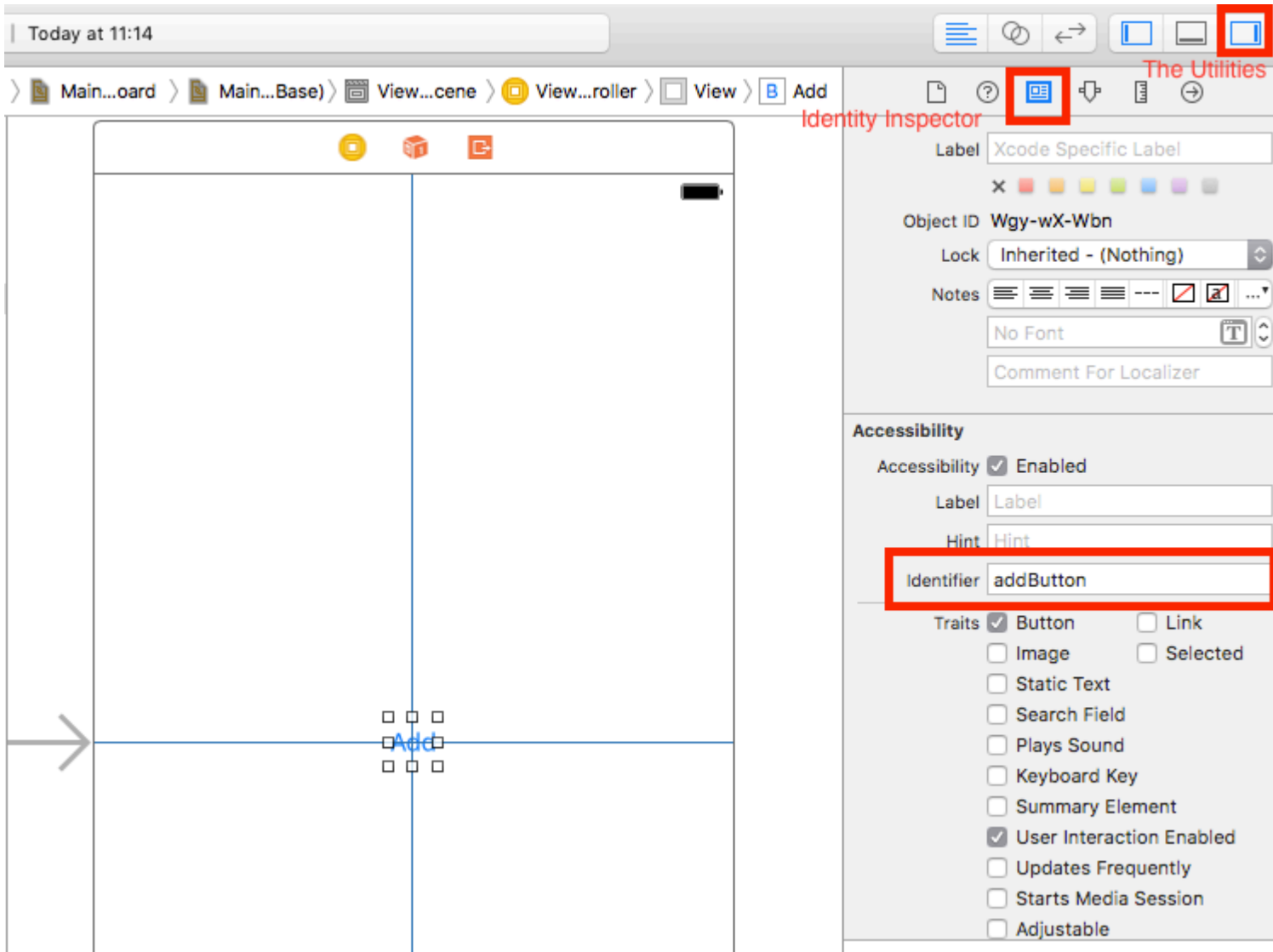
UI target。

### Setps

- File -> New -> Target
- iOS UI Testing Bundle

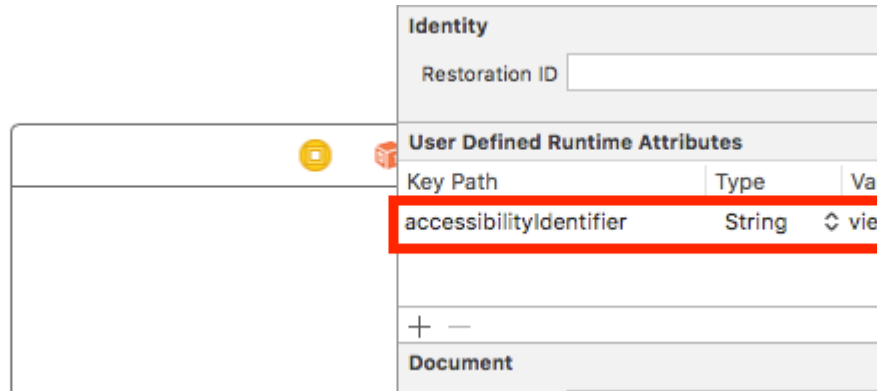
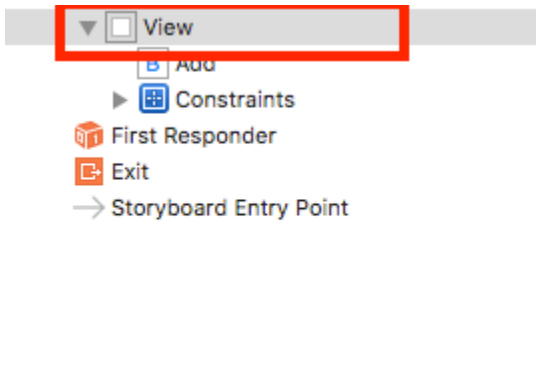


- storyboard。
- the Utilities
- Identity Inspector
- 
- addButtonon



“”””””

- storyboard ◦
- the Utilities
- Identity Inspector
- 
- “ User Defined Runtime Attributes User Defined Runtime Attributes
- Key Path - accessibilityIdentifier
- Type - `String
- For Value - view



## UITest

```
import XCTest

class StackOverFlowUITests: XCTestCase {

 private let app = XCUIApplication()

 //Views

 private var view: XCUIElement!

 //Buttons

 private var addButton: XCUIElement!

 override func setUp() {
 super.setUp()

 app.launch()

 //Views

 view = app.otherElements["view"]

 //Buttons

 addButton = app.buttons["addButton"]
 }

 func testMyApp() {

 addButton.tap()
 view.tap()
 }
}
```

[ ]°

## UIImageView UIScrollView

```
let imageView = app.images["imageView"]
```

```
let scrollView = app.scrollViews["scrollView"]
let view = app.otherElements["view"]
```

## UILabel

```
let label = app.staticTexts["label"]
```

## UIStackView

```
let stackView = app.otherElements["stackView"]
```

## UITableView

```
let tableView = app.tables["tableView"]
```

## UITableViewCell

```
let tableViewCell = tableView.cells["tableViewCell"]
```

## UITableViewCell

```
let tableViewCellButton = tableView.cells.element(boundBy: 0).buttons["button"]
```

## UICollectionView

```
let collectionView = app.collectionViews["collectionView"]
```

## UIButtonUIBarButtonItem

```
let button = app.buttons["button"]
let barButtonItem = app.buttons["barButtonItem"]
```

## UITextField

- UITextField

```
let textField = app.textFields["textField"]
```

- UITextField

```
let passwordTextField = app.secureTextFields["passwordTextField"]
```

## UITextView

```
let textView = app.textViews["textView"]
```

## UISwitch

```
let switch = app.switches["switch"]
```

```
let alert = app.alerts["About yourself"] // Title of presented alert
```

## UI

setUp

```
app.launchEnvironment = ["animations": "0"]
```

appXCUIApplication◦

```
override func setUp() {
 super.setUp()

 let app = XCUIApplication()

 app.launch()
}
```

```
func testStacOverFlowApp() {

 app.terminate()
}
```

```
XCUIDevice.shared().orientationorientationXCUIDevice.shared().orientation
```

```
XCUIDevice.shared().orientation = .landscapeLeft
XCUIDevice.shared().orientation = .portrait
```

[UI https://riptutorial.com/zh-TW/ios/topic/7526/ui](https://riptutorial.com/zh-TW/ios/topic/7526/ui)

# 102: UUID

[UUIDSSKeychainUtility](#) ◦ [Github](#)

## Examples

UUID

## UUID

```
func randomUUID() -> NSString{
 return NSUUID.UUID().UUIDString()
}
```

## Objective-C

```
+ (NSString *)randomUUID {
 if(NSClassFromString(@"NSUUID")) { // only available in iOS >= 6.0
 return [[NSUUID UUID] UUIDString];
 }
 CFUUIDRef uuidRef = CFUUIDCreate(kCFAllocatorDefault);
 CFStringRef cfuuid = CFUUIDCreateString(kCFAllocatorDefault, uuidRef);
 CFRelease(uuidRef);
 NSString *uuid = [((__bridge NSString *) cfuuid) copy];
 CFRelease(cfuuid);
 return uuid;
}
```

iOS 6

UUID

```
let UDIDString = UIDevice.currentDevice().identifierForVendor?.UUIDString
```

## Objective-C

```
NSString *UDIDString = [[[UIDevice currentDevice] identifierForVendor] UUIDString];
```

identifierForVendor ◦ [Apple](#) UUID ◦

## AppleIFAIFFVApple

- IFAIFV ◦
- IFA ◦ IFV ◦

- ASIdentifierManager
  - **advertisingIdentifierUUID** ◦
  - **isAdvertisingTrackingEnabled** ◦

## IFV

- ASIdentifierManager
  - **identifierForVendorUUID** ◦

IFAIFV◦

## iOSUUID

UUID String ◦

UUID◦

---

# Swift 3.0

```
print (UUID ().uuidString)
```

ID◦

UUID <https://riptutorial.com/zh-TW/ios/topic/3629/uuid-->

# 103: WCSessionDelegate

```
WCSessionDelegate WatchConnectivityOS2 + ◦ var watchSession WCSession func
startWatchSession{if WCSession.isSupported{watchSession = WCSession.default watchSession
.delegate = self watchSession.activate}} - didReceiveApplicationContext
```

## Examples

### WKInterfaceController

```
import WatchConnectivity

var watchSession : WCSession?

override func awake(withContext context: Any?) {
 super.awake(withContext: context)
 // Configure interface objects here.
 startWatchSession()
}

func startWatchSession() {

 if(WCSession.isSupported()){
 watchSession = WCSession.default()
 watchSession!.delegate = self
 watchSession!.activate()
 }
}

//Callback in below delegate method when iOS app triggers event
func session(_ session: WCSession, didReceiveApplicationContext applicationContext: [String :
Any]) {
 print("did ReceiveApplicationContext at watch")
}
```

**WCSessionDelegate** <https://riptutorial.com/zh-TW/ios/topic/8289/wcsessiondelegate>



# 104: WKWebView

WKWebView iOS 8 OS X Yosemite WebKit API。UIKit UIWebView AppKit WebView API。

WKWebView 60fps Safari JavaScript WWDC 2014。

## Examples

### WebBrowser

```
import UIKit
import WebKit

class ViewController: UIViewController, UISearchBarDelegate, WKNavigationDelegate, WKUIDelegate {

 var searchBar: UISearchBar! //All web-browsers have a search-bar.
 var webView: WKWebView! //The WKWebView we'll use.
 var toolbar: UIToolbar! //Toolbar at the bottom just like in Safari.
 var activityIndicator: UIActivityIndicatorView! //Activity indicator to let the user know the page is loading.

 override func viewDidLoad() {
 super.viewDidLoad()

 self.initControls()
 self.setTheme()
 self.doLayout()
 }

 override func didReceiveMemoryWarning() {
 super.didReceiveMemoryWarning()
 }

 func initControls() {
 self.searchBar = UISearchBar()

 //WKUserContentController allows us to add Javascript scripts to our webView that will run either at the beginning of a page load OR at the end of a page load.

 let configuration = WKWebViewConfiguration()
 let contentController = WKUserContentController()
 configuration.userContentController = contentController

 //create the webView with the custom configuration.
 self.webView = WKWebView(frame: .zero, configuration: configuration)

 self.toolbar = UIToolbar()
 self.layoutToolbar()

 self.activityIndicator = UIActivityIndicatorView(activityIndicatorStyle: .gray)
 self.activityIndicator.hidesWhenStopped = true
 }
}
```

```

func setTheme() {
 self.edgesForExtendedLayout = UIRectEdge(rawValue: 0)
 self.navigationController?.navigationBar.barTintColor = UIColor.white()

 //Theme the keyboard and searchBar. Setup delegates.
 self.searchbar.delegate = self
 self.searchbar.returnKeyType = .go
 self.searchbar.searchBarStyle = .prominent
 self.searchbar.placeholder = "Search or enter website name"
 self.searchbar.autocapitalizationType = .none
 self.searchbar.autocorrectionType = .no

 //Set the WebView's delegate.
 self.webView.navigationDelegate = self //Delegate that handles page navigation
 self.webView.uiDelegate = self //Delegate that handles new tabs, windows, popups,
layout, etc..

 self.activityIndicator.transform = CGAffineTransform(scaleX: 1.5, y: 1.5)
}

func layoutToolbar() {
 //Browsers typically have a back button, forward button, refresh button, and
newTab/newWindow button.

 var items = Array<UIBarButtonItem>()

 let space = UIBarButtonItem(barButtonItemSystemItem: .flexibleSpace, target: nil, action:
nil)

 items.append(UIBarButtonItem(title: "<", style: .plain, target: self, action:
#selector(onBackButtonPressed)))
 items.append(space)
 items.append(UIBarButtonItem(title: ">", style: .plain, target: self, action:
#selector(onForwardButtonPressed)))
 items.append(space)
 items.append(UIBarButtonItem(barButtonItemSystemItem: .refresh, target: self, action:
#selector(onRefreshPressed)))
 items.append(space)
 items.append(UIBarButtonItem(barButtonItemSystemItem: .organize, target: self, action:
#selector(onTabPressed)))

 self.toolbar.items = items
}

func doLayout() {
 //Add the searchBar to the navigationBar.
 self.navigationItem.titleView = self.searchbar

 //Add all other subViews to self.view.
 self.view.addSubview(self.webView)
 self.view.addSubview(self.toolbar)
 self.view.addSubview(self.activityIndicator)

 //Setup which views will be constrained.

 let views: [String: AnyObject] = ["webView": self.webView, "toolbar": self.toolbar,
"activityIndicator": self.activityIndicator];
 var constraints = Array<String>();

 constraints.append("H:|-0-[webView]-0-|")
 constraints.append("H:|-0-[toolbar]-0-|")

```

```

constraints.append("V:|-0-[webView]-0-[toolbar(50)]-0-|")

//constrain the subviews using the above visual constraints.

for constraint in constraints {
 self.view.addConstraints(NSLayoutConstraint.constraints(withVisualFormat:
constraint, options: NSLayoutConstraintOptions(rawValue: 0), metrics: nil, views: views))
}

for view in self.view.subviews {
 view.translatesAutoresizingMaskIntoConstraints = false
}

//constraint the activity indicator to the center of the view.
self.view.addConstraint(NSLayoutConstraint(item: self.activityIndicator, attribute:
.centerX, relatedBy: .equal, toItem: self.view, attribute: .centerX, multiplier: 1.0,
constant: 0.0))
self.view.addConstraint(NSLayoutConstraint(item: self.activityIndicator, attribute:
.centerY, relatedBy: .equal, toItem: self.view, attribute: .centerY, multiplier: 1.0,
constant: 0.0))
}

//Searchbar Delegates

func searchBarSearchButtonClicked(_ searchBar: UISearchBar) {
 self.searchbar.resignFirstResponder()

 if let searchText = self.searchbar.text, url = URL(string: searchText) {
 //Get the URL from the search bar. Create a new NSURLRequest with it and tell the
webView to navigate to that URL/Page. Also specify a timeout for if the page takes too long.
Also handles cookie/caching policy.

 let request = URLRequest(url: url, cachePolicy: .useProtocolCachePolicy,
timeoutInterval: 30)
 self.webView.load(request)
 }
}

//Toolbar Delegates

func onBackButtonPressed(button: UIBarButtonItem) {
 if (self.webView.canGoBack) { //allow the user to go back to the previous page.
 self.webView.goBack()
 }
}

func onForwardButtonPressed(button: UIBarButtonItem) {
 if (self.webView.canGoForward) { //allow the user to go forward to the next page.
 self.webView.goForward()
 }
}

func onRefreshPressed(button: UIBarButtonItem) {
 self.webView.reload() //reload the current page.
}

func onTabPressed(button: UIBarButtonItem) {
 //TODO: Open a new tab or web-page.
}

```

```

}

//WebView Delegates

func webView(_ webView: WKWebView, decidePolicyFor navigationAction: WKNavigationAction,
decisionHandler: (WKNavigationActionPolicy) -> Void) {

 decisionHandler(.allow) //allow the user to navigate to the requested page.
}

func webView(_ webView: WKWebView, decidePolicyFor navigationResponse:
WKNavigationResponse, decisionHandler: (WKNavigationResponsePolicy) -> Void) {

 decisionHandler(.allow) //allow the webView to process the response.
}

func webView(_ webView: WKWebView, didStartProvisionalNavigation navigation:
WKNavigation!) {
 self.activityIndicator.startAnimating()
}

func webView(_ webView: WKWebView, didFailProvisionalNavigation navigation: WKNavigation!,
withError error: NSError) {
 self.activityIndicator.stopAnimating()

 //Handle the error. Display an alert to the user telling them what happened.

 let alert = UIAlertController(title: "Error", message: error.localizedDescription,
preferredStyle: .alert)
 let action = UIAlertAction(title: "OK", style: .default) { (action) in
 alert.dismiss(animated: true, completion: nil)
 }
 alert.addAction(action)
 self.present(alert, animated: true, completion: nil)
}

func webView(_ webView: WKWebView, didFinish navigation: WKNavigation!) {
 self.activityIndicator.stopAnimating()

 //Update our search bar with the webPage's final endpoint-URL.
 if let url = self.webView.url {
 self.searchbar.text = url.absoluteString ?? self.searchbar.text
 }
}

func webView(_ webView: WKWebView, didReceiveServerRedirectForProvisionalNavigation
navigation: WKNavigation!) {
 //When the webview receives a "Redirect" to a different page or endpoint, this is
called.
}

func webView(_ webView: WKWebView, didCommit navigation: WKNavigation!) {
 //When the content for the webpage starts arriving, this is called.
}

func webView(_ webView: WKWebView, didFail navigation: WKNavigation!, withError error:
NSError) {
}

```

```

func webView(_ webView: WKWebView, didReceive challenge: URLAuthenticationChallenge,
completionHandler: (URLSession.AuthChallengeDisposition, URLCredential?) -> Void) {

 completionHandler(.performDefaultHandling, .none) //Handle SSL connections by default.
 We aren't doing SSL pinning or custom certificate handling.

}

//WebView's UINavigation Delegates

//This is called when a webView or existing loaded page wants to open a new window/tab.
func webView(_ webView: WKWebView, createWebViewWith configuration:
WKWebViewConfiguration, for navigationAction: WKNavigationAction, windowFeatures:
WKWindowFeatures) -> WKWebView? {

 //The view that represents the new tab/window. This view will have an X button at the
 top left corner + a webView.
 let container = UIView()

 //New tabs need an exit button.
 let XButton = UIButton()
 XButton.addTarget(self, action: #selector(onWebViewExit), for: .touchUpInside)
 XButton.layer.cornerRadius = 22.0

 //Create the new webView window.
 let webView = WKWebView(frame: .zero, configuration: configuration)
 webView.navigationDelegate = self
 webView.uiDelegate = self

 //Layout the tab.
 container.addSubview(XButton)
 container.addSubview(webView)

 let views: [String: AnyObject] = ["XButton": XButton, "webView": webView];
 var constraints = Array<String>()

 constraints.append("H:|-(22)-[XButton(44)]")
 constraints.append("H:|-0-[webView]-0-|")
 constraints.append("V:|-(22)-[XButton(44)]-0-[webView]-0-|")

 //constrain the subviews.
 for constraint in constraints {
 container.addConstraints(NSLayoutConstraint.constraints(withVisualFormat:
constraint, options: NSLayoutFormatOptions(rawValue: 0), metrics: nil, views: views))
 }

 for view in container.subviews {
 view.translatesAutoresizingMaskIntoConstraints = false
 }

 //TODO: Add the containerView to self.view or present it with a new controller. Keep
 track of tabs..

 return webView
}

func onWebViewExit(button: UIButton) {
 //TODO: Destroy the tab. Remove the new tab from the current window or controller.
}

```

}

---

GO

 <https://stackoverflow.com/>



[Questions](#) [Tags](#) [Users](#) [Badges](#) [Unanswered](#) [Ask](#)

All Questions 

Show [Interesting](#)

0 

0 

**Mysql error, "Specified key was too long; max key length is 767 bytes" need workaround**

mysql

6 secs ago [rerat](#)

0 

0 

**Error Code: 1305. FUNCTION or PROCEDURE does not exist**

q w e r t y u i o p

a s d f g h j k l

 z x c v b n m 

# 105: Xcode

- `xcodebuild [-project name.xcodeproj] -scheme schemename [[-destination destinationspecifier] ...] [-destination-timeout value] [-configuration configurationname] [-sdk [sdkfullpath | sdkname]] [action ...] [buildsetting=value ...] [-userdefault=value ...]`

|      |            |
|------|------------|
| -    | .xcodeproj |
| -    | o          |
| -    |            |
| -    |            |
| -sdk | SDK        |

`xcodebuild` Xcode Xcode-workspace-scheme `xcodebuild`

## Examples

```
xcodebuild -exportArchive -exportFormat ipa \
-archivePath "/Users/username/Desktop/MyiOSApp.xcarchive" \
-exportPath "/Users/username/Desktop/MyiOSApp.ipa" \
-exportProvisioningProfile "MyCompany Distribution Profile"
```

```
xcodebuild -project <ProjectName.xcodeproj>
-scheme <ProjectName>
-sdk iphonesimulator
-configuration Debug
-destination "platform=iOS Simulator,name=<Device>,OS=9.3"
clean build
```

Xcode <https://riptutorial.com/zh-TW/ios/topic/5027/xcode>



# 106: XCTest -

## Examples

### Xcode

“”。

Language:

Devices:

Use Core Data

Include Unit Tests

Include UI Tests

。

### 1-Xcode

### 2-“”

### 3-“”

### 4-“”“Cocoa Touch Unit Test Testing Bundle”

[Your app name]Tests.swift ◦ Objective-C [Your app name]Tests.h [Your app name]Tests.m ◦

[Your app name]Tests.swift or .m

- XCTest
- A [Your app name]TestsXCTestCase
- setUp tearDown testExample testPerformanceExample

```
import XCTest

class MyProjectTests: XCTestCase {

 override func setUp() {
 super.setUp()
 // Put setup code here. This method is called before the invocation of each test method in
 the class.
 }

 override func tearDown() {
 // Put teardown code here. This method is called after the invocation of each test method
 in the class.
 }
}
```

```

 super.tearDown()
}

func testExample() {
 // This is an example of a functional test case.
 // Use XCTAssert and related functions to verify your tests produce the correct results.
}

func testPerformanceExample() {
 // This is an example of a performance test case.
 self.measure {
 // Put the code you want to measure the time of here.
 }
}
}

```

## Objective-C

```

#import <XCTest/XCTest.h>

@interface MyProjectTests : XCTestCase

@end

@implementation MyProjectTests

- (void)setUp {
 [super setUp];
 // Put setup code here. This method is called before the invocation of each test method in the
 class.
}

- (void)tearDown {
 // Put teardown code here. This method is called after the invocation of each test method in
 the class.
 [super tearDown];
}

- (void)testExample {
 // This is an example of a functional test case.
 // Use XCTAssert and related functions to verify your tests produce the correct results.
}

- (void)testPerformanceExample {
 // This is an example of a performance test case.
 [self measureBlock:^(
 // Put the code you want to measure the time of here.
)];
}

@end

```

## StoryboardView Controller

View ControllerStoryboard

```
var viewController : ViewController!
```

## StoryboardView Controller

setUp()

```
let storyboard = UIStoryboard(name: "Main", bundle: nil)
viewController = storyboard.instantiateInitialViewController() as! ViewController
```

## Objective-C

```
UINavigationController *storyboard = [UINavigationController storyboardWithName:@"Main" bundle:nil];
viewController = (ViewController *) [storyboard instantiateInitialViewController];
```

- View ControllerStoryboard◦

Apple

testvoidvoidtestColorIsRed◦ API◦ ◦

“test”

```
func testSomething() {
}
```

## Objective-C

```
- (void)testSomething {
}
```

XCTAssert()true◦

View Controller<sub>sum()</sub>◦

```
func testSum(){
 let result = viewController.sum(4, and: 5)
 XCTAssertEqual(result, 9)
}
```

## Objective-C

```
- (void)testSum {
 int result = [viewController sum:4 and:5];
 XCTAssertEqual(result, 9);
}
```

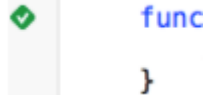
StoryboardView ControllerUI. View ControllerloadView()◦ loadView()viewviewController  
◦ UI

```
XCTAssertNotNil(viewController.view)
```

◦

◦

◦



```
func
}
```

◦



```
func
}
```

```
Product -> Test OR Cmd + U
```

```
internal◦ ◦ ◦ @testable◦
```

```
ToDo◦
```

```
@testable import ToDo
```

```
importToDointernalinternal◦
```

◦

```
loadView()viewDidLoad()◦ ◦ sut
```

```
XCTAssertNotNil(sut.view)
```

---

```
viewWillAppear(_:)viewDidAppear(_:)
```

```
sut.beginAppearanceTransition(true, animated: true)
sut.endAppearanceTransition()
```

```
import XCTest
@testable import PersonApp

class PersonTests: XCTestCase {
 func test_completeName() {
 let person = Person(firstName: "Josh", lastName: "Brown")
 XCTAssertEqual(person.completeName(), "Josh Brown")
 }
}
```

◦ import XCTestXCTestCaseXCTAssertEqual ◦ XCTestCasetestXcodeU > ◦ @testable import PersonApp  
PersonAppPerson ◦ XCTAssertEqualperson.completeName()"Josh Brown" ◦

**XCTest** - <https://riptutorial.com/zh-TW/ios/topic/5075/xctest--->

# 107: NSBundle

## Examples

### 1. Cocoa bundle.

#### Cocoa **NSBundle mainBundle**.

```
NSBundle *mainBundle;
// Get the main bundle for the app;
mainBundle = [NSBundle mainBundle];
```

### 2. Core Foundation.

#### **CFBundleGetMainBundleC**.

```
CFBundleRef mainBundle;
// Get the main bundle for the app
mainBundle = CFBundleGetMainBundle();
```

### 1. Cocoa

#### Cocoa bundle **NSBundle bundleWithPath** class

```
NSBundle *myBundle;
// obtain a reference to a loadable bundle
myBundle = [NSBundle bundleWithPath:@"~/Library/MyBundle.bundle"];
```

### 2. Cocoa Foundation

#### Core Foundation **CFBundleCreateCFURLRef**.

```
CFURLRef bundleURL;
CFBundleRef myBundle;
// Make a CFURLRef from the CFString representation of the bundle's path.
bundleURL = CFURLCreateWithFileSystemPath(kCFAllocatorDefault,
CFSTR("~/Library/MyBundle.bundle"), kCFURLPOSIXPathStyle, true);
// Make a bundle instance using the URLRef.
myBundle = CFBundleCreate(kCFAllocatorDefault, bundleURL);
// You can release the URL now.
CFRelease(bundleURL);
// Use the bundle ...
// Release the bundle when done.
CFRelease(myBundle);
```

**NSBundle** <https://riptutorial.com/zh-TW/ios/topic/5862/nsbundle>

# 108:

## Grand Central Dispatch

- `dispatch_async` - ◦ `dispatch_async``dispatch_async`
- `dispatch_sync` - ◦ `dispatch_async`

```
◦ ;◦ dispatch_get_main_queuedispatch_queue_create("QUEUE_NAME",
DISPATCH_QUEUE_CONCURRENT) ◦ ◦ dispatch_asyncdispatch_sync ◦ ; DISPATCH_QUEUE_CONCURRENT
DISPATCH_QUEUE_SERIAL
```

```
queue;◦ XcodeXcode
```

- `UIView`/◦

## Examples

-

“Foo”“Bar”◦ ◦

```
dispatch_async(dispatch_queue_create("Foo", DISPATCH_QUEUE_CONCURRENT), ^{
 for (int i = 0; i < 100; i++) {
 NSLog(@"Foo");
 usleep(100000);
 }
});

for (int i = 0; i < 100; i++) {
 NSLog(@"Bar");
 usleep(50000);
}
```

“Foo”100100ms◦ Foo “Bar”50ms◦ “Foo”“Bars”

- REST API `UILabel`◦ `UILabel`

```
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{
 //Perform expensive tasks
 //...

 //Now before updating the UI, ensure we are back on the main thread
 dispatch_async(dispatch_get_main_queue(), ^{
 label.text = //....
 });
})
```

◦

- o

```
dispatch_group_t preapreWaitingGroup = dispatch_group_create();

dispatch_group_enter(preapreWaitingGroup);
[self doAsynchronousTaskWithComplete:^(id someResults, NSError *error) {
 // Notify that this task has been completed.
 dispatch_group_leave(preapreWaitingGroup);
}]

dispatch_group_enter(preapreWaitingGroup);
[self doOtherAsynchronousTaskWithComplete:^(id someResults, NSError *error) {
 dispatch_group_leave(preapreWaitingGroup);
}]

dispatch_group_notify(preapreWaitingGroup, dispatch_get_main_queue(), ^{
 // This block will be executed once all above threads completed and call
 dispatch_group_leave
 NSLog(@"Prepare completed. I'm readyyyy");
});
```

## 1. Swift 3.

```
let prepareGroup = DispatchGroup()
prepareGroup.enter()
doAsynchronousTaskWithComplete() { (someResults, error) in
 // Notify that this task has been completed.
 prepareGroup.leave()
}

prepareGroup.enter()
doOtherAsynchronousTaskWithComplete() { (someResults, error) in
 // Notify that this task has been completed.
 prepareGroup.leave()
}

prepareGroup.notify(queue: DispatchQueue.main) {
 // This block will be executed once all above threads completed and call
 dispatch_group_leave
 print("Prepare completed. I'm readyyyy")
}
```

<https://riptutorial.com/zh-TW/ios/topic/1090/>



---

# 109:

## Examples

XCodeIPA◦ <https://developer.apple.com/account/ios/profile/create>

- 
- iOS/ tvOS - ◦
  - App Store / tvOS App Store - ◦
  - - ◦
  - Ad Hoc / tvOS Ad Hoc - UDID◦

<https://riptutorial.com/zh-TW/ios/topic/6055/>

# 110: CoreBluetooth

- 
- ◦
- iPhoneLittle EndianLittle Endian◦
  - intel CPU◦
  - ARM3big-endian◦
- ◦

---

## UUID

```
func SearchBLE(){
 cb_manager.scanForPeripherals(withServices:[service_uuid], options: nil)
 StopSearchBLE()
}
```

---

## SERVICE UUID

```
func centralManager(_ central: CBCentralManager, didConnect peripheral:
CBPeripheral) {
 peripheral.delegate = self
 peripheral.discoverServices(nil)
}

func peripheral(_ peripheral: CBPeripheral, didDiscoverServices error: Error?) {
 for service in peripheral.services! {
 print("Service: \(service)\n error: \(error)")
 }
}
```

- discoverServicesnil - NIL◦ 3
- SERVICE UUID

```
Service: <CBService: 0x171e75280, isPrimary = YES, UUID = Battery>
error: nil
Service: <CBService: 0x171e74c40, isPrimary = YES, UUID = Device Information>
error: nil
Service: <CBService: 0x171e75300, isPrimary = YES, UUID = FFF0>
error: nil
```

- 3FFF0
- uuid
- FFF0SERVICE UUID

---

## UInt16

```

protocol DataConvertible {
 init?(data: Data)
 var data: Data { get }
}

extension DataConvertible {

 init?(data: Data) {
 guard data.count == MemoryLayout<Self>.size else { return nil }
 self = data.withUnsafeBytes { $0.pointee }
 }

 var data: Data {
 var value = self
 return Data(buffer: UnsafeBufferPointer(start: &value, count: 1))
 }
}

extension UInt16 : DataConvertible {
 init?(data: Data) {
 guard data.count == MemoryLayout<UInt16>.size else { return nil }
 self = data.withUnsafeBytes { $0.pointee }
 }
 var data: Data {
 var value = CFSwapInt16HostToBig(self)
 return Data(buffer: UnsafeBufferPointer(start: &value, count: 1))
 }
}

```

## Examples

### BLE

- BLE。
- CBCentralManagerDelegate。
- centralManagerDidUpdateState\_ centralCBCentralManager。
- 。
- CBCentralManagercentralManagerDidUpdateState。

```

class BLEController: CBCentralManagerDelegate{

var cb_manager: CBCentralManager!
var bles : [CBPeripheral] = []

 override func viewDidLoad() {
 super.viewDidLoad()
 cb_manager = CBCentralManager(delegate: self, queue: DispatchQueue.global())
 }

 func centralManagerDidUpdateState(_ central: CBCentralManager) {
 print("UPDATE STATE - \(central)")
 }
}

```

centralManagerDidUpdateStateCoreBluetoothBLE。 centralManagerDidUpdateStateBLE。

```

func centralManagerDidUpdateState(_ central: CBCentralManager) {
 print("UPDATE STATE - \(central)")
 SearchBLE()
}

func SearchBLE() {
 cb_manager.scanForPeripherals(withServices: nil, options: nil)
 StopSearchBLE()
}

func StopSearchBLE() {
 let when = DispatchTime.now() + 5 // change 5 to desired number of seconds
 DispatchQueue.main.asyncAfter(deadline: when) {
 self.cb_manager.stopScan()
 }
}

```

- SearchBLEBLE5s
- cb\_manager.scanForPeripheralswithServicesniloptionsnilBLE。
- StopSearchBLE5s。
- BLEfunc centralManager\_ centralCBCentralManagerdidDiscover peripheralCBPeripheral advertisementData[StringAny]rssi RSSINSNumber

```

func centralManager(_ central: CBCentralManager, didDiscover peripheral:
 CBPeripheral, advertisementData: [String : Any], rssi RSSI: NSNumber) {
 guard let name = peripheral.name else {
 return
 }
 print(name)
 bles.append(peripheral)
}

```

- IBEACON。
- BLEControllerCBPeripheralDelegate
- BLE。
- StopSearchBLE

```

class BLEController: CBCentralManagerDelegate, CBPeripheralDelegate {
//...
 func StopSearchMiniewBeacon() {
 let when = DispatchTime.now() + 5 // change 2 to desired number of seconds
 DispatchQueue.main.asyncAfter(deadline: when) {
 self.cb_manager.stopScan()
 self.cb_manager.connect(bles.first)
 }
 }
//...
}

```

- BLESERVICE UUIDMAJOR UUID CHARACTERISTIC

```

var service_uuid = CBUUID(string: "0000fff0-0000-1000-8000-00805f9b34fb")
var major_uuid = CBUUID(string: "0000fff2-0000-1000-8000-00805f9b34fb")

```

```

func centralManager(_ central: CBCentralManager, didConnect peripheral:
CBPeripheral) {
 peripheral.delegate = self
 peripheral.discoverServices([service_uuid])
}

func peripheral(_ peripheral: CBPeripheral, didDiscoverServices error: Error?) {
 print("Service: \(service)\n error: \(error)")
 peripheral.discoverCharacteristics([major_uuid], for: (peripheral.services?[0]!))
}

```

- 'service\_uuid' major\_uuid'. '-0000-1000-8000-00805f9b34fb'. 'fff0'UUID'fff2'MAJOR UUID '0000'4uuid1°.
- discoverCharacteristics[major\_uuid]forperipheral.services[0]gattNIL°.
- peripheral.services?[0] - peripheral.discoverServices[service\_uuid]0 beacuse

```

func peripheral(_ peripheral: CBPeripheral, didDiscoverCharacteristicsFor service: CBService,
error: Error?) {
 for characteristic in service.characteristics! {
 print("Characteristic: \(characteristic)\n error: \(error)")
 if(characteristic.uuid.uuidString == "FFF2"){
 peripheral.readValue(for: characteristic)
 }
 }
}

func peripheral(_ peripheral: CBPeripheral, didUpdateValueFor characteristic:
CBCharacteristic, error: Error?) {
 print("Characteristic read: \(characteristic)\n error: \(error)")
 let major = UInt16.init(bigEndian: UInt16(data: characteristic.value!))
 print("major: \(major)")
}

```

- peripheral.readValueforcharacteristic
- readValuefunc\_ peripheralCBPeripheraldidUpdateValueForCBCharacteristicvalue°.
- °.
- ° func\_ peripheralCBPeripheraldidUpdateValueForCBCharacteristic
- new\_majorreset\_characteristic

```

var reset_characteristic : CBCharacteristic!
func peripheral(_ peripheral: CBPeripheral, didDiscoverCharacteristicsFor service: CBService,
error: Error?) {
 for characteristic in service.characteristics! {
 print("Characteristic: \(characteristic)\n error: \(error)")
 if(characteristic.uuid.uuidString == "FFF2"){
 peripheral.readValue(for: characteristic)
 }
 if(characteristic.uuid.uuidString == "FFFF"){
 reset_characteristic = characteristic
 }
 }
}
let new_major : UInt16 = 100

```

```
func peripheral(_ peripheral: CBPeripheral, didUpdateValueFor characteristic:
CBCharacteristic, error: Error?) {
 print("Characteristic read: \(characteristic)\n error: \(error)")
 let major = UInt16.init(bigEndian: UInt16(data: characteristic.value!))
 print("major: \(major)")
 peripheral.writeValue(new_major.data, for: characteristic, type:
CBCharacteristicWriteType.withResponse)
}
```

- `defaultiPhoneLittle EndianMINEWNR51822ARMBig Endian`。
- `BLECBCharacteristicWriteType.withResponse`。

```
func peripheral(_ peripheral: CBPeripheral, didWriteValueFor characteristic: CBCharacteristic,
error: Error?) {
 print("Characteristic write: \(characteristic)\n error: \(error)")
 if(characteristic.uuid.uuidString == "FFF2"){
 print("Resetting")
 peripheral.writeValue("minew123".data(using: String.Encoding.utf8)!, for:
reset_characteristic, type: CBCharacteristicWriteType.withResponse)
 }
 if(characteristic.uuid.uuidString == "FFFF"){
 print("Reboot finish")
 cb_manager.cancelPeripheralConnection(peripheral)
 }
}
```

- `gatt`。
- `FFFF`。
- `'minew123'`。
- `.`

```
func peripheral(_ peripheral: CBPeripheral, didUpdateValueFor characteristic:
CBCharacteristic, error: Error?) {
 print("Characteristic read: \(characteristic)\n error: \(error)")
 let major = UInt16.init(bigEndian: UInt16(data: characteristic.value!))
 print("major: \(major)")
 //peripheral.writeValue(new_major.data, for: characteristic, type:
CBCharacteristicWriteType.withResponse)
```

```
}
```

- `didUpdateValueFor`。

**CoreBluetooth** <https://riptutorial.com/zh-TW/ios/topic/9488/corebluetooth>

# 111: CoreImage / OpenCV

## Examples

### Objective-C

#### ViewController

```
#import <CoreImage/CoreImage.h>
#import <CoreImage/CoreImage.h>
#import <QuartzCore/QuartzCore.h>
```

```
[self faceDetector];
```

```
-(void)faceDetector
{
 // Load the picture for face detection
 UIImageView* image = [[UIImageView alloc] initWithImage:[UIImage
imageNamed:@"download.jpeg"]];

 // Draw the face detection image
 [self.view addSubview:image];

 // Execute the method used to markFaces in background
 [self performSelectorInBackground:@selector(markFaces:) withObject:image];

 // flip image on y-axis to match coordinate system used by core image
 [image setTransform:CGAffineTransformMakeScale(1, -1)];

 // flip the entire window to make everything right side up
 [self.view setTransform:CGAffineTransformMakeScale(1, -1)];
}
```

```
//Adds face squares and color masks to eyes and mouth
-(void)markFaces:(UIImageView *)facePicture
{
 // draw a CI image with the previously loaded face detection picture
 CIImage* image = [CIImage imageWithCGImage:facePicture.image.CGImage];

 // create a face detector - since speed is not an issue we'll use a high accuracy
 // detector
 CIDetector* detector = [CIDetector detectorOfType:CIDetectorTypeFace
context:nil options:[NSDictionary
dictionaryWithObject:CIDetectorAccuracyHigh forKey:CIDetectorAccuracy]];

 // create an array containing all the detected faces from the detector
 NSArray* features = [detector featuresInImage:image];
 NSLog(@"Number of faces %d",[features count]);

 // we'll iterate through every detected face. CIFaceFeature provides us
 // with the width for the entire face, and the coordinates of each eye
 // and the mouth if detected. Also provided are BOOL's for the eye's and
```

```

// mouth so we can check if they already exist.
// for (features in image)
// {
for(CIFaceFeature* faceFeature in features)
{
 // get the width of the face
 CGFloat faceWidth = faceFeature.bounds.size.width;

 // create a UIView using the bounds of the face
 UIView* faceView = [[UIView alloc] initWithFrame:faceFeature.bounds];

 // add a border around the newly created UIView
 faceView.layer.borderWidth = 1;
 faceView.layer.borderColor = [[UIColor redColor] CGColor];

 // add the new view to create a box around the face
 [self.view addSubview:faceView];

 if(faceFeature.hasLeftEyePosition)
 {
 // create a UIView with a size based on the width of the face
 UIView* leftEyeView = [[UIView alloc]
initWithFrame:CGRectMake(faceFeature.leftEyePosition.x-faceWidth*0.15,
faceFeature.leftEyePosition.y-faceWidth*0.15, faceWidth*0.3, faceWidth*0.3)];
 // change the background color of the eye view
 [leftEyeView setBackgroundColor:[UIColor blueColor]
colorWithAlphaComponent:0.3]];
 // set the position of the leftEyeView based on the face
 [leftEyeView setCenter:faceFeature.leftEyePosition];
 // round the corners
 leftEyeView.layer.cornerRadius = faceWidth*0.15;
 // add the view to the window
 [self.view addSubview:leftEyeView];
 }

 if(faceFeature.hasRightEyePosition)
 {
 // create a UIView with a size based on the width of the face
 UIView* leftEye = [[UIView alloc]
initWithFrame:CGRectMake(faceFeature.rightEyePosition.x-faceWidth*0.15,
faceFeature.rightEyePosition.y-faceWidth*0.15, faceWidth*0.3, faceWidth*0.3)];
 // change the background color of the eye view
 [leftEye setBackgroundColor:[UIColor blueColor] colorWithAlphaComponent:0.3]];
 // set the position of the rightEyeView based on the face
 [leftEye setCenter:faceFeature.rightEyePosition];
 // round the corners
 leftEye.layer.cornerRadius = faceWidth*0.15;
 // add the new view to the window
 [self.view addSubview:leftEye];
 }

 if(faceFeature.hasMouthPosition)
 {
 // create a UIView with a size based on the width of the face
 UIView* mouth = [[UIView alloc]
initWithFrame:CGRectMake(faceFeature.mouthPosition.x-faceWidth*0.2,
faceFeature.mouthPosition.y-faceWidth*0.2, faceWidth*0.4, faceWidth*0.4)];
 // change the background color for the mouth to green
 [mouth setBackgroundColor:[UIColor greenColor] colorWithAlphaComponent:0.3]];
 // set the position of the mouthView based on the face
 [mouth setCenter:faceFeature.mouthPosition];
 }
}
}

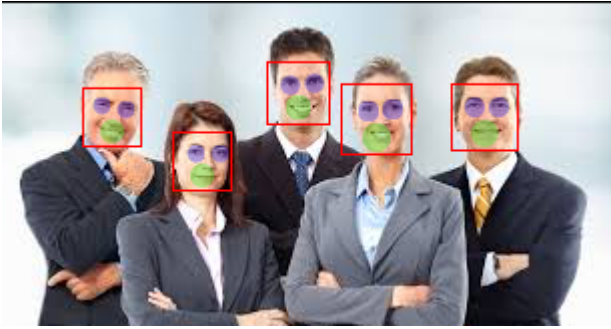
```



```
 // round the corners
 mouth.layer.cornerRadius = faceWidth*0.2;
 // add the new view to the window
 [self.view addSubview:mouth];
 }
}

// }
}
```

## ScreenShot



CoreImage / OpenCV <https://riptutorial.com/zh-TW/ios/topic/7298/coreimage---opencv>

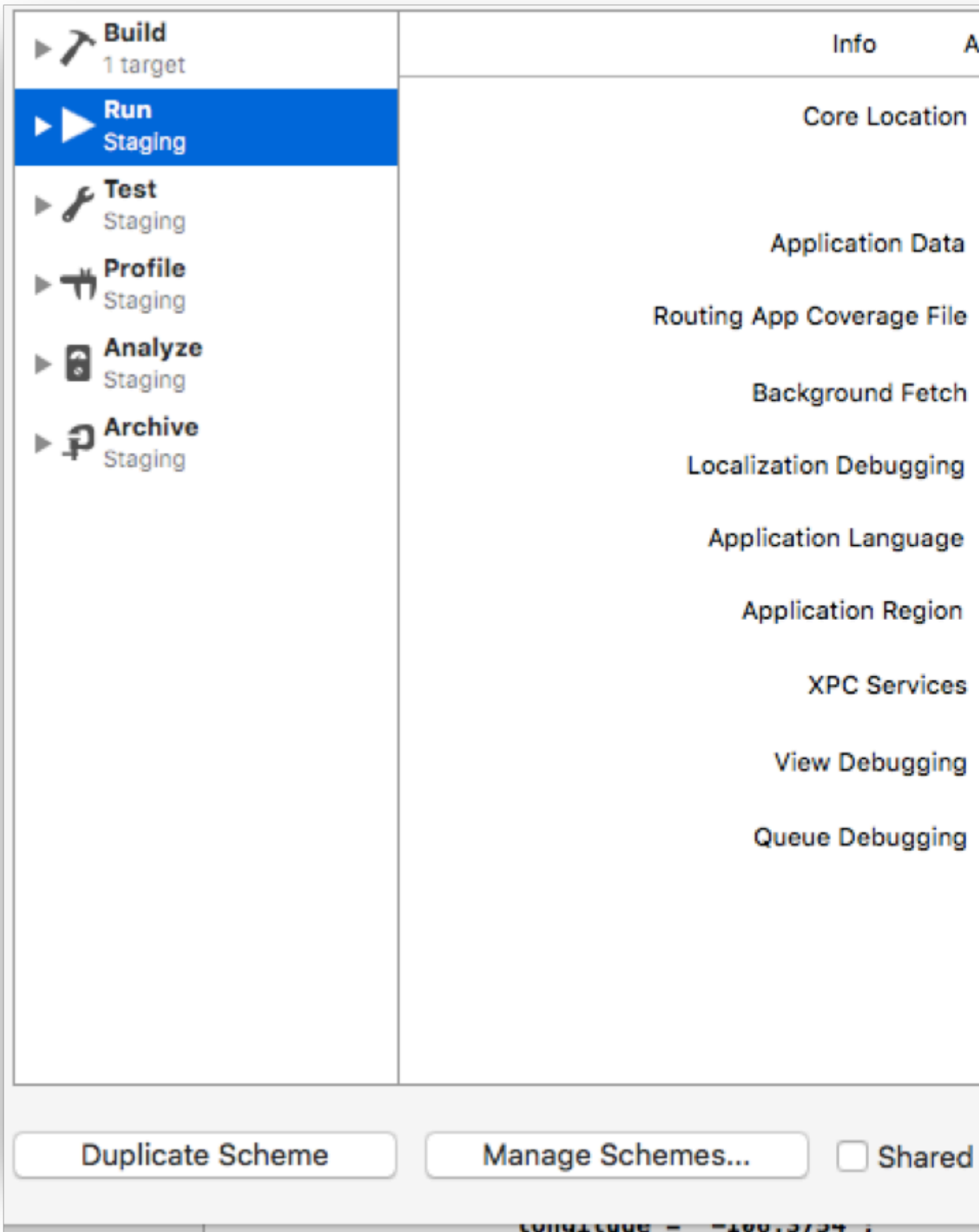
# 112: GPXios

## Examples

### .gpxMPS\_HQ.gpx

```
<gpx xmlns="http://www.topografix.com/GPX/1/1"
 xmlns:gpxx = "http://www.garmin.com/xmlschemas/GpxExtensions/v3"
 xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.topografix.com/GPX/1/1
http://www.topografix.com/GPX/1/1/gpx.xsd
http://www.garmin.com/xmlschemas/GpxExtensions/v3
http://www8.garmin.com/xmlschemas/GpxExtensions/v3/GpxExtensionsv3.xsd"
 version="1.1"
 creator="gpx-poi.com">
<wpt lat="38.9072" lon="77.0369">38.9072/-77.0369
<time>2015-04-16T22:20:29Z</time>
 <name>Washington, DC</name>
 <extensions>
 <gpxx:WaypointExtension>
 <gpxx:Proximity>10</gpxx:Proximity>
 <gpxx:Address>
 <gpxx:StreetAddress>Washington DC</gpxx:StreetAddress>
 <gpxx:City>Washington</gpxx:City>
 <gpxx:State>DC</gpxx:State>
 <gpxx:Country>United States</gpxx:Country>
 <gpxx:PostalCode> 20005 </gpxx:PostalCode>
 </gpxx:Address>
 </gpxx:WaypointExtension>
 </extensions>
```

1. °
2. Run - > Options°
3. ""°
4. ""\* .GPX°



GPXiOS <https://riptutorial.com/zh-TW/ios/topic/9883/gpxios>

# 113: UICollectionViewUIKit Dynamics

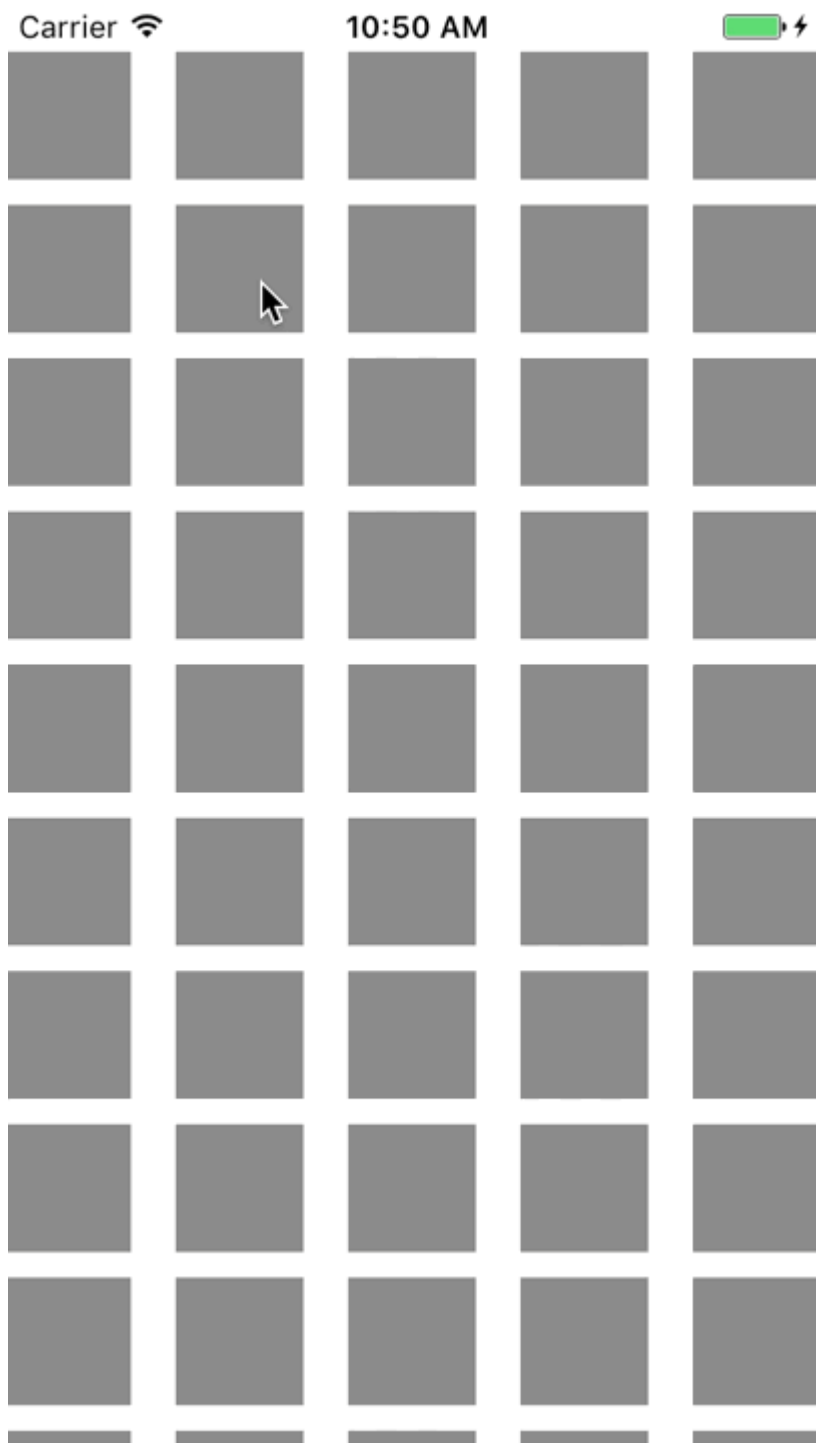
UIKit DynamicsUIKit ◦ UIKit DynamicsAPIUICollectionViewUICollectionViewLayout

## Examples

### UIDynamicAnimator

Subclassing UIDynamicBehaviorUICollectionViewFlowLayout ◦ UICollectionView ◦ UIDynamicAnimator“”

◦



UIAttachmentBehaviorUIDynamicItemUIDynamicItems ◦

RectangleAttachmentBehavior

```
final class RectangleAttachmentBehavior: UIDynamicBehavior
{
 init(item: UIDynamicItem, point: CGPoint)
 {
 // Higher frequency more "ridged" formation
 let frequency: CGFloat = 8.0

 // Lower damping longer animation takes to come to rest
 let damping: CGFloat = 0.6

 super.init()

 // Attachment points are four corners of item
 let points = self.attachmentPoints(for: point)

 let attachmentBehaviors: [UIAttachmentBehavior] = points.map
 {
 let attachmentBehavior = UIAttachmentBehavior(item: item, attachedToAnchor: $0)
 attachmentBehavior.frequency = frequency
 attachmentBehavior.damping = damping
 return attachmentBehavior
 }

 attachmentBehaviors.forEach
 {
 addChildBehavior($0)
 }
 }

 func updateAttachmentLocation(with point: CGPoint)
 {
 // Update anchor points to new attachment points
 let points = self.attachmentPoints(for: point)
 let attachments = self.childBehaviors.flatMap { $0 as? UIAttachmentBehavior }
 let pairs = zip(points, attachments)
 pairs.forEach { $0.1.anchorPoint = $0.0 }
 }

 func attachmentPoints(for point: CGPoint) -> [CGPoint]
 {
 // Width and height should be close to the width and height of the item
 let width: CGFloat = 40.0
 let height: CGFloat = 40.0

 let topLeft = CGPoint(x: point.x - width * 0.5, y: point.y - height * 0.5)
 let topRight = CGPoint(x: point.x + width * 0.5, y: point.y - height * 0.5)
 let bottomLeft = CGPoint(x: point.x - width * 0.5, y: point.y + height * 0.5)
 let bottomRight = CGPoint(x: point.x + width * 0.5, y: point.y + height * 0.5)
 let points = [topLeft, topRight, bottomLeft, bottomRight]
 return points
 }
}
```

## Objective-C

```

@implementation RectangleAttachmentBehavior

- (instancetype)initWithItem:(id<UIDynamicItem>)item point:(CGPoint)point
{
 CGFloat frequency = 8.0f;
 CGFloat damping = 0.6f;
 self = [super init];
 if (self)
 {
 NSArray <NSValue *> *pointValues = [self attachmentPointValuesForPoint:point];
 for (NSValue *value in pointValues)
 {
 UIAttachmentBehavior *attachment = [[UIAttachmentBehavior alloc] initWithItem:item
attachedToAnchor:[value CGPointValue]];
 attachment.frequency = frequency;
 attachment.damping = damping;
 [self addChildBehavior:attachment];
 }
 }
 return self;
}

- (void)updateAttachmentLocationWithPoint:(CGPoint)point
{
 NSArray <NSValue *> *pointValues = [self attachmentPointValuesForPoint:point];
 for (NSInteger i = 0; i < pointValues.count; i++)
 {
 NSValue *pointValue = pointValues[i];
 UIAttachmentBehavior *attachment = self.childBehaviors[i];
 attachment.anchorPoint = [pointValue CGPointValue];
 }
}

- (NSArray <NSValue *> *)attachmentPointValuesForPoint:(CGPoint)point
{
 CGFloat width = 40.0f;
 CGFloat height = 40.0f;

 CGPoint topLeft = CGPointMake(point.x - width * 0.5, point.y - height * 0.5);
 CGPoint topRight = CGPointMake(point.x + width * 0.5, point.y - height * 0.5);
 CGPoint bottomLeft = CGPointMake(point.x - width * 0.5, point.y + height * 0.5);
 CGPoint bottomRight = CGPointMake(point.x + width * 0.5, point.y + height * 0.5);

 NSArray <NSValue *> *pointValues = @[[NSValue valueWithCGPoint:topLeft], [NSValue
valueWithCGPoint:topRight], [NSValue valueWithCGPoint:bottomLeft], [NSValue
valueWithCGPoint:bottomRight]];
 return pointValues;
}

@end

```

RectangleAttachmentBehavior◊

```

final class DragBehavior: UIDynamicBehavior
{
 init(items: [UIDynamicItem], point: CGPoint)
 {
 super.init()
 items.forEach
 {

```

```

 let rectAttachment = RectangleAttachmentBehavior(item: $0, point: point)
 self.addChildBehavior(rectAttachment)
 }
}

func updateDragLocation(with point: CGPoint)
{
 // Tell low-level behaviors location has changed
 self.childBehaviors.flatMap { $0 as? RectangleAttachmentBehavior }.forEach {
$0.updateAttachmentLocation(with: point) }
}
}

```

## Objective-C

```

@implementation DragBehavior

- (instancetype)initWithItems:(NSArray <id<UIDynamicItem>> *)items point: (CGPoint)point
{
 self = [super init];
 if (self)
 {
 for (id<UIDynamicItem> item in items)
 {
 RectangleAttachmentBehavior *rectAttachment = [[RectangleAttachmentBehavior
alloc] initWithItem:item point:point];
 [self addChildBehavior:rectAttachment];
 }
 }
 return self;
}

- (void)updateDragLocationWithPoint:(CGPoint)point
{
 for (RectangleAttachmentBehavior *rectAttachment in self.childBehaviors)
 {
 [rectAttachment updateAttachmentLocationWithPoint:point];
 }
}

@end

```

- UICollectionViewFlowLayout ◦ layoutAttributesForElementsInRectUIDynamicAnimator's itemsInRect ◦

```

final class DraggableLayout: UICollectionViewFlowLayout
{
 // Array that holds dragged index paths
 var indexPathsForDraggingElements: [IndexPath]?

 // The dynamic animator that will animate drag behavior
 var animator: UIDynamicAnimator?

 // Custom high-level behavior that dictates drag animation
 var dragBehavior: DragBehavior?

 // Where dragging starts so can return there once dragging ends
 var startDragPoint = CGPoint.zero
}

```

```

// Bool to keep track if dragging has ended
var isFinishedDragging = false

// Method to inform layout that dragging has started
func startDragging(indexPaths selectedIndexPaths: [IndexPath], from point: CGPoint)
{
 indexPathsForDraggingElements = selectedIndexPaths
 animator = UIDynamicAnimator(collectionViewLayout: self)
 animator?.delegate = self

 // Get all of the draggable attributes but change zIndex so above other cells
 let draggableAttributes: [UICollectionViewLayoutAttributes] =
selectedIndexPaths.flatMap {
 let attribute = super.layoutAttributesForItem(at: $0)
 attribute?.zIndex = 1
 return attribute
 }

 startDragPoint = point

 // Add them to high-level behavior
 dragBehavior = DragBehavior(items: draggableAttributes, point: point)

 // Add high-level behavior to animator
 animator?.addBehavior(dragBehavior!)
}

func updateDragLocation(_ point: CGPoint)
{
 // Tell high-level behavior that point has updated
 dragBehavior?.updateDragLocation(with: point)
}

func endDragging()
{
 isFinishedDragging = true

 // Return high-level behavior to starting point
 dragBehavior?.updateDragLocation(with: startDragPoint)
}

func clearDraggedIndexPaths()
{
 // Reset state for next drag event
 animator = nil
 indexPathsForDraggingElements = nil
 isFinishedDragging = false
}

override func layoutAttributesForElements(in rect: CGRect) ->
[UICollectionViewLayoutAttributes]?
{
 let existingAttributes: [UICollectionViewLayoutAttributes] =
super.layoutAttributesForElements(in: rect) ?? []
 var allAttributes = [UICollectionViewLayoutAttributes]()

 // Get normal flow layout attributes for non-drag items
 for attributes in existingAttributes
 {

```



```

 if (indexPathsForDraggingElements?.contains(attributes.indexPath) ?? false) ==
false
 {
 allAttributes.append(attributes)
 }
 }

 // Add dragged item attributes by asking animator for them
 if let animator = self.animator
 {
 let animatorAttributes: [UICollectionViewLayoutAttributes] = animator.items(in:
rect).flatMap { $0 as? UICollectionViewLayoutAttributes }
 allAttributes.append(contentsOf: animatorAttributes)
 }
 return allAttributes
}
}
extension DraggableLayout: UIDynamicAnimatorDelegate
{
 func dynamicAnimatorDidPause(_ animator: UIDynamicAnimator)
 {
 // Animator has paused and done dragging; reset state
 guard isFinishedDragging else { return }
 clearDraggedIndexPaths()
 }
}
}

```

## Objective-C

```

@interface DraggableLayout () <UIDynamicAnimatorDelegate>
@property (nonatomic, strong) NSArray <NSIndexPath *> *indexPathsForDraggingElements;
@property (nonatomic, strong) UIDynamicAnimator *animator;
@property (nonatomic, assign) CGPoint startDragPoint;
@property (nonatomic, assign) BOOL finishedDragging;
@property (nonatomic, strong) DragBehavior *dragBehavior;
@end

@implementation DraggableLayout

- (void)startDraggingWithIndexPaths:(NSArray <NSIndexPath *> *)selectedIndexPaths
fromPoint:(CGPoint)point
{
 self.indexPathsForDraggingElements = selectedIndexPaths;
 self.animator = [[UIDynamicAnimator alloc] initWithCollectionViewLayout:self];
 self.animator.delegate = self;
 NSMutableArray *draggableAttributes = [[NSMutableArray
alloc] initWithCapacity:selectedIndexPaths.count];
 for (NSIndexPath *indexPath in selectedIndexPaths)
 {
 UICollectionViewLayoutAttributes *attributes = [super
layoutAttributesForItemAtIndexPath:indexPath];
 attributes.zIndex = 1;
 [draggableAttributes addObject:attributes];
 }
 self.startDragPoint = point;
 self.dragBehavior = [[DragBehavior alloc] initWithItems:draggableAttributes point:point];
 [self.animator addBehavior:self.dragBehavior];
}
}

```

```

- (void)updateDragLoactionWithPoint:(CGPoint)point
{
 [self.dragBehavior updateDragLocationWithPoint:point];
}

- (void)endDragging
{
 self.finishedDragging = YES;
 [self.dragBehavior updateDragLocationWithPoint:self.startDragPoint];
}

- (void)clearDraggedIndexPath
{
 self.animator = nil;
 self.indexPathsForDraggingElements = nil;
 self.finishedDragging = NO;
}

- (void)dynamicAnimatorDidPause:(UIDynamicAnimator *)animator
{
 if (self.finishedDragging)
 {
 [self clearDraggedIndexPath];
 }
}

- (NSArray<UICollectionViewLayoutAttributes *>
*)layoutAttributesForElementsInRect:(CGRect)rect
{
 NSArray *existingAttributes = [super layoutAttributesForElementsInRect:rect];
 NSMutableArray *allAttributes = [[NSMutableArray
alloc] initWithCapacity:existingAttributes.count];
 for (UICollectionViewLayoutAttributes *attributes in existingAttributes)
 {
 if (![self.indexPathsForDraggingElements containsObject:attributes.indexPath])
 {
 [allAttributes addObject:attributes];
 }
 }
 [allAttributes addObjectsFromArray:[self.animator itemsInRect:rect]];
 return allAttributes;
}

@end

```

UICollectionView

```

final class ViewController: UIViewController
{
 // Collection view that displays cells
 lazy var collectionView: UICollectionView =
 {
 let collectionView = UICollectionView(frame: .zero, collectionViewLayout:
DraggableLayout())
 collectionView.backgroundColor = .white
 collectionView.translatesAutoresizingMaskIntoConstraints = false
 self.view.addSubview(collectionView)
 collectionView.topAnchor.constraint(equalTo:
self.topAnchor).isActive = true
 collectionView.leadingAnchor.constraint(equalTo: self.view.leadingAnchor).isActive =

```

```

true
 collectionView.trailingAnchor.constraint(equalTo: self.view.trailingAnchor).isActive =
true
 collectionView.bottomAnchor.constraint(equalTo:
self.bottomLayoutGuide.topAnchor).isActive = true

 return collectionView
}()

// Gesture that drives dragging
lazy var longPress: UILongPressGestureRecognizer =
{
 let longPress = UILongPressGestureRecognizer(target: self, action:
#selector(self.handleLongPress(sender:)))
 return longPress
}()

// Array that holds selected index paths
var selectedIndexPaths = [IndexPath]()

override func viewDidLoad()
{
 super.viewDidLoad()
 collectionView.delegate = self
 collectionView.dataSource = self
 collectionView.register(UICollectionViewCell.self, forCellWithReuseIdentifier: "Cell")
 collectionView.addGestureRecognizer(longPress)
}

func handleLongPress(sender: UILongPressGestureRecognizer)
{
 guard let draggableLayout = collectionView.collectionViewLayout as? DraggableLayout
else { return }
 let location = sender.location(in: collectionView)
 switch sender.state
 {
 case .began:
 draggableLayout.startDragging(indexPaths: selectedIndexPaths, from: location)
 case .changed:
 draggableLayout.updateDragLocation(location)
 case .ended, .failed, .cancelled:
 draggableLayout.endDragging()
 case .possible:
 break
 }
}
}

extension ViewController: UICollectionViewDelegate, UICollectionViewDataSource
{
 func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section:
Int) -> Int
 {
 return 1000
 }

 func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath:
IndexPath) -> UICollectionViewCell
 {
 let cell = collectionView.dequeueReusableCell(withReuseIdentifier: "Cell", for:
indexPath)
 cell.backgroundColor = .gray
 }
}

```

```

 if selectedIndexPaths.contains(indexPath) == true
 {
 cell.backgroundColor = .red
 }
 return cell
 }

 func collectionView(_ collectionView: UICollectionView, didSelectItemAt indexPath:
IndexPath)
 {
 // Bool that determines if cell is being selected or unselected
 let isSelected = !selectedIndexPaths.contains(indexPath)
 let cell = collectionView.cellForItem(at: indexPath)
 cell?.backgroundColor = isSelected ? .red : .gray
 if isSelected
 {
 selectedIndexPaths.append(indexPath)
 }
 else
 {
 selectedIndexPaths.remove(at: selectedIndexPaths.index(of: indexPath)!)
 }
 }
}

```

## Objective-C

```

@interface ViewController () <UICollectionViewDelegate, UICollectionViewDataSource>
@property (nonatomic, strong) UICollectionView *collectionView;
@property (nonatomic, strong) UILongPressGestureRecognizer *longPress;
@property (nonatomic, strong) NSMutableArray <NSIndexPath *> *selectedIndexPaths;
@end

@implementation ViewController

- (void)viewDidLoad
{
 [super viewDidLoad];
 self.collectionView.delegate = self;
 self.collectionView.dataSource = self;
 [self.collectionView registerClass:[UICollectionViewCell class]
forCellWithReuseIdentifier:@"Cell"];
 [self.collectionView addGestureRecognizer:self.longPress];
 self.selectedIndexPaths = [[NSMutableArray alloc] init];
}

- (UICollectionView *)collectionView
{
 if (!_collectionView)
 {
 _collectionView = [[UICollectionView alloc] initWithFrame:CGRectZero
collectionViewLayout:[[DraggableLayout alloc] init]];
 _collectionView.backgroundColor = [UIColor whiteColor];
 _collectionView.translatesAutoresizingMaskIntoConstraints = NO;
 [self.view addSubview:_collectionView];
 [_collectionView.topAnchor
constraintEqualToAnchor:self.topLayoutGuide.bottomAnchor].active = YES;
 [_collectionView.leadingAnchor constraintEqualToAnchor:self.view.leadingAnchor].active
= YES;
 }
}

```

```

 [_collectionView.trailingAnchor
constraintEqualToAnchor:self.view.trailingAnchor].active = YES;
 [_collectionView.bottomAnchor
constraintEqualToAnchor:self.bottomLayoutGuide.topAnchor].active = YES;
 }
 return _collectionView;
}

- (UILongPressGestureRecognizer *)longPress
{
 if (!_longPress)
 {
 _longPress = [[UILongPressGestureRecognizer alloc] initWithTarget:self
action:@selector(handleLongPress:)];
 }
 return _longPress;
}

- (void)handleLongPress:(UILongPressGestureRecognizer *)sender
{
 DraggableLayout *draggableLayout = (DraggableLayout
*)self.collectionView.collectionViewLayout;
 CGPoint location = [sender locationInView:self.collectionView];
 if (sender.state == UIGestureRecognizerStateBegan)
 {
 [draggableLayout startDraggingWithIndexPaths:self.selectedIndexPaths
fromPoint:location];
 }
 else if (sender.state == UIGestureRecognizerStateChanged)
 {
 [draggableLayout updateDragLoactionWithPoint:location];
 }
 else if (sender.state == UIGestureRecognizerStateEnded || sender.state ==
UIGestureRecognizerStateCancelled || sender.state == UIGestureRecognizerStateFailed)
 {
 [draggableLayout endDragging];
 }
}

- (NSInteger)collectionView:(UICollectionView *)collectionView
numberOfItemsInSection:(NSInteger)section
{
 return 1000;
}

- (UICollectionViewCell *)collectionView:(UICollectionView *)collectionView
cellForItemAtIndexPath:(NSIndexPath *)indexPath
{
 UICollectionViewCell *cell = [collectionView
dequeueReusableCellWithReuseIdentifier:@"Cell" forIndexPath:indexPath];
 cell.backgroundColor = [UIColor grayColor];
 if ([self.selectedIndexPaths containsObject:indexPath])
 {
 cell.backgroundColor = [UIColor redColor];
 }
 return cell;
}

- (void)collectionView:(UICollectionView *)collectionView
didSelectItemAtIndexPath:(NSIndexPath *)indexPath
{

```

```
 BOOL isSelected = ![self.selectedIndexPaths containsObject:indexPath];
 UICollectionViewCell *cell = [collectionView cellForItemAtIndexPath:indexPath];
 if (isSelected)
 {
 cell.backgroundColor = [UIColor redColor];
 [self.selectedIndexPaths addObject:indexPath];
 }
 else
 {
 cell.backgroundColor = [UIColor grayColor];
 [self.selectedIndexPaths removeObject:indexPath];
 }
}

@end
```

## 2013 WWDC“UIKit Dynamics”

**UICollectionViewUIKit Dynamics** <https://riptutorial.com/zh-TW/ios/topic/10079/uicollectionviewuikit-dynamics>

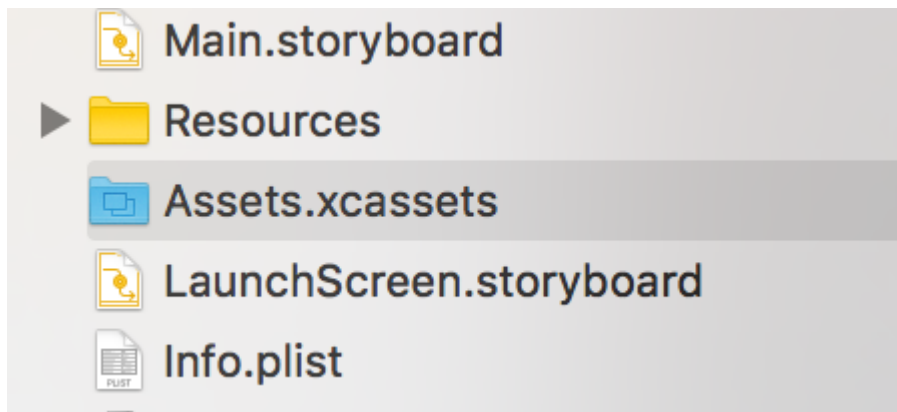
# 114: Aseets

XcodeiOS.

o

## Examples

Xcodeplist. Assets.xcassets. o







o o



◦

AppIcons◦

▼ **App Icons and Launch Images**

App Icons Source

Launch Images Source

Launch Screen File

▼ **App Icons and Launch Images**

App Icons Source  ▾ ➔

Launch Images Sourc  ▾ ➔

Launch Screen File

---

XcodeLaunchImage



AppIcon



LaunchImage

## LaunchImage

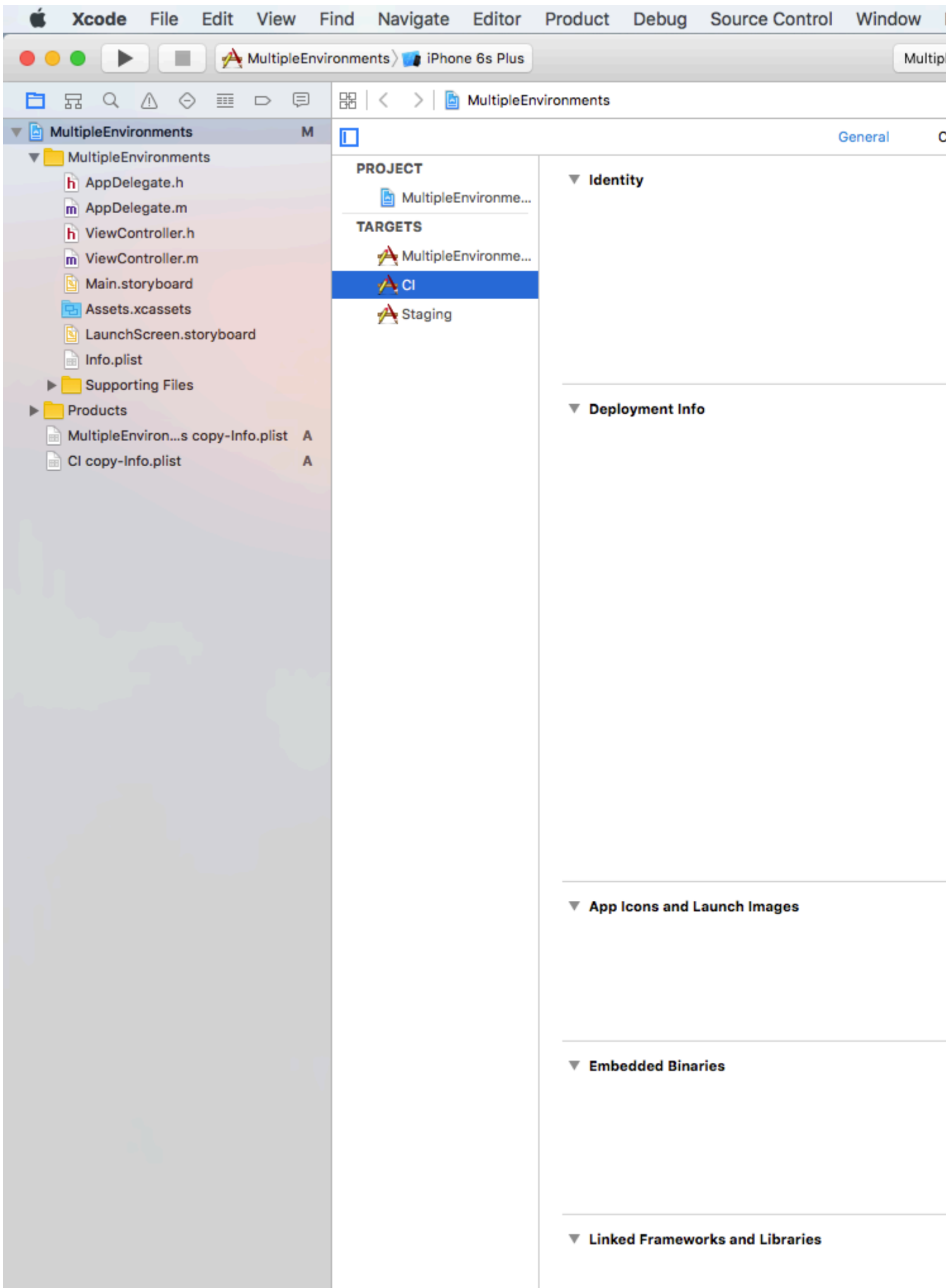
---

**115:**

## Examples

CI - ◦ URL◦

2



- CI/SERVER\_URL<http://ci.api.example.com/>
- STAGING/SERVER\_URL<http://stg.api.example.com/>



Xcode

File

Edit

View

Find

Navigate



CI >



iPhone 6s



M



MultipleEnvironments

M



MultipleEnvironments



AppDelegate.h



AppDelegate.m



ViewController.h



ViewController.m



Main.storyboard



Assets.xcassets



LaunchScreen.storyboard



Info.plist



Supporting Files



AppConfigurations.h

A



Products



MultipleEnviron...s copy-Info.plist

A



CI copy-Info.plist

A



### PROJECT



MultipleEnv

### TARGETS



MultipleEnv



CI



Staging







MultipleEn...





App CI





App STG

# 116: UIView

## Examples

### CUiView

```
#import <QuartzCore/QuartzCore.h>ViewController
```

```
UIView *view1=[[UIView alloc]init];
view1.backgroundColor=[UIColor colorWithRed:255/255.0 green:193/255.0 blue:72/255.0
alpha:1.0];
CGRect view1Frame = view1.frame;
view1Frame.size.width = SCREEN_WIDTH*0.97;
view1Frame.size.height = SCREEN_HEIGHT*0.2158;
view1Frame.origin.x = 0;
view1Frame.origin.y = 0.1422*SCREEN_HEIGHT-10;
view1.frame = view1Frame;
[self setMaskTo:view1 byRoundingCorners:UIRectCornerBottomRight|UIRectCornerTopRight];
[self.view addSubview:view1];
```

```
- (void)setMaskTo:(UIView*)view byRoundingCorners:(UIRectCorner)corners
{
 UIBezierPath *rounded = [UIBezierPath bezierPathWithRoundedRect:view.bounds
 byRoundingCorners:corners
 cornerRadii:CGSizeMake(20.0, 20.0)];

 CAShapeLayer *shape = [[CAShapeLayer alloc] init];
 [shape setPath:rounded.CGPath];
 view.layer.mask = shape;
}
```

**UIView** <https://riptutorial.com/zh-TW/ios/topic/7224/uiview>

# 117: XIBUIViews

AppleInterface Builder

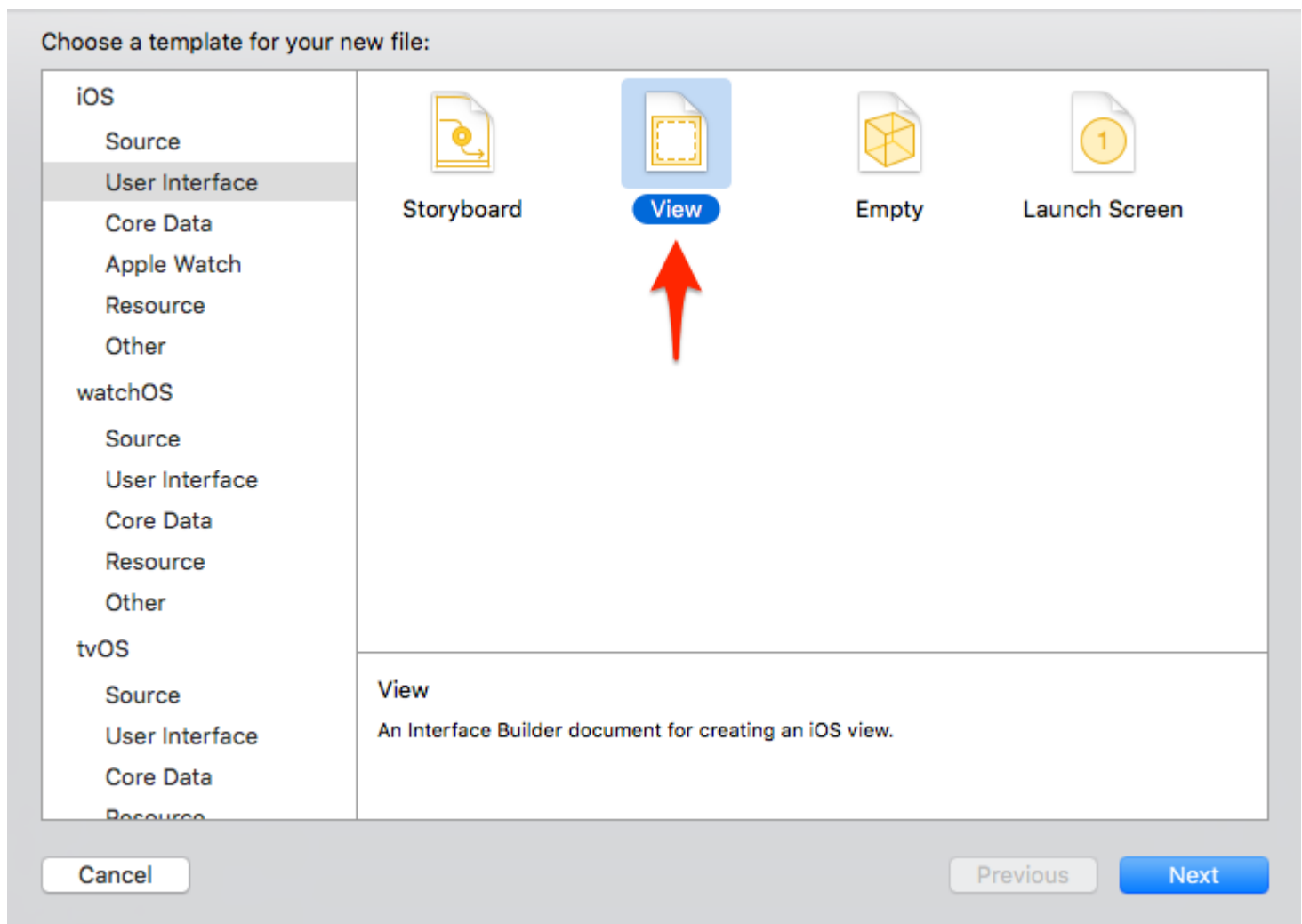
- XIBUILabelUITextField“XIB。

## Examples

XIB

Xcode>>>。

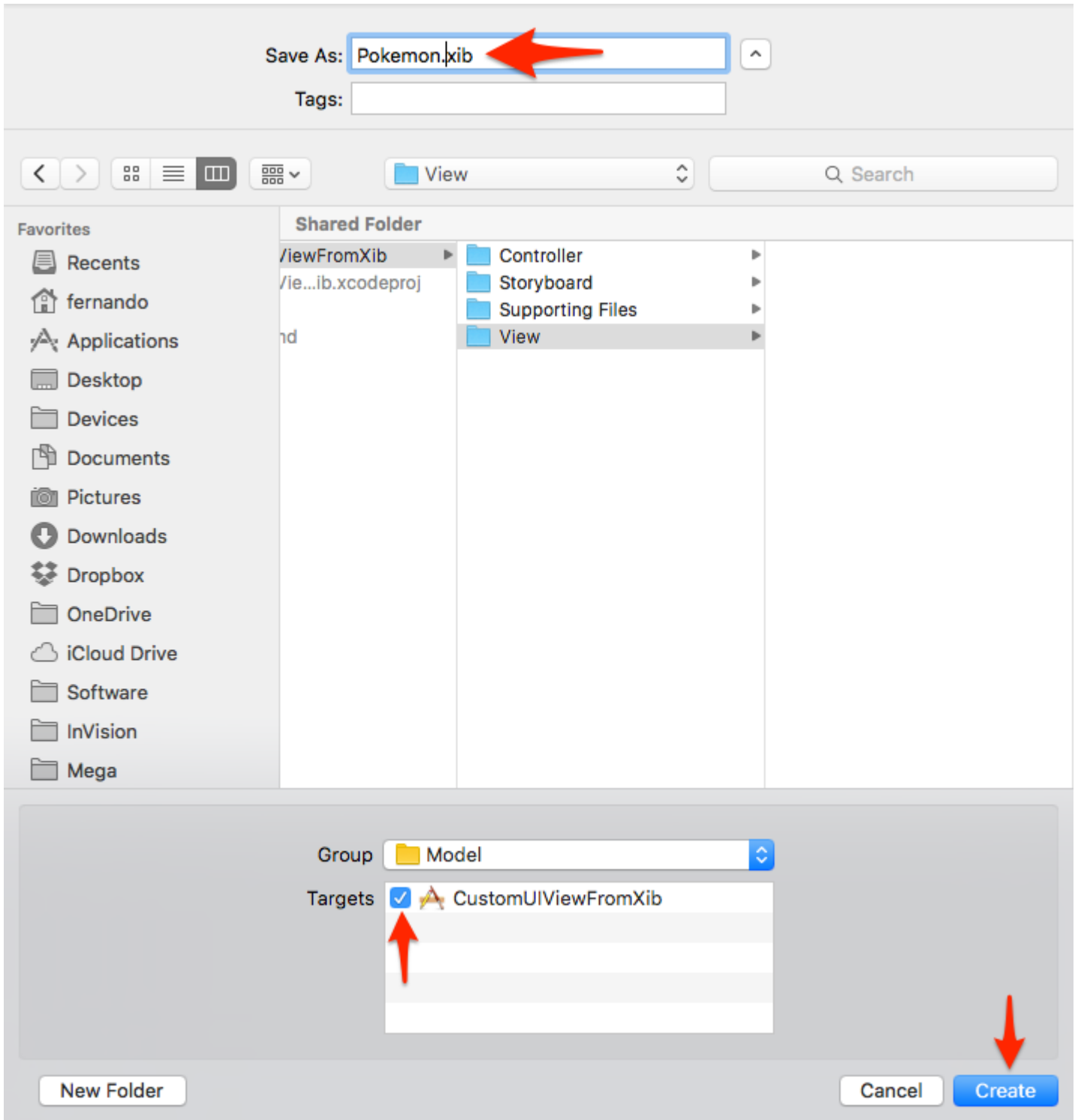
iOS“



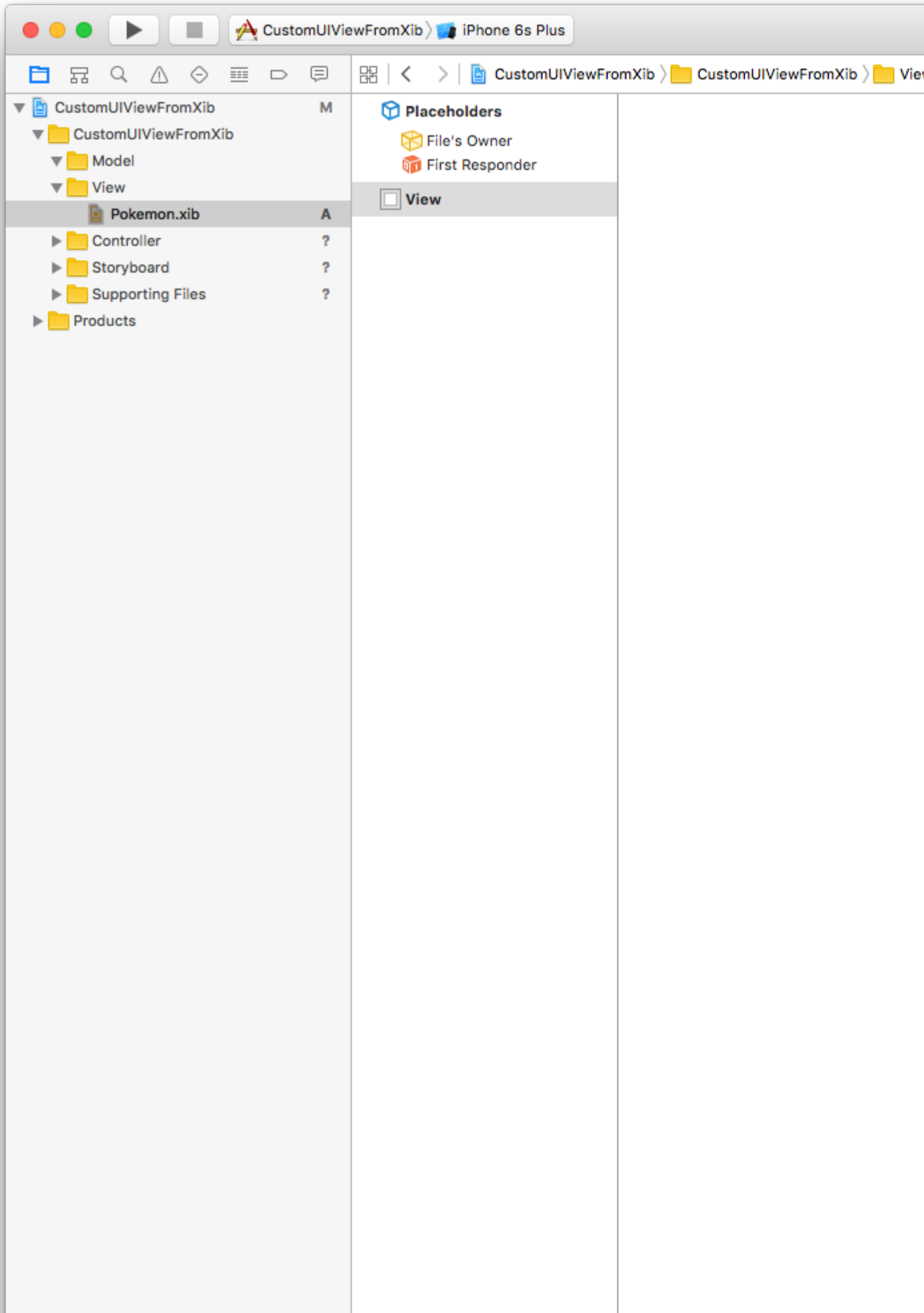
XIB。

“”。

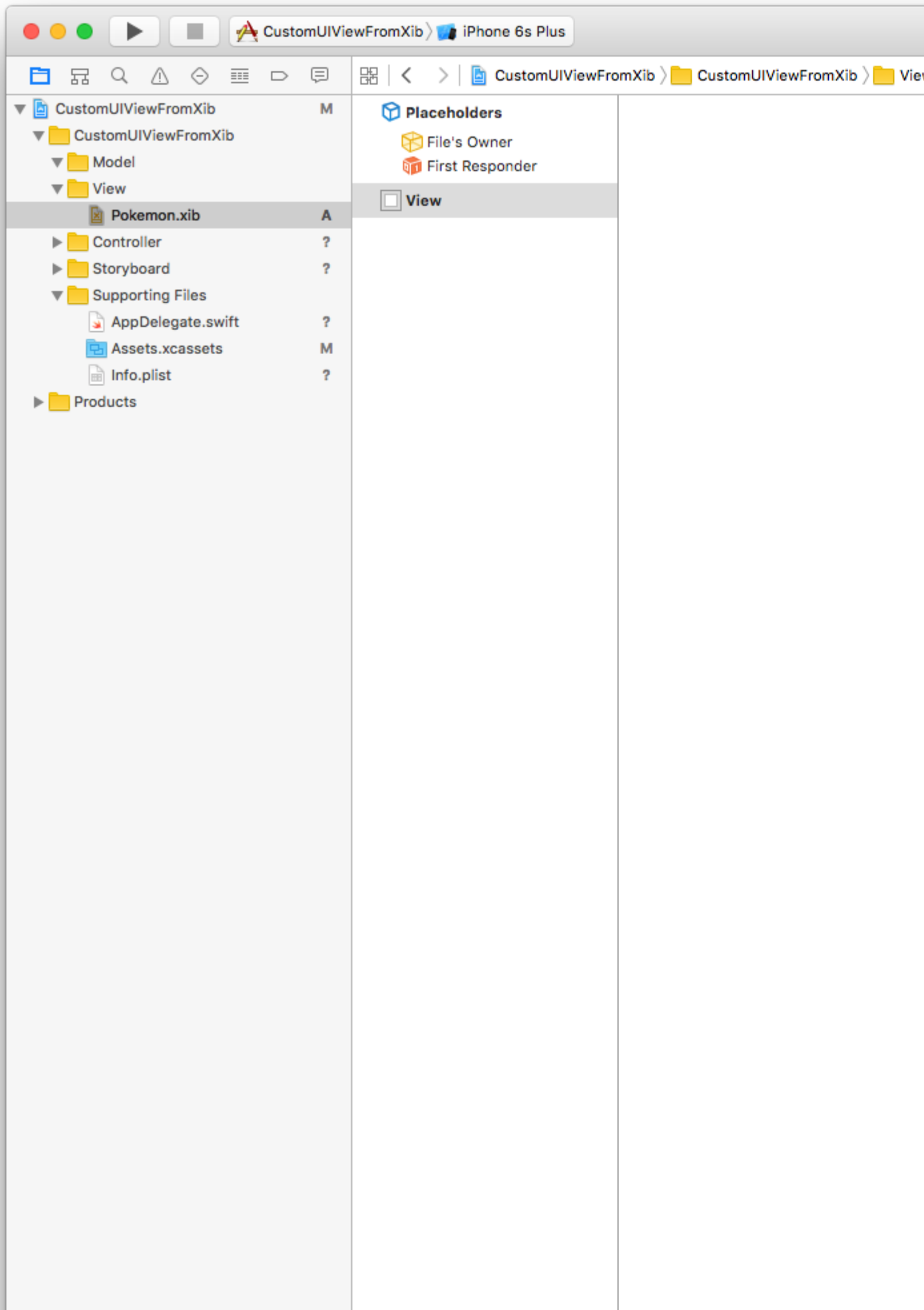




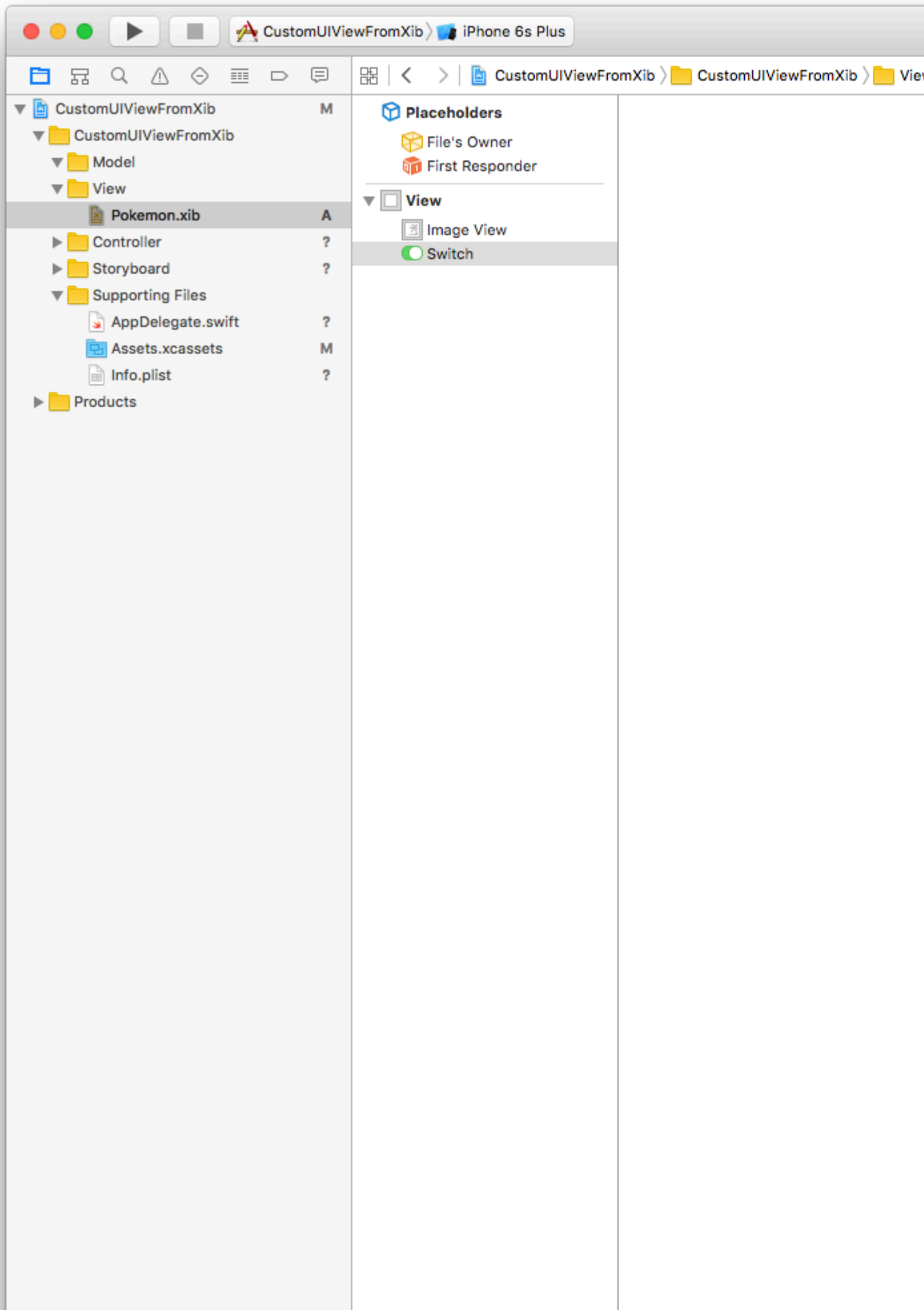
- 
- 
- 
-



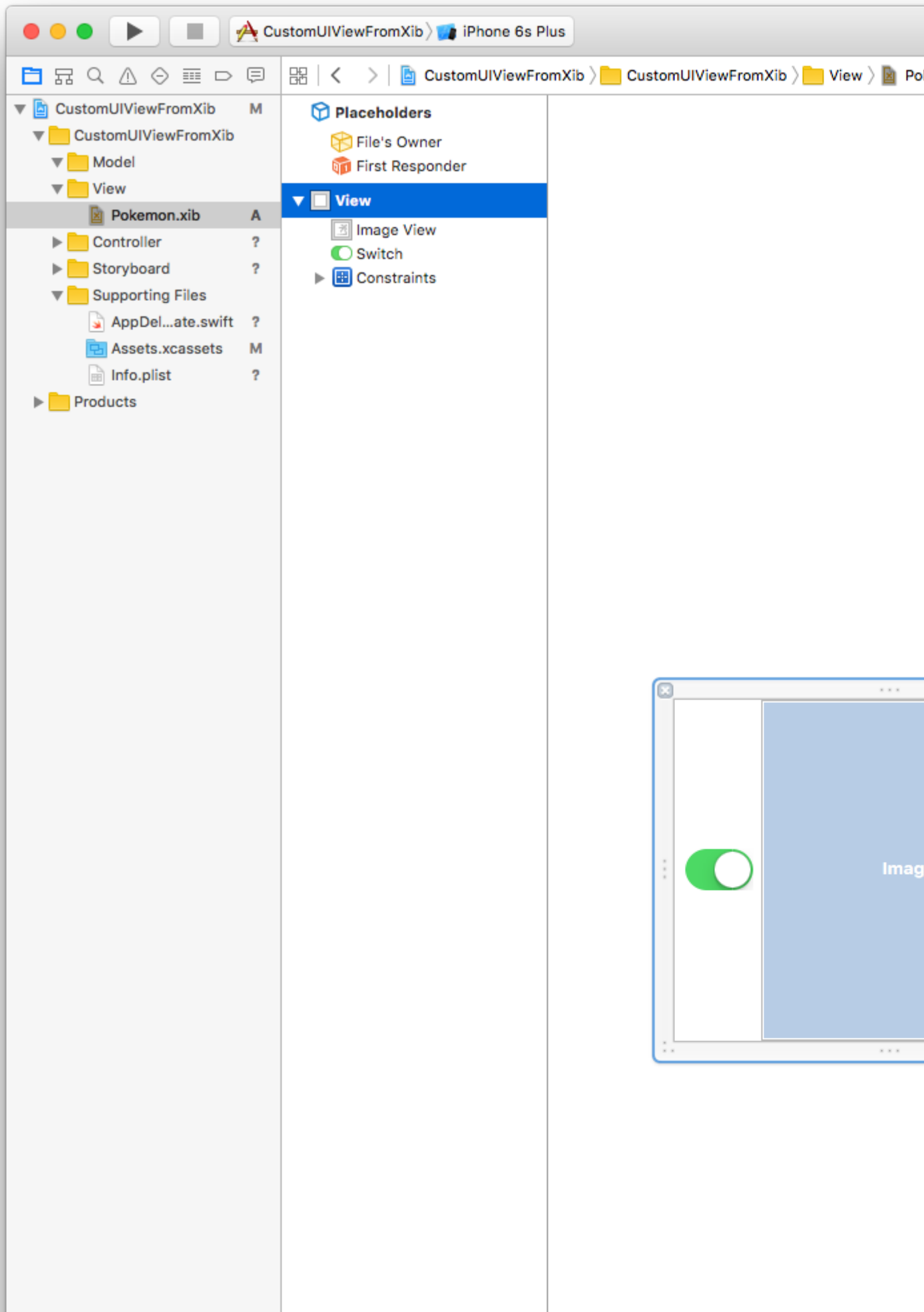
321256。



256x256 ◦



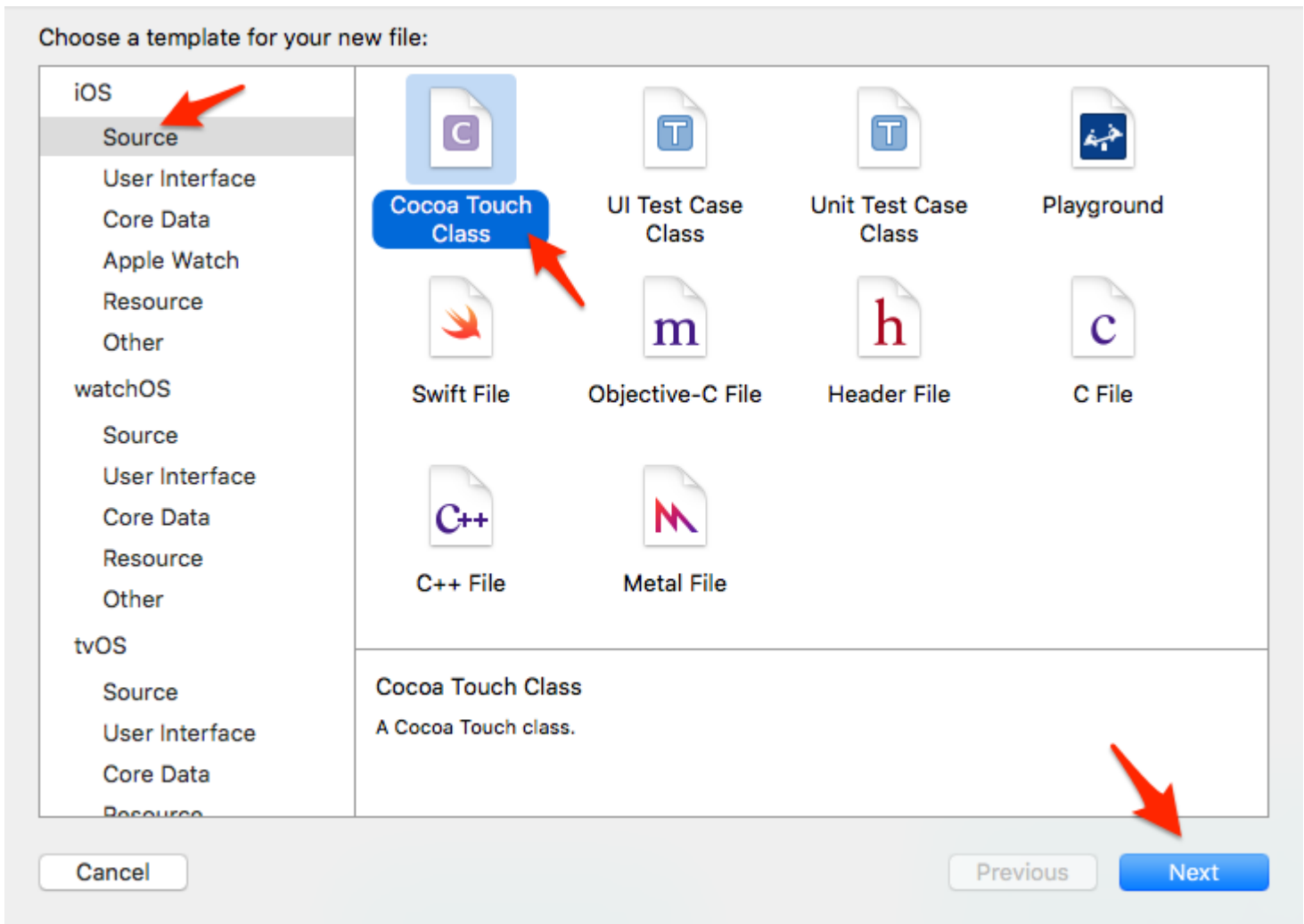
“iPhone”





Xcode>>>。

iOS / Source / Cocoa Touch Class。 “”。



XIBPokemon。

UIView“”。

Choose options for your new file:

Class:

Subclass of:

Also create XIB file

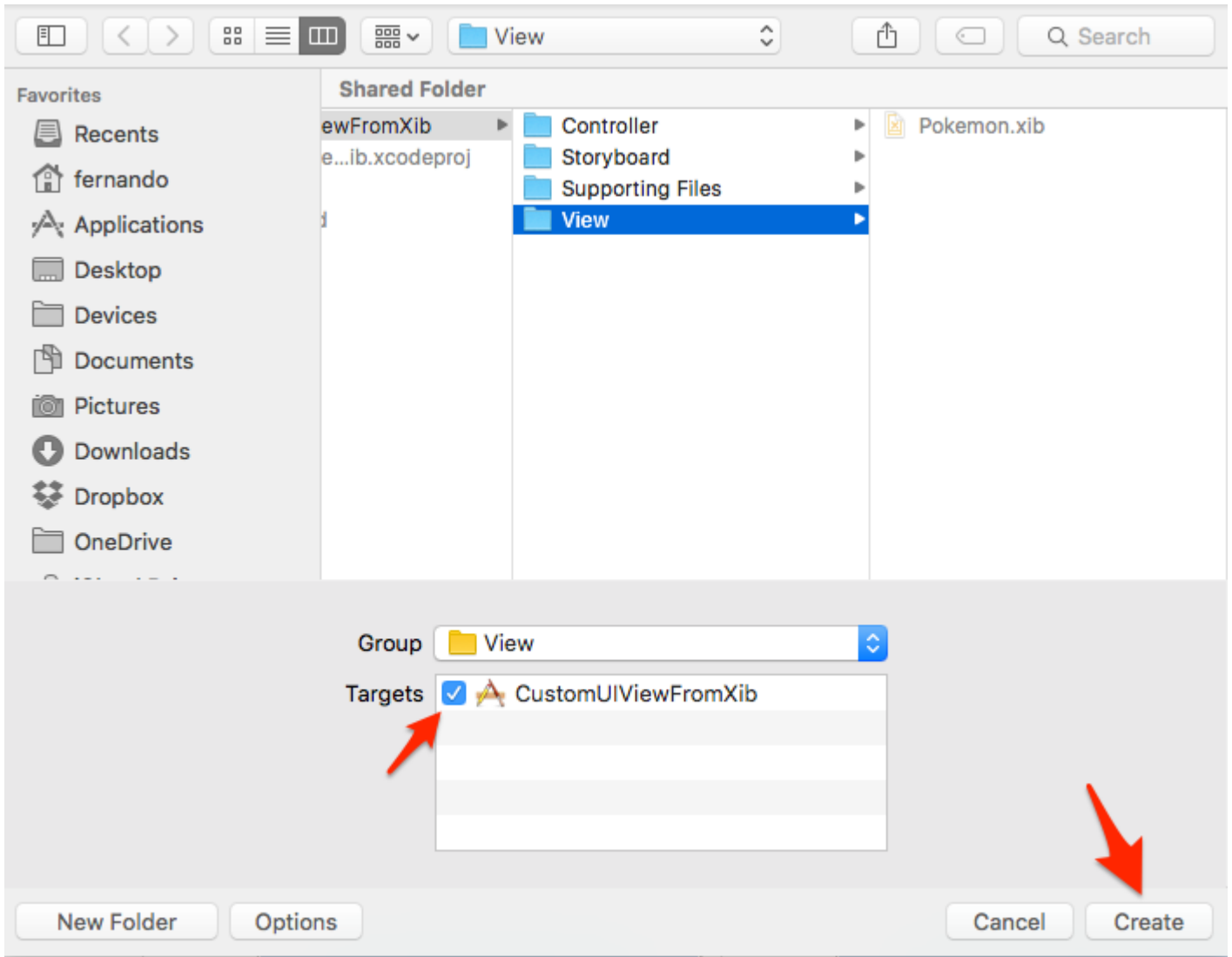
Language:

Cancel

Previous

Next

“”。

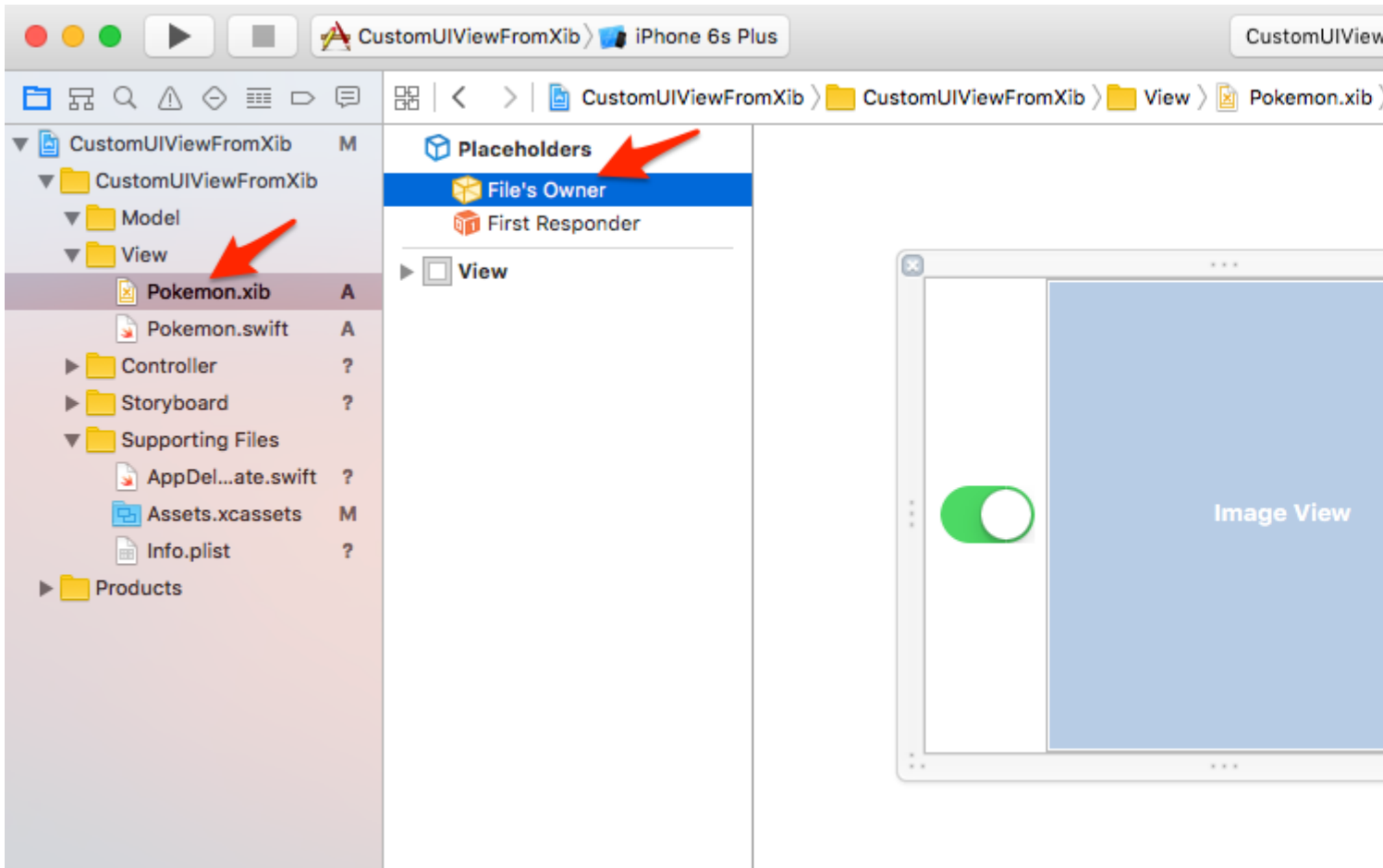


“Pokemon.xibPokemon.swift

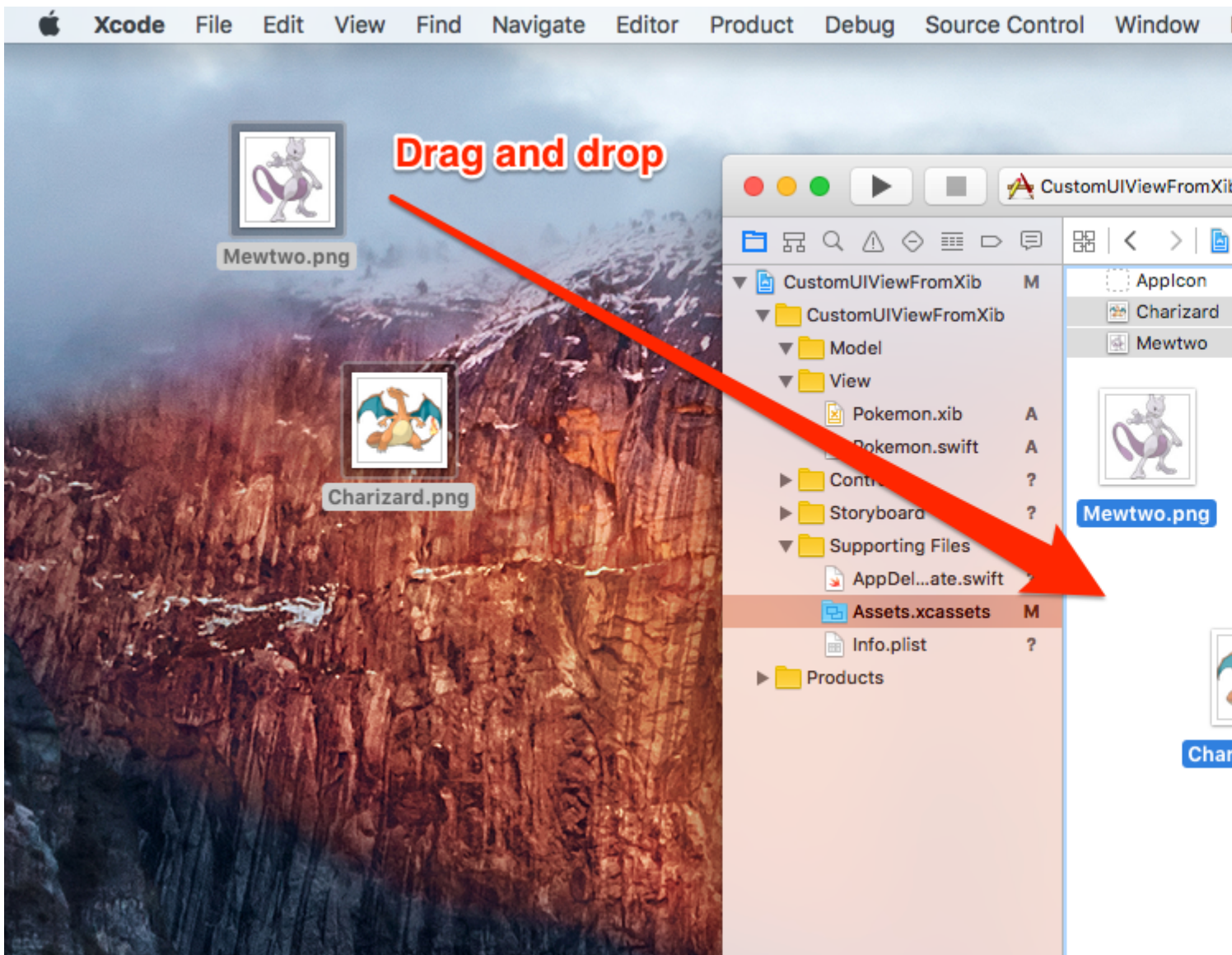
XcodePokemon.xib。

“”。

“ClassPokemon.swift。



!Pokemons""。  
PGN256x256。



Pokemon.swift

- 1.
2. XIB
3. XIB
- 4.

Pokemon.swift

```
import UIKit

class Pokemon: UIView {

 // MARK: - Initializers

 override init(frame: CGRect) {
 super.init(frame: frame)
 setupView()
 }
}
```

```

}

required init?(coder aDecoder: NSCoder) {
 super.init(coder: aDecoder)
 setupView()
}

// MARK: - Private Helper Methods

// Performs the initial setup.
private func setupView() {
 let view = viewFromNibForClass()
 view.frame = bounds

 // Auto-layout stuff.
 view.autoresizingMask = [
 UIViewAutoresizing.flexibleWidth,
 UIViewAutoresizing.flexibleHeight
]

 // Show the view.
 addSubview(view)
}

// Loads a XIB file into a view and returns this view.
private func viewFromNibForClass() -> UIView {

 let bundle = Bundle(for: type(of: self))
 let nib = UINib(nibName: String(describing: type(of: self)), bundle: bundle)
 let view = nib.instantiate(withOwner: self, options: nil).first as! UIView

 /* Usage for swift < 3.x
 let bundle = NSBundle(forClass: self.dynamicType)
 let nib = UINib(nibName: String(self.dynamicType), bundle: bundle)
 let view = nib.instantiateWithOwner(self, options: nil)[0] as! UIView
 */

 return view
}
}

```

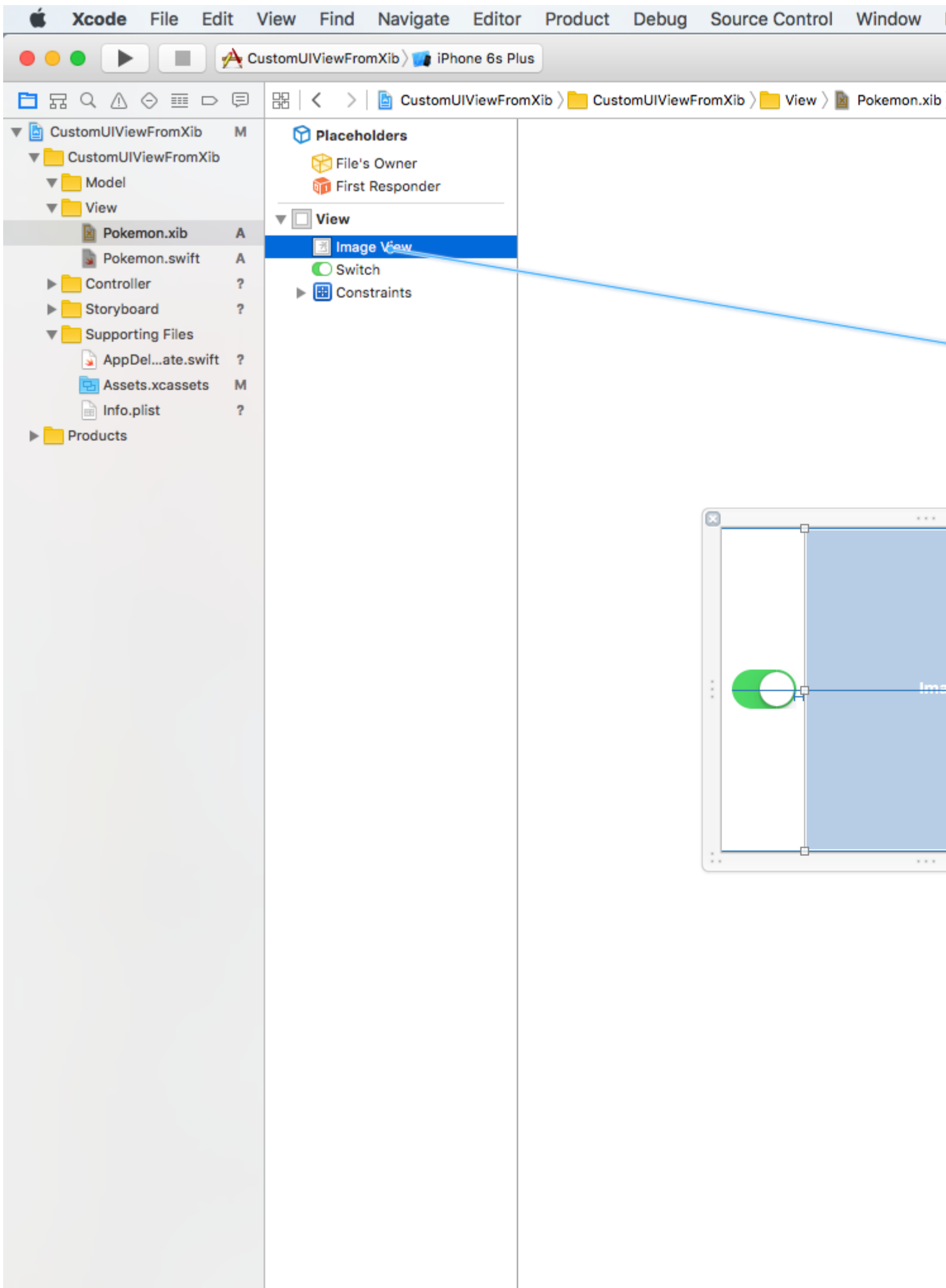
## @IBDesignable@IBInspectable

@IBDesignable Interface Builder◦

@IBInspectable Interface Builder◦

Image View “Inspectable”◦

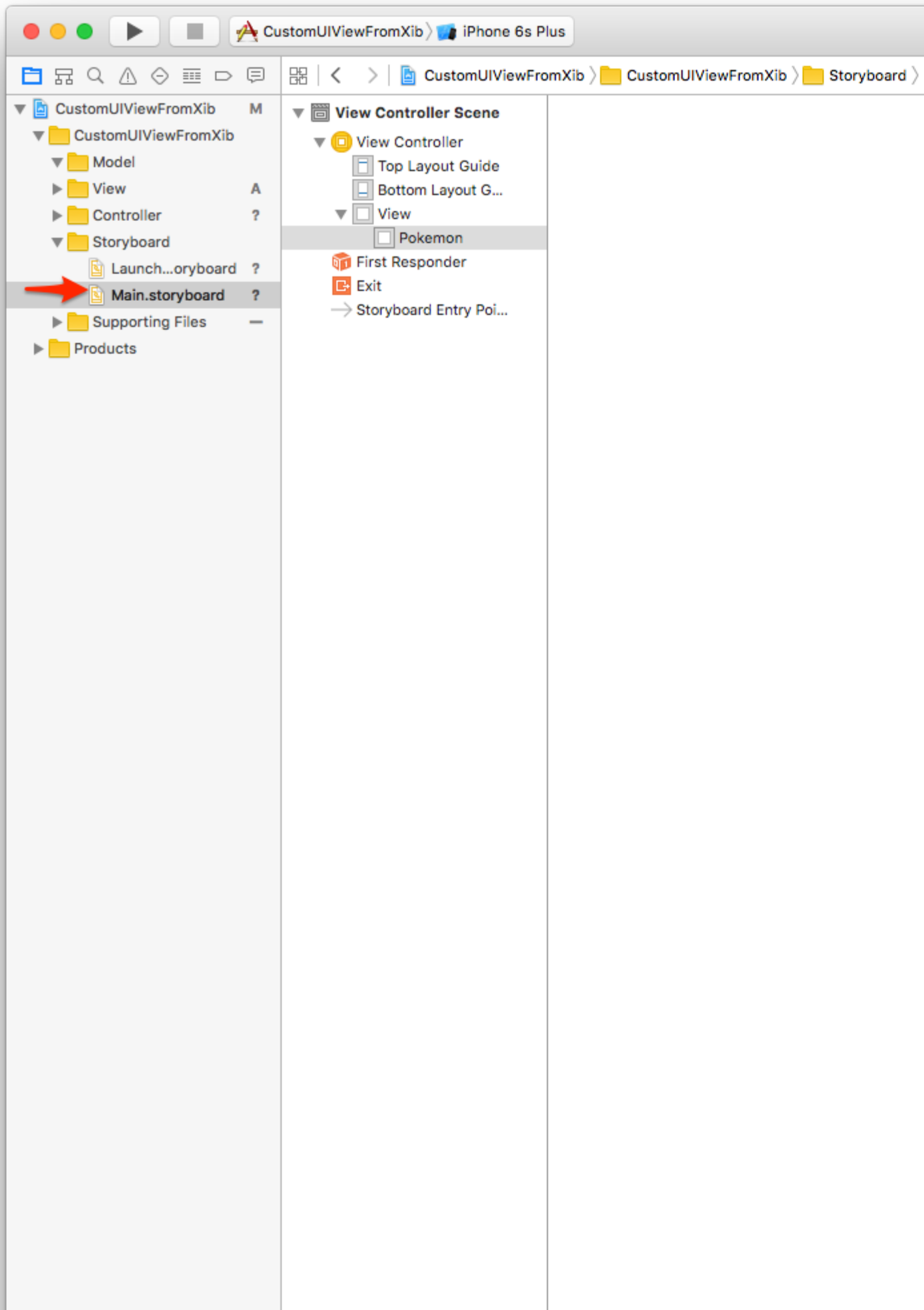
Image View Pokemon.xib Pokemon.swift◦



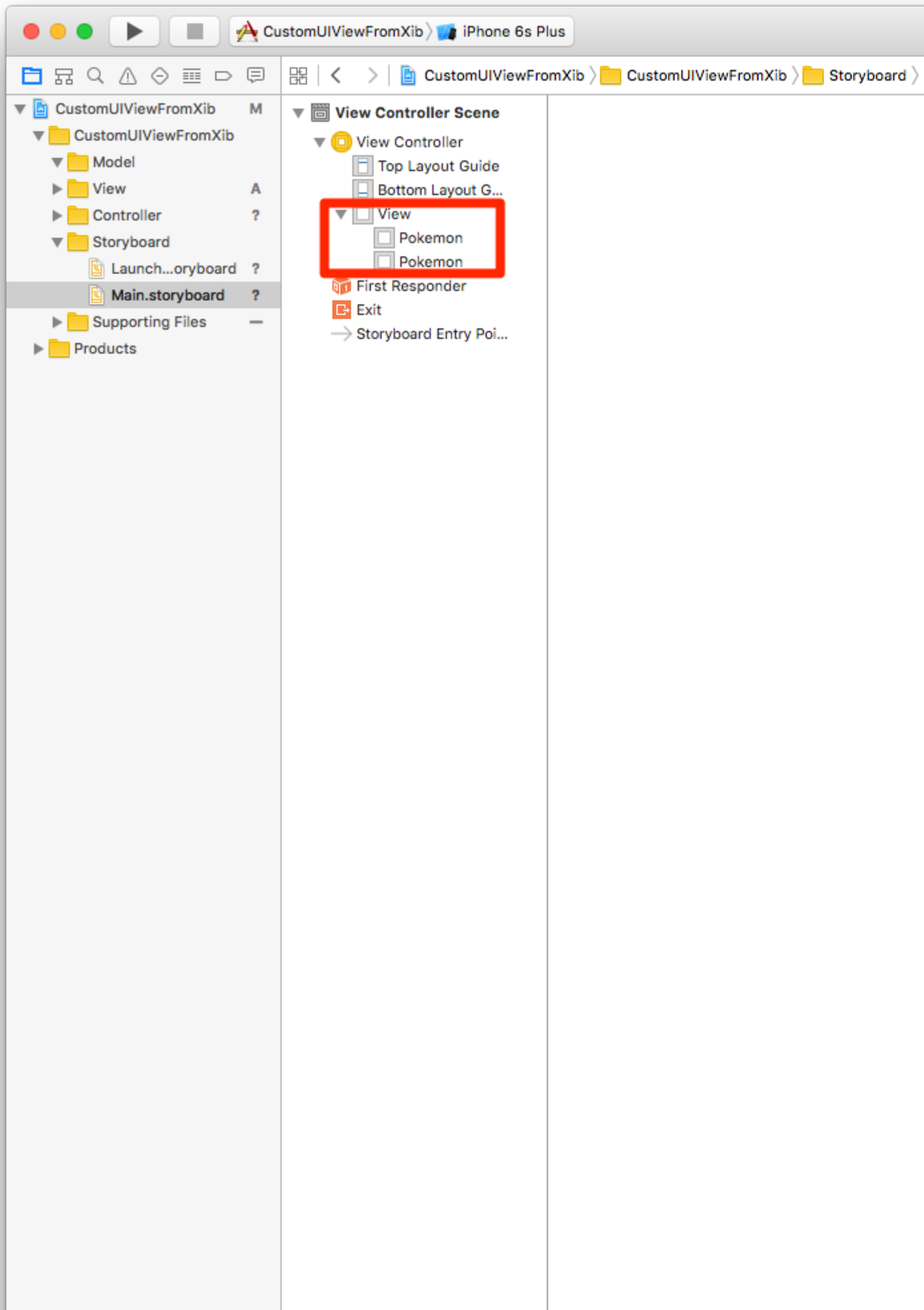
200x200。。

IdentityClassPokemon。





150x150.  
Pokemon



/Pokemons。


SwitchIBActionPokemon.swift。

switchTapped。

```
// MARK: - Actions

@IBAction func switchTapped(sender: UISwitch) {
 imageView.alpha = sender.on ? 1.0 : 0.2
}

// MARK: - Initializers
...
```

Carrier 

3:54 PM



Game Center



Extras



Watch



CustomUIV...



UI。

[Github](#)。  
**Swift 3.1**

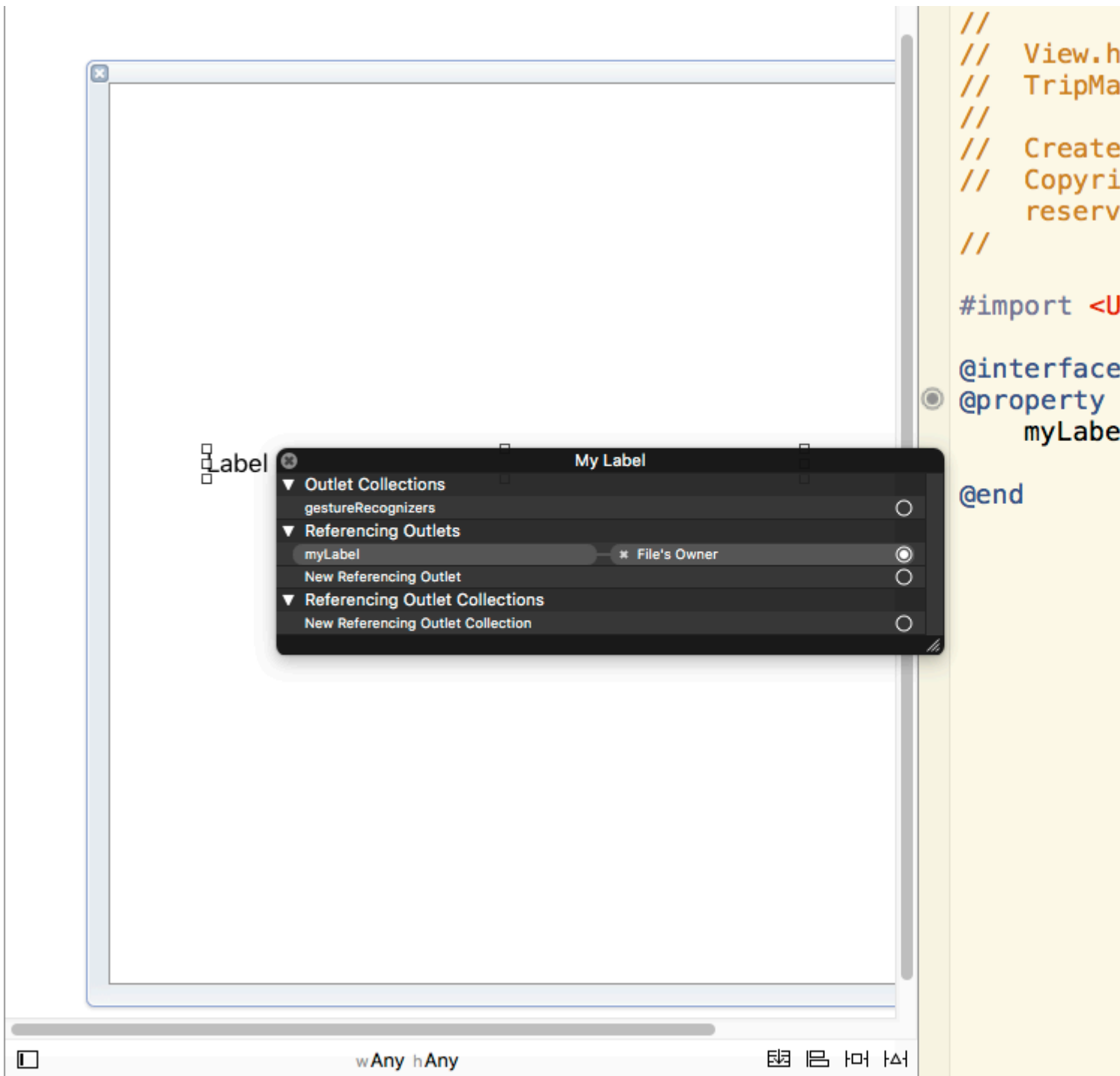
## XIBUIView

XIB。

。

[loadNibNamed](#)

1. XIB
2. .h.m
3. .h
4. .hXIB



5. `.initWithCoderloadNibNamed:` UIViewstoryboard / Parent UIView XIB. storyboard / parent XIB. XCodeObjective C.

XIBUIViews <https://riptutorial.com/zh-TW/ios/topic/1362/xibuiviews>

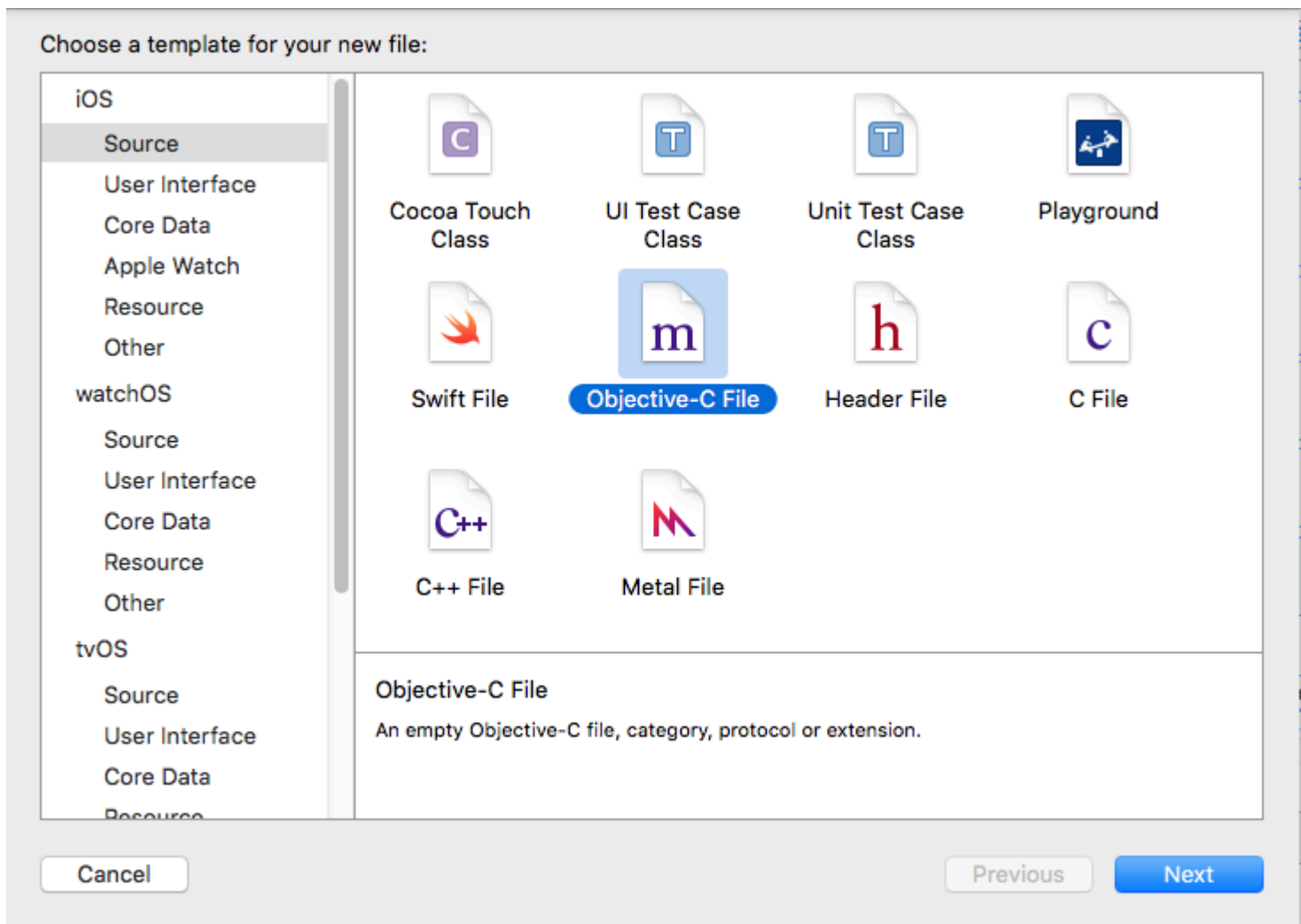
# 118:

◦ ◦ ◦ ◦

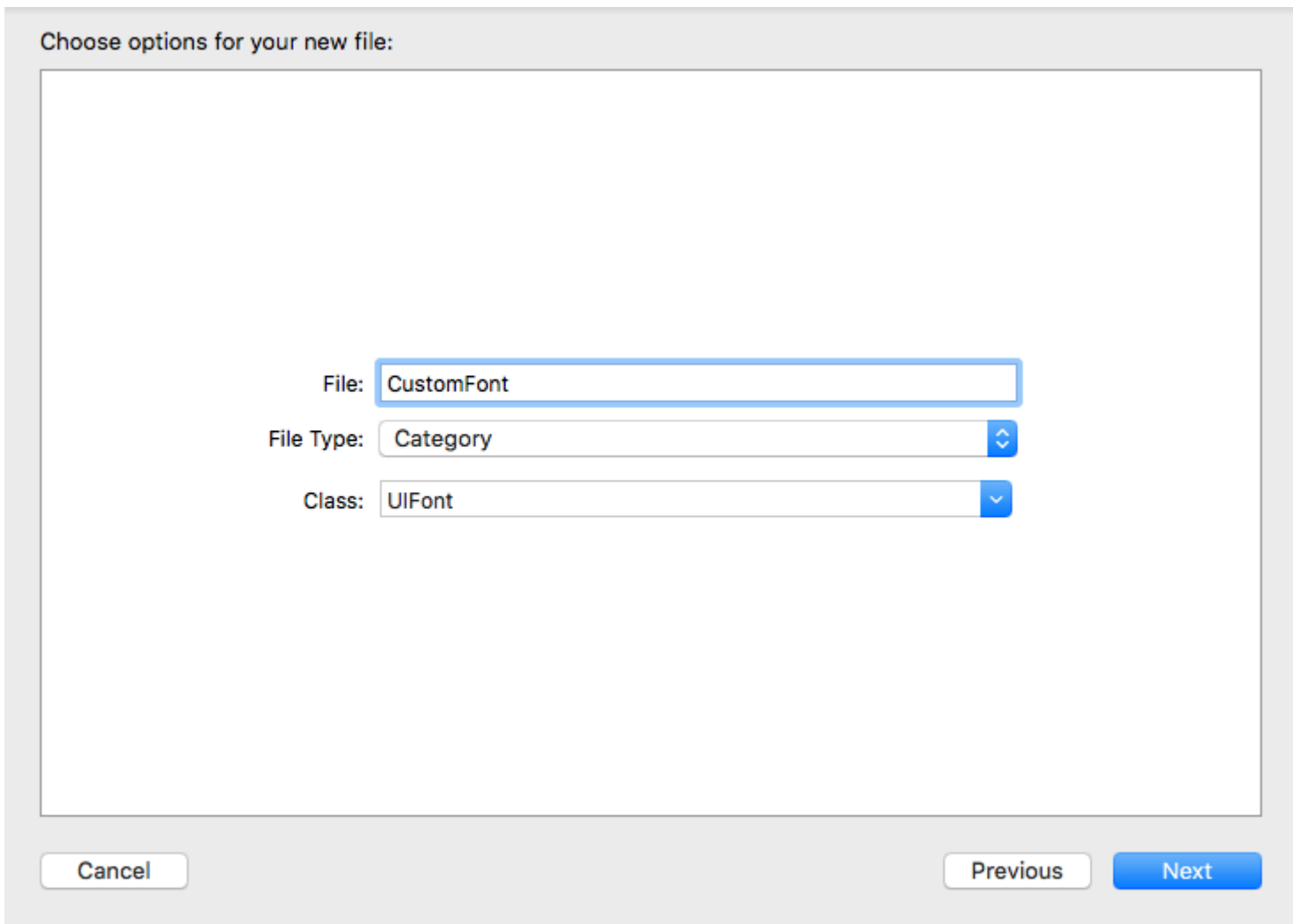
## Examples

◦

◦ UIFont ◦ XcodeFile - > New - > FileObjective-CNext“CustomFont”CategoryClassUIFont“Next”  
“Create”◦ “







-

“UIFont + CustomFonts.h”。

```
@interface UIFont (CustomFonts)
+ (UIFont *)productSansRegularFontWithSize:(CGFloat)size;
@end
```

-

“UIFont + CustomFonts.m”。

```
+ (UIFont *)productSansRegularFontWithSize:(CGFloat)size{
 return [UIFont fontWithName:@"ProductSans-Regular" size:size];
}
```

```
#import "UIFont+CustomFonts.h"
```

Label

```
[self.label setFont:[UIFont productSansRegularFontWithSize:16.0]];
```

<https://riptutorial.com/zh-TW/ios/topic/3633/>

# 119:

## Examples

### one liner

```
class Contact: UIView
{
 private var message: UILabel
 private var phone: UITextView

 required init?(coder aDecoder: NSCoder) {
 (message, phone) = self.dynamicType.setUp()
 super.init(coder: aDecoder)
 }

 override func awakeFromNib() {
 (message, phone) = self.dynamicType.setUp()
 super.awakeFromNib()
 }

 override init(frame: CGRect) {
 (message, phone) = self.dynamicType.setUp()
 super.init(frame: frame)
 }

 private static func setUp(){
 let message = UILabel() // ...
 let phone = UITextView() // ...
 return (message, phone)
 }
}
```

```
let mySwitch: UISwitch = {
 view.addSubview($0)
 $0.addTarget(self, action: "action", forControlEvents: .TouchUpInside)
 return $0
}(UISwitch())
```

### didSet

```
@IBOutlet weak var title: UILabel! {
 didSet {
 label.textColor = UIColor.redColor()
 label.font = UIFont.systemFontOfSize(20)
 label.backgroundColor = UIColor.blueColor()
 }
}
```

```
private var loginButton = UIButton() {
 didSet(oldValue) {
 loginButton.addTarget(self, action: #selector(LoginController.didClickLogin),
 forControlEvents: .TouchUpInside)
```

```
}
}
```

## NSObject

### NSObject。Object。

```
class ContactFormStyle: NSObject
{
 @IBOutlet private weak var message: UILabel! {
 didSet {
 message.font = UIFont.systemFontOfSize(12)
 message.textColor = UIColor.blackColor()
 }
 }
}
}

class ContactFormVC: UIViewController
{
 @IBOutlet private var style: ContactFormStyle!
}
```

<https://github.com/devxoul/Then>Then。

```
let label = UILabel().then {
 $0.textAlignment = .Center
 $0.textColor = UIColor.blackColor()
 $0.text = "Hello, World!"
}
```

Then

```
import Foundation

public protocol Then {}

extension Then
{
 public func then(@noescape block: inout Self -> Void) -> Self {
 var copy = self
 block(©)
 return copy
 }
}

extension NSObject: Then {}
```

```
internal func Init<Type>(value : Type, block: @noescape (object: Type) -> Void) -> Type
{
 block(object: value)
 return value
}
```

```
Init(UILabel(frame: CGRect.zero)) {
```

```
$0.backgroundColor = UIColor.blackColor()
}
```

<https://riptutorial.com/zh-TW/ios/topic/3513/>

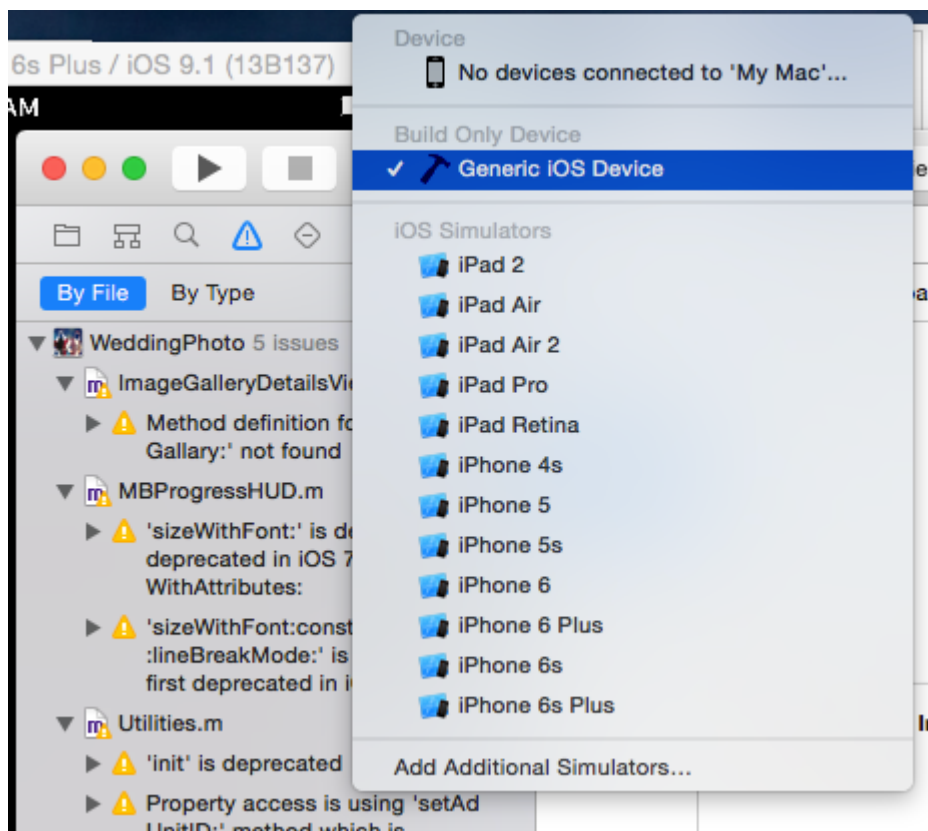
# 120: .ipaApplicationloaderappstore

## Examples

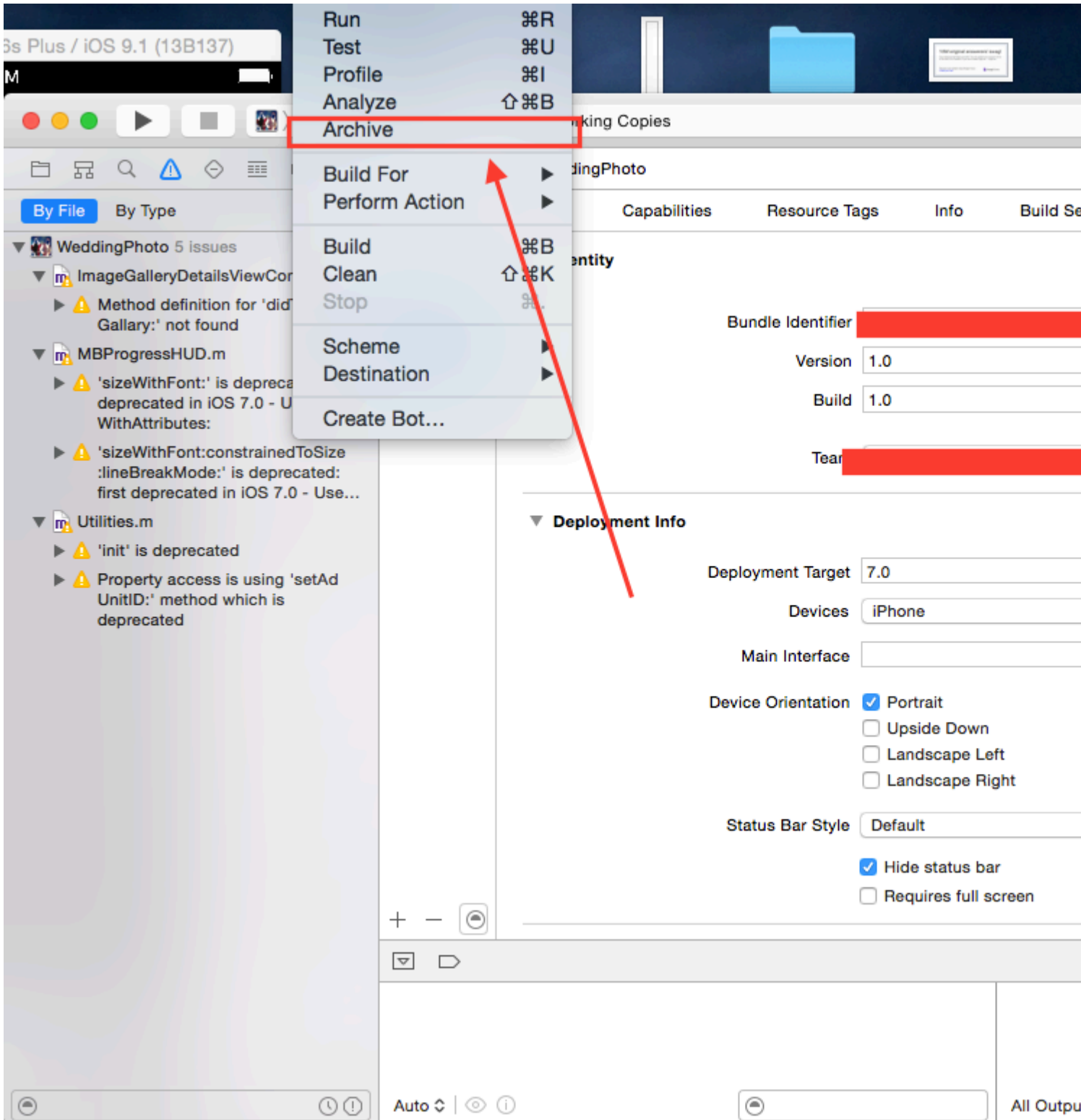
.ipaApplication Loaderappstore

.ipaitunesconnect Xcode ◦ iTunes.ipa ◦

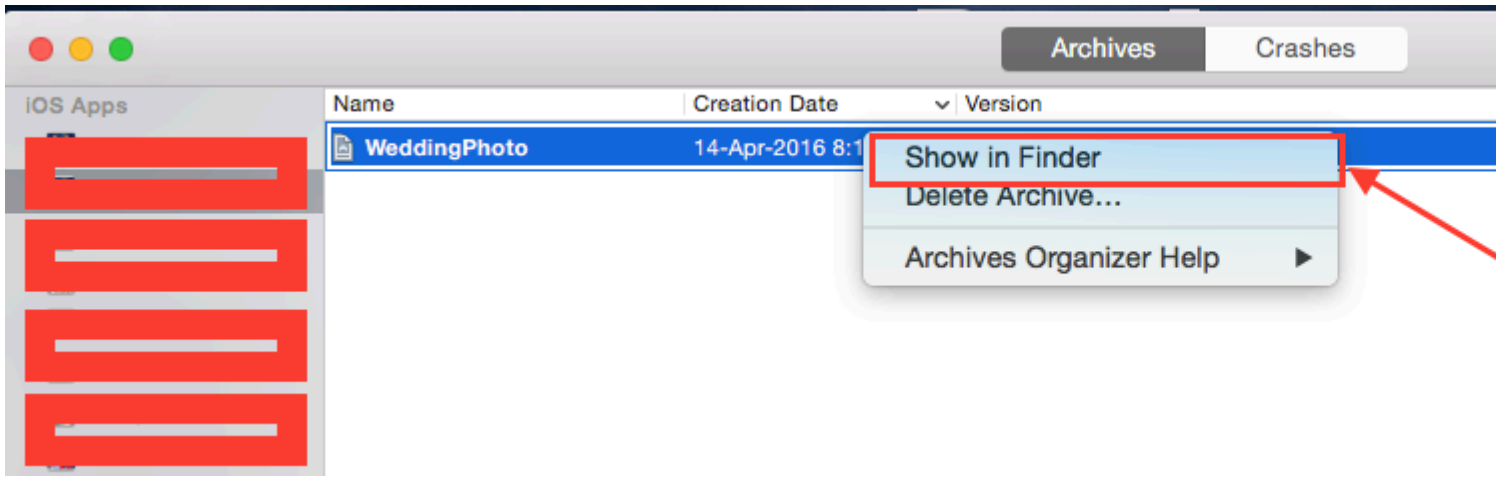
1 -◦



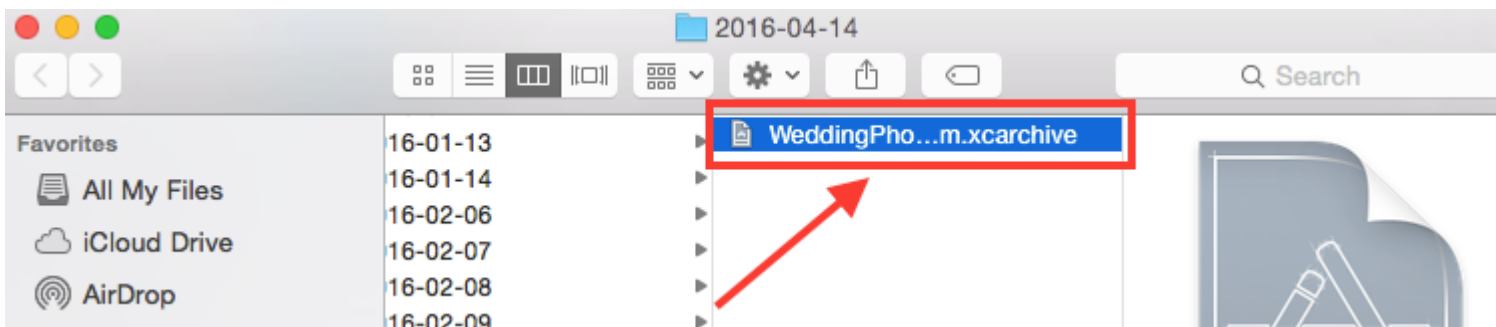
2 - - >



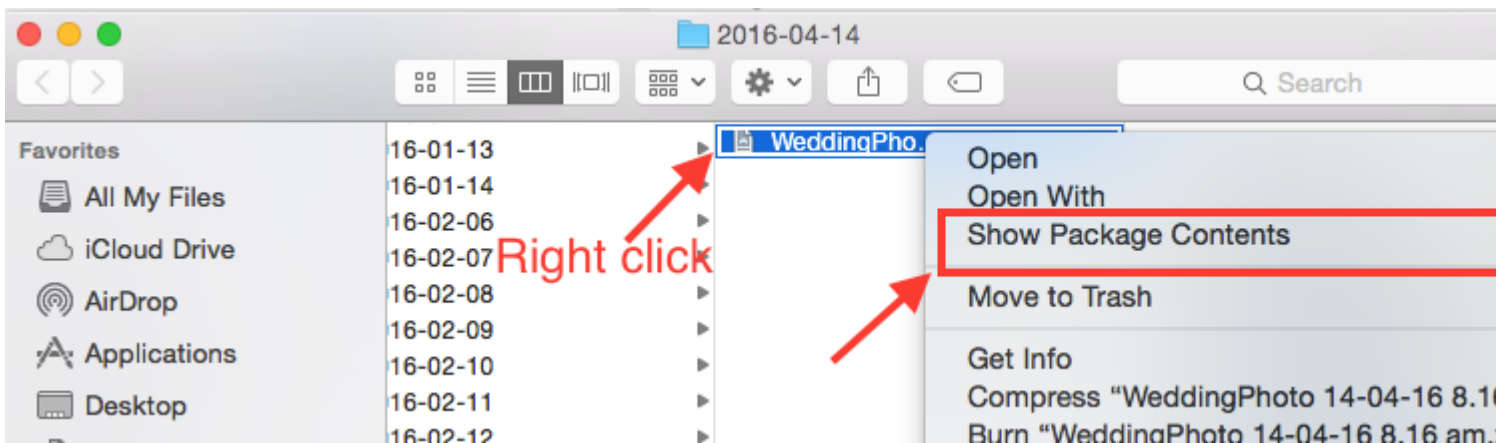
3 - ->Finder



4 -findershowArchive

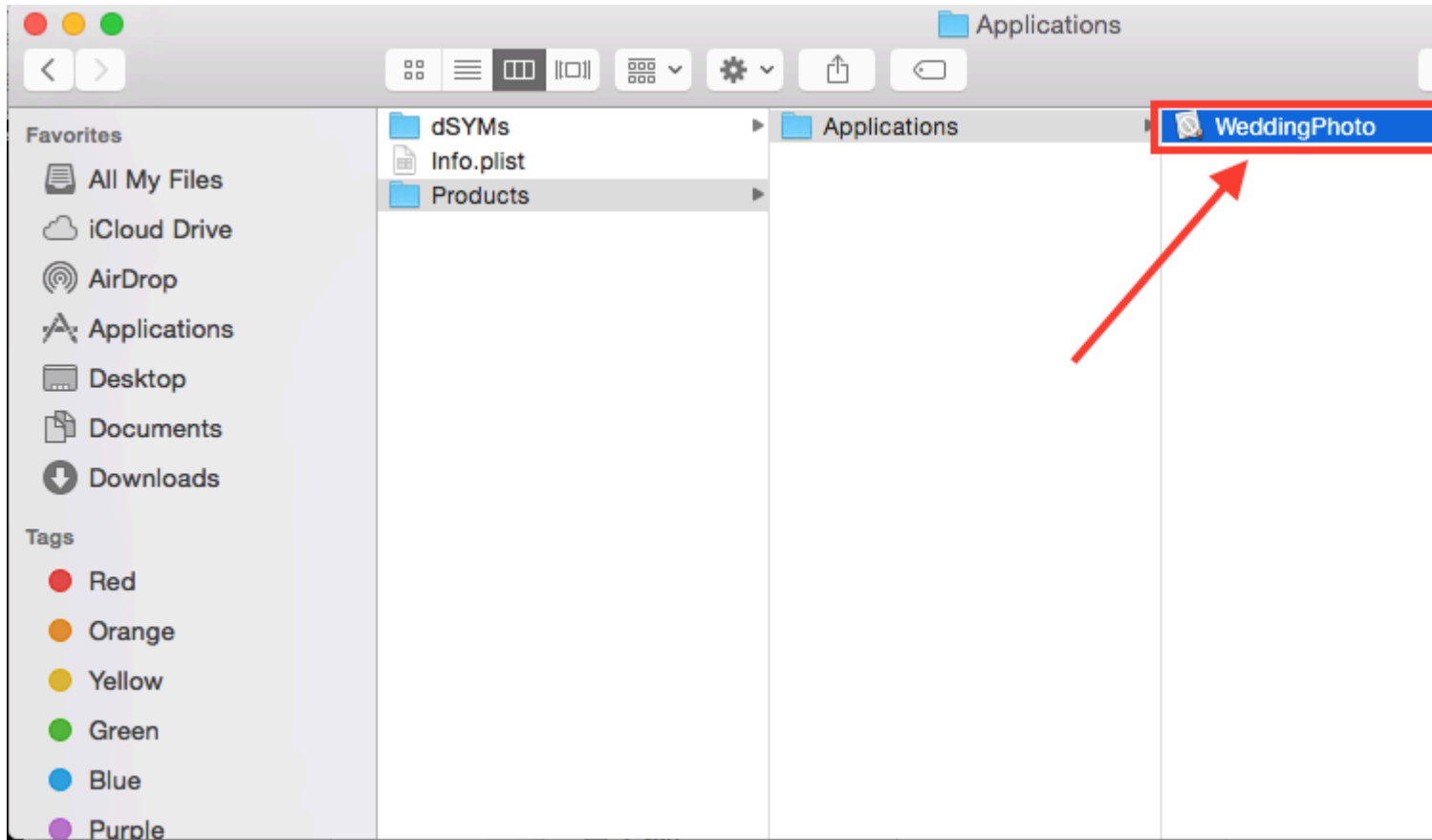


5 -.xcarchive ->Show in Finder.

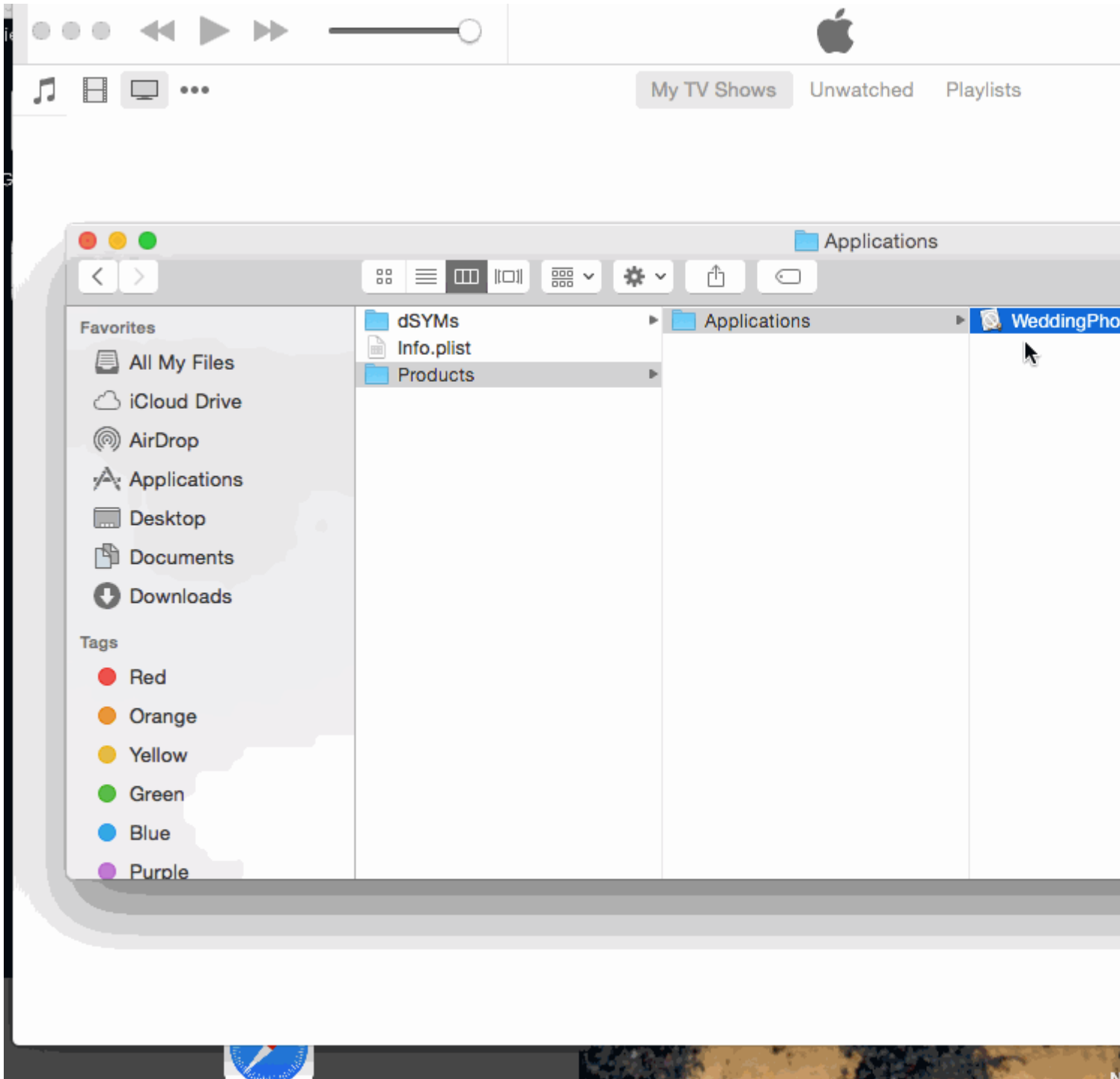


6 - -> ->yourprojectname.app





7 -.app.ipaiTunes.



8 -.ipa。

-

Loader

-

-.ipa.aap.zip.zip.ipa。

compress .app.zip.ipa。。

IPA。

[.ipaApplicationloaderappstore](https://riptutorial.com/zh-TW/ios/topic/6119/-ipaapplicationloader-appstore) <https://riptutorial.com/zh-TW/ios/topic/6119/-ipaapplicationloader-appstore>

# 121: ID

## Examples

- IAPiTunes Connect. ◦ App StoreiTunes. IAP

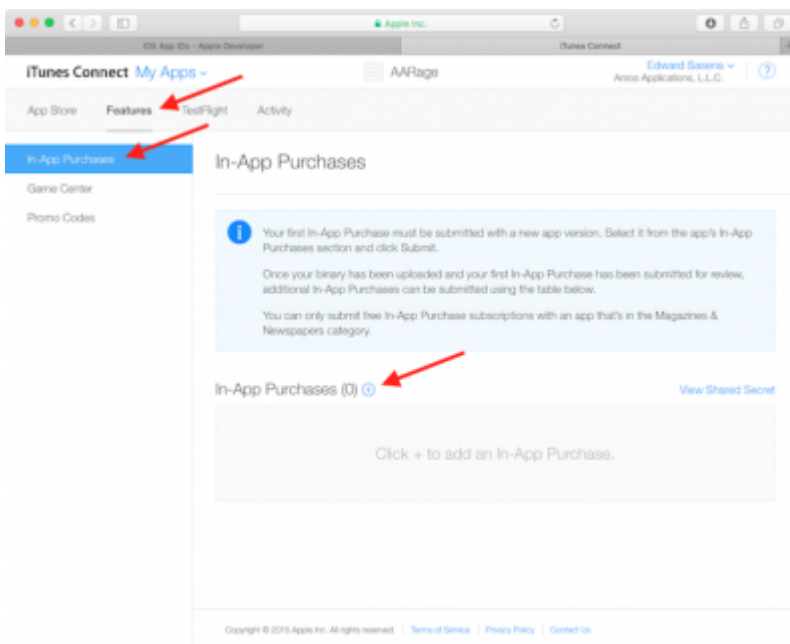
- ◦ ◦

- ◦

- ◦

- raywenderlich.com.

- Apple. iTunes Connect“””. IAP+.



## Select the In-App Purchase you want to create.

**Consumable**

A product that is used once, after which it becomes depleted and must be purchased again.

**Example:** Fish food for a fishing app.

---

**Non-Consumable**

A product that is purchased once and does not expire or decrease with use.

**Example:** Race track for a game app.

---

**Auto-Renewable Subscription**

A product that allows users to purchase dynamic content for a set period. This type of subscription renews automatically unless cancelled by the user.

**Example:** Monthly subscription for an app offering a streaming service.

---

**Non-Renewing Subscription**

A product that allows users to purchase a service with a limited duration. The content of this in-app purchase can be static. This type of subscription does not renew automatically.

**Example:** Annual subscription to a catalog of archived articles.

---

[Learn more about In-App Purchases.](#)

Cancel

Create

“” IAP

- iTunes Connect IAP. “”
- ID IAP. Bundle ID. “GirlfriendOfDrummerRage”.  
com.theNameYouPickedEarlier.Rage.GirlFriendOfDrummerRage.
- IAP.
- IAP. 1.

“” “” IAP.

### Localizations +

English (U.S.)

Display Name ?

Girlfriend of Drummer

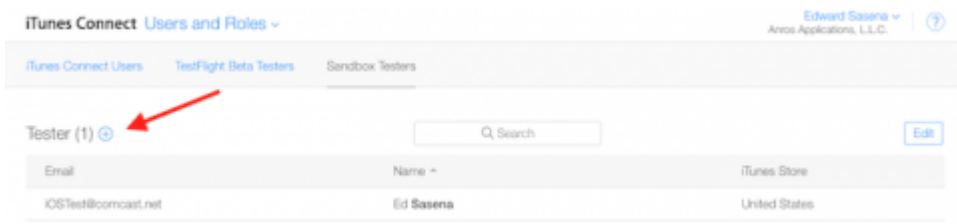
Description ?

Girlfriend of Drummer

234

• Apple “” IAP.

iTunes ConnectiTunes Connect. “”“”。“Tester”+。



“” Apple. Apple ID. Gmail

ID <https://riptutorial.com/zh-TW/ios/topic/10854/id>

# 122:

## Examples

[AlamofireUIImageViewExtension](#) ◦ [imageViewtableView](#) ◦ [imageView](#) ◦

[tableViewcellForRowAt](#)

```
let url = URL(string: "https://httpbin.org/image/png")!
let placeholderImage = UIImage(named: "placeholder")!

imageView.af_setImage(withURL: url, placeholderImage: placeholderImage)
```

[urlplaceholder](#) ◦ [af\\_setImageimageViewURL](#) ◦

◦ ◦ [UIImageView](#) ◦

[Alamofire](#) ◦

[tableViewimageView](#) ◦ [cellForRowAt](#)

```
let placeholderImage = UIImage(named: "placeholder")!
imageView.image = placeholderImage

let url = URL(string: "https://httpbin.org/image/png")!

Alamofire.request(url!, method: .get).responseImage { response in
 guard let image = response.result.value else { return }

 if let updateCell = tableView.cellForRow(at: indexPath) {
 updateCell.imageView.image = image
 }
}
```

◦ [Alamofire](#) ◦ [request](#) ◦ [url.get HTTP](#) ◦ [.responseImage](#) ◦

◦ [cellForRowatindexPathnil](#) ◦ ◦

[updateCellnilifnilif](#) ◦ ◦

<https://riptutorial.com/zh-TW/ios/topic/10793/>

# 123: UIStackView

## Examples

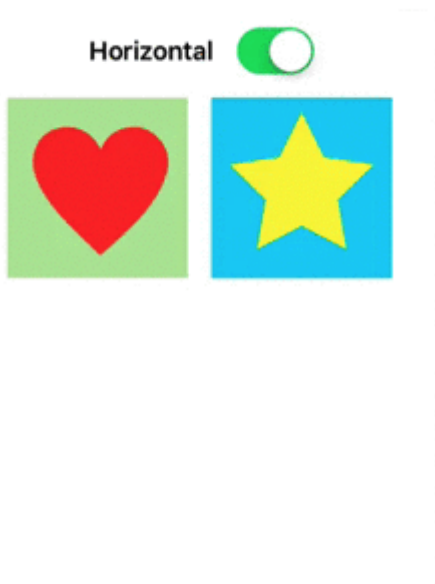
### UISwitch

```
@IBAction func axisChange(sender: UISwitch) {
 UIView.animateWithDuration(1.0) {
 self.updateConstraintsForAxis()
 }
}
```

### updateConstraintForAxis

```
private func updateConstraintsForAxis() {
 if (axisSwitch.on) {
 stackView.axis = .Horizontal
 } else {
 stackView.axis = .Vertical
 }
}
```

gif



[UIStackView](https://riptutorial.com/zh-TW/ios/topic/5884/uistackview) <https://riptutorial.com/zh-TW/ios/topic/5884/uistackview>



---

# 124:

```
// Content size category constants
UIContentSizeCategoryExtraSmall
UIContentSizeCategorySmall
UIContentSizeCategoryMedium
UIContentSizeCategoryLarge
UIContentSizeCategoryExtraLarge
UIContentSizeCategoryExtraExtraLarge
UIContentSizeCategoryExtraExtraExtraLarge

// Accessibility sizes
UIContentSizeCategoryAccessibilityMedium
UIContentSizeCategoryAccessibilityLarge
UIContentSizeCategoryAccessibilityExtraLarge
UIContentSizeCategoryAccessibilityExtraExtraLarge
UIContentSizeCategoryAccessibilityExtraExtraExtraLarge
```

---

## Examples

```
UIApplication.sharedApplication().preferredContentSizeCategory
```

---

## Objective-C

```
[UIApplication sharedApplication].preferredContentSizeCategory;
```

◦

◦

---

```
NSNotificationCenter.defaultCenter().addObserver(self, selector: #selector(updateFont), name:
name:UIContentSizeCategoryDidChangeNotification, object: nil)
```

---

## Objective-C

```
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(updateFont)
name:UIContentSizeCategoryDidChangeNotification object:nil];
```

userInfo:UIContentSizeCategoryNewValueKey◦

## WKWebView

```
// build HTML header for dynamic type and responsive design
func buildHTMLHeader() -> String {

 // Get preferred dynamic type font sizes for html styles
 let bodySize = UIFont.preferredFont(forTextStyle: UIFontTextStyle.body).pointSize
 let h1Size = UIFont.preferredFont(forTextStyle: UIFontTextStyle.title1).pointSize
 let h2Size = UIFont.preferredFont(forTextStyle: UIFontTextStyle.title2).pointSize
 let h3Size = UIFont.preferredFont(forTextStyle: UIFontTextStyle.title3).pointSize

 // On iPad, landscape text is larger than preferred font size
 var portraitMultiplier = CGFloat(1.0)
 var landscapeMultiplier = CGFloat(0.5)

 // iPhone text is shrunken
 if UIDevice.current.model.range(of: "iPhone") != nil {
 portraitMultiplier = CGFloat(3.0)
 landscapeMultiplier = CGFloat(1.5)
 }

 // Start HTML header text
 let patternText = "<html> <head> <style> "

 // Match Dynamic Type for this page.
 + "body { background-color: \(backgroundColor); } "
 + "@media all and (orientation:portrait) {img {max-width: 90%; height: auto;} "
 + "p, li { font: -apple-system-body; font-family: Georgia, serif; font-
size:calc(\(bodySize * portraitMultiplier)px + 1.0vw); font-weight: normal; color:
\(fontColor) } "
 + "h1 { font: -apple-system-headline; font-family: Verdana, sans-serif; font-
size:calc(\(h1Size * portraitMultiplier)px + 1.0vw); font-weight: bold; color:
\(headFontColor) } "
 + "h2 { font: -apple-system-headline; font-family: Verdana, sans-serif; font-
size:calc(\(h2Size * portraitMultiplier)px + 1.0vw); font-weight: bold; color:
\(headFontColor) } "
 + "h3, h4 { font: -apple-system-headline; font-family: Verdana, sans-serif; font-
size:calc(\(h3Size * portraitMultiplier)px + 1.0vw); font-weight: bold; color:
\(headFontColor) } } "
 + "@media all and (orientation:landscape) {img {max-width: 65%; height: auto;}"
 + "p, li { font: -apple-system-body; font-family: Georgia, serif; font-
size:calc(\(bodySize * landscapeMultiplier)px + 1.0vw); font-weight: normal; color:
\(fontColor) }"
 + "h1 { font: -apple-system-headline; font-family: Verdana, sans-serif; font-
size:calc(\(h1Size * landscapeMultiplier)px + 1.0vw); font-weight: bold; color:
\(headFontColor) } "
 + "h2 { font: -apple-system-headline; font-family: Verdana, sans-serif; font-
size:calc(\(h2Size * landscapeMultiplier)px + 1.0vw); font-weight: bold; color:
\(headFontColor) } "
 + "h3, h4 { font: -apple-system-headline; font-family: Verdana, sans-serif; font-
size:calc(\(h3Size * landscapeMultiplier)px + 1.0vw); font-weight: bold; color:
\(headFontColor) } } </style>"
 + "</head><body>"
 + "<meta name=\"viewport\" content=\"width: device-width\">"

 return patternText
}
```

## iOS 10

UILabel UITextField UITextView **iOS 10** adjustsFontForContentSizeCategory°

---

```
@IBOutlet var label:UILabel!

if #available(iOS 10.0, *) {
 label.adjustsFontForContentSizeCategory = true
} else {
 // Observe for UIContentSizeCategoryDidChangeNotification and handle it manually
 // since the adjustsFontForContentSizeCategory property isn't available.
}
```

<https://riptutorial.com/zh-TW/ios/topic/4466/>

# 125:

Codable Xcode 9 iOS 11 Swift 4 Codable JSON。

Codable Codable Codable Encodable Decodable。。

## Examples

### Swift 4 Codable JSON Encoder JSON Decoder

Codable。。

```
struct Movie: Codable {
 enum MovieGenre: String, Codable {
 case horror, skifi, comedy, adventure, animation
 }

 var name : String
 var moviesGenre : [MovieGenre]
 var rating : Int
}
```

```
let upMovie = Movie(name: "Up", moviesGenre: [.comedy , .adventure, .animation], rating : 4)
```

upMovie "Up" movieGenre 4。

JSON Encoder JSON。 JSON Encoder Codable。

```
// Encode data
let jsonEncoder = JSONEncoder()
do {
 let jsonData = try jsonEncoder.encode(upMovie)
 let jsonString = String(data: jsonData, encoding: .utf8)
 print("JSON String : " + jsonString!)
}
catch {
}
```

JSON Encoder JSON JSON。

```
{
 "name": "Up",
 "moviesGenre": [
 "comedy",
 "adventure",
 "animation"
],
 "rating": 4
}
```

## JSONDecoderJSON。 JSON。

```
do {
 // Decode data to object

 let jsonDecoder = JSONDecoder()
 let upMovie = try jsonDecoder.decode(Movie.self, from: jsonData)
 print("Rating : \(upMovie.name)")
 print("Rating : \(upMovie.rating)")
}
catch {
}
```

## JSONDataMovie。 ◦

```
Name : Up
Rating : 4
```

<https://riptutorial.com/zh-TW/ios/topic/10639/>

# 126: Coreplot

## Examples

### CorePlot

Core Plotpodspeccocoapods

cocoapods

```
pod init Podfile
```

```
Podfilepod'CorePlot"> 1.6'
```

```
cdpod install
```

```
Cocoapodsxcworkspace.xcodeprojpod
```

```
CocoaPods.xcworkspace
```

### ViewController.h

```
#import <CorePlot/ios/CorePlot.h>
//#import "CorePlot-CocoaTouch.h" or the above import statement
@interface ViewController : UIViewController<CPTPlotDataSource>
```

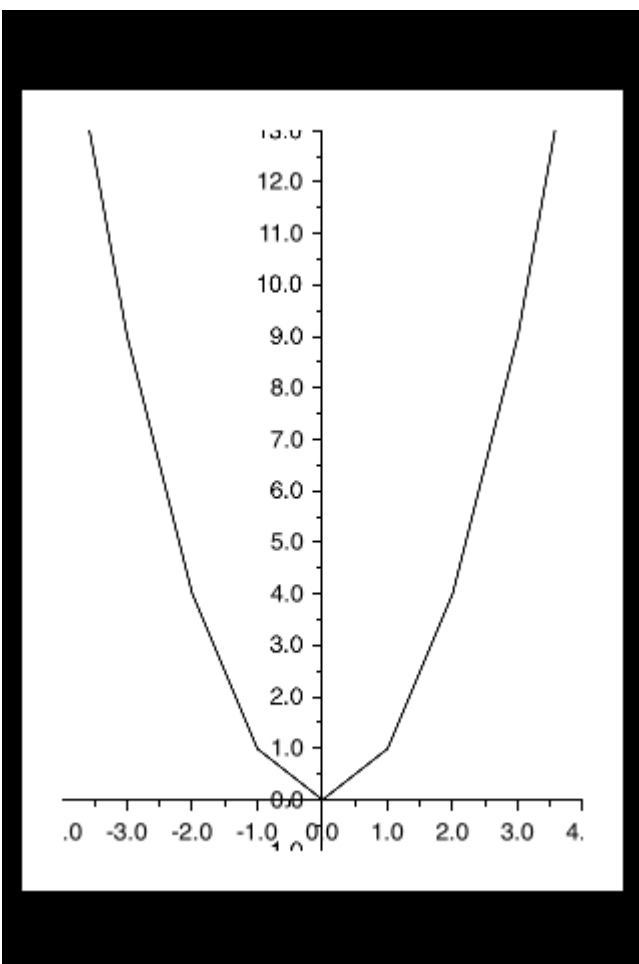
### ViewController.m

```
-(void)loadView
{
 [super loadView];
 // We need a hostview, you can create one in IB (and create an outlet) or just do this:
 CPTGraphHostingView* hostView = [[CPTGraphHostingView alloc] initWithFrame:CGRectMake(10,
 40, 300, 400)];
 hostView.backgroundColor=[UIColor whiteColor];
 self.view.backgroundColor=[UIColor blackColor];
 [self.view addSubview: hostView];
 // Create a CPTGraph object and add to hostView
 CPTGraph* graph = [[CPTXYGraph alloc] initWithFrame:CGRectMake(10, 40, 300, 400)];
 hostView.hostedGraph = graph;
 // Get the (default) plotspace from the graph so we can set its x/y ranges
 CPTXYPlotSpace *plotSpace = (CPTXYPlotSpace *) graph.defaultPlotSpace;
 // Note that these CPTPlotRange are defined by START and LENGTH (not START and END) !!
 [plotSpace setYRange: [CPTPlotRange plotRangeWithLocation:CPTDecimalFromFloat(0)
 length:CPTDecimalFromFloat(20)]];
 [plotSpace setXRange: [CPTPlotRange plotRangeWithLocation:CPTDecimalFromFloat(-4)
 length:CPTDecimalFromFloat(8)]];
 // Create the plot (we do not define actual x/y values yet, these will be supplied by the
 datasource...)
 CPTScatterPlot* plot = [[CPTScatterPlot alloc] initWithFrame:CGRectMakeZero];
 // Let's keep it simple and let this class act as datasource (therefore we implemtn
 <CPTPlotDataSource>)
 plot.dataSource = self;
```

```

// Finally, add the created plot to the default plot space of the CPTGraph object we
created before
[graph addPlot:plot toPlotSpace:graph.defaultPlotSpace];
}
// This method is here because this class also functions as datasource for our graph
// Therefore this class implements the CPTPlotDataSource protocol
-(NSUInteger)numberOfRecordsForPlot:(CPTPlot *)plotnumberOfRecords
{
 return 9; // Our sample graph contains 9 'points'
}
// This method is here because this class also functions as datasource for our graph
// Therefore this class implements the CPTPlotDataSource protocol
-(NSNumber *)numberForPlot:(CPTPlot *)plot field:(NSUInteger)fieldEnum
recordIndex:(NSUInteger)index
{
 // We need to provide an X or Y (this method will be called for each) value for every
index
 int x = index - 4;
 // This method is actually called twice per point in the plot, one for the X and one for
the Y value
 if(fieldEnum == CPTScatterPlotFieldX)
 {
 // Return x value, which will, depending on index, be between -4 to 4
 return [NSNumber numberWithInt: x];
 } else
 {
 // Return y value, for this example we'll be plotting y = x * x
 return [NSNumber numberWithInt: x * x];
 }
}
}

```



Coreplot <https://riptutorial.com/zh-TW/ios/topic/7302/-coreplot->



# 127: iOS

## Examples

### Swift

#### Swift-IOs

1. ◦ Xcode
2. iOS / FrameworkLibrary / Cocoa Touch Framework
3. ""productName
4. nextProject
- 5.

""""

- 1.
2. product""\* .framework""Embedded Binaries"
3. ViewController

iOS <https://riptutorial.com/zh-TW/ios/topic/7331/ios>

# 128:

## Examples

### Swift 2 IAP

#### iTunesConnectIAP

```
import StoreKit
```

```
class ViewController: UIViewController, SKProductsRequestDelegate, SKPaymentTransactionObserver {
```

#### iTunesConnectID

```
var product_id: NSString?

override func viewDidLoad() {

 product_id = "YOUR_PRODUCT_ID"
 super.viewDidLoad()
 SKPaymentQueue.defaultQueue().addTransactionObserver(self)

 //Check if product is purchased
 if (NSUserDefaults.standardUserDefaults().boolForKey("purchased")) {

 // Hide ads
 adView.hidden = true

 } else {
 print("Should show ads...")
 }

}
```

#### IAP

```
@IBAction func unlockAction(sender: AnyObject) {

 print("About to fetch the product...")

 // Can make payments
 if (SKPaymentQueue.canMakePayments())
 {
 let productID:NSSet = NSSet(object: self.product_id!);
 let productsRequest:SKProductsRequest = SKProductsRequest(productIdentifiers:
productID as! Set<NSString>);
 productsRequest.delegate = self;
 productsRequest.start();
 println("Fetching Products");
 }else{
```

```

 print("Can't make purchases");
 }
}

```

```

func buyProduct(product: SKProduct){
 println("Sending the Payment Request to Apple");
 let payment = SKPayment(product: product)
 SKPaymentQueue.defaultQueue().addPayment(payment);
}

```

```

func productsRequest (request: SKProductsRequest, didReceiveResponse response:
SKProductsResponse) {

 let count : Int = response.products.count
 if (count>0) {
 var validProduct: SKProduct = response.products[0] as SKProduct
 if (validProduct.productIdentifier == self.product_id) {
 print(validProduct.localizedTitle)
 print(validProduct.localizedDescription)
 print(validProduct.price)
 buyProduct(validProduct);
 } else {
 print(validProduct.productIdentifier)
 }
 } else {
 print("nothing")
 }
}

func request(request: SKRequest!, didFailWithError error: NSError!) {
 print("Error Fetching product information");
}

func paymentQueue(_ queue: SKPaymentQueue,
updatedTransactions transactions: [SKPaymentTransaction])
{
 print("Received Payment Transaction Response from Apple");

 for transaction:AnyObject in transactions {
 if let trans:SKPaymentTransaction = transaction as? SKPaymentTransaction{
 switch trans.transactionState {
 case .Purchased:
 print("Product Purchased");
 SKPaymentQueue.defaultQueue().finishTransaction(transaction as!
SKPaymentTransaction)
 // Handle the purchase
 UserDefaults.standardUserDefaults().setBool(true , forKey: "purchased")
 adView.hidden = true
 break;
 case .Failed:
 print("Purchased Failed");
 SKPaymentQueue.defaultQueue().finishTransaction(transaction as!
SKPaymentTransaction)
 break;
 }
 }
 }
}

```

```
 case .Restored:
 print("Already Purchased");
 SKPaymentQueue.defaultQueue().restoreCompletedTransactions()

 // Handle the purchase
 UserDefaults.standardUserDefaults().setBool(true , forKey: "purchased")
 adView.hidden = true
 break;
 default:
 break;
 }
}
}
```

```
if (SKPaymentQueue.canMakePayments()) {
 SKPaymentQueue.defaultQueue().restoreCompletedTransactions()
}
```

## iTunesConnect

[iTunesConnect IAP](#)。

# In-App Purchases (0)

Click

。 IAP。

IAP。

## In-App Purchase Summary

Enter a reference name and a product ID for this In-App Purchase

Reference Name

Product ID

### Pricing and Availability

Enter the pricing and availability details for this In-App Purchase

Cleared for Sale

Price

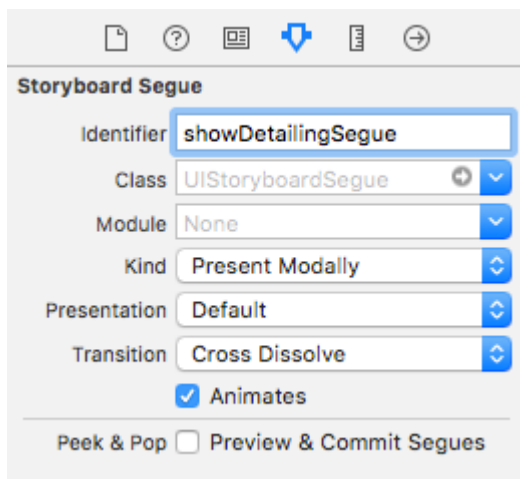
# 129:

## Examples

### Segues

`segue segue.prepareForSegue segue`

`segue`



`ctrl + segue segue`

### Objective-C

```
- (void) showDetail {
 [self performSegueWithIdentifier:@"showDetailingSegue" sender:self];
}
```

```
func showDetail() {
 self.performSegue(withIdentifier: "showDetailingSegue", sender: self)
}
```

`prepareForSegue segue`

### Objective-C

```
- (void) prepareForSegue: (UIStoryboardSegue *) segue sender: (id) sender {
 if ([segue.identifier isEqualToString:@"showDetailingSegue"]) {
 DetailViewController *controller = (DetailViewController
*) segue.destinationViewController;
 controller.isDetailingEnabled = YES;
 }
}
```

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
```

```
if segue.identifier == "showDetailingSegue" {
 let controller = segue.destinationViewController as! DetailViewController
 controller.isDetailingEnabled = true
}
}
```

DetailViewController isDetailingEnabled°

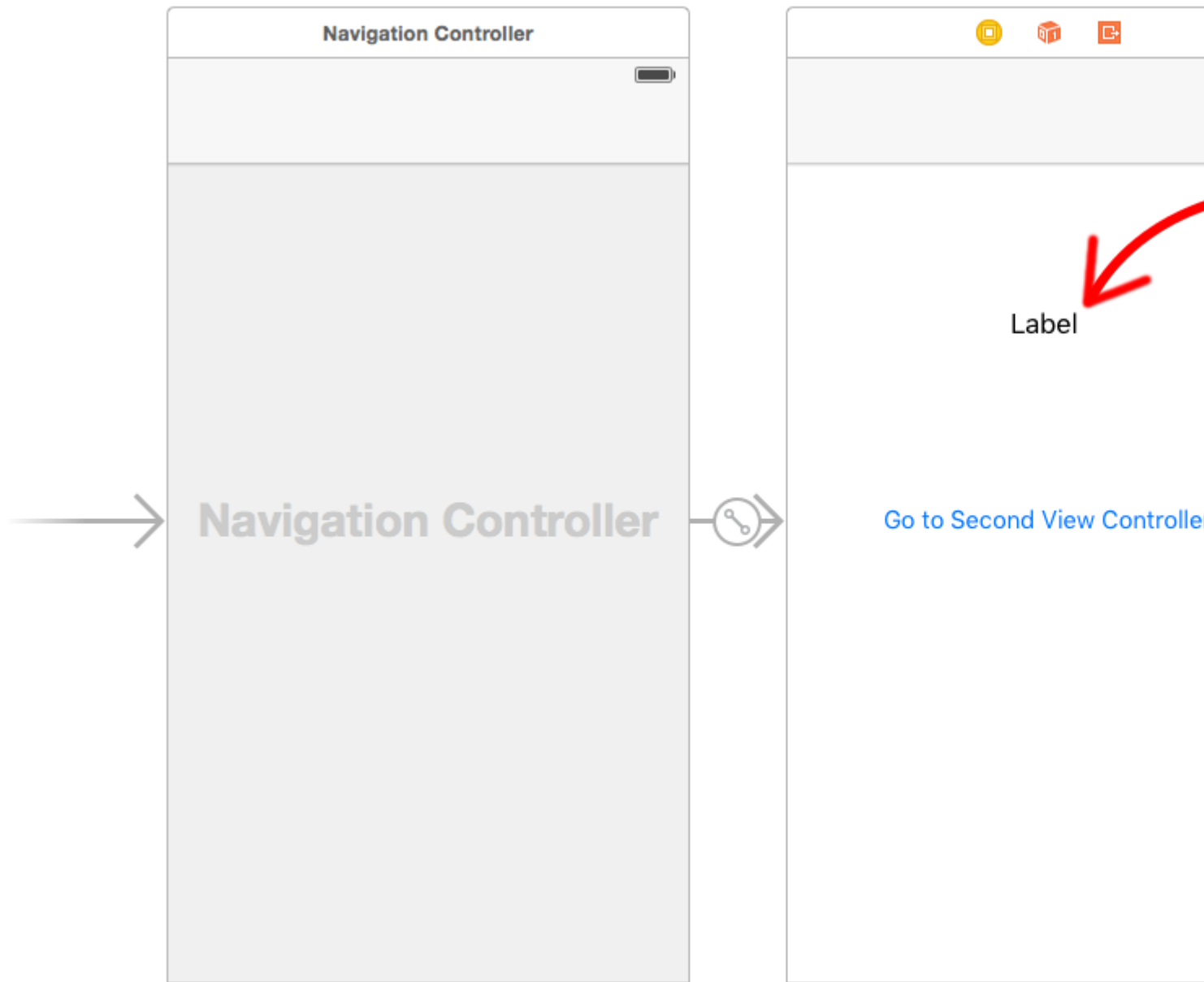
DetailViewController° DetailViewController° °

## Objective-C

```
- (void)initVC:(BOOL *)isDetailingEnabled {
 self.isDetailingEnabled = isDetailingEnabled
}
```

```
func initVC(isDetailingEnabled: Bool) {
 self.isDetailingEnabled = isDetailingEnabled
}
```

°



Interface Builder segue segue showSecondViewController ◦ ◦

```
class FirstViewController: UIViewController, DataEnteredDelegate {

 @IBOutlet weak var label: UILabel!

 override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
 if segue.identifier == "showSecondViewController", let secondViewController =
segue.destinationViewController as? SecondViewController {
 secondViewController.delegate = self
 }
 }

 // required method of our custom DataEnteredDelegate protocol
 func userDidEnterInformation(info: String) {
 label.text = info
 navigationController?.popViewControllerAnimated(true)
 }
}
```



## Objective-C

```
@interface FirstViewController : UIViewController <DataEnteredDelegate>
@property (weak, nonatomic) IBOutlet UILabel *label;
@end

@implementation FirstViewController
- (void)viewDidLoad {
 [super viewDidLoad];
}
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
 SecondViewController *secondViewController = segue.destinationViewController;
 secondViewController.delegate = self;
}
-(void)userDidEnterInformation:(NSString *)info {
 _label.text = info
 [self.navigationController popViewControllerAnimated:YES];
}
@end
```

DataEnteredDelegate°

```
// protocol used for sending data back
protocol DataEnteredDelegate: class {
 func userDidEnterInformation(info: String)
}

class SecondViewController: UIViewController {

 // making this a weak variable so that it won't create a strong reference cycle
 weak var delegate: DataEnteredDelegate?

 @IBOutlet weak var textField: UITextField!

 @IBAction func sendTextBackButton(sender: AnyObject) {

 // call this method on whichever class implements our delegate protocol (the first
 view controller)
 delegate?.userDidEnterInformation(textField.text ?? "")
 }
}
```

## Objective-C

```
@protocol DataEnteredDelegate <NSObject>
-(void)userDidEnterInformation:(NSString *)info;
@end

@interface SecondViewController : UIViewController
@property (nonatomic) id <DataEnteredDelegate> delegate;
@property (weak, nonatomic) IBOutlet UITextField *textField;
@end

@implementation SecondViewController
```

```

- (void) viewDidLoad {
 [super viewDidLoad];
}

- (IBAction) sendTextBackButton:(id) sender {
 [_delegate userDidEnterInformation:textField.text];
}
@end

```

protocol View Controller.

## segue

segue""

VC1 -> VC2

""

VC1 <- VC2

VC2.

VC1VC1

```
@IBAction func unwindToPresentingViewController(segue:UIStoryboardSegue)
```

unwind ""Xcode.

## segue

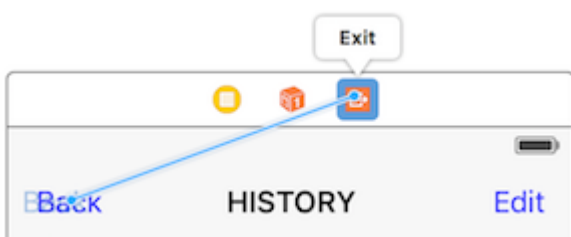
```

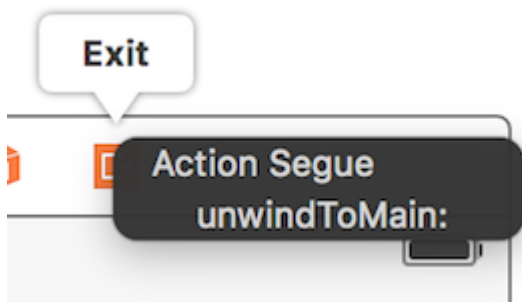
@IBAction func unwindToPresentingViewController(segue:UIStoryboardSegue)
{
 if segue.identifier == "YourCustomIdentifier"
 {
 if let VC2 = segue.sourceViewController as? VC2
 {
 // Your custom code in here to access VC2 class member
 }
 }
}

```

2

1. "" self.performSegueWithIdentifier("YourCustomIdentifier", sender: self)  
performSegueWithIdentifier.
2. storyboard ""Ctrl""





UIViewController closures • UIStoryboardSegue prepareForSegue

```
final class DestinationViewController: UIViewController {
 var onComplete: ((success: Bool) -> ())?

 @IBAction func someButtonTapped(sender: AnyObject?) {
 onComplete?(success: true)
 }
}

final class MyViewController: UIViewController {
 override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {

 guard let destinationController = segue.destinationViewController as?
 DestinationViewController else { return }

 destinationController.onCompletion = { success in
 // this will be executed when `someButtonTapped(_:)` will be called
 print(success)
 }
 }
}
```

## SwiftObjective-C

iOS • delegate pattern closure closure

delegate pattern prepareForSegue userDidEnterInformation prepareForSegue

## callbackViewController

textString

delegate

```
class SecondViewController: UIViewController {

 //weak var delegate: DataEnteredDelegate? = nil
 var callback: ((String?) -> ())?

 @IBOutlet weak var textField: UITextField!

 @IBAction func sendTextBackButton(sender: AnyObject) {

 //delegate?.userDidEnterInformation(textField.text!)
 callback?(input.text)
 }
}
```

```

 self.navigationController?.popViewControllerAnimated(true)
 }
}

```

delegate pattern

```

//no more DataEnteredDelegate
class FirstViewController: UIViewController {

 @IBOutlet weak var label: UILabel!

 override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
 if segue.identifier == "showSecondViewController" {
 let secondViewController = segue.destinationViewController as!
SecondViewController
 //secondViewController.delegate = self
 secondViewController.callback = { text in self.label.text = text }
 }
 }

 // required method of our custom DataEnteredDelegate protocol
 //func userDidEnterInformation(info: String) {
 // label.text = info
 //}
}

```

o

```

class FirstViewController: UIViewController {

 func openSecondViewController() {

 // Here we initialize SecondViewController and set the id property to 492
 let secondViewController = SecondViewController()
 secondViewController.id = 492

 // Once it was assign we now push or present the view controller
 present(secondViewController, animated: true, completion: nil)
 }

}

class SecondViewController: UIViewController {

 var id: Int?

 override func viewDidLoad() {
 super.viewDidLoad()

 // Here we unwrapped the id and will get the data from the previous view controller.
 if let id = id {
 print("Id was set: \(id)")
 }
 }

}

```

<https://riptutorial.com/zh-TW/ios/topic/434/>

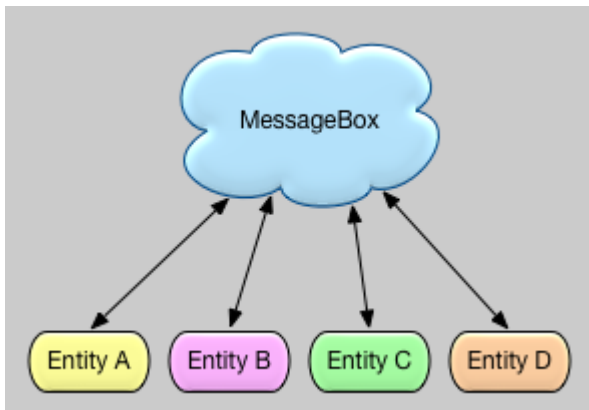
# 130: MessageBox-Concept

MessageBox。

AB。

。

## Examples



```
let messageBox:MessageBox = MessageBox()

// set
messageBox.setObject("TestObject1", forKey:"TestKey1")

// get
// but don't remove it, keep it stored, so that it can still be retrieved later
let someObject:String = messageBox.getObject(forKey:"TestKey1", removeIfFound:false)

// get
// and remove it
let someObject:String = messageBox.getObject(forKey:"TestKey1", removeIfFound:true)
```

**MessageBox-Concept** <https://riptutorial.com/zh-TW/ios/topic/9118/-messagebox-concept->

# 131: I / O.

## Examples

### Documents

#### 3

```
import UIKit

// Save String to file
let fileName = "TextFile"
let documentDirectory = try FileManager.default.urlForDirectory(.documentDirectory, in:
.userDomainMask, appropriateFor: nil, create: true)

var fileURL = try
documentDirectory.appendingPathComponent(fileName).appendingPathExtension("txt")

print("FilePath: \(fileURL.path)")

var toFileString = "Text to write"
do {
 // Write to file
 try toFileString.writeToURL(fileURL, atomically: true, encoding: NSUTF8StringEncoding)
} catch let error as NSError {
 print("Failed writing to URL: \(fileURL), Error:\(error.localizedDescription)")
}

// Reading
var fromFileString = ""
do {
 fromFileString = try String(contentsOfURL: fileURL)
} catch let error as NSError {
 print("Failed reading from URL: \(fileURL), Error: " + error.localizedDescription)
}
print("Text input from file: \(fromFileString)")
```

#### 2

```
import UIKit

// Save String to file
let fileName = "TextFile"
let DocumentDirectoryURL = try!
NSFileManager.defaultManager().URLForDirectory(.DocumentDirectory, inDomain: .UserDomainMask,
appropriateForURL: nil, create: true)

let fileURL =
DocumentDirectoryURL.URLByAppendingPathComponent(fileName).URLByAppendingPathExtension("txt")
print("FilePath: \(fileURL.path)")

var toFileString = "Text to write"
do {
 // Write to file
 try toFileString.writeToURL(fileURL, atomically: true, encoding: NSUTF8StringEncoding)
```

```
} catch let error as NSError {
 print("Failed writing to URL: \(fileURL), Error:\(error.localizedDescription)")
}

// Reading
var fromFileString = ""
do {
 fromFileString = try String(contentsOfURL: fileURL)
} catch let error as NSError {
 print("Failed reading from URL: \(fileURL), Error: " + error.localizedDescription)
}
print("Text input from file: \(fromFileString)")
```

I / O. <https://riptutorial.com/zh-TW/ios/topic/8892/i---o->

# 132:

- returnType ^ blockName parameterTypes = ^ returnType parameters{...};
- @property nonatomic copy returnType ^ blockName parameterTypes;
- - void methodWithBlock returnType ^ parameterTypes blockName;
- typedef

```
typedef returnType ^ TypeName parameterTypes;
```

```
TypeName blockName = ^ returnType parameters{...};
```

## Examples

### UIView

```
[UIView animateWithDuration:1.0
 animations:^{
 someView.alpha = 0;
 otherView.alpha = 1;
 }
 completion:^(BOOL finished) {
 [someView removeFromSuperview];
 }];
```

“^”. ^{ ... }. “void”. “ - void something;”。

。 ^BOOL someArg NSString someStr{...} \*。 APINURLConnection。

### 1-

```
typedef void(^myCustomCompletion)(BOOL);
```

### 2-

```
-(void) customMethodName:(myCustomCompletion) compblock{
 //do stuff
 // check if completion block exist; if we do not check it will throw an exception
 if(compblock)
 compblock(YES);
}
```

### 3-Method

```
[self customMethodName:^(BOOL finished) {
 if(finished) {
 NSLog(@"success");
 }
}
```



```
};
```

## Block. “const”

```
int val = 10;
void (^blk)(void) = ^{
 val = 20; // Error! val is a constant value and cannot be modified!
};
```

## \_\_block.

```
__block int val = 10;
void (^blk)(void) = ^{
 val = 20; // Correct! val now can be modified as an ordinary variable.
};
```

<https://riptutorial.com/zh-TW/ios/topic/6888/>

# 133:

## Examples

Apple

UIStoryboardSegue ◦ segue ◦ **Segue** ◦ segue storyboardprepareForSegue sender ◦

```
sourceViewController: UIViewController {get}
destinationViewController: UIViewController {get}
identifier: String? {get}
```

- [UIViewController](#)
- [UIStoryboardSegue](#)

Segue

## PrepareForSegue

```
func prepareForSegue(_ segue: UIStoryboardSegue, sender sender: AnyObject?)
```

segue

*segue* segue ◦

*sender* segue ◦

## Swift

segue“SomeSpecificIdentifier”

```
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
 if segue.identifier == "SomeSpecificIdentifier" {
 //- Do specific task
 }
}
```

Segue ◦

## ShouldPerformSegueWithIdentifier

```
func shouldPerformSegueWithIdentifier(_ identifier:String, sender sender:AnyObject?) -> Bool
```

segue.

segue

segue.

## Swift

“SomeSpecificIdentifier”segue

```
override func shouldPerformSegueWithIdentifier(identifier:String, sender:AnyObject?) -> Bool {
 if identifier == "SomeSpecificIdentifier" {
 return true
 }
 return false
}
```

## Segues

---

Unwind Segues“”。 Interface Builder。 **IBActionUIStoryboardSegue** 。 。 。  
UIViewControllerUnwind Segue。 [] [1]

segue

C

```
-(IBAction)prepareForUnwind:(UIStoryboardSegue *)segue {
}
```

```
@IBAction func prepareForUnwind(segue: UIStoryboardSegue) {
}
```

## Segue

## PerformSegueWithIdentifier

```
func performSegueWithIdentifier(_ identifier:String, sender sender:AnyObject?)
```

storyboardsegue

segue

segue.

# Swift

## “SomeSpecificIdentifier”segue

```
func tableView(tableView: UITableView, didSelectRowAtIndexPath indexPath: NSIndexPath) {
 performSegueWithIdentifier("SomeSpecificIdentifier", sender: indexPath.item)
}
```

<https://riptutorial.com/zh-TW/ios/topic/5575/>

# 134:

iOS。

## Examples

。

**1** -RRMulticastDelegate NSObject

**2** -RRMulticastDelegate.h

```
#import <Foundation/Foundation.h>

@interface RRMulticastDelegate : NSObject

{
 //Handle multiple observers of delegate
 NSMutableArray* _delegates;
}

// Delegate method implementation to the list of observers
- (void)addDelegate:(id)delegate;
- (void)removeDelegate:(id)delegate;

// Get multiple delegates
-(NSArray *)delegatesObjects;

@end
```

**3** -RRMulticastDelegate.m

```
#import "RRMulticastDelegate.h"

@implementation RRMulticastDelegate

- (id)init
{
 if (self = [super init])
 {
 _delegates = [NSMutableArray array];
 }
 return self;
}

-(NSArray *)delegatesObjects
{
 return _delegates;
}

- (void)removeDelegate:(id)delegate
{
 if ([_delegates containsObject:delegate])
 [_delegates removeObject:delegate];
}
```

```

}

- (void)addDelegate:(id)delegate
{
 if (![delegates containsObject:delegate])
 [delegates addObject:delegate];
}

- (BOOL)respondsToSelector:(SEL)aSelector
{
 if ([super respondsToSelector:aSelector])
 return YES;

 // if any of the delegates respond to this selector, return YES
 for(id delegate in _delegates)
 {
 if (!delegate)
 continue;

 if ([delegate respondsToSelector:aSelector])
 {
 return YES;
 }
 }
 return NO;
}

- (NSMethodSignature *)methodSignatureForSelector:(SEL)aSelector
{
 // can this class create the signature?
 NSMethodSignature* signature = [super methodSignatureForSelector:aSelector];

 // if not, try our delegates
 if (!signature)
 {
 for(id delegate in _delegates)
 {
 if (!delegate)
 continue;

 if ([delegate respondsToSelector:aSelector])
 {
 return [delegate methodSignatureForSelector:aSelector];
 }
 }
 }
 return signature;
}

- (void)forwardInvocation:(NSInvocation *)anInvocation
{
 // forward the invocation to every delegate
 for(id delegate in _delegates)
 {
 if (!delegate)
 continue;

 if ([delegate respondsToSelector:[anInvocation selector]])
 {
 [anInvocation invokeWithTarget:delegate];
 }
 }
}

```

```
 }
}

@end
```

#### 4 -RRProperty NSObject

#### 5 -NSObject+RRProperty.h

```
#import <Foundation/Foundation.h>

#import "RRMulticastDelegate.h"

@interface NSObject (RRProperty)<UITextFieldDelegate,UITableViewDataSource>

-(void)setObject:(id)block forKey:(NSString *)key;
-(id)objectForKey:(NSString *)key;

#pragma mark - Multicast Delegate

-(RRMulticastDelegate *)multicastDelegate;
-(RRMulticastDelegate *)multicastDatasource;

-(void)addDelegate:(id)delegate;
-(void)addDataSource:(id)datasource;

@end
```

#### 6 -NSObject+RRProperty.m

```
#import "NSObject+RRProperty.h"

#import <objc/message.h>
#import <objc/runtime.h>

#pragma GCC diagnostic ignored "-Wprotocol"

static NSString *const MULTICASTDELEGATE = @"MULTICASTDELEGATE";
static NSString *const MULTICASTDATASOURCE = @"MULTICASTDATASOURCE";

@implementation NSObject (RRProperty)

-(void)setObject:(id)block forKey:(NSString *)key
{
 objc_setAssociatedObject(self, (__bridge const void *)key), block,
 OBJC_ASSOCIATION_RETAIN);
}

-(id)objectForKey:(NSString *)key
{
 return objc_getAssociatedObject(self, (__bridge const void *)key);
}

#pragma mark - Multicast Delegate

-(RRMulticastDelegate *)multicastDelegate
{
 id multicastDelegate = [self objectForKey:MULTICASTDELEGATE];
```

```

 if (multicastDelegate == nil) {
 multicastDelegate = [[RRMulticastDelegate alloc] init];

 [self setObject:multicastDelegate forKey:MULTICASTDELEGATE];
 }

 return multicastDelegate;
}

- (RRMulticastDelegate *)multicastDatasource
{
 id multicastDatasource = [self objectForKey:MULTICASTDATASOURCE];
 if (multicastDatasource == nil) {
 multicastDatasource = [[RRMulticastDelegate alloc] init];

 [self setObject:multicastDatasource forKey:MULTICASTDATASOURCE];
 }

 return multicastDatasource;
}

-(void)addDelegate:(id)delegate
{
 [self.multicastDelegate addDelegate:delegate];

 UITextField *text = (UITextField *) self;
 text.delegate = self.multicastDelegate;
}

-(void)addDataSource:(id)datasource
{
 [self.multicastDatasource addDelegate:datasource];
 UITableView *text = (UITableView *) self;
 text.dataSource = self.multicastDatasource;
}

@end

```

multicast dalegate.....

...

NSObject+RRProperty.h **viewController/** ◦

```

UITextView *txtView = [[UITextView alloc] initWithFrame:txtframe];
[txtView addDelegate:self];

UITableView *tblView = [[UITableView alloc] initWithFrame:tblframe];
[tblView addDelegate:self];
[tblView addDataSource:self];

```

<https://riptutorial.com/zh-TW/ios/topic/10081/>



# 135:

◦ ◦

“” ◦ `UIViewController.viewWillTransition(to:with:)` ◦

## Examples

iOS ◦ ◦

Apple ◦ ◦ ◦ “Compact / Regular” ◦

### UITraitEnvironment

```
class MyViewController: UIViewController {
 override func viewDidLoad(_ animated: Bool) {
 super.viewDidLoad(animated)
 print("Horizontal size class: \(traitCollection.horizontalSizeClass)")
 print("Vertical size class: \(traitCollection.verticalSizeClass)")
 }

 override func traitCollectionDidChange(_ previousTraitCollection: UITraitCollection?) {
 super.traitCollectionDidChange(previousTraitCollection)
 print("Trait collection changed; size classes may be different.")
 }
}
```

### UIViewUITraitEnvironment

- - Auto Layout ◦ ◦

### UIStackViewUILabel

```
class ViewController: UIViewController {
 var stackView: UIStackView!

 override func viewDidLoad() {
 super.viewDidLoad()

 stackView = UIStackView()
 for text in ["foo", "bar"] {
 let label = UILabel()
 label.translatesAutoresizingMaskIntoConstraints = false
 label.text = text
 stackView.addArrangedSubview(label)
 }

 view.addSubview(stackView)
 stackView.translatesAutoresizingMaskIntoConstraints = false
 stackView.centerXAnchor.constraint(equalTo: view.centerXAnchor).isActive = true
 stackView.centerYAnchor.constraint(equalTo: view.centerYAnchor).isActive = true
 }
}
```

```

override func viewDidLoad(_ animated: Bool) {
 super.viewDidLoad(animated)
 updateAxis(forTraitCollection: traitCollection)
}

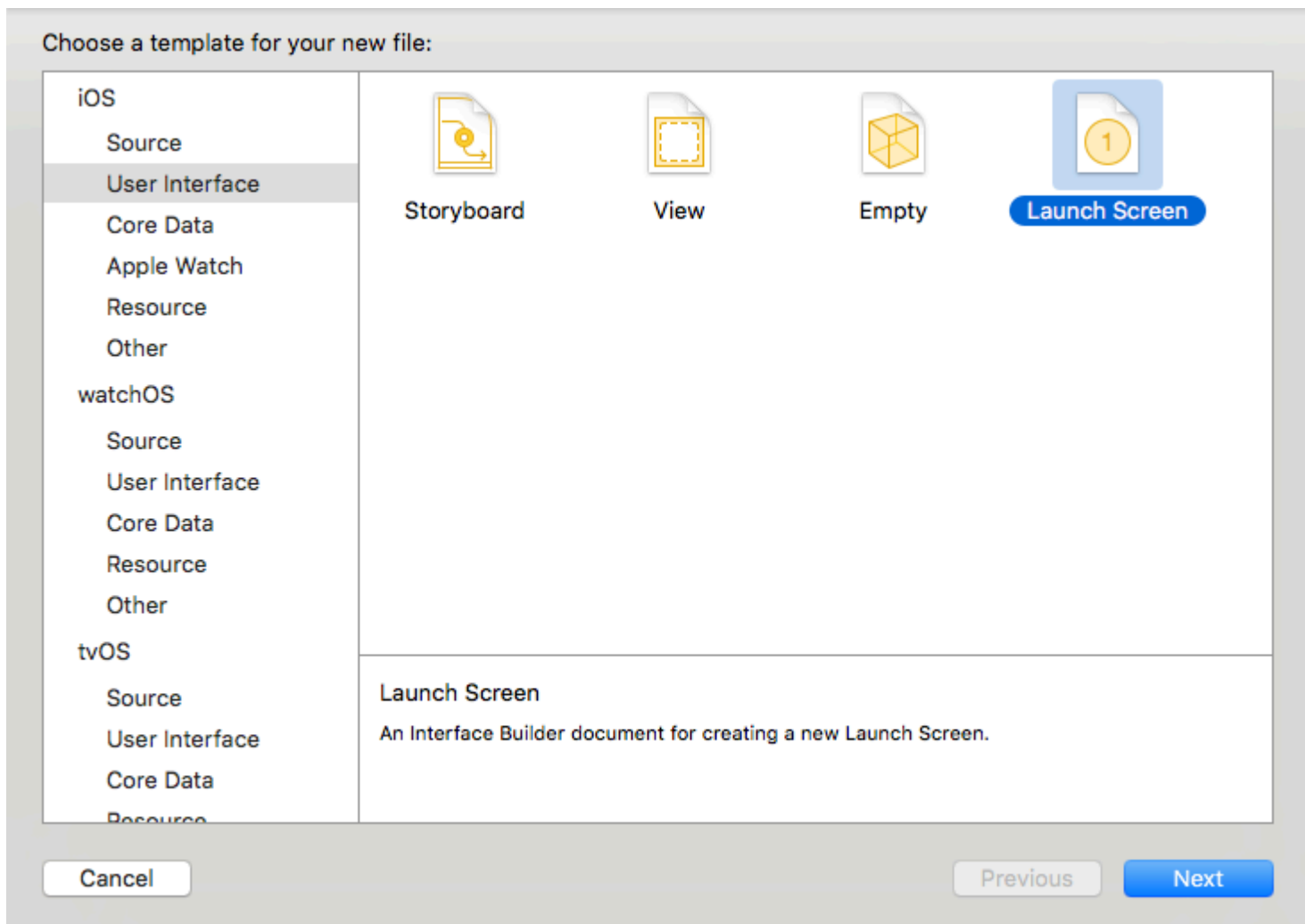
override func traitCollectionDidChange(_ previousTraitCollection: UITraitCollection?) {
 super.traitCollectionDidChange(previousTraitCollection)
 updateAxis(forTraitCollection: traitCollection)
}

private func updateAxis(forTraitCollection traitCollection: UITraitCollection) {
 switch traitCollection.horizontalSizeClass {
 case .regular:
 stackView.axis = .horizontal
 case .compact:
 stackView.axis = .vertical
 case .unspecified:
 print("Unspecified size class!")
 stackView.axis = .horizontal
 }
}
}
}

```

## iPadiOS

iOSiPad. Xcode 7LaunchScreen.storyboard.



App Icons Source   

Launch Images Source

Launch Screen File  

iPad。

<https://riptutorial.com/zh-TW/ios/topic/4628/>

---

# 136:

iOSStoryboard [Apple](#) ◦

[Apple](#) Visual Format Language ◦

## Examples

### Storyboard

nib [UIView](#) ◦

◦

- 1.



# 137:

iOS

## Examples

### SSL

iOS。

SSL。

WebSSL over HTTPHTTPS 。

https://server.com/partWebhttp://server.com/part 。

SSLserver.com 。

-

URLSessionDelegate

```
func urlSession(_ session: URLSession, didReceive challenge: URLAuthenticationChallenge,
completionHandler: @escaping (URLSession.AuthChallengeDisposition, URLCredential?) -> Void) {

 if challenge.protectionSpace.authenticationMethod == NSURLAuthenticationMethodServerTrust
 {
 let serverTrust:SecTrust = challenge.protectionSpace.serverTrust!

 func acceptServerTrust() {
 let credential:URLCredential = URLCredential(trust: serverTrust)
 challenge.sender?.use(credential, for: challenge)
 completionHandler(.useCredential, URLCredential(trust:
challenge.protectionSpace.serverTrust!))
 }

 let success = SSLTrustManager.shouldTrustServerTrust(serverTrust, forCert:
"Server_Public_SSL_Cert")
 if success {
 acceptServerTrust()
 return
 }
 }
 else if challenge.protectionSpace.authenticationMethod ==
NSURLAuthenticationMethodClientCertificate {
 completionHandler(.rejectProtectionSpace, nil);
 return
 }
 completionHandler(.cancelAuthenticationChallenge, nil)
}
```

:(Swift

```
@implementation SSLTrustManager
+ (BOOL)shouldTrustServerTrust:(SecTrustRef)serverTrust forCert:(NSString*)certName {
```

```

// Load up the bundled certificate.
NSString *certPath = [[NSBundle mainBundle] pathForResource:certName ofType:@"der"];
NSData *certData = [[NSData alloc] initWithContentsOfFile:certPath];
CFDataRef certDataRef = (__bridge_retained CFDataRef)certData;
SecCertificateRef cert = SecCertificateCreateWithData(NULL, certDataRef);

// Establish a chain of trust anchored on our bundled certificate.
CFArrayRef certArrayRef = CFArrayCreate(NULL, (void *)&cert, 1, NULL);
SecTrustSetAnchorCertificates(serverTrust, certArrayRef);

// Verify that trust.
SecTrustResultType trustResult;
SecTrustEvaluate(serverTrust, &trustResult);

// Clean up.
CFRelease(certArrayRef);
CFRelease(cert);
CFRelease(certDataRef);

// Did our custom trust chain evaluate successfully?
return trustResult == kSecTrustResultUnspecified;
}
@end

```

## Server\_Public\_SSL\_Cert.derSSL。

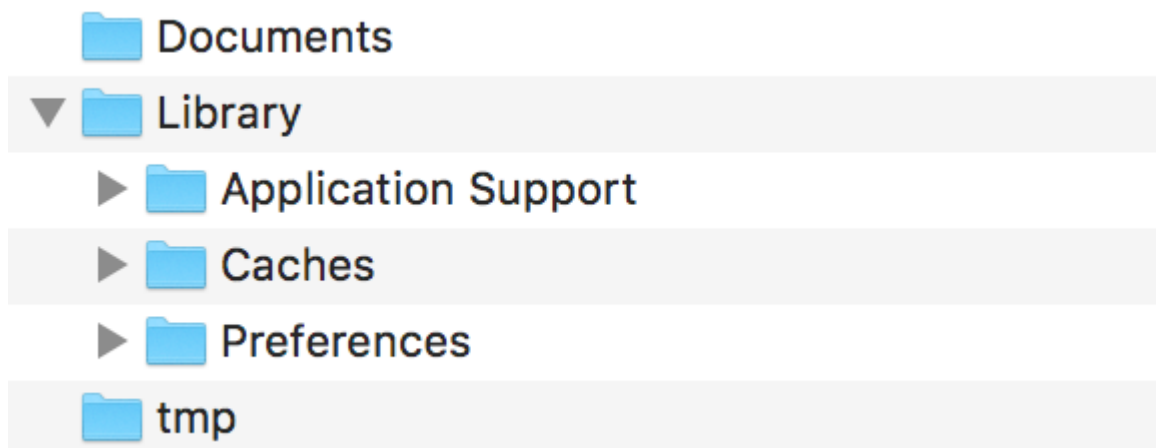
- 。

## iTunes

iTunesiTunes。

macOSiTunesiOS。

URLResourceKey.isExcludedFromBackupKey。



“”。

## Application Support

```

let urls = FileManager.default.urls(for: .applicationSupportDirectory, in:
.userDomainMask)
let baseURL = urls[urls.count-1];

```

```
let bundleIdentifier = Bundle.main.object(forKey: "CFBundleIdentifier") as!
String
let pathURL = baseURL.appendingPathComponent(bundleIdentifier)
let persistentStoreDirectoryPath = pathURL.path
if !FileManager.default.fileExists(atPath: persistentStoreDirectoryPath) {
 do {
 try FileManager.default.createDirectory(atPath: path, withIntermediateDirectories:
true, attributes: nil)
 }catch {
 //handle error
 }
}
let dirURL = URL.init(fileURLWithPath: persistentStoreDirectoryPath, isDirectory: true)
do {
 try (dirURL as NSURL).setResourceValue((true), forKey: .isExcludedFromBackupKey)
} catch {
 //handle error
}
```

iTunes。

iExploreriTunes。

<https://riptutorial.com/zh-TW/ios/topic/9999/>



---

# 138: HTMLNSAttributed

## Examples

### Objective C HTMLNSAttributedString Vice Versa

#### HTMLNSAttributedString -

```
//HTML String
NSString *htmlString=[[NSString alloc] initWithFormat:@"<!DOCTYPE html><html><body><h1>My
First Heading</h1><p>My first paragraph.</p></body></html>"];
//Converting HTML string with UTF-8 encoding to NSAttributedString
NSAttributedString *attributedString = [[NSAttributedString alloc]
initWithData: [htmlString
dataUsingEncoding:NSUTF8StringEncoding]
options: @{ NSDocumentTypeDocumentAttribute:
NSHTMLTextDocumentType }
documentAttributes: nil
error: nil];
```

#### NSAttributedStringHTML -

```
//Dictionary to hold all the attributes of NSAttributedString
NSDictionary *documentAttributes = @{NSDocumentTypeDocumentAttribute:
NSHTMLTextDocumentType};
//Saving the NSAttributedString with all its attributes as a NSData Entity
NSData *htmlData = [attributedString dataFromRange:NSMakeRange(0, attributedString.length)
documentAttributes:documentAttributes error:NULL];
//Convert the NSData into HTML String with UTF-8 Encoding
NSString *htmlString = [[NSString alloc] initWithData:htmlData
encoding:NSUTF8StringEncoding];
```

HTMLNSAttributed <https://riptutorial.com/zh-TW/ios/topic/7225/htmlnsattributed->

# 139: NSAttributedStringUIImage

## Examples

### NSAttributedStringUIImage

#### Objective-C

```
NSMutableAttributedString *str = [[NSMutableAttributedString alloc] initWithString:@"Hello.
That is a test attributed string."];
[str addAttribute:NSBackgroundColorAttributeName value:[UIColor yellowColor]
range:NSMakeRange(3,5)];
[str addAttribute:NSForegroundColorAttributeName value:[UIColor greenColor]
range:NSMakeRange(10,7)];
[str addAttribute:NSFontAttributeName value:[UIFont fontWithName:@"HelveticaNeue-Bold"
size:20.0] range:NSMakeRange(20, 10)];
UIImage *customImage = [self imageFromAttributedString:str];
```

imageFromAttributedString

```
- (UIImage *)imageFromAttributedString:(NSAttributedString *)text
{
 UIGraphicsBeginImageContextWithOptions(text.size, NO, 0.0);

 // draw in context
 [text drawAtPoint:CGPointMake(0.0, 0.0)];

 // transfer image
 UIImage *image = [UIGraphicsGetImageFromCurrentImageContext()
imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];
 UIGraphicsEndImageContext();

 return image;
}
```

[NSAttributedStringUIImage https://riptutorial.com/zh-TW/ios/topic/7242/nsattributedStringUIImage](https://riptutorial.com/zh-TW/ios/topic/7242/nsattributedStringUIImage)

# 140: UIImage

## Examples

- C

```
import #include <math.h>
```

```
viewDidLoadloadView
```

```
- (void)loadView
{
 [super loadView];
 UIImageView *imageView=[[UIImageView alloc]initWithFrame:CGRectMake(0, 50, 320, 320)];
 [self.view addSubview:imageView];
 UIImage *image=[UIImage imageNamed:@"Dubai-Photos-Images-Travel-Tourist-Images-Pictures-800x600.jpg"];
 imageView.image=[self circularScaleAndCropImage:[UIImage imageNamed:@"Dubai-Photos-Images-Travel-Tourist-Images-Pictures-800x600.jpg"] frame:CGRectMake(0, 0, 320, 320)];
}
```

```
circularScaleAndCropImage
```

```
- (UIImage*)circularScaleAndCropImage:(UIImage*)image frame:(CGRect)frame {
 // This function returns a newImage, based on image, that has been:
 // - scaled to fit in (CGRect) rect
 // - and cropped within a circle of radius: rectWidth/2

 //Create the bitmap graphics context
 UIGraphicsBeginImageContextWithOptions(CGSizeMake(frame.size.width, frame.size.height),
 NO, 0.0);
 CGContextRef context = UIGraphicsGetCurrentContext();

 //Get the width and heights
 CGFloat imageWidth = image.size.width;
 CGFloat imageHeight = image.size.height;
 CGFloat rectWidth = frame.size.width;
 CGFloat rectHeight = frame.size.height;

 //Calculate the scale factor
 CGFloat scaleFactorX = rectWidth/imageWidth;
 CGFloat scaleFactorY = rectHeight/imageHeight;

 //Calculate the centre of the circle
 CGFloat imageCentreX = rectWidth/2;
 CGFloat imageCentreY = rectHeight/2;

 // Create and CLIP to a CIRCULAR Path
 // (This could be replaced with any closed path if you want a different shaped clip)
 CGFloat radius = rectWidth/2;
 CGContextBeginPath (context);
 CGContextAddArc (context, imageCentreX, imageCentreY, radius, 0, 2*M_PI, 0);
 CGContextClosePath (context);
 CGContextClip (context);
```

```

//Set the SCALE factor for the graphics context
//All future draw calls will be scaled by this factor
CGContextScaleCTM (context, scaleFactorX, scaleFactorY);

// Draw the IMAGE
CGRect myRect = CGRectMake(0, 0, imageWidth, imageHeight);
[image drawInRect:myRect];

UIImage *newImage = UIGraphicsGetImageFromCurrentImageContext();
UIGraphicsEndImageContext();

return newImage;
}

```

## SWIFT 3

```

override func viewDidLoad() {
 super.viewDidLoad()
 // Do any additional setup after loading the view, typically from a nib.
 let imageView = UIImageView(frame: CGRect(x: CGFloat(0), y: CGFloat(50), width:
CGFloat(320), height: CGFloat(320)))
 view.addSubview(imageView)
 let image = UIImage(named: "Dubai-Photos-Images-Travel-Tourist-Images-Pictures-
800x600.jpg")
 imageView.image = circularScaleAndCropImage(UIImage(named: "Dubai-Photos-Images-
Travel-Tourist-Images-Pictures-800x600.jpg")!, frame: CGRect(x: CGFloat(0), y: CGFloat(0),
width: CGFloat(100), height: CGFloat(100)))
}

```

### circularScaleAndCropImage

```

func circularScaleAndCropImage(_ image: UIImage, frame: CGRect) -> UIImage{
 // This function returns a newImage, based on image, that has been:
 // - scaled to fit in (CGRect) rect
 // - and cropped within a circle of radius: rectWidth/2
 //Create the bitmap graphics context
 UIGraphicsBeginImageContextWithOptions(CGSize(width: CGFloat(frame.size.width),
height: CGFloat(frame.size.height)), false, 0.0)
 let context: CGContext? = UIGraphicsGetCurrentContext()
 //Get the width and heights
 let imageWidth: CGFloat = image.size.width
 let imageHeight: CGFloat = image.size.height
 let rectWidth: CGFloat = frame.size.width
 let rectHeight: CGFloat = frame.size.height
 //Calculate the scale factor
 let scaleFactorX: CGFloat = rectWidth / imageWidth
 let scaleFactorY: CGFloat = rectHeight / imageHeight
 //Calculate the centre of the circle
 let imageCentreX: CGFloat = rectWidth / 2
 let imageCentreY: CGFloat = rectHeight / 2
 // Create and CLIP to a CIRCULAR Path
 // (This could be replaced with any closed path if you want a different shaped clip)
 let radius: CGFloat = rectWidth / 2
 context?.beginPath()
 context?.addArc(center: CGPoint(x: imageCentreX, y: imageCentreY), radius: radius,
startAngle: CGFloat(0), endAngle: CGFloat(2 * Float.pi), clockwise: false)
 context?.closePath()
 context?.clip()
}

```

```
//Set the SCALE factor for the graphics context
//All future draw calls will be scaled by this factor
context?.scaleBy(x: scaleFactorX, y: scaleFactorY)
// Draw the IMAGE
let myRect = CGRect(x: CGFloat(0), y: CGFloat(0), width: imageWidth, height:
imageHeight)
image.draw(in: myRect)
let newImage: UIImage? = UIGraphicsGetImageFromCurrentImageContext()
UIGraphicsEndImageContext()
return newImage!
}
```

UIImage <https://riptutorial.com/zh-TW/ios/topic/7222/uiimage>

# 141:

## Examples

o

```
// Default UINavigationController appearance throughout the app
[[UINavigationController appearance] setTitleTextAttributes:@{NSForegroundColorAttributeName:
[UIColor whiteColor],
 NSFontAttributeName : [UIFont
fontWithName:@"HelveticaNeue-CondensedBold" size:17],
 }
];

[[UINavigationController appearance] setTintColor:[UIColor whiteColor]];
[[UINavigationController appearance] setBarTintColor:[UIColor KNGRed]];
[[UINavigationController appearance] setTranslucent:NO];
[[UINavigationController appearance] setBarStyle:UIBarStyleBlack];
[[UIBarButtonItem appearanceWhenContainedIn: [UISearchBar class], nil] setTintColor:[UIColor
KNGGray]];
```

## SWIFT

```
navigationController?.navigationBar.titleTextAttributes = [NSForegroundColorAttributeName:
UIColor.white, NSFontAttributeName:UIFont(name: "HelveticaNeue-CondensedBold", size: 17)!,]
navigationController?.navigationBar.tintColor = .white
navigationController?.navigationBar.barTintColor = .red
navigationController?.navigationBar.isTranslucent = false
navigationController?.navigationBar.barStyle = .black
```

<https://riptutorial.com/zh-TW/ios/topic/7066/>

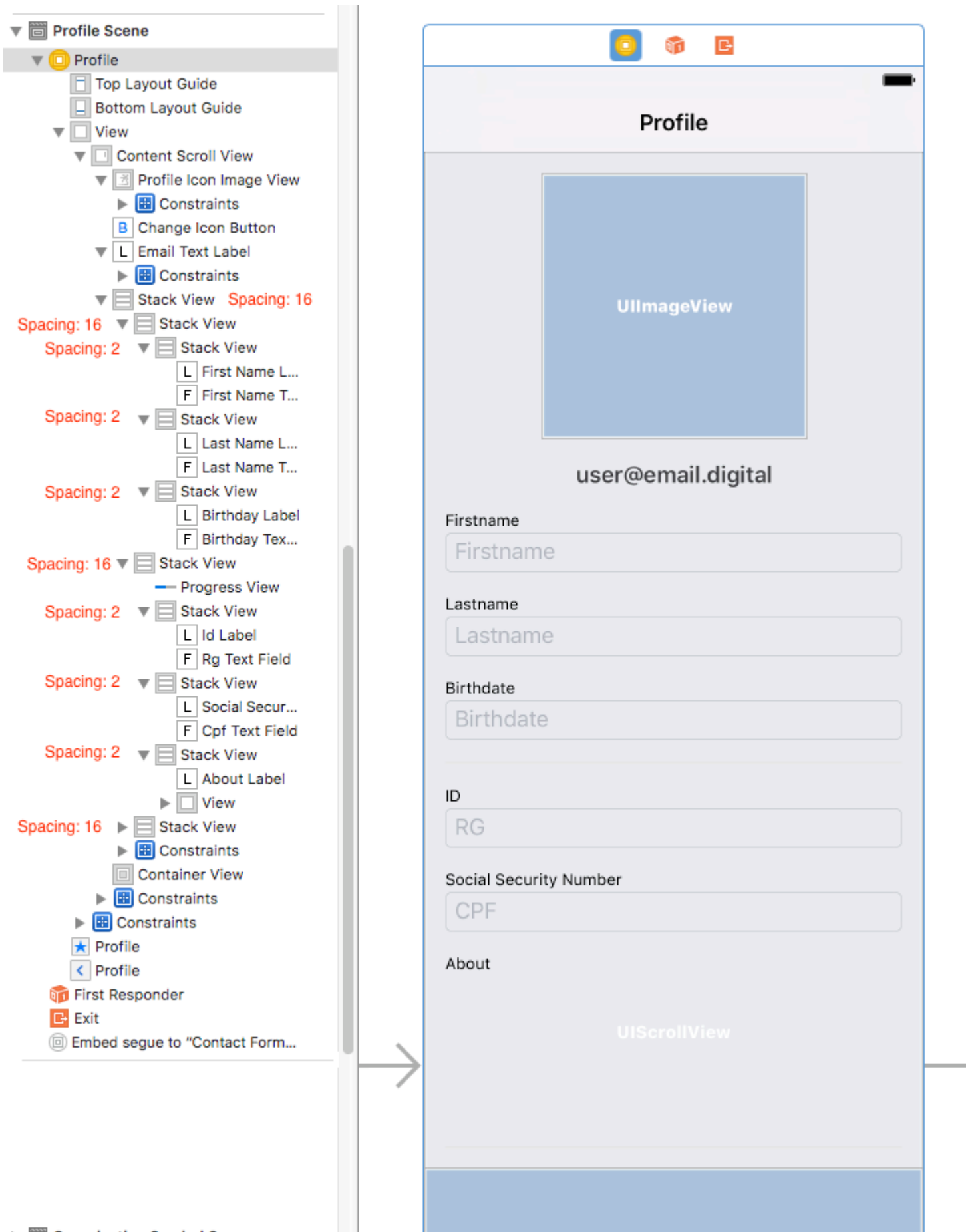
---

# 142: StackViewUIScrollView

## Examples

ScrollViewStackView

StackViews◦



ScrollViewStackViews/.

1.



```
UIScrollViewUIViewcontentScrollView;
2. contentScrollView0
3. ;
```

## StackViews

### ScrollView StackViewTextField。

```
TextField.frame.minY0 minYStackView。 /。
```

#### 1 - ScrollView

```
extension UIScrollView {

 func scrollToShowView(view: UIView){
 var offset = view.frame.minY
 var superview = view.superview
 while((superview != nil)){
 offset += (superview?.frame.minY)!
 superview = superview?.superview
 }

 offset -= 100 //optional margin added on offset

 self.contentOffset = CGPoint.init(x: 0, y: offset)
 }

}
```

#### TextField

```
func textViewDidBeginEditing(_ textView: UITextView) {
 self.contentOffset.scrollToShowView(view: textView)
}
```

[StackViewUIScrollView](https://riptutorial.com/zh-TW/ios/topic/9404/stackviewuiscrollview) <https://riptutorial.com/zh-TW/ios/topic/9404/stackviewuiscrollview>

# 143:

AVFoundation

## Examples

### UIImage

AVAssetWriter

```
NSError *error = nil;
NSURL *outputURL = <#NSURL object representing the URL where you want to save the video#>;
AVAssetWriter *assetWriter = [AVAssetWriter assetWriterWithURL:outputURL
 fileType:AVFileTypeQuickTimeMovie error:&error];
if (!assetWriter) {
 // handle error
}
```

AVAssetWriterAVAssetWriter

```
NSDictionary *writerInputParams = [NSDictionary dictionaryWithObjectsAndKeys:
 AVVideoCodecH264, AVVideoCodecKey,
 [NSNumber numberWithInt:renderSize.width],
 AVVideoWidthKey,
 [NSNumber numberWithInt:renderSize.height],
 AVVideoHeightKey,
 AVVideoScalingModeResizeAspectFill,
 AVVideoScalingModeKey,
 nil];

AVAssetWriterInput *assetWriterInput = [AVAssetWriterInput
assetWriterInputWithMediaType:AVMediaTypeVideo outputSettings:writerInputParams];
if ([assetWriter canAddInput:assetWriterInput]) {
 [assetWriter addInput:assetWriterInput];
} else {
 // show error message
}
```

CVPixelBufferRefAVAssetWriterInputAVAssetWriterInputPixelFormatAdaptor

```
NSDictionary *attributes = [NSDictionary dictionaryWithObjectsAndKeys:
 [NSNumber numberWithInt:kCVPixelFormatType_32ARGB],
 (NSString*)kCVPixelBufferPixelFormatTypeKey,
 [NSNumber numberWithBool:YES], (NSString
*)kCVPixelBufferCGImageCompatibilityKey,
 [NSNumber numberWithBool:YES], (NSString
*)kCVPixelBufferCGBitmapContextCompatibilityKey,
 nil];

AVAssetWriterInputPixelFormatAdaptor *writerAdaptor = [AVAssetWriterInputPixelFormatAdaptor
assetWriterInputPixelFormatAdaptorWithAssetWriterInput:assetWriterInput
sourcePixelFormatAttributes:attributes];
```

```
[assetWriter startWriting];
```

```

[assetWriter startSessionAtSourceTime:kCMTimeZero];
[assetWriterInput requestMediaDataWhenReadyOnQueue:exportingQueue usingBlock:^(
 for (int i = 0; i < images.count; ++i) {
 while (![assetWriterInput isReadyForMoreMediaData]) {
 [NSThread sleepForTimeInterval:0.01];
 // can check for attempts not to create an infinite loop
 }

 UIImage *uImage = images[i];

 CVPixelBufferRef buffer = NULL;
 CVReturn err = PixelBufferCreateFromImage(uImage.CGImage, &buffer);
 if (err) {
 // handle error
 }

 // frame duration is duration of single image in seconds
 CMTime presentationTime = CMTimeMakeWithSeconds(i * frameDuration, 1000000);

 [writerAdaptor appendPixelBuffer:buffer withPresentationTime:presentationTime];

 CVPixelBufferRelease(buffer);
 }

[assetWriterInput markAsFinished];
[assetWriter finishWritingWithCompletionHandler:^(
 if (assetWriter.error) {
 // show error message
 } else {
 // outputURL
 }
}]];
});

```

CGImageRefCVPixelBufferRef

```

CVReturn PixelBufferCreateFromImage(CGImageRef imageRef, CVPixelBufferRef *outBuffer) {
 CIContext *context = [CIContext context];
 CIImage *ciImage = [CIImage imageWithCGImage:imageRef];

 NSDictionary *attributes = [NSDictionary dictionaryWithObjectsAndKeys:
 [NSNumber numberWithBool:YES], (NSString
*)kCVPixelBufferCGBitmapContextCompatibilityKey,
 [NSNumber numberWithBool:YES], (NSString
*)kCVPixelBufferCGImageCompatibilityKey
 ,nil];

 CVReturn err = CVPixelBufferCreate(kCFAllocatorDefault, CGImageGetWidth(imageRef),
 CGImageGetHeight(imageRef), kCVPixelFormatType_32ARGB, (__bridge CFDictionaryRef
 _Nullable) (attributes), outBuffer);
 if (err) {
 return err;
 }

 if (outBuffer) {
 [context render:ciImage toCVPixelBuffer:*outBuffer];
 }

 return kCVReturnSuccess;
}

```



---

# 144:

## Examples

[fastlane](#) [Android](#) [iOS](#) [Ruby](#) [Mac](#) [Ruby](#)

---

## fastlane

1. `sudo gem install fastlane --verbose`
2. `fastlane init`
3. `fastlane init`
4. `fastlane init`
5. `fastlane init`
6. Fastlane

## iOS

- App Store
- iOS
- [frameit](#)
- [pem](#)
- iTunes Connect Dev Portal iOS
- [cert](#) iOS
- iOS
- Git
- [scan](#) iOS Mac
- Apple Dev Center iTunes Connect Ruby

## iOS TestFlight

- [pilot](#) TestFlight
- TestFlight beta

## Android

- Android Google Play
- [screengrab](#) Android

<https://riptutorial.com/zh-TW/ios/topic/3574/>

# 145:

## Examples

### UIViewController

UIViewController UIWindow RootViewController ◦

```
extension UIApplication {

 func topViewController(_ base: UIViewController? =
 UIApplication.shared.keyWindow?.rootViewController) -> UIViewController {

 if let nav = base as? UINavigationController {
 return topViewController(nav.visibleViewController)
 }

 if let tab = base as? UITabBarController {
 if let selected = tab.selectedViewController {
 return topViewController(selected)
 }
 }

 if let presented = base?.presentedViewController {
 return topViewController(presented)
 }

 return base!
 }
}
```

### iOSNotificationCenter

```
NotificationCenter.default.addObserver(
 self,
 selector: #selector(ViewController.do(_:)),
 name: NSNotification.Name.UIApplicationDidBecomeActive,
 object: nil)
```

<https://developer.apple.com/reference/foundation/nsnotification.name> ◦

<https://riptutorial.com/zh-TW/ios/topic/7188/>

# 146:

iOSApp Store

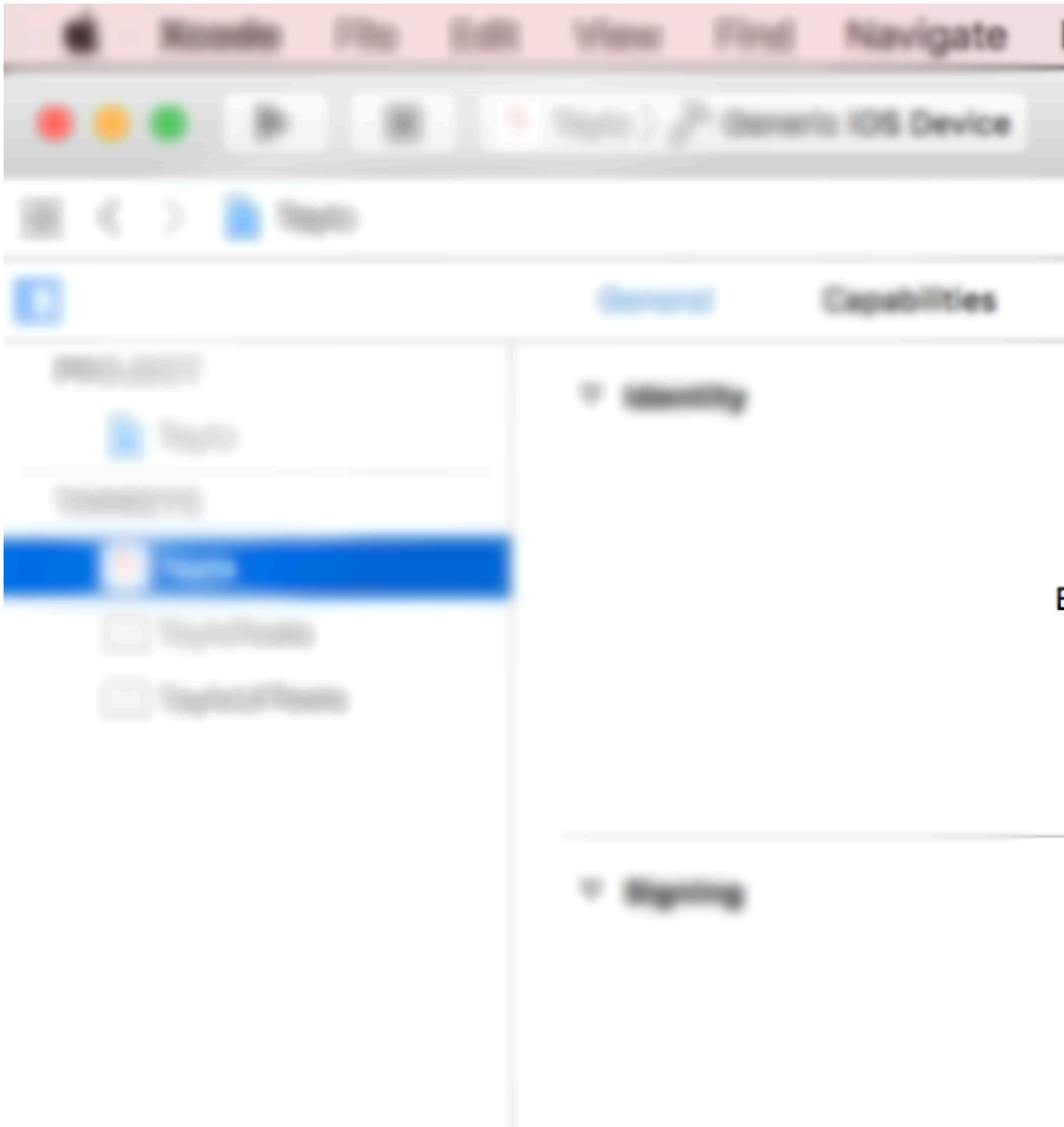
## Examples

Xcode 8. " - >"

### ▼ Signing

Automatically ma  
Xcode will create a  
certificates.

"iOS" "



◦ 301.0 ◦ 311.0.1 ◦ ◦

## IPA

Xcode ◦ ◦ “App Store .....”9/10Xcode ◦ Application LoaderXcode ◦ Application loaderIPAApp Store ◦ ◦ “”◦



## Select a method for export:

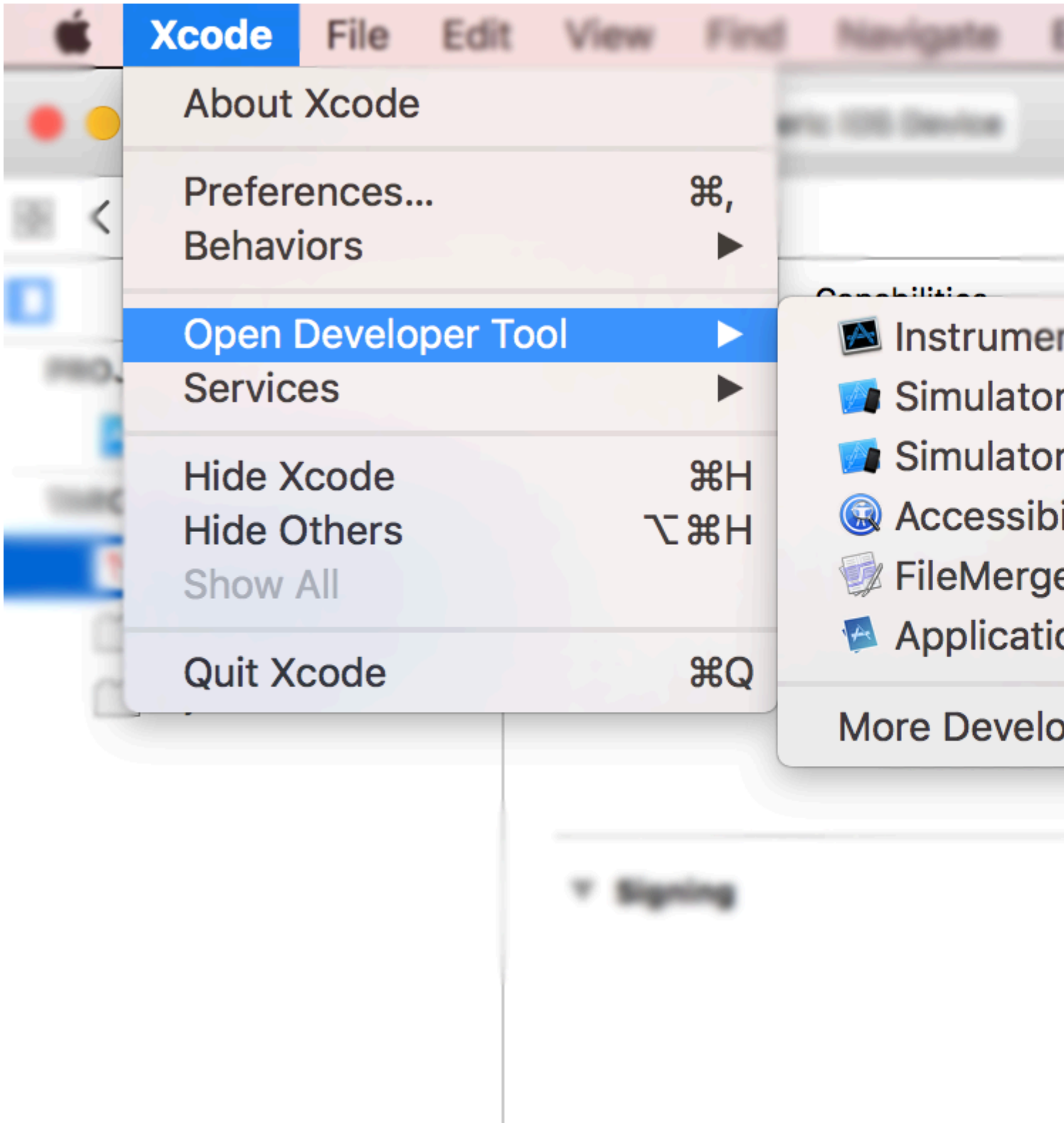
- Save for iOS App Store Deployment**  
Sign and package application for distribution in the iOS App Store
- Save for Ad Hoc Deployment**  
Sign and package application for Ad Hoc distribution outside the App Store
- Save for Enterprise Deployment**  
Sign and package application for enterprise distribution outside the App Store
- Save for Development Deployment**  
Sign and package application for development distribution outside the App Store

Cancel

App Store™ . IPA .

**Application Loader**IPA

IPAXcodeApplication Loader.



Xcode。 。 “”。 iTunes Connect。



**Deliver Your App**

Open Recent ▾

Import

<https://riptutorial.com/zh-TW/ios/topic/8765/>

# 147: ATS

| NSAppTransportSecurity                       |                                    | ATS                                                                                                                                  |
|----------------------------------------------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| NSAllowsArbitraryLoads                       | YES                                | ATS。 iOS 10 macOS 10.12 Info.plist<br>NSAllowsArbitraryLoadsInMedia<br>NSAllowsArbitraryLoadsInWebContent<br>NSAllowsLocalNetworking |
| NSAllowsArbitraryLoadsInMedia                | YES                                | AV Foundation API ATS。 <b>iOS 10 + macOS 10.12+</b>                                                                                  |
| NSAllowsArbitraryLoadsInWebContent           | YES                                | Web WKWebView UIWebView WebView ATS<br>NSURLSession。 <b>iOS 10 + macOS 10.12+</b>                                                    |
| NSAllowsLocalNetworking                      | YES                                | .local。 <b>iOS 10 + macOS 10.12+</b>                                                                                                 |
| NSExceptionDomains                           |                                    |                                                                                                                                      |
| NSIncludesSubdomains                         | YES                                | 。                                                                                                                                    |
| NSRequiresCertificateTransparency            | YES                                | CT CTX.509。 <b>iOS 10 + macOS 10.12+</b>                                                                                             |
| NSExceptionAllowsInsecureHTTPLoads           | YES                                | HTTP。                                                                                                                                |
| NSExceptionRequiresForwardSecrecy            | YES ; NO                           | Forward Secrecy。                                                                                                                     |
| NSExceptionMinimumTLSVersion                 | TLSv1.2 ;                          | TLSv1.0 TLSv1.1 TLSv1.2                                                                                                              |
| NSThirdPartyExceptionAllowsInsecureHTTPLoads | NSExceptionAllowsInsecureHTTPLoads |                                                                                                                                      |
| NSThirdPartyExceptionRequiresForwardSecrecy  | NSExceptionRequiresForwardSecrecy  |                                                                                                                                      |
| NSThirdPartyExceptionMinimumTLSVersion       | NSExceptionMinimumTLSVersion       |                                                                                                                                      |

[App Transport Security](#) iOS macOS。 。 HTTPS。

HTTP Info.plist。

2017 Apple ATS。 Info.plist ATS。 HTTP Apple。 [WWDC 2016 - 706](#)

App Transport Security [CocoaKeys](#) 。

## Examples

HTTP

Apple iOS 9 ATS Web。 ATS HTTPS。 。

ATS Info.plist NSAppTransportSecurity Xcode Info.plist App Transport Security Settings。 HTTP Allow Arbitrary Loads NSAllowsArbitraryLoads YES。 HTTP。

## HTTP

HTTP App Transport Security Settings。 Exception Domains NSExceptionDomains ATS。

。 NSExceptionAllowsInsecureHTTPLoads YES HTTPS。

## SSL

iOS 9 HTTPS。

SSL。 。

Info.plist

### 1. testdomain.com

```
<key>NSAppTransportSecurity</key>
<dict>
<key>NSExceptionDomains</key>
<dict>
 <key>testdomain.com</key>
 <dict>
 <key>NSIncludesSubdomains</key>
 <true/>
 <key>NSExceptionAllowsInsecureHTTPLoads</key>
 <true/>
 </dict>
</dict>
</dict>
```

NSExceptionAllowsInsecureHTTPLoads。 HTTP testdomain.com HTTP。

NSIncludesSubdomains testdomain.com。

### 2.

```
<key>NSAppTransportSecurity</key>
<dict>
 <key>NSAllowsArbitraryLoads</key>
 <true/>
</dict>
```

HTTP。 2017 11 App Store。

- 。
- 。
- 。

ATS <https://riptutorial.com/zh-TW/ios/topic/5435/-ats->

# 148:

- `UIUserNotificationSettings.typesUIUserNotificationType //`
- `UIUserNotificationSettings.categories//`



## Examples

didFinishLaunchingWithOptionsAppDelegate

```
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[NSObject: AnyObject]?) -> Bool {
 // Override point for customization after application launch.
 if UIDevice.currentDevice().systemVersion.compare(v, options: .NumericSearch) ==
NSOrderedAscending {
 // Register for Push Notifications, if running iOS < 8
 if application.respondsToSelector("registerUserNotificationSettings:") {
 let types:UIUserNotificationType = (.Alert | .Badge | .Sound)
 let settings:UIUserNotificationSettings = UIUserNotificationSettings(forTypes:
types, categories: nil)

 application.registerUserNotificationSettings(settings)
 application.registerForRemoteNotifications()
 } else {
 // Register for Push Notifications before iOS 8
 application.registerForRemoteNotificationTypes(.Alert | .Badge | .Sound)
 }
 } else {
 var center = UNUserNotificationCenter.currentNotificationCenter()
 center.delegate = self
 center.requestAuthorizationWithOptions((UNAuthorizationOptionSound |
UNAuthorizationOptionAlert | UNAuthorizationOptionBadge)) {(granted: Bool, error: NSError) ->
Void in
 if !error {
 UIApplication.sharedApplication().registerForRemoteNotifications()
 // required to get the app to do anything at all about push notifications
 print("Push registration success.")
 } else {
 print("Push registration FAILED")
 print("ERROR: \(error.localizedDescription) -
\(error.localizedDescription)")
 print("SUGGESTIONS: \(error.localizedRecoveryOptions) -
\(error.localizedRecoverySuggestion)")
 }
 })
 }

 return true
}
```

## Objective-C

```

#define SYSTEM_VERSION_LESS_THAN(v) ([[UIDevice currentDevice] systemVersion] compare:v
options:NSNumericSearch] == NSOrderedAscending)

if(SYSTEM_VERSION_LESS_THAN(@"10.0"))
{
 if ([application respondsToSelector:@selector(isRegisteredForRemoteNotifications)])
 {
 // iOS 8 Notifications
 [application registerUserNotificationSettings:[UIUserNotificationSettings
settingsForTypes:(UIUserNotificationTypeSound | UIUserNotificationTypeAlert |
UIUserNotificationTypeBadge) categories:nil]];

 [application registerForRemoteNotifications];
 }
 else
 {
 // iOS < 8 Notifications
 [application registerForRemoteNotificationTypes:
(UIRemoteNotificationTypeBadge | UIRemoteNotificationTypeAlert |
UIRemoteNotificationTypeSound)];
 }
}
else
{
 UNUserNotificationCenter *center = [UNUserNotificationCenter currentNotificationCenter];
 center.delegate = self;
 [center requestAuthorizationWithOptions:(UNAuthorizationOptionSound |
UNAuthorizationOptionAlert | UNAuthorizationOptionBadge) completionHandler:^(BOOL granted,
NSError * _Nullable error)
 {
 if(!error)
 {
 [[UIApplication sharedApplication] registerForRemoteNotifications]; // required
to get the app to do anything at all about push notifications
 NSLog(@"Push registration success.");
 }
 else
 {
 NSLog(@"Push registration FAILED");
 NSLog(@"ERROR: %@ - %@", error.localizedFailureReason,
error.localizedDescription);
 NSLog(@"SUGGESTIONS: %@ - %@", error.localizedRecoveryOptions,
error.localizedRecoverySuggestion);
 }
 }
 }];
}

//to check if your App lunch from Push notification
//-----
//Handel Push notification
if (launchOptions != nil)
{
 // Here app will open from pushnotification
 //RemoteNotification
 NSDictionary* dictionary1 = [launchOptions
objectForKey:UIApplicationLaunchOptionsRemoteNotificationKey];
 //LocalNotification
 NSDictionary* dictionary2 = [launchOptions
objectForKey:UIApplicationLaunchOptionsLocalNotificationKey];
}

```



```

 if (dictionary1 != nil)
 {
 //RemoteNotification Payload
 NSLog(@"Launched from push notification: %@", dictionary1);
 //here handle your push notification
 }
 if (dictionary2 != nil)
 {
 NSLog(@"Launched from dictionary2dictionary2dictionary2 notification: %@",
dictionary2);
 double delayInSeconds = 7;
 dispatch_time_t popTime = dispatch_time(DISPATCH_TIME_NOW,
(int64_t)(delayInSeconds * NSEC_PER_SEC));
 dispatch_after(popTime, dispatch_get_main_queue(), ^(void){
 // [self addMessageFromRemoteNotification:dictionary2 updateUI:NO];
 });
 }

}
else
 {}
//-----

```

APNs iOS APN.

APNs.

AppDelegate

```

func application(application: UIApplication, didRegisterForRemoteNotificationsWithDeviceToken
deviceToken: NSData) {
 print("DEVICE TOKEN = \(deviceToken)")
}

func application(application: UIApplication, didFailToRegisterForRemoteNotificationsWithError
error: NSError) {
 print(error)
}

```

## Objective-C

```

- (void)application:(UIApplication *)application
didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken
{
 NSString * deviceTokenString = [[[deviceToken description]
 stringByReplacingOccurrencesOfString:@"<" withString:@""]
 stringByReplacingOccurrencesOfString:@">" withString:@""]
 stringByReplacingOccurrencesOfString:@" " withString:@""];

 NSLog(@"The generated device token string is : %@",deviceTokenString);
}

- (void)application:(UIApplication*)application
didFailToRegisterForRemoteNotificationsWithError:(NSError*)error
{
 NSLog(@"Failed to get token, error: %@", error.description);
}

```

```
func application(application: UIApplication, didRegisterForRemoteNotificationsWithDeviceToken
deviceToken: NSData) {
 print("DEVICE TOKEN = \(deviceToken)")
}
```

## Swift3

```
@objc(userNotificationCenter:willPresentNotification:withCompletionHandler:) @available(iOS
10.0, *)
func userNotificationCenter(_ center: UNUserNotificationCenter, willPresent notification:
UNNotification, withCompletionHandler completionHandler: @escaping
(UNNotificationPresentationOptions) -> Void)
{
 //To show notifications in foreground.
 print("Userinfo2 \(notification.request.content.userInfo)")
}
```

## Objective-C

```
- (void)application:(UIApplication *)application
didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken
{
 if(application.applicationState == UIApplicationStateInactive) {
 NSLog(@"Inactive - the user has tapped in the notification when app was closed or in
background");
 //do some tasks
 [self handelPushNotification:userInfo];
 }
 else if (application.applicationState == UIApplicationStateBackground) {
 NSLog(@"application Background - notification has arrived when app was in background");
 [self handelPushNotification:userInfo];
 }
 else {
 NSLog(@"application Active - notication has arrived while app was opened");
 //Show an in-app banner
 //do tasks
 }
}
```

```
func application(application: UIApplication, didFailToRegisterForRemoteNotificationsWithError
error: NSError) {
 print(error)
}
```

## Objective-C

```
- (void)application:(UIApplication*)application
didFailToRegisterForRemoteNotificationsWithError:(NSError*)error
```

## APNInternet。

```
let isPushEnabled = UIApplication.sharedApplication().isRegisteredForRemoteNotifications()
```

AppDelegate.swift。

```
let application = UIApplication.sharedApplication()
let settings = UIUserNotificationSettings(forTypes: [.Alert, .Badge, .Sound], categories: nil)
application.registerUserNotificationSettings(settings)
```

didRegisterUserNotificationSettings

```
func application(application: UIApplication, didRegisterUserNotificationSettings
notificationSettings: UIUserNotificationSettings) {
 application.registerForRemoteNotifications()
}
```

。

```
func application(application: UIApplication, didRegisterForRemoteNotificationsWithDeviceToken
deviceToken: NSData) {
 let tokenChars = UnsafePointer<CChar>(deviceToken.bytes)
 var tokenString = ""

 for i in 0..
```

。 InternetAPNS Sandbox。 APNSAPI。 [Apple](#)。

。 JSON

```
func application(application: UIApplication, didReceiveRemoteNotification userInfo: [NSObject
: AnyObject]) {
 print("Received notification: \(userInfo)")
}
```

## C.

```
- (void)application:(UIApplication *)application didReceiveRemoteNotification: (NSDictionary
*)userInfo
{
 NSLog(@"Received notification: %@", userInfo);
}
```

## iOS 10

```
#define SYSTEM_VERSION_GREATER_THAN_OR_EQUAL_TO(v) ([[UIDevice currentDevice] systemVersion]
compare:v options:NSNumericSearch] != NSOrderedAscending)

-(void) application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary
*)userInfo fetchCompletionHandler:(void
(^)(UIBackgroundFetchResult))completionHandler
{
 // iOS 10 will handle notifications through other methods
 NSLog(@"Received notification: %@", userInfo);

 if(SYSTEM_VERSION_GREATER_THAN_OR_EQUAL_TO(@"10.0"))
 {
 NSLog(@"iOS version >= 10. Let NotificationCenter handle this one.");
 // set a member variable to tell the new delegate that this is background
 return;
 }
 NSLog(@"HANDLE PUSH, didReceiveRemoteNotification: %@", userInfo);

 // custom code to handle notification content

 if([UIApplication sharedApplication].applicationState == UIApplicationStateInactive)
 {
 NSLog(@"INACTIVE");
 completionHandler(UIBackgroundFetchResultNewData);
 }
 else if([UIApplication sharedApplication].applicationState == UIApplicationStateBackground
)
 {
 NSLog(@"BACKGROUND");
 completionHandler(UIBackgroundFetchResultNewData);
 }
 else
 {
 NSLog(@"FOREGROUND");
 completionHandler(UIBackgroundFetchResultNewData);
 }
}

-(void)userNotificationCenter:(UNUserNotificationCenter *)center
willPresentNotification:(UNNotification *)notification
withCompletionHandler:(void (^)(UNNotificationPresentationOptions
options))completionHandler
{
 NSLog(@"Handle push from foreground");
 // custom code to handle push while app is in the foreground
 NSLog(@"%@", notification.request.content.userInfo);
}

-(void)userNotificationCenter:(UNUserNotificationCenter *)center
didReceiveNotificationResponse:(UNNotificationResponse *)response
withCompletionHandler:(void (^)(void))completionHandler
{
 NSLog(@"Handle push from background or closed");
 // if you set a member variable in didReceiveRemoteNotification, you will know if this is
from closed or background
 NSLog(@"%@", response.notification.request.content.userInfo);
}
```

## App ID

---

- Apple
- appApp IDcom.example.MyApp
- [developer.apple.com](https://developer.apple.com)
- iOS

## AppleApp IDAPN

1-developer.apple.com

Account

2-“”

3-“App ID”

ID Identifiers

■ App IDs

4-“+”



5-ID

6-ID

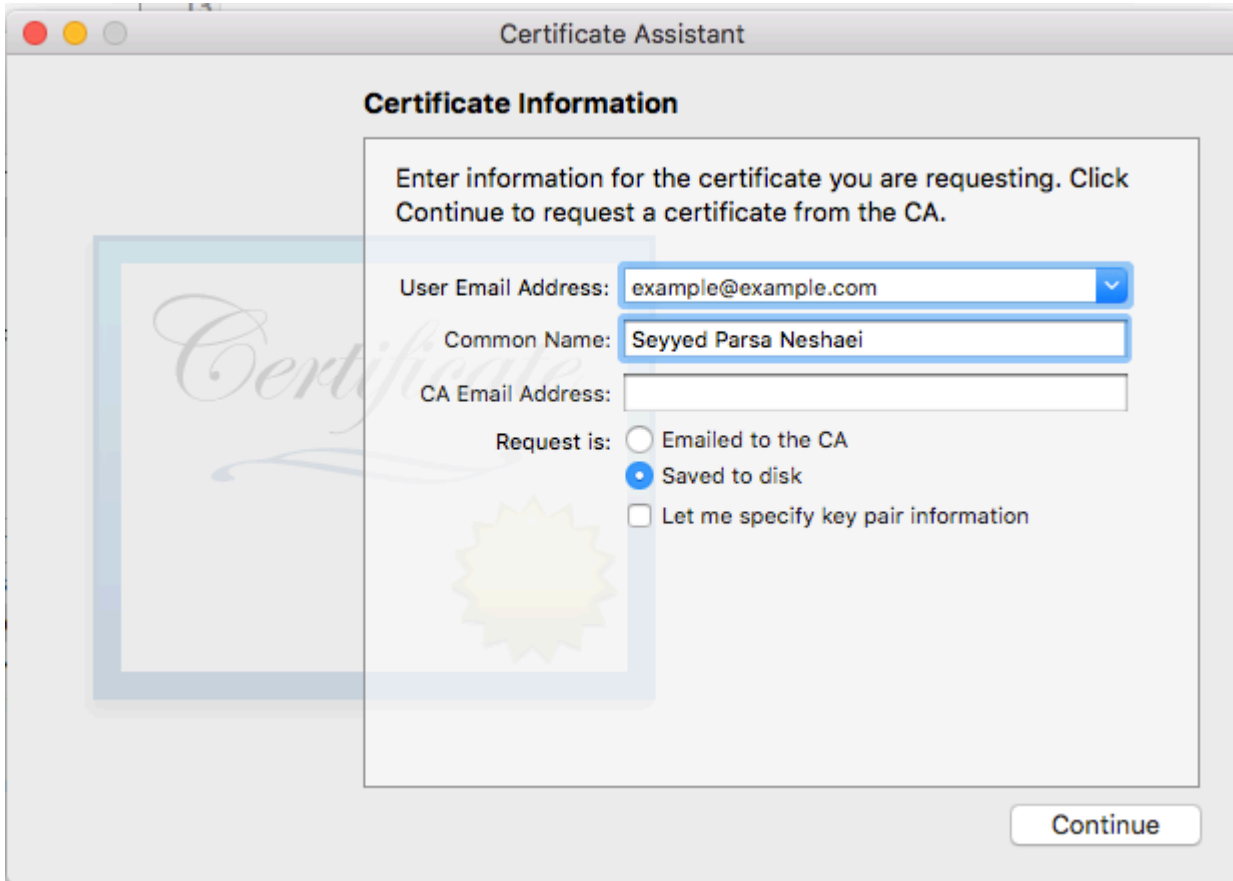
7-“”“”

8-MacKeychain Access

9-Keychain AccessCertificate Assistant - >

10-

11-



12-CA

13-CA.

14-""

15-AppleKeychain Access

---

## XcodeAPN

1-

2-

3-

4 -

## Objective-C

```
[[UIApplication sharedApplication] unregisterForRemoteNotifications];
```

```
UIApplication.sharedApplication().unregisterForRemoteNotifications()
```

- 
- ◦ ◦

someNumber

## Objective-C

```
[UIApplication sharedApplication].applicationIconBadgeNumber = someNumber;
```

```
UIApplication.shared.applicationIconBadgeNumber = someNumber
```

someNumber = 0 ◦

## ◦ PHP◦

1. send\_push.php
- 2.
3. dev\_pathprod\_path“APNS”
4. cd'php send\_push'
- 5.

```
<?php

// Put your device token here (without spaces):
$deviceToken = '20128697f872d7d39e48c4a61f50cb11d77789b39e6fc6b4cd7ec80582ed5229';
// Put your final pem cert name here. it is supposed to be in the same folder as this script
$cert_name = 'final_cert.pem';
// Put your private key's passphrase here:
$passphrase = '1234';

// sample point
$alert = 'Hello world!';
$event = 'new_incoming_message';

// You can choose either of the paths, depending on what kind of certificate you are using
$dev_path = 'ssl://gateway.sandbox.push.apple.com:2195';
$prod_path = 'ssl://gateway.push.apple.com:2195';

////////////////////////////////////

$ctx = stream_context_create();
stream_context_set_option($ctx, 'ssl', 'local_cert', $cert_name);
stream_context_set_option($ctx, 'ssl', 'passphrase', $passphrase);

// Open a connection to the APNS server
$fp = stream_socket_client(
 $dev_path, $err,
 $errstr, 60, STREAM_CLIENT_CONNECT|STREAM_CLIENT_PERSISTENT, $ctx);

if (!$fp)
 exit("Failed to connect: $err $errstr" . PHP_EOL);

echo 'Connected to APNS' . PHP_EOL;

// Create the payload body
// it should be as short as possible
```

```

// if the notification doesnt get delivered that is most likely
// because the generated message is too long
$body['aps'] = array(
 'alert' => $alert,
 'sound' => 'default',
 'event' => $event
);

// Encode the payload as JSON
$payload = json_encode($body);

// Build the binary notification
$msg = chr(0) . pack('n', 32) . pack('H*', $deviceToken) . pack('n', strlen($payload)) .
$payload;

// Send it to the server
$result = fwrite($fp, $msg, strlen($msg));

if (!$result)
 echo 'Message not delivered' . PHP_EOL;
else
 echo 'Message successfully delivered' . PHP_EOL;

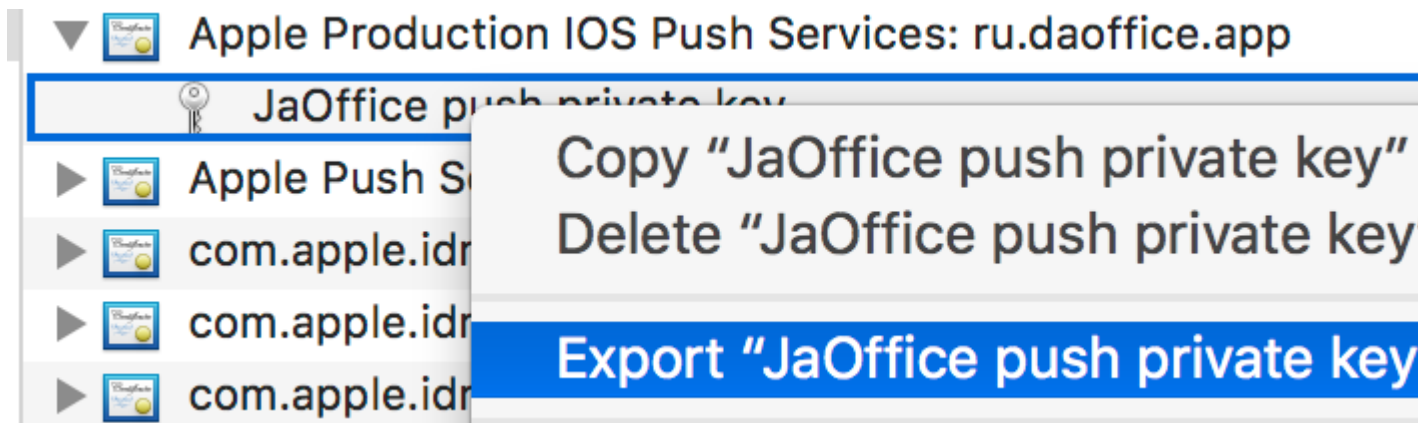
// Close the connection to the server
fclose($fp);

```

## .cer.pem

1. aps.cer

2. “Keychain access”.p12key.p12。 “”。 1。 。 。



3. cd

4. .cer.pem

```
openssl x509 -in aps.cer -inform der -out aps.pem
```

5. .pem。 2。 。 。

```
openssl pkcs12 -nocerts -out key.pem -in key.p12
```

6.



```
cat key.pem aps.pem > final_cert.pem
```

7. final\_cert.pem。 5。

<https://riptutorial.com/zh-TW/ios/topic/3492/>

---

# 149: - iOS 10.

iOS 10 `UserNotifications.framework /API` .

- 50 MB/ GIF /.

## Examples

- 
- ◦

1. **xCode** NavigatorTargets ◦ Add New Target ◦
2. Notification Content Extension

Choose a template for your new target:

ios watchOS tvOS macOS Cross-Platform

Call Directory Extension

Content Blocker Extension

iMessage

Intents Extension

Notification Service Extension

Photo Editing Extension

Spotlight Index

Sticker Pack

Cancel

## 5. category key Info.plist3

```
{
 aps: {
 alert: { ... },
 category: 'io.swifting.notification-category'
 }
}
```

```
let mutableNotificationContent = UNMutableNotificationContent()
mutableNotificationContent.category = "io.swifting.notification-category"
mutableNotificationContent.title = "Swifting.io Notifications"
mutableNotificationContent.subtitle = "Swifting.io presents"
mutableNotificationContent.body = "Custom notifications"
```

**API** <https://developer.apple.com/reference/usernotificationsui/unnotificationcontentextension>  
`utm_source = swing.ioutm_medium = webutm_campaign = blog20post`

- iOS 10: <https://riptutorial.com/zh-TW/ios/topic/9501/----ios-10->

# 150:

◦ ◦ ◦

## Examples

```
//Swift
let storyboard = UIStoryboard(name: "Main", bundle: NSBundle.mainBundle())

//Objective-c
UINavigationController *storyboard = [UINavigationController storyboardWithName:@"Main" bundle:[NSBundle mainBundle]];
```

## ViewController

```
//Swift
let mainScreen = storyboard.instantiateInitialViewController()

//Objective-c
UIViewController *initialScreen = [storyboard instantiateInitialViewController];
```

## ViewController

```
//Swift
let viewController = storyboard.instantiateViewControllerWithIdentifier("identifier")

//Objective-c
UIViewController *viewController = [storyboard instantiateViewControllerWithIdentifier:@"identifier"];
```

<https://riptutorial.com/zh-TW/ios/topic/3514/>

# 151:

## Examples

### 1024NSMutableData socket

```
// MARK: - Get file data as chunks Methode.
func getFileDataInChunks() {

 let documentDirectoryPath = NSSearchPathForDirectoriesInDomains(.documentDirectory,
 .userDomainMask, true)[0] as NSString
 let filePath = documentDirectoryPath.appendingPathComponent("video.mp4")

 //Check file exists at path or not.
 if FileManager.default.fileExists(atPath: filePath) {

 let chunkSize = 1024 // divide data into 1 kb

 //Create NSMutableData object to save read data.
 let ReadData = NSMutableData()

 do {

 //open file for reading.
 outputFileHandle = try FileHandle(forReadingFrom: URL(fileURLWithPath: filePath))

 // get the first chunk
 var datas = outputFileHandle?.readData(ofLength: chunkSize)

 //check next chunk is empty or not.
 while !(datas?.isEmpty)! {

 //here I write chunk data to ReadData or you can directly write to socket.
 ReadData.append(datas!)

 // get the next chunk
 datas = outputFileHandle?.readData(ofLength: chunkSize)

 print("Running: \(ReadData.length)")
 }

 //close outputFileHandle after reading data complete.
 outputFileHandle?.closeFile()

 print("File reading complete")

 }catch let error as NSError {
 print("Error : \(error.localizedDescription)")
 }
 }
}
```

ReadData>DataoutputFileHandleFileHandle

```
var outputFileHandle:FileHandle?
```

<https://riptutorial.com/zh-TW/ios/topic/10665/>

# 152:

## Examples

### UINavigationController

1. info.plist view controller-based status bar appearance YES
2. UINavigationController

### Objective-C

```
- (UIStatusBarStyle)preferredStatusBarStyle
{
 return UIStatusBarStyleLightContent;
}
```

### Swift

```
override func preferredStatusBarStyle() -> UIStatusBarStyle {
 return UIStatusBarStyle.LightContent
}
```

### UINavigationController

### UINavigationController

### Objective-C

```
- (UIStatusBarStyle)preferredStatusBarStyle
{
 return UIStatusBarStyleLightContent;
}
```

### Swift

```
override func preferredStatusBarStyle() -> UIStatusBarStyle {
 return .lightContent
}
```

UINavigationController

### C

```
// e.g. in your view controller's viewDidLoad method:
self.navigationController.navigationBar.barStyle = UIBarStyleBlack; // this will give you a
white status bar
```

```
// e.g. in your view controller's viewDidLoad method:
```



```
navigationController?.navigationBar.barStyle = .black // this will give you a white status bar
```

UIBarStyledefault black blackOpaque blackTranslucent ◦ **3**◦

◦

## ViewController

### AwesomeViewController

```
let awesomeViewController = AwesomeViewController()
awesomeViewController.navigationBar.barStyle = .blackTranslucent // or other style
```

## ViewController

UIViewControllerContainment◦

```
class RootViewController: UIViewController {

 private let messageBarViewController = MessageBarViewController()

 override func childViewControllerForStatusBarStyle() -> UIViewController? {
 return messageBarViewController
 }

 override func viewDidLoad() {
 super.viewDidLoad()

 //add child vc code here...

 setNeedsStatusBarAppearanceUpdate()
 }
}

class MessageBarViewController: UIViewController {

 override func preferredStatusBarStyle() -> UIStatusBarStyle {
 return .Default
 }
}
```

# 1

## Info.plist

```
View controller-based status bar appearance
```

NO

| Key                                    | Type       | Value      |
|----------------------------------------|------------|------------|
| ▼ Information Property List            | Dictionary | (15 items) |
| View controller-based status bar style | Boolean    | NO         |

## 2

### AppDelegate.swift didFinishLaunchingWithOptions

```
UIApplication.shared.statusBarStyle = .lightContent
```

```
UIApplication.shared.statusBarStyle = .default
```

- `.lightContentstatusBar`.
- `.defaultstatusBar`.

---

## Objective-C

### SWIFT。 AppDelegate.m

```
[[UIApplication sharedApplication] setStatusBarStyle:UIStatusBarStyleLightContent];
```

```
[[UIApplication sharedApplication] setStatusBarStyle:UIStatusBarStyleDefault];
```

<https://riptutorial.com/zh-TW/ios/topic/378/>

# 153:

iOS◦ ◦ iOS◦

## Examples

### iOS

Localizable.strings◦ ◦

```
"str" = "str-language";
```

### Objective-Cstr

```
//Try to provide description on the localized string to be able to create a proper
documentation if needed
NSString *str = NSLocalizedString(@"string", @"description of the string");
```

### Swiftstr

```
let str = NSLocalizedString("string", comment: "language");
```

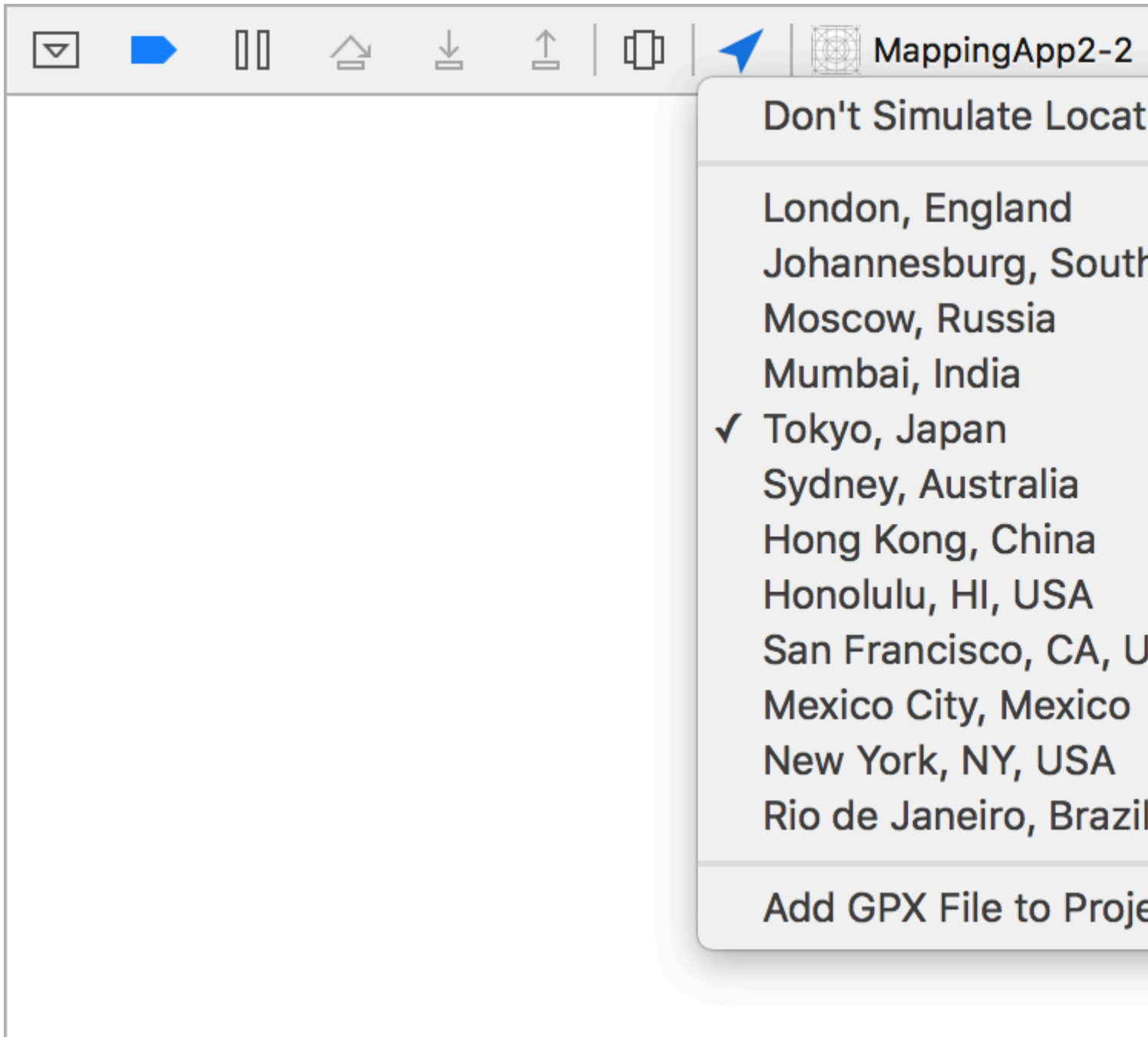
<https://riptutorial.com/zh-TW/ios/topic/1579/>

---

## 154:

1. desiredAccuracy
2. distanceFilter
3. requestLocation
4. startUpdatingLocation
5. allowDeferredLocationUpdatesuntilTraveled:)
6. startMonitoringSignificantLocationChanges
7. allowDeferredLocationUpdatesuntilTraveled:)
8. authorizedAlways
9. authorizedWhenInUse
10. locationManager\_didChangeAuthorization :)

- 
1. Xcode。
  2. ""。
  3. 。



## Examples

CoreLocation



MappingApp



General

Cap

**PROJECT**



MappingApp2-2

**TARGETS**



MappingApp2-2

Choose frameworks and libraries

CoreL

▼ iOS 9.2



CoreLocation.framework



Developer Frameworks

Add Other...



- GPX

```
<?xml version="1.0"?>
<gpx version="1.1" creator="Xcode">
<!--
 Provide one or more waypoints containing a latitude/longitude pair. If you provide one
 waypoint, Xcode will simulate that specific location. If you provide multiple
 waypoints,
 Xcode will simulate a route visitng each waypoint.
-->
<wpt lat="52.599878" lon="4.702029">
 <name>location name (eg. Florida)</name>
</wpt>
```

- -> Scheme -> Edit SchemeRUNGPX.

Background Modes Location updates ◦

### Info.plist◦

```
<key>NSLocationAlwaysUsageDescription</key>
<string>I want to get your location Information in background</string>

<key>UIBackgroundModes</key>
<array>
 <string>location</string>
</array>
```

### CLLocationManager

#### C.

```
//The Location Manager must have a strong reference to it.
_locationManager = [[CLLocationManager alloc] init];
_locationManager.delegate = self;

//Request Always authorization (iOS8+)
if ([_locationManager respondsToSelector:@selector(requestAlwaysAuthorization)]) {
 [_locationManager requestAlwaysAuthorization];
}

//Allow location updates in the background (iOS9+)
if ([_locationManager respondsToSelector:@selector(allowsBackgroundLocationUpdates)]) {
 _locationManager.allowsBackgroundLocationUpdates = YES;
}

[_locationManager startUpdatingLocation];
```

```
self.locationManager.delegate = self

if #available (iOS 8.0,*) {
 self.locationManager.requestAlwaysAuthorization()
}

if #available (iOS 9.0,*) {
 self.locationManager.allowsBackgroundLocationUpdates = true
}
```

```
}

self.locationManager.startUpdatingLocation()
```

<https://riptutorial.com/zh-TW/ios/topic/2937/>



# 155:

## Examples

### Core Motion

```
import CoreMotion
```

```
CMAltimeterViewDidLoad() ◦ ◦ viewDidLoad() CMAltimeter
```

```
let altimeter = CMAltimeter()
```

1. `relativeAltitude` `CMAltimeter.isRelativeAltitudeAvailable` ◦
2. `true` `startRelativeAltitudeUpdatesToQueue`
3. `relativeAltitude` **pressure** ◦

◦

```
@IBAction func start(sender: AnyObject){
if CMAltimeter.isRelativeAltitudeAvailable() {
 // 2
 altimeter.startRelativeAltitudeUpdatesToQueue(NSOperationQueue.mainQueue(), withHandler: {
data, error in
 // 3
 if (error == nil) {
 println("Relative Altitude: \(data.relativeAltitude)")
 println("Pressure: \(data.pressure)")
 }
 })
}
```

<https://riptutorial.com/zh-TW/ios/topic/7636/>

# 156:

## Examples

### Core Graphics

```
UIGraphicsBeginImageContextWithOptions() C° UIGraphicsEndImageContext()
```

```
let size = CGSize(width: 256, height: 256)

UIGraphicsBeginImageContextWithOptions(size, false, 0)

let context = UIGraphicsGetCurrentContext()

// drawing code here

UIGraphicsEndImageContext()
```

## Objective-C

```
CGSize size = [CGSize width:256 height:256];

UIGraphicsBeginImageContextWithOptions(size, NO, 0);

CGContext *context = UIGraphicsGetCurrentContext();

// drawing code here

UIGraphicsEndImageContext();
```

3 UIGraphicsBeginImageContextWithOptions()

1. CGSize
2. true
3. 123° 0°

```
let image = UIGraphicsGetImageFromCurrentImageContext()
imageView.image = image //assuming imageView is a valid UIImageView object
```

## Objective-C

```
UIImage *image = UIGraphicsGetImageFromCurrentImageContext();
imageView.image = image; //assuming imageView is a valid UIImageView object
```

<https://riptutorial.com/zh-TW/ios/topic/5530/>

# 157:

- - - iOS SDK◦

Core DataAPI◦ ◦ ◦ ◦

## Examples

```
NSManagedObjectContext *context = ((AppDelegate*)[[UIApplication sharedApplication] delegate]).persistentContainer.viewContext;
```

```
NSFetchRequest<EntityName *> *fetchRequest = [EntityName fetchRequest];
NSError *error ;
NSArray *resultArray= [context executeFetchRequest:fetchRequest error:&error];
```

```
NSFetchRequest<EntityName *> *fetchRequest = [EntityName fetchRequest];
NSSortDescriptor *sortDescriptor = [NSSortDescriptor sortDescriptorWithKey:@"someKey"
ascending:YES];
fetchRequest.sortDescriptors = @[sortDescriptor];
NSError *error ;
NSArray *resultArray= [context executeFetchRequest:fetchRequest error:&error];
```

```
NSManagedObject *entityNameObj = [NSEntityDescription
insertNewObjectForEntityForName:@"EntityName" inManagedObjectContext:context];
[entityNameObj setValue:@"someValue" forKey:@"someKey"];
```

```
[((AppDelegate*)[[UIApplication sharedApplication] delegate]) saveContext];
```

<https://riptutorial.com/zh-TW/ios/topic/9489/>

---

# 158:

iOSwatchOStvOS。 ◦

- 
- iOS
  - watchOS
  - tvOS
  -

macOSXcodemacOS。

---

->



Xcode

File

Edit

View

Find

Navigate



- ▶ Swift
- ▶ Objective-C
- ▶ JavaScript

# Abc

Impo  
chang

Simulat  
Simulat  
of the s

Simulat  
simulat  
These s

---

# 159:

/ Library / Developer / CoreSimulator / Devices /

////

iOS。

iPhonei386iPadx8

## Examples

```
xcrun simctl install booted *.app
```

<https://riptutorial.com/zh-TW/ios/topic/9813/>

---

# 160: iOS

## Examples

### iOS 8

#### 3

```
let minimumVersion = OperatingSystemVersion(majorVersion: 8, minorVersion: 1, patchVersion: 2)
if ProcessInfo().isOperatingSystemAtLeast(minimumVersion) {
 //current version is >= (8.1.2)
} else {
 //current version is < (8.1.2)
}
```

```
let minimumVersionString = "3.1.3"
let versionComparison = UIDevice.current.systemVersion.compare(minimumVersionString, options:
.numeric)
switch versionComparison {
 case .orderedSame, .orderedDescending:
 //current version is >= (3.1.3)
 break
 case .orderedAscending:
 //current version is < (3.1.3)
 fallthrough
 default:
 break;
}
```

---

## Objective-C

```
NSString *version = @"3.1.3";
NSString *currentVersion = @"3.1.1";
NSComparisonResult result = [currentVersion compare:version options:NSNumericSearch];
switch(result){
 case: NSOrderedAscending:
 //less than the current version
 break;
 case: NSOrderedDescending:
 case: NSOrderedSame:
 // equal or greater than the current version
 break;
}
```

### Swift 2.0

```
if #available(iOS 9, *) {
 // iOS 9
} else {
 // iOS 8 or earlier
}
```



```
}
```

## iOS

◦

---

# Objective-C

```
NSString *version = [[UIDevice currentDevice] systemVersion]
```

---

```
let version = UIDevice.currentDevice().systemVersion
```

---

## 3

```
let version = UIDevice.current.systemVersion
```

iOS <https://riptutorial.com/zh-TW/ios/topic/2194/ios>

# 161:

Apple [Reachability.h](#) [Reachability.m](#)

Apple [API iOS](#)

Github / [Cocoapod](#)

Apple [Reachability](#) / [SCNetworkReachability](#)

## Examples

### Reachability

Apple [Reachability](#)

```
Reachability *internetReachability = [Reachability reachabilityForInternetConnection];
[internetReachability startNotifier];
```

Reachability [NSNotification](#)

```
[[NSNotificationCenter defaultCenter] addObserver:self
 selector:@selector(reachabilityChanged:) name:kReachabilityChangedNotification object:nil];
```

```
- (void) reachabilityChanged:(NSNotification *)note {
 //code which reacts to network changes
}
```

```
- (void)reachabilityChanged:(NSNotification *)note {
 Reachability* reachability = [note object];
 NetworkStatus netStatus = [reachability currentReachabilityStatus];

 if (netStatus == NotReachable) {
 NSLog(@"Network unavailable");
 }
}
```

### WIFI

```
- (void)reachabilityChanged:(NSNotification *)note {
 Reachability* reachability = [note object];
 NetworkStatus netStatus = [reachability currentReachabilityStatus];

 switch (netStatus) {
 case NotReachable:
 NSLog(@"Network unavailable");
 break;
 case ReachableViaWWAN:
```

```

 NSLog(@"Network is cellular");
 break;
 case ReachableViaWiFi:
 NSLog(@"Network is WIFI");
 break;
 }
}

```

```

import SystemConfiguration

/// Class helps to code reuse in handling internet network connections.
class NetworkHelper {

 /**
 Verify if the device is connected to internet network.
 - returns: true if is connected to any internet network, false if is not
 connected to any internet network.
 */
 class func isConnectedToNetwork() -> Bool {
 var zeroAddress = sockaddr_in()

 zeroAddress.sin_len = UInt8(sizeofValue(zeroAddress))
 zeroAddress.sin_family = sa_family_t(AF_INET)

 let defaultRouteReachability = withUnsafePointer(&zeroAddress) {
 SCNetworkReachabilityCreateWithAddress(nil, UnsafePointer($0))
 }

 var flags = SCNetworkReachabilityFlags()

 if !SCNetworkReachabilityGetFlags(defaultRouteReachability!, &flags) {
 return false
 }

 let isReachable = (flags.rawValue & UInt32(kSCNetworkFlagsReachable)) != 0
 let needsConnection = (flags.rawValue & UInt32(kSCNetworkFlagsConnectionRequired)) != 0

 return (isReachable && !needsConnection)
 }
}

if NetworkHelper.isConnectedToNetwork() {
 // Is connected to network
}

```

## Objective-C

```

-(BOOL)isConnectedToNetwork
{
 Reachability *networkReachability = [Reachability reachabilityForInternetConnection];
 NetworkStatus networkStatus = [networkReachability currentReachabilityStatus];
 if (networkStatus == NotReachable)
 {
 NSLog(@"There IS NO internet connection");
 return false;
 } else
 {

```

```
 NSLog(@"There IS internet connection");
 return true;
 }
}
```

<https://riptutorial.com/zh-TW/ios/topic/704/>

---

# 162: SWIFT BRIDGING HEADER

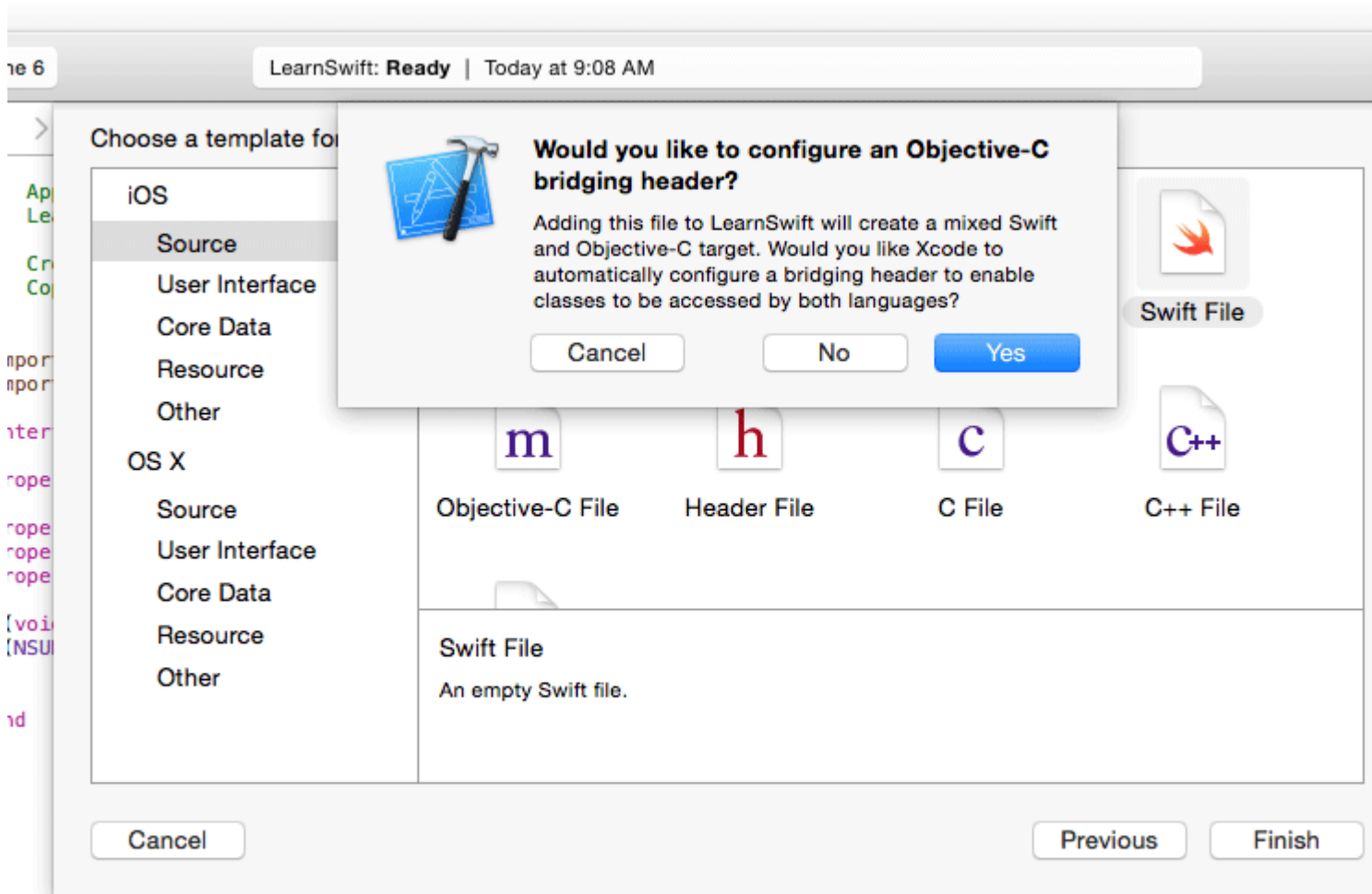
## Examples

### Swift

- Xcode>>“”。
- “YourProjectName-Bridging-Header.h”。 app Station“Station-Bridging-Header”。
- 。
- “Swift - ”。 “Swift Compiler”。 “Swift - ”Swift。 Swift。
- “Objective-C Bridging Header”/。 。“ProjectName / ProjectName-Bridging-Header.h”  
“ProjectName-Bridging-Header.h”。
- #importObjective-C。 swift。

### Xcode

SwiftXcode。 。



SWIFT BRIDGING HEADER <https://riptutorial.com/zh-TW/ios/topic/10851/swift-bridging-header>

# 163:

iOSVoiceOveriOS。 iOS。

## Examples

UIViewVoiceOver。

```
myView.isAccessibilityElement = YES;
```

。 Apple“ ”。

VoiceOverVoiceOver。

```
myElement.accessibilityFrame = frameInScreenCoordinates;
```

accessibilityFrame**rect**。 。 iOS 10API accessibilityFrameInContainerSpace 。

VoiceOver。 。 VoiceOver。 VoiceOver。 VoiceOver

```
UIAccessibilityPostNotification(UIAccessibilityScreenChangedNotification, firstElement);
```

。 。 VoiceOver。 VoiceOver。 。

。 。 “” 。

```
UIAccessibilityPostNotification(UIAccessibilityLayoutChangedNotification, firstElement);
```

“”“” 。

```
UIAccessibilityPostNotification(UIAccessibilityAnnouncementNotification, @"The thing happened!");
```

VoiceOver。 “F”。 VoiceOver。 。 “” 。

VoiceOverVoiceOver。 shouldGroupAccessibiltyChildrenshouldGroupAccessibiltyChildren

```
myView.shouldGroupAccessibilityChildren = YES;
```

UIKit。

VoiceOveriOSUIKitUIAccessibilityProtocol。 UIViewCore GraphicsMetal。 iOS 8.0UIView

```
myInaccessibleContainerView.accessibilityElements = @[elements, that, should, be, accessible];
```

UIAccessibilityElementUIAccessibilityProtocol ◦ ◦ VoiceOver ◦ UIViewUIAccessibilityProtocol  
UIAccessibilityElementUIView ◦ VoiceOver ◦

◦ iOS ◦ accessibilityViewIsModalVoiceOver ◦ ◦

```
myModalView.accessibilityViewIsModal = YES;
```

VoiceOver ◦ VoiceOver ◦

UIKitUIViewUIAccessibilityProtocol ◦ UIViewVoiceOver ◦ ◦ ◦ VoiceOverUIKit ◦ VoiceOver ◦ UIKit

```
myViewFullofButtons.hidden = YES;
```

```
myViewFullofButtons.accessibilityElementsHidden = YES;
```

◦ ◦ VoiceOver ◦ accessibilityElementsHidden ◦

<https://riptutorial.com/zh-TW/ios/topic/773/>



---

# 164:

XcodeInstruments. CPU/. Xcode.

## Examples

Time Profiler ◦ Instruments. Xcode. Time Profiler -



# Time Profiler



Call Tree

Call Tre

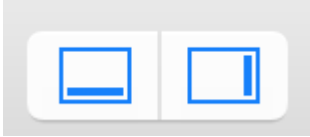
Running Time	Self	Symbol Name
5838.0ms 46.9%	0.0	▼Main Thread 0xa2db0
5234.0ms 42.0%	0.0	▶ ext.InstrumentsTutorial.ObjectiveC.CIContext
315.0ms 2.5%	0.0	▶ top_level_code InstrumentsTutorial
115.0ms 0.9%	0.0	▶ <Unknown Address>
63.0ms 0.5%	0.0	▶ InstrumentsTutorial.FlickrPhoto
15.0ms 0.1%	0.0	▶ @!objc ext.UIKit.ObjectiveC.CIContext
15.0ms 0.1%	0.0	▶ InstrumentsTutorial.ViewContro
12.0ms 0.0%	0.0	▶ InstrumentsTutorial.SearchResu
10.0ms 0.0%	0.0	▶ @!objc ObjectiveC.UImage.init
10.0ms 0.0%	0.0	▶ @!objc ObjectiveC.CIContext.____
8.0ms 0.0%	0.0	▶ InstrumentsTutorial.FlickrPhoto
8.0ms 0.0%	0.0	▶ InstrumentsTutorial.SearchResu
3.0ms 0.0%	0.0	▶ InstrumentsTutorial.SearchResu
3.0ms 0.0%	0.0	▶ InstrumentsTutorial.SearchResu
3.0ms 0.0%	0.0	▶ InstrumentsTutorial.ViewContro
3.0ms 0.0%	0.0	▶ InstrumentsTutorial.ViewContro
2.0ms 0.0%	0.0	▶ InstrumentsTutorial.Flickr.searc
2.0ms 0.0%	0.0	▶ InstrumentsTutorial.FlickrPhoto
1.0ms 0.0%	0.0	▶ swift_getEnumCaseSinglePaylo
1.0ms 0.0%	1.0	▶ Swift.HeapBufferStorage.__dea
1.0ms 0.0%	0.0	▶ swift_getExistentialTypeMetada
1.0ms 0.0%	0.0	▶ InstrumentsTutorial.SearchResu
1.0ms 0.0%	1.0	▶ _swift_retain_(swift::HeapObjec
1.0ms 0.0%	1.0	▶ swift_unknownRelease libswif
1.0ms 0.0%	0.0	▶ InstrumentsTutorial.SearchResu
1.0ms 0.0%	0.0	▶ swift_getGenericClassObjCNan
1.0ms 0.0%	1.0	▶ _swift_release_(swift::HeapObjec
1.0ms 0.0%	0.0	▶ InstrumentsTutorial.ViewContro

◦

Time Profiler“” Instruments ◦ ◦ - “ ” ◦ ◦

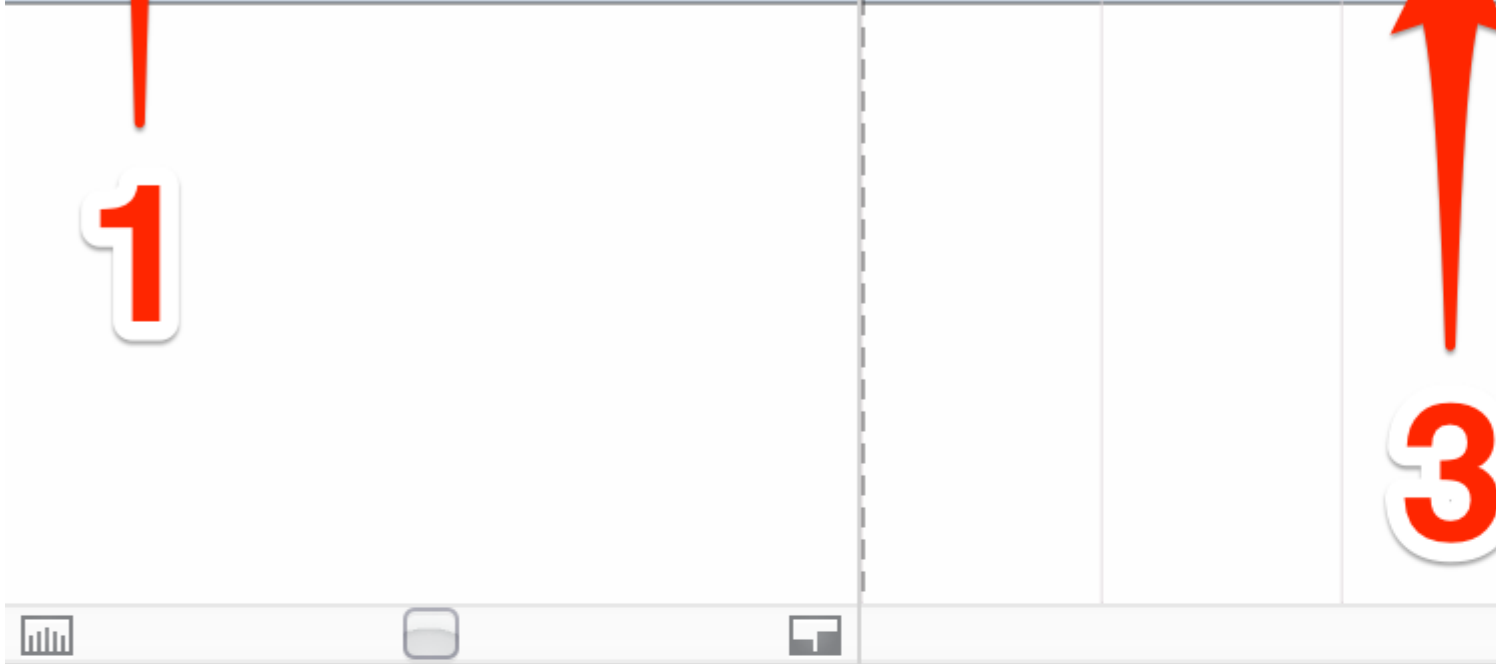
◦ ◦ -

## Instruments◦



◦

Time Profiler

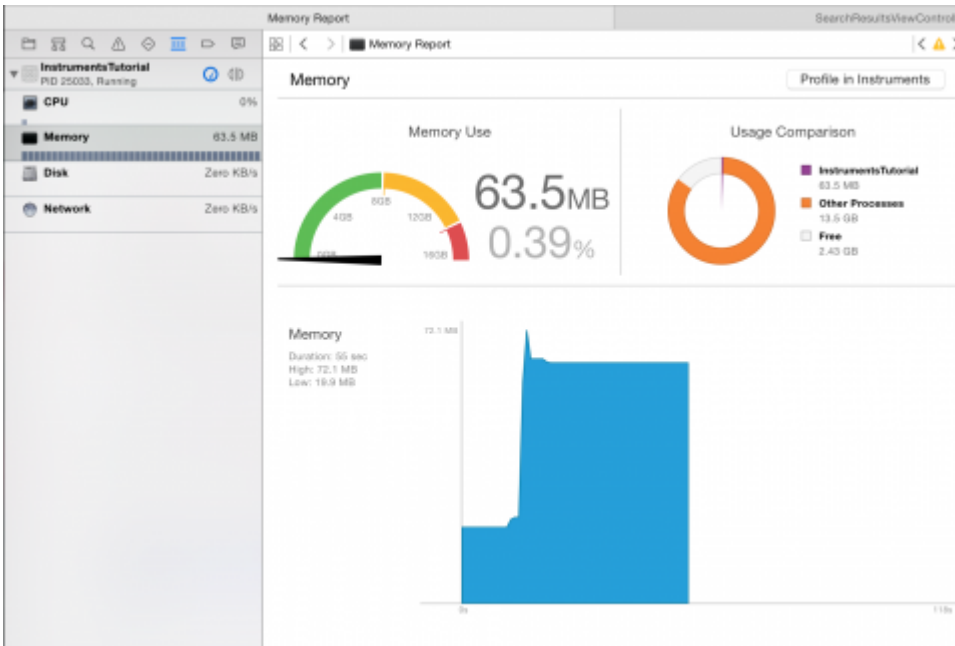


1

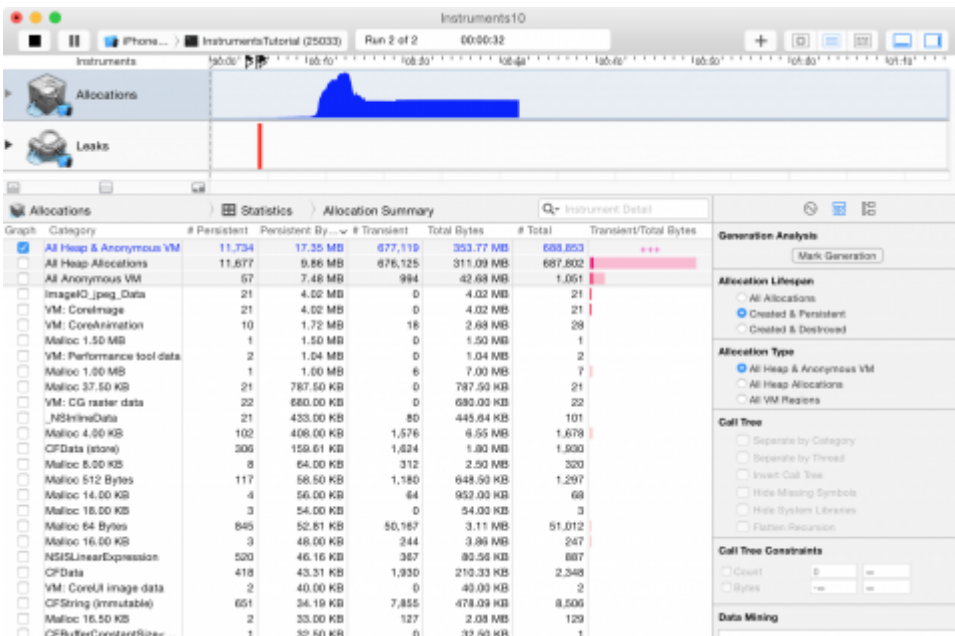
3

Running Time	Self	Symbol Name
4500.0ms 48.6%	0.0	▼ Main Thread 0xa4f45
3903.0ms 42.1%	0.0	▶ ext.InstrumentsTutorial.ObjectiveC.CIContext.
387.0ms 4.1%	0.0	▶ top_level_code InstrumentsTutorial
76.0ms 0.8%	0.0	▶ <Unknown Address>
39.0ms 0.4%	0.0	▶ InstrumentsTutorial.FlickrPhoto
16.0ms 0.1%	0.0	▶ @!objc ext.UIKit.ObjectiveC.CIContext.
10.0ms 0.1%	0.0	▶ InstrumentsTutorial.ViewContro
7.0ms 0.0%	0.0	▶ @!objc ObjectiveC.CIContext.
6.0ms 0.0%	0.0	▶ @!objc ObjectiveC.UILImage.init
5.0ms 0.0%	0.0	▶ InstrumentsTutorial.SearchResu
3.0ms 0.0%	0.0	▶ InstrumentsTutorial SearchRes

# Memory



◦ ◦ Profile in Instruments Profile in Instruments ◦ ◦



◦ ◦ ; LeaksObjective-C ◦ ◦

- ◦ Swift ARC retain cycle or strong reference cycle ◦ ◦

◦ system's memory ◦ iOS ◦

Allocations ◦ ◦

Allocations ◦ ◦ unbounded memory growth ◦

generation analysis ◦ Mark Generation. ""

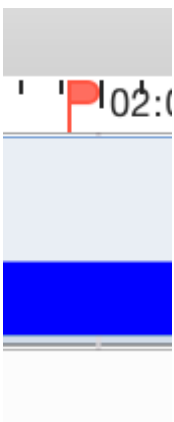
## Generation Analysis

Mark Generation

---

### Allocation Lifespan

- All Allocations
- Created & Persistent
- Created & Destroyed



generation analysis unbounded fashion

Snapshot	Timestamp	Growth	# Persistent
Generation A	00:41.300.205	3.15 MB	11,462
Generation B	01:02.001.909	32.43 MB	18,601
Generation C	01:15.020.335	11.44 MB	7,398
Generation D	01:27.413.265	10.35 MB	7,851
Generation E	01:39.070.377	11.71 MB	8,521
Generation F	01:50.576.231	28.91 MB	78,296

memory warnings? memory warnings? Memory warnings iOS

UIKit

memory warning Instrument\Simulate Memory Warning Hardware\Simulate Memory Warnings simulator's

◦ ◦

<https://riptutorial.com/zh-TW/ios/topic/9629/>

## 165: /

iOS 10.3 Apple/。 ApplestorekitSKStoreReviewController。 SKStoreReviewController  
requestReview。

App Store。

## Examples

/iOS

/。

SKStoreReviewController.requestReview

[/ https://riptutorial.com/zh-TW/ios/topic/9678/-](https://riptutorial.com/zh-TW/ios/topic/9678/)



---

# 166: MKMapView

## Examples

### MKMapView

```
let mapView = MKMapView(frame: CGRect(x: 0, y: 0, width: 320, height: 500))
```

mapView<sup>ViewController</sup>

### C.

```
self.map = [[MKMapView alloc] initWithFrame:CGRectMake(0, 0, self.view.frame.size.width,
self.view.frame.size.height)];
[self.view addSubview:self.map];
```

5 <sup>MKMapType</sup> MKMapView

### iPhone OS 3



○

◦

### 2

```
mapView.mapType = .Standard
```

### 3

```
mapView.mapType = .standard
```

## Objective-C

```
_mapView.mapType = MKMapTypeStandard;
```



iPhone OS 3

\_\_\_\_\_

○

○

2

```
mapView.mapType = .Satellite
```

3

```
mapView.mapType = .satellite
```

## Objective-C

```
_mapView.mapType = MKMapTypeSatellite;
```



iOS 9

---

**.satelliteFlyover**

◦

2

```
mapView.mapType = .SatelliteFlyover
```

3

```
mapView.mapType = .satelliteFlyover
```

## Objective-C

```
_mapView.mapType = MKMapTypeSatelliteFlyover;
```

iPhone OS 3

---

## .hybrid

◦

2

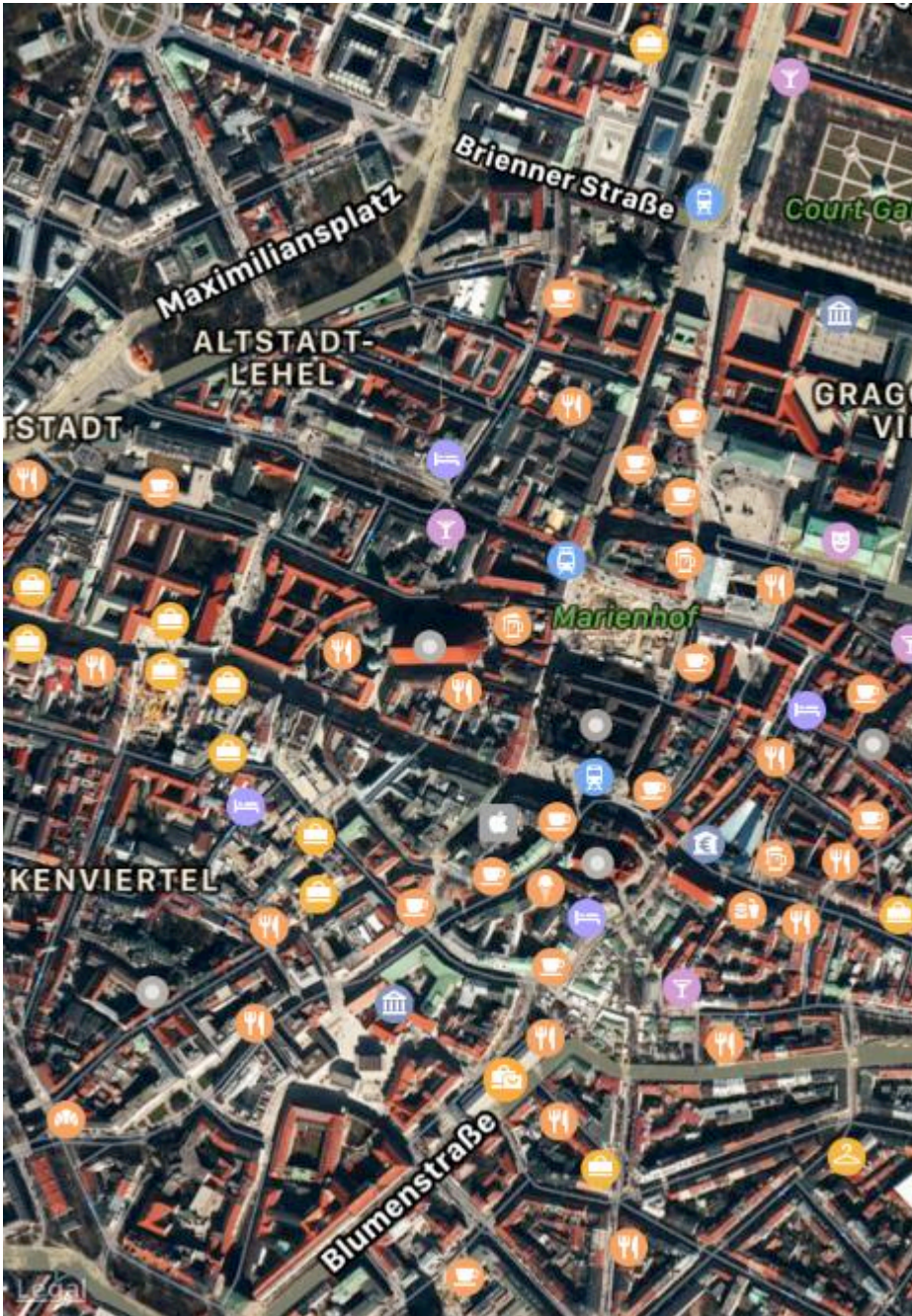
```
mapView.mapType = .Hybrid
```

3

```
mapView.mapType = .hybrid
```

## Objective-C

```
_mapView.mapType = MKMapTypeHybrid;
```



iOS 9

---

## .hybridFlyover

◦

2

```
mapView.mapType = .HybridFlyover
```

3

```
mapView.mapType = .hybridFlyover
```

## Objective-C

```
_mapView.mapType = MKMapTypeHybridFlyover;
```

/

2km。

```
MKUserLocation *userLocation = _mapView.userLocation;
MKCoordinateRegion region = MKCoordinateRegionMakeWithDistance
(userLocation.location.coordinate, 2000, 2000);
[_mapView setRegion:region animated:NO];
```

## MKLocalSearch

MKLocalSearch“”。

MKLocalSearchResponseMKMapItem。

```
MKLocalSearchRequest *request =
 [[MKLocalSearchRequest alloc] init]; //initialising search request
request.naturalLanguageQuery = @"Gym"; // adding query
request.region = _mapView.region; //setting region
MKLocalSearch *search =
 [[MKLocalSearch alloc] initWithRequest:request]; //initiate search

[search startWithCompletionHandler:^(MKLocalSearchResponse
 *response, NSError *error)
{
 if (response.mapItems.count == 0)
 NSLog(@"No Matches");
 else
 for (MKMapItem *item in response.mapItems)
 {
 NSLog(@"name = %@", item.name);
 NSLog(@"Phone = %@", item.phoneNumber);
 }
}];
```

## OpenStreetMap Tile-Overlay

Apple。

mapView [OpenStreetMap](#)。

```
self.mapViewViewController MKMapView。
```

```
ViewControllerMKMapViewDelegate。
```

```
class MyViewController: UIViewController, MKMapViewDelegate
```

```
ViewControllerMapView
```

```
mapView.delegate = self
```

◦ **URL** ◦ **URL** `http://tile.openstreetmap.org/{z}/{x}/{y}.png`

`http://tile.openstreetmap.org/{z}/{x}/{y}.png`

```
let urlTeplate = "http://tile.openstreetmap.org/{z}/{x}/{y}.png"
```

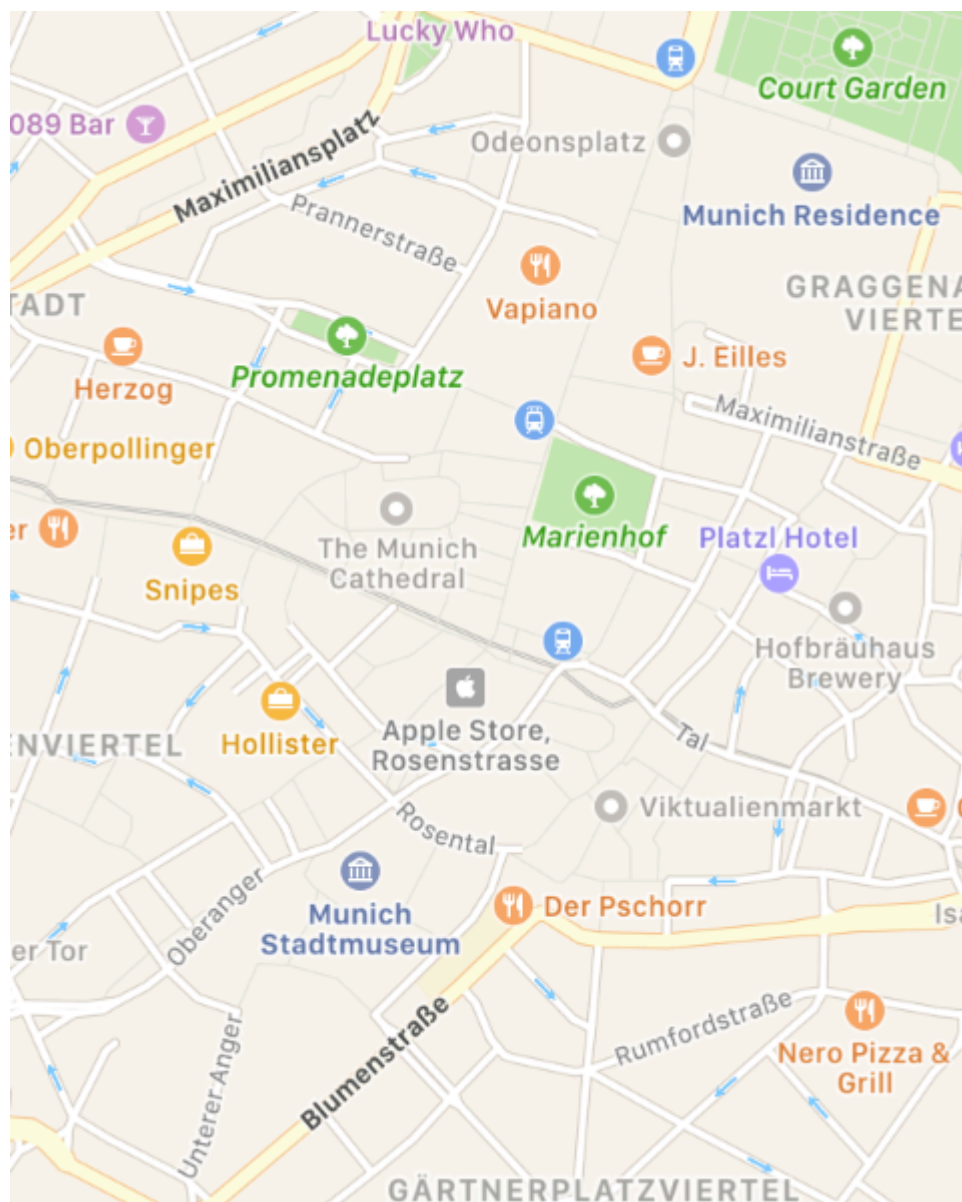
```
let overlay = MKTileOverlay(urlTemplate: urlTeplate)
```

```
overlay.canReplaceMapContent = true
```

mapView ◦

```
mapView.add(overlay, level: .aboveLabels)
```

`.aboveLabelslevel` ◦ ◦ `.aboveRoads` ◦



```
mapView ◦ ◦ func mapView(_ mapView: MKMapView, rendererFor overlay: MKOverlay) -> MKOverlayRenderer
```

```
func mapView(_ mapView: MKMapView, rendererFor overlay: MKOverlay) -> MKOverlayRenderer {
 if overlay is MKTileOverlay {
 let renderer = MKTileOverlayRenderer(overlay: overlay)
 return renderer
 } else {
 return MKTileOverlayRenderer()
 }
}
```

MKOverlayRenderermapView ◦



URL ◦ [OSM Wiki](#)◦ [tile-servers](#) ◦

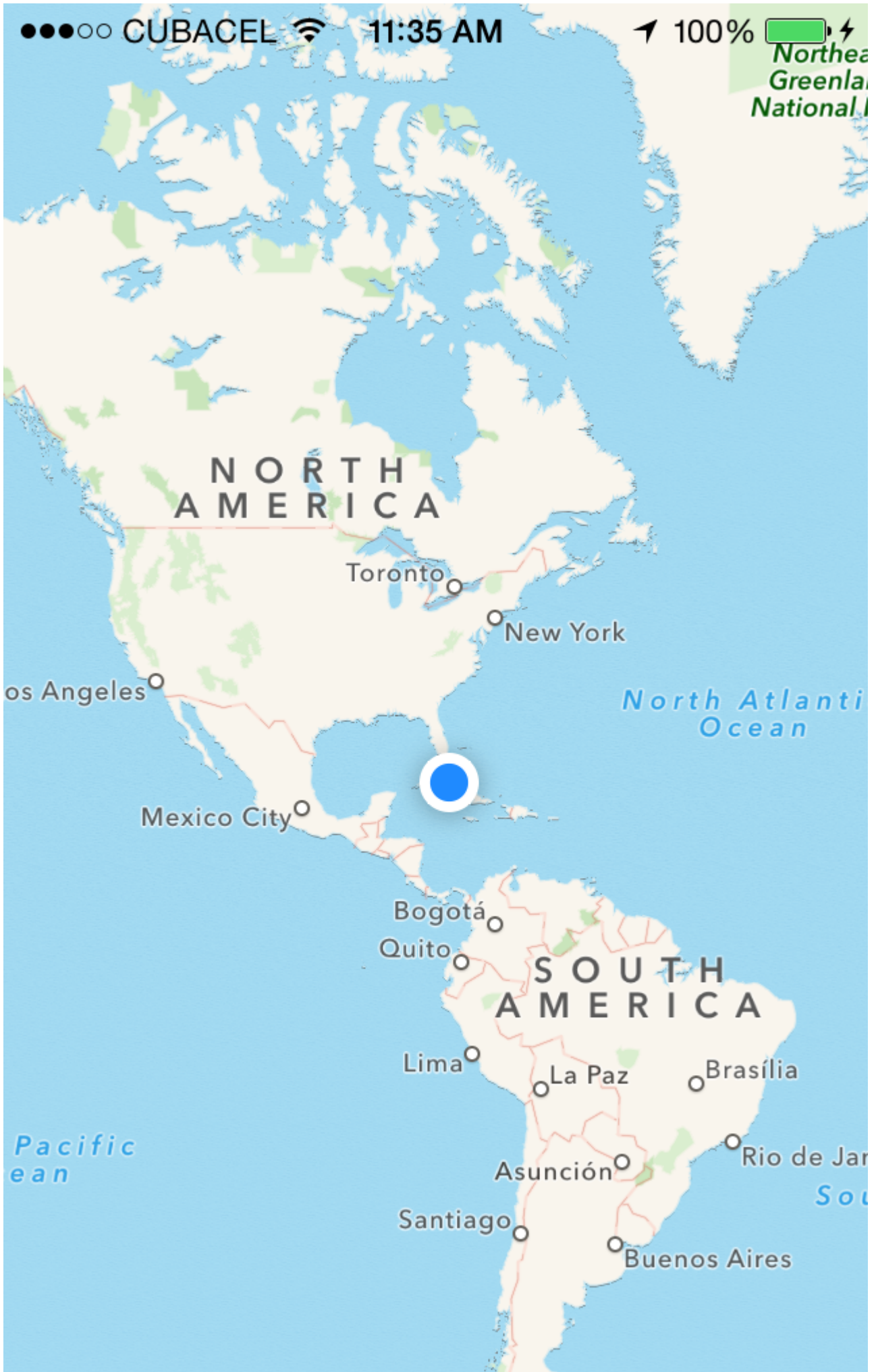
UserLocationUserTracking

## Objective-C

```
[self.map setShowsUserLocation:YES];
```



```
self.map?.showsUserLocation = true
```



---

# 167: NSData

---

[AppleNSData](#)

[NSData.dataWithContentsOfFile](#)

[NSData.bytes](#)

## Examples

### NSData

---

```
let data = NSData(contentsOfFile: filePath) //assuming filePath is a valid path
```

### Objective-C

```
NSData *data = [NSData dataWithContentsOfFile:filePath]; //assuming filePath is a valid path
```

---

## String

```
let data = (string as NSString).dataUsingEncoding(NSUTF8StringEncoding) //assuming string is a String object
```

### Objective-C

```
NSData *data = [string dataUsingEncoding:NSUTF8StringEncoding]; //assuming string is a String object
```

### NSData

---

```
let string = String(NSString(data: data, encoding: NSUTF8StringEncoding)) //assuming data is a valid NSData object
```

### Objective-C

```
NSString *string = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];
//assuming data is a valid NSData object
[string release];
```

---

```
let array = data.bytes as! NSMutableArray //assuming data is a valid NSData object
```

## Objective-C

```
NSMutableArray *array = (NSMutableArray *)[data bytes]; //assuming data is a valid NSData object
```

---

```
let byteArray = data.bytes as! UInt8 //assuming data is a valid NSData object
```

## Objective-C

```
UInt8 *byteArray = (UInt8 *)data.bytes; //assuming data is a valid NSData object
```

## NSDataHEX

NSDatadescription°

```
extension NSData {

 func hexString() -> String {
 return UnsafeBufferPointer<UInt8>(start: UnsafePointer<UInt8>(bytes), count: length)
 .reduce("") { $0 + String(format: "%02x", $1) }
 }

}
```

## Objective-C

```
@implementation NSData (HexRepresentation)

- (NSString *)hexString {
 const unsigned char *bytes = (const unsigned char *)self.bytes;
 NSMutableString *hex = [NSMutableString new];
 for (NSUInteger i = 0; i < self.length; i++) {
 [hex appendFormat:@"%02x", bytes[i]];
 }
 return [hex copy];
}

@end
```

[NSData](https://riptutorial.com/zh-TW/ios/topic/5084/nsdata) <https://riptutorial.com/zh-TW/ios/topic/5084/nsdata>

# 168: NSDate

- NSDate// NSDateinit
- NSDate。 TimeIntervalSince1970 //19701100:00:00 UTC。
- NSDate。 compareNSDate//NSComparisonResult

	/	1	
ÿ	。	175 AD→“175”	2016 AD→“2016”
YY	2。	5 AD→“05”	2016 AD→“16”
YYY	3。	5 AD→“005”	2016 AD→“2016”
YYYY	4。	5 AD→“0005”	2016 AD→“2016”
	。	→“7”	“” →“”
MM	2。	→“07”	“” →“”
MMM	。	→“”	“” →“”
MMMM	。	→“”	“” →“”
MMMMM	JanJuneJuly'J'。	→“J”	“” →“N”
d	。	8→“8”	29→“29”
DD	。	8→“08”	29→“29”
“E”“EE”“EEE”	3。	→“”	→“”
EEEE	。	→“”	→“”
EEEEEE	1。 'T'	→“M”	→“T”
EEEEEE	2。	→“”	→“Th”
	/。	10→“PM”	2→“”
H	1-121。	10→“10”	2→“2”
HH	1-122。	10→“10”	2→“02”
H	0-231。	10→“14”	2→“2”

	/	1	
HH	0-232。	10→“14”	2→“02”
	1。	7→“7”	29→“29”
	2。	7→“07”	29→“29”
	。	7→“7”	29→“29”
SS	2。	7→“07”	29→“29”

zS。

## Examples

- NSDate

```
var date = NSDate()
```

### 3

```
var date = Date()
```

## Objective-C

```
NSDate *date = [NSDate date];
```

### NSDateN

- ◦

dateWithTimeIntervalSinceNow(seconds: NSTimeInterval) -> NSDate **Swift+**  
 (NSDate\*)dateWithTimeIntervalSinceNow:(NSTimeInterval)seconds **Objective-C**。

- 

```
let totalSecondsInWeek:NSTimeInterval = 7 * 24 * 60 * 60;
//Using negative value for previous date from today
let nextWeek = NSDate().dateWithTimeIntervalSinceNow(totalSecondsInWeek)

//Using positive value for future date from today
let lastWeek = NSDate().dateWithTimeIntervalSinceNow(-totalSecondsInWeek)
```

### 3

```
let totalSecondsInWeek:TimeInterval = 7 * 24 * 60 * 60;

//Using positive value to add to the current date
let nextWeek = Date(timeIntervalSinceNow: totalSecondsInWeek)

//Using negative value to get date one week from current date
let lastWeek = Date(timeIntervalSinceNow: -totalSecondsInWeek)
```

## Objective-C

```
NSTimeInterval totalSecondsInWeek = 7 * 24 * 60 * 60;
//Using negative value for previous date from today
NSDate *lastWeek = [NSDate dateWithTimeIntervalSinceNow:-totalSecondsInWeek];

//Using positive value for future date from today
NSDate *nextWeek = [NSDate dateWithTimeIntervalSinceNow:totalSecondsInWeek];

NSLog(@"Last Week: %@", lastWeek);
NSLog(@"Right Now: %@", now);
NSLog(@"Next Week: %@", nextWeek);
```

### 4

- isEqualToDate(anotherDate: NSDate) -> Bool
- earlierDate(anotherDate: NSDate) -> NSDate
- laterDate(anotherDate: NSDate) -> NSDate
- compare(anotherDate: NSDate) -> NSComparisonResult

## Objective-C

- - (BOOL)isEqualToDate:(NSDate \*)anotherDate
- - (NSDate \*)earlierDate:(NSDate \*)anotherDate
- - (NSDate \*)laterDate:(NSDate \*)anotherDate
- - (NSComparisonResult)compare:(NSDate \*)anotherDate

### 2

```
let date1: NSDate = ... // initialized as July 7, 2016 00:00:00
let date2: NSDate = ... // initialized as July 2, 2016 00:00:00
```

## Objective-C

```
NSDate *date1 = ... // initialized as July 7, 2016 00:00:00
NSDate *date2 = ... // initialized as July 2, 2016 00:00:00
```

```
if date1.isEqualToDate(date2) {
 // returns false, as both dates aren't equal
}
```

```
earlierDate: NSDate = date1.earlierDate(date2) // returns the earlier date of the two (date 2)
```

```

laterDate: NSDate = date1.laterDate(date2) // returns the later date of the two (date1)

result: NSComparisonResult = date1.compare(date2)

if result == .OrderedAscending {
 // true if date1 is earlier than date2
} else if result == .OrderedSame {
 // true if the dates are the same
} else if result == .OrderedDescending {
 // true if date1 is later than date1
}

```

## Objective-C

```

if ([date1 isEqualToDate:date2]) {
 // returns false, as both date are not equal
}

NSDate *earlierDate = [date1 earlierDate:date2]; // returns date which comes earlier from both
date, here it will return date2
NSDate *laterDate = [date1 laterDate:date2]; // returns date which comes later from both date,
here it will return date1

NSComparisonResult result = [date1 compare:date2];
if (result == NSOrderedAscending) {
 // fails
 // comes here if date1 is earlier then date2, in our case it will not come here
} else if (result == NSOrderedSame){
 // fails
 // comes here if date1 is same as date2, in our case it will not come here
} else{ // NSOrderedDescending
 // succeeds
 // comes here if date1 is later than date2, in our case it will come here
}

```

## 3

```

let dateStringUTC = "2016-10-22 12:37:48 +0000"
let dateFormatter = DateFormatter()
dateFormatter.locale = Locale(identifier: "en_US_POSIX")
dateFormatter.dateFormat = "yyyy-MM-dd HH:mm:ss X"
let date = dateFormatter.date(from: dateStringUTC)!

let now = Date()

let formatter = DateComponentsFormatter()
formatter.unitsStyle = .full
formatter.maximumUnitCount = 2
let string = formatter.string(from: date, to: Date())! + " " + NSLocalizedString("ago",
comment: "added after elapsed time to say how long before")

```

```

// get the current date and time
let currentDateTime = Date()

// get the user's calendar

```



```

let userCalendar = Calendar.current

// choose which date and time components are needed
let requestedComponents: Set<Calendar.Component> = [
 .year,
 .month,
 .day,
 .hour,
 .minute,
 .second
]

// get the components
let dateTimeComponents = userCalendar.dateComponents(requestedComponents, from:
currentDateTime)

// now the components are available
dateTimeComponents.year
dateTimeComponents.month
dateTimeComponents.day
dateTimeComponents.hour
dateTimeComponents.minute
dateTimeComponents.second

```

## Unix

**Unix Epoch Time** `timeIntervalSince1970`

```

let date = NSDate() // current date
let unixtime = date.timeIntervalSince1970

```

## Objective-C

```

NSDate *date = [NSDate date]; // current date
int unixtime = [date timeIntervalSince1970];

```

## NSDateFormatter

NSDate3◦

### 1. NSDateFormatter

```

let dateFormatter = NSDateFormatter()

```

### 3

```

let dateFormatter = DateFormatter()

```

## Objective-C

```
NSDateFormatter *dateFormatter = [[NSDateFormatter alloc] init];
```

---

### 2.

```
dateFormatter.dateFormat = "yyyy-MM-dd 'at' HH:mm"
```

## Objective-C

```
dateFormatter.dateFormat = @"yyyy-MM-dd 'at' HH:mm";
```

---

### 3.

```
let date = NSDate() // your NSDate object
let dateString = dateFormatter.stringFromDate(date)
```

### 3

```
let date = Date() // your NSDate object
let dateString = dateFormatter.stringFromDate(date)
```

## Objective-C

```
NSDate *date = [NSDate date]; // your NSDate object
NSString *dateString = [dateFormatter stringFromDate:date];
```

2001-01-02 at 13:00

---

NSDateFormatter

o

```
extension Date {
 func toString() -> String {
 let dateFormatter = DateFormatter()
 dateFormatter.dateFormat = "MMMM dd yyyy"
 return dateFormatter.string(from: self)
 }
}
```

[swiftly-getting-human-readable-date-nsdateformatter](#) ◦

◦

## NSDateNSDate

NSDate08:12NSDate◦

NSDate""NSDate""◦

hourAndMinuteNSDate

## Objective-C

```
NSDateComponents *hourAndMinuteComponents = [calendar components:NSCalendarUnitHour |
NSDateComponents *componentsOfDate = [[NSCalendar currentCalendar]
NSDateComponents *components = [[NSDateComponents alloc] init];
[components setDay: componentsOfDate.day];
[components setMonth: componentsOfDate.month];
[components setYear: componentsOfDate.year];
[components setHour: [hourAndMinuteComponents hour]];
[components setMinute: [hourAndMinuteComponents minute]];
[components setSecond: 0];
[calendar setTimeZone: [NSTimeZone defaultTimeZone]];

NSDate *yourFullNSDateObject = [calendar dateFromComponents:components];
```

""◦

## UTCTimeZoneNSDate

UTC◦

```
+(NSTimeInterval)getUTCOffsetIntervalWithCurrentTimeZone:(NSTimeZone *)current forDate:(NSDate
*)date {
 NSTimeZone *utcTimeZone = [NSTimeZone timeZoneWithAbbreviation:@"UTC"];
 NSInteger currentGMTOffset = [current secondsFromGMTForDate:date];
 NSInteger gmtOffset = [utcTimeZone secondsFromGMTForDate:date];
 NSTimeInterval gmtInterval = currentGMTOffset - gmtOffset;
 return gmtInterval;
}
```

1224

---

**AMPM**

## Objective-C

```
NSDateFormatter *formatter = [[NSDateFormatter alloc] init];
[formatter setLocale:[NSLocale currentLocale]];
[formatter setDateStyle:NSDateFormatterNoStyle];
[formatter setTimeStyle:NSDateFormatterShortStyle];
NSString *dateString = [formatter stringFromDate:[NSDate date]];
NSRange amRange = [dateString rangeOfString:[formatter AMSymbol]];
NSRange pmRange = [dateString rangeOfString:[formatter PMSymbol]];
BOOL is24h = (amRange.location == NSNotFound && pmRange.location == NSNotFound);
```

**NSDateFormatter**

## Objective-C

```
NSString *formatStringForHours = [NSDateFormatter dateFormatFromTemplate:@"%j" options:0
locale:[NSLocale currentLocale]];
NSRange containsA = [formatStringForHours rangeOfString:@"%a"];
BOOL is24h = containsA.location == NSNotFound;
```

“j” ICU Spec ...

[...]hHKk。 [...]API'j'1224。

- “”◦ NSDateFormatterNSCalendarICU◦

◦

JSONNSDate“/1268123281843/”

Json.NET 4.5Microsoft“/ Date1198908717056/”◦ NSDate

## Objective-C

```
(NSDate*) getDateFromJSON:(NSString *)dateString
{
 // Expect date in this format "/Date(1268123281843)/"
 int startPos = [dateString rangeOfString:@"("].location+1;
 int endPos = [dateString rangeOfString:@")"].location;
 NSRange range = NSMakeRange(startPos, endPos-startPos);
 unsigned long long milliseconds = [[dateString substringWithRange:range] longLongValue];
 NSLog(@"%llu", milliseconds);
 NSTimeInterval interval = milliseconds/1000;
 NSDate *date = [NSDate dateWithTimeIntervalSince1970:interval];
 // add code for date formatter if need NSDate in specific format.
 return date;
}
```

NSDate5s ago2m ago3h ago

RSS

## Objective-C

```
- (NSString *)getHistoricTimeText:(NSDate *)since
{
 NSString *str;
 NSTimeInterval interval = [[NSDate date] timeIntervalSinceDate:since];
 if(interval < 60)
 str = [NSString stringWithFormat:@"%is ago", (int)interval];
 else if(interval < 3600)
 {
 int minutes = interval/60;
 str = [NSString stringWithFormat:@"%im ago", minutes];
 }
 else if(interval < 86400)
 {
 int hours = interval/3600;

 str = [NSString stringWithFormat:@"%ih ago", hours];
 }
 else
 {
 NSDateFormatter *dateFormatter=[[NSDateFormatter alloc]init];
 [dateFormatter setLocale:[NSLocale currentLocale]];
 NSString *dateFormat = [NSDateFormatter dateFormatFromTemplate:@"MMM d, YYYY"
options:0 locale:[NSLocale currentLocale]];
 [dateFormatter setDateFormat:dateFormat];
 str = [dateFormatter stringFromDate:since];

 }
 return str;
}
```

NSDate <https://riptutorial.com/zh-TW/ios/topic/1502/nsdate>

# 169: NSTimer

interval	;◦
target	selector
selector	Swift Selector◦target
repeats	false ◦ true interval◦

[NSTimer◦](#)

## Examples

[doSomethingself 5◦](#)

```
let timer = NSTimer.scheduledTimerWithTimeInterval(5,
 target: self,
 selector: Selector(doSomething()),
 userInfo: nil,
 repeats: false)
```

### 3

```
let timer = Timer.scheduledTimer(timeInterval: 1,
 target: self,
 selector: #selector(doSomething()),
 userInfo: nil,
 repeats: true)
```

## Objective-C

```
NSTimer *timer = [NSTimer scheduledTimerWithTimeInterval:5.0 target:self
selector:@selector(doSomething) userInfo:nil repeats:NO];
```

[false/NO◦ true/YES ◦](#)

```
timer.fire()
```

## Objective-C

```
[timer fire];
```

[fireNSTimer◦](#)

◦ [fire◦](#)

◦

```
timer.invalidate()
```

## Objective-C

```
[timer invalidate];
```

### ◦ Apple

◦ ◦

### ◦ fire◦

```
class ViewController: UIViewController {

 var timer = NSTimer()

 override func viewDidLoad() {
 NSTimer.scheduledTimerWithTimeInterval(1.0, target: self, selector:
Selector(self.timerMethod()), userInfo: nil, repeats: true)
 }

 func timerMethod() {
 print("Timer method called")
 }

 func endTimer() {
 timer.invalidate()
 }
}
```

## 3

```
class ViewController: UIViewController {

 var timer = Timer()

 override func viewDidLoad() {
 Timer.scheduledTimer(timeInterval: 1.0, target: self, selector:
#selector(self.timerMethod()), userInfo: nil, repeats: true)
 }

 func timerMethod() {
 print("Timer method called")
 }

 func endTimer() {
 timer.invalidate()
 }
}
```

◦

```
NSTimer.scheduledTimerWithTimeInterval(3.0, target: self, selector:
Selector(self.timerMethod()), userInfo: nil, repeats: false)
```

3

```
Timer.scheduledTimer(timeInterval: 3.0, target: self, selector:
#selector(self.timerMethod()), userInfo: nil, repeats: false)
```

3. ◦

## Timer

userInfo◦

Timer◦

[ 3 ]

```
Timer.scheduledTimer(timeInterval: 1.0, target: self, selector:#selector(iGotCall(sender:)),
userInfo: ["Name": "i am iOS guy"], repeats:true)
```

[ - C ]

```
NSTimer* timer = [NSTimer scheduledTimerWithTimeInterval:1.0
 target:self
 selector:@selector(iGotCall:)
 userInfo:@"i am iOS guy" repeats:YES];
```

userInfo ["Name": "i am iOS guy"] ◦ iGotCall◦

[ 3 ]

```
func iGotCall(sender: Timer) {
 print((sender.userInfo!))
}
```

[ - C ]

```
- (void)iGotCall:(NSTimer*)theTimer {
 NSLog(@"%@", (NSString*)[theTimer userInfo]);
}
```

**NSTimer** <https://riptutorial.com/zh-TW/ios/topic/2624/nstimer>



---

# 170: UIBarButtonItem

	UIBarButtonItem
	UIBarButtonItem
	UIBarButtonItem
	UIBarButtonItem

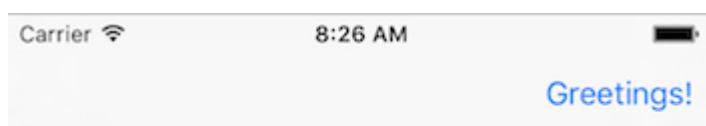
self.navigationItem:UIViewController UINavigationController ◦

## Examples

### UIBarButtonItem

```
//Swift
let barButtonItem = UIBarButtonItem(title: "Greetings!", style: .Plain, target: self, action:
#selector(barButtonTapped))
self.navigationItem.rightBarButtonItem = barButtonItem

//Objective-C
UIBarButtonItem *barButtonItem = [[UIBarButtonItem alloc] initWithTitle:@"Greetings!"
style:UIBarButtonItemStylePlain target:self action:@selector(barButtonTapped)];
self.navigationItem.rightBarButtonItem = barButtonItem;
```

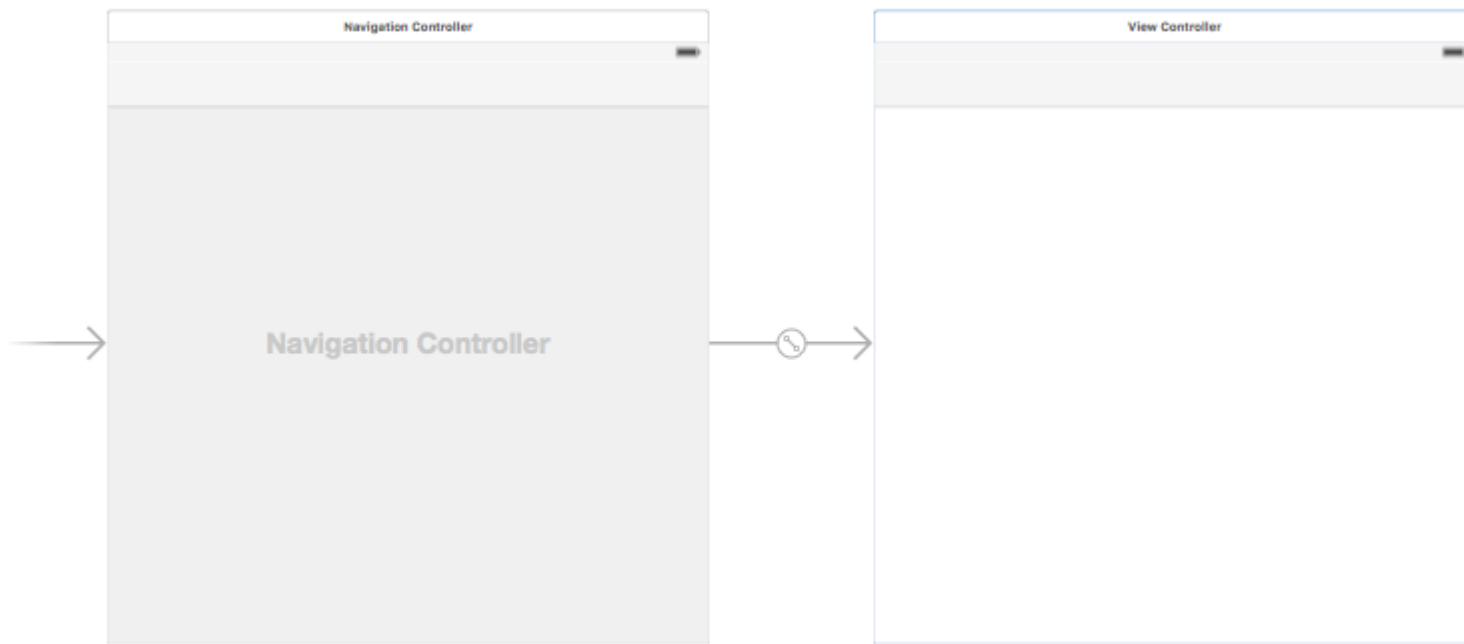


### Interface BuilderUIBarButtonItem

Interface BuilderUIBarButtonItem ◦

---

View ControllerXcodeEditor> Embed In> Navigation Controller ◦



UINavigationController ◦

UIBarButtonItem UIBarButtonItem ◦

File Code Lock List

**Item** - Represents an item on a UIToolbar or UINavigationController object.

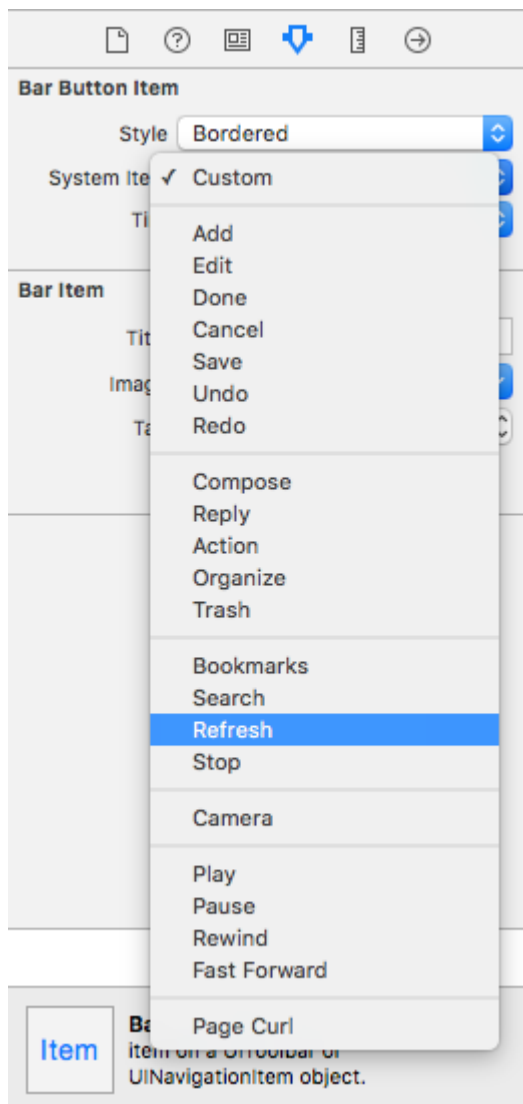
**Fixed Space Bar Button Item** - Represents a fixed space item on a UIToolbar object.

**Flexible Space Bar Button Item** - Represents a flexible space item on a UIToolbar object.

uibarbuttonitem



“” ◦ UIBarButtonItemAttributes Inspector System Item Refresh ◦



“” ◦



**IB**

UIBarButtonItem View Controller@IBAction ◦

```
class ViewController: UIViewController {

 @IBAction func refreshBarButtonItemTap(sender: UIBarButtonItem) {

 print("How refreshing!")
 }

}
```

◦

- [Stack Overflow](#) ◦

UIBarButtonItem nullInterface Builder ◦

## Objective-C

```
UIBarButtonItem.image = [UIBarButtonItem.image
imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];
```

[UIBarButtonItem](https://riptutorial.com/zh-TW/ios/topic/1543/uiBarButtonItem) <https://riptutorial.com/zh-TW/ios/topic/1543/uiBarButtonItem>

---

# 171: UIButton

UIButton UIControl ◦ ◦ ◦

◦ ◦

- UIButtonTypeCustom

```
No button style.
```

- UIButtonTypeSystem

```
A system style button, such as those shown in navigation bars and toolbars.
```

- UIButtonTypeDetailDisclosure

```
A detail disclosure button.
```

- UIButtonTypeInfoLight

```
An information button that has a light background.
```

- UIButtonTypeInfoDark

```
An information button that has a dark background.
```

- UIButtonTypeContactAdd

```
A contact add button.
```

0,0,0,0 ◦

## Examples

### UIButton

### UIButtons

```
let button = UIButton(frame: CGRect(x: x, y: y, width: width, height: height))
```

### C.

```
UIButton *button = [[UIButton alloc] initWithFrame:CGRectMake(x, y, width, height)];
```

## UIButton

```
let button = UIButton(type: .Custom)
```

### C.

```
UIButton *button = [UIButton buttonWithType:UIButtonTypeCustom];
```

typeUIButtonType

```
enum UIButtonType : Int {
 case Custom
 case System
 case DetailDisclosure
 case InfoLight
 case InfoDark
 case ContactAdd
 static var RoundedRect: UIButtonType { get }
}
```

```
button.setTitle(titleString, forState: controlState)
```

### C.

```
[button setTitle:(NSString *) forState:(UIControlState)];
```

### “HelloWorld”

```
button.setTitle("Hello, World!", forState: .normal)
```

### C.

```
[button setTitle:@"Hello, World!" forState:UIControlStateNormal];
```

```
//Swift
button.setTitleColor(color, forState: controlState)
```

```
//Objective-C
[button setTitleColor:(nullable UIColor *) forState:(UIControlState)];
```

```
//Swift
button.setTitleColor(.blue, for: .normal)
```

```
//Objective-C
[button setTitleColor:[UIColor blueColor] forState:UIControlStateNormal]
```

```
//Align contents to the left of the frame
button.contentHorizontalAlignment = .left
```

```
//Align contents to the right of the frame
```

```
button.contentHorizontalAlignment = .right

//Align contents to the center of the frame
button.contentHorizontalAlignment = .center

//Make contents fill the frame
button.contentHorizontalAlignment = .fill
```

### C.

```
//Align contents to the left
button.contentHorizontalAlignment = UIControlContentHorizontalAlignmentLeft;

//Align contents to the right
button.contentHorizontalAlignment = UIControlContentHorizontalAlignmentRight;

//Align contents to the center
button.contentHorizontalAlignment = UIControlContentHorizontalAlignmentCenter;

//Align contents to fill the frame
button.contentHorizontalAlignment = UIControlContentHorizontalAlignmentFill;
```

```
var label: UILabel? = button.titleLabel
```

### C.

```
UILabel *label = button.titleLabel;
```

```
button.titleLabel?.font = UIFont.boldSystemFontOfSize(12)
```

### C.

```
button.titleLabel.font = [UIFont boldSystemFontOfSize:12];
```

## UIButton

```
myButton.isEnabled = false
```

### Objective-C

```
myButton.enabled = NO;
```

# Button

adjustsImageWhenDisabledfalse / NO

# UIButton

## Objective-C

```
-(void)someButtonAction:(id)sender {
 // sender is the object that was tapped, in this case its the button.
 NSLog(@"Button is tapped");
}
```

```
func someButtonAction() {
 print("Button is tapped")
}
```

## C.

```
[yourButtonInstance addTarget:self action:@selector(someButtonAction)
forControlEvents:UIControlEventTouchUpInside];
```

```
yourButtonInstance.addTarget(self, action: #selector(someButtonAction), forControlEvents:
.TouchUpInside)
```

## ControlEvents ENUM [UIControlEvents](#)

```
myButton.titleLabel?.font = UIFont(name: "YourFontName", size: 20)
```

## C.

```
myButton.titleLabel.font = [UIFont fontWithName:@"YourFontName" size:20];
```

## Objective-C

```
-(void) someButtonAction{
 NSLog(@"Button is tapped");
}
```

```
func someButtonAction() {
 print("Button is tapped")
}
```

## C.

```
[yourButtonInstance addTarget:self action:@selector(someButtonAction)
forControlEvents:UIControlEventTouchUpInside];
```

```
yourButtonInstance.addTarget(self, action: #selector(someButtonAction), forControlEvents:
.touchUpInside)
```



ControlEvents ENUM [UIButtonEvents](#) ◦

## UIButton

UIButtonfunctionalContentSize intrinsicContentSize ◦

```
button.intrinsicContentSize.width
```

## Objective-C

```
button.intrinsicContentSize.width;
```

```
button.setImage (UIImage (named:"test-image"), forState: .normal)
```

## C.

```
[self.button setImage:[UIImage imageNamed:@"test-image"] forState:UIControlStateNormal];
```

---

**UIControlStatesSelectedHighlighted**

```
button.setImage (UIImage (named:"test-image"), forState:[.selected, .highlighted])
```

## C.

```
[self.button setImage:[UIImage imageNamed:@"test-image"]
forState:UIControlStateSelected|UIControlStateHighlighted];
```

**UIButton** <https://riptutorial.com/zh-TW/ios/topic/516/uibutton>

# 172: UIColor

## Examples

### UIColor

UIColor

- ```
let redColor = UIColor.redColor()
let blueColor: UIColor = .blueColor()

// In Swift 3, the "Color()" suffix is removed:
let redColor = UIColor.red
let blueColor: UIColor = .blue
```

UIColor

```
let view = UIView()
view.backgroundColor = .yellowColor()
```

- alpha

```
let grayscaleColor = UIColor(white: 0.5, alpha: 1.0)
```

- alpha

```
let hsbColor = UIColor(
    hue: 0.4,
    saturation: 0.3,
    brightness: 0.7,
    alpha: 1.0
)
```

- RGBA

```
let rgbColor = UIColor(
    red: 30.0 / 255,
    green: 70.0 / 255,
    blue: 200.0 / 255,
    alpha: 1.0
)
```

- ```
let patternColor = UIColor(patternImage: UIImage(named: "myImage")!)
```

### Objective-C

- ```
UIColor *redColor = [UIColor redColor];
```

- alpha

```
UIColor *grayscaleColor = [UIColor colorWithWhite: 0.5 alpha: 1.0];
```

- alpha

```
UIColor *hsbColor = [UIColor  
    colorWithHue: 0.4  
    saturation: 0.3  
    brightness: 0.7  
    alpha: 1.0  
];
```

- RGBA

```
UIColor *rgbColor = [UIColor  
    colorWithRed: 30.0 / 255.0  
    green: 70.0 / 255.0  
    blue: 200.0 / 255.0  
    alpha: 1.0  
];
```

- ```
UIColor *pattenColor = [UIColor colorWithPatternImage:[UIImage
 imageNamed:@"myImage.png"]];
```

UIColor ◦ UIColor ◦ styleString()\_systemDestructiveTintColor() ◦

### styleString

iOS 2.0 UIColor styleString RGB RGB ARGB whiteColor whiteColor ◦

### Objective-C

```
@interface UIColor (Private)

- (NSString *)styleString;

@end

// ...

[[UIColor whiteColor] styleString]; // rgb(255,255,255)
[[UIColor redColor] styleString]; // rgb(255,0,0)
[[UIColor lightTextColor] styleString]; // rgba(255,255,255,0.600000)
```

Swift ◦ Swift @objc unsafeBitCast UIColor

```
@objc protocol UIColorPrivate {
 func styleString() -> String
}

let white = UIColor.whiteColor()
let red = UIColor.redColor()
```

```

let lightTextColor = UIColor.lightTextColor()

let whitePrivate = unsafeBitCast(white, UIColorPrivate.self)
let redPrivate = unsafeBitCast(red, UIColorPrivate.self)
let lightTextColorPrivate = unsafeBitCast(lightTextColor, UIColorPrivate.self)

whitePrivate.styleString() // rgb(255,255,255)
redPrivate.styleString() // rgb(255,0,0)
lightTextColorPrivate.styleString() // rgba(255,255,255,0.600000)

```

## **`_systemDestructiveTintColor()`**

`UIColor_systemDestructiveTintColor`

```
let red = UIColor.performSelector("_systemDestructiveTintColor").takeUnretainedValue()
```

`.takeUnretainedValue()` ◦

## API

```

if UIColor.respondsToSelector("_systemDestructiveTintColor") {
 if let red = UIColor.performSelector("_systemDestructiveTintColor").takeUnretainedValue()
as? UIColor {
 // use the color
 }
}

```

```

@objc protocol UIColorPrivateStatic {
 func _systemDestructiveTintColor() -> UIColor
}

```

```

let privateClass = UIColor.self as! UIColorPrivateStatic
privateClass._systemDestructiveTintColor() // UIDeviceRGBColorSpace 1 0.231373 0.188235 1

```

## Alpha

`init(red:_, green:_, blue:_, alpha:_) UIColorUIColor` ◦

```
let colorWithAlpha = UIColor.redColor().colorWithAlphaComponent(0.1)
```

## 3

```

//In Swift Latest Version
_ colorWithAlpha = UIColor.red.withAlphaComponent(0.1)

```

## Objective-C

```
UIColor * colorWithAlpha = [[UIColor redColor] colorWithAlphaComponent:0.1];
```

## CGColor

Interface BuilderCGColorCGColor ;

### Swift

```
extension CALayer {
 func borderUIColor() -> UIColor? {
 return borderColor != nil ? UIColor(CGColor: borderColor!) : nil
 }

 func setBorderUIColor(color: UIColor) {
 borderColor = color.CGColor
 }
}
```

## borderUIColor

### User Defined Runtime Attributes

Key Path	Type	Value
layer.cornerRadius	Number	6.5
layer.borderWidth	Number	1
layer.clipsToBounds	Boolean	<input checked="" type="checkbox"/>
layer.borderColor	Color	<input type="color"/>

## UIColor

UIColor 0xff00cc“FFFFFF”

```
extension UIColor {
 convenience init(hex: Int, alpha: CGFloat = 1.0) {
 let r = CGFloat((hex >> 16) & 0xff) / 255
 let g = CGFloat((hex >> 08) & 0xff) / 255
 let b = CGFloat((hex >> 00) & 0xff) / 255
 self.init(red: r, green: g, blue: b, alpha: alpha)
 }
}
```

```
let color = UIColor(hex: 0xff00cc, alpha: 1.0)
```

alpha 1.0

```
let color = UIColor(hex: 0xff00cc)
```

```
extension UIColor {
 convenience init(hexCode: String) {
 let hex =
hexCode.stringByTrimmingCharactersInSet(NSCharacterSet.alphanumericCharacterSet().invertedSet)
 var int = UInt32()
 NSScanner(string: hex).scanHexInt(&int)
```

```

let a, r, g, b: UInt32

switch hex.characters.count {
case 3:
 (a, r, g, b) = (255, (int >> 8) * 17, (int >> 4 & 0xF) * 17, (int & 0xF) * 17)
case 6:
 (a, r, g, b) = (255, int >> 16, int >> 8 & 0xFF, int & 0xFF)
case 8:
 (a, r, g, b) = (int >> 24, int >> 16 & 0xFF, int >> 8 & 0xFF, int & 0xFF)
default:
 (a, r, g, b) = (1, 1, 1, 0)
}

self.init(red: CGFloat(r) / 255, green: CGFloat(g) / 255, blue: CGFloat(b) / 255,
alpha: CGFloat(a) / 255)
}
}

```

## alpha

```
let color = UIColor("#80FFFFFF")
```

## alpha color alpha1.0

```
let color = UIColor("#FFFFFF")
let color = UIColor("#FFF")
```

## Objective-C

```

@interface UIColor (Hex)
+ (UIColor *)colorWithHex:(NSUInteger)hex alpha:(CGFloat)alpha;
@end

@implementation UIColor (Hex)
+ (UIColor *)colorWithHex:(NSUInteger)hex alpha:(CGFloat)alpha {
 return [UIColor colorWithRed:((CGFloat)((hex & 0xFF0000) >> 16))/255.0
 green:((CGFloat)((hex & 0xFF00) >> 8))/255.0
 blue:((CGFloat)(hex & 0xFF))/255.0
 alpha:alpha];
}
@end

```

```
UIColor *color = [UIColor colorWithHex:0xff00cc alpha:1.0];
```

```

- (UIColor*) hex:(NSString*)hexCode {

 NSString *noHashString = [hexCode stringByReplacingOccurrencesOfString:@"#"
withString:@""];
 NSScanner *scanner = [NSScanner scannerWithString:noHashString];
 [scanner setCharactersToBeSkipped:[NSCharacterSet symbolCharacterSet]];

 unsigned hex;
 if (![scanner scanHexInt:&hex]) return nil;
 int a;

```

```

int r;
int g;
int b;

switch (noHashString.length) {
 case 3:
 a = 255;
 r = (hex >> 8) * 17;
 g = ((hex >> 4) & 0xFF) * 17;
 b = ((hex >> 0) & 0xFF) * 17;
 break;
 case 6:
 a = 255;
 r = (hex >> 16);
 g = (hex >> 8) & 0xFF;
 b = (hex) & 0xFF;
 break;
 case 8:
 a = (hex >> 24);
 r = (hex >> 16) & 0xFF;
 g = (hex >> 8) & 0xFF;
 b = (hex) & 0xFF;
 break;

 default:
 a = 255.0;
 r = 255.0;
 b = 255.0;
 g = 255.0;
 break;
}

return [UIColor colorWithRed:r / 255.0f green:g / 255.0f blue:b / 255.0f alpha:a / 255];
}

```

## alpha

```
UIColor* color = [self hex:@"#80FFFFFF"];
```

## alpha color alpha1

```
UIColor* color = [self hex:@"#FFFFFF"];
UIColor* color = [self hex:@"#FFF"];
```

## UIColor

。

## Objective-C

```

+ (UIColor *)adjustedColorForColor:(UIColor *)c : (double)percent
{
 if (percent < 0) percent = 0;

 CGFloat r, g, b, a;
 if ([c getRed:&r green:&g blue:&b alpha:&a])

```

```
 return [UIColor colorWithRed:MAX(r * percent, 0.0)
 green:MAX(g * percent, 0.0)
 blue:MAX(b * percent, 0.0)
 alpha:a];
 }
 return nil;
}
```

```
func adjustedColorForColor(c: UIColor, var percent: CGFloat) -> UIColor {
 if percent < 0 {
 percent = 0
 }

 var r,g,b,a: CGFloat
 r = 0.0
 g = 0.0
 b = 0.0
 a = 0.0

 if c.getRed(&r, green: &g, blue: &b, alpha: &a) {
 return UIColor(red: max(r * percent, 0.0), green: max(g * percent, 0.0), blue: max(b *
percent, 0.0), alpha: a)
 }

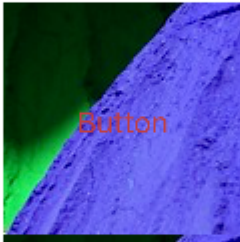
 return UIColor()
}
```

## UIColor

UIColor(patternImage:\_) UIColor°

```
btn.backgroundColor = UIColor(patternImage: UIImage(named: "image")!)
```





---

## UIColor

```
+ (UIColor *)darkerColorForColor:(UIColor *)c
{
 CGFloat r, g, b, a;
 if ([c getRed:&r green:&g blue:&b alpha:&a])
 return [UIColor colorWithRed:MAX(r - 0.2, 0.0)
 green:MAX(g - 0.2, 0.0)
 blue:MAX(b - 0.2, 0.0)
 alpha:a];

 return nil;
}
```

```
+ (UIColor *)lighterColorForColor:(UIColor *)c
{
 CGFloat r, g, b, a;
 if ([c getRed:&r green:&g blue:&b alpha:&a])
 return [UIColor colorWithRed:MIN(r + 0.2, 1.0)
 green:MIN(g + 0.2, 1.0)
 blue:MIN(b + 0.2, 1.0)
 alpha:a];

 return nil;
}
```

```
[UIColor orangeColor]
```



UIColor <https://riptutorial.com/zh-TW/ios/topic/956/uicolor>

# 173: UIDatePicker

UIDatePicker UIPickerView

## Examples

```
let datePicker = UIDatePicker(frame: CGRect(x: 0, y: 0, width: 320, height: 200))
```

## Objective-C

```
UIDatePicker *datePicker = [[UIDatePicker alloc] initWithFrame:CGRectMake(x: 0, y: 0, width: 320, height: 200)];
```

UIDatePicker.

```
[datePicker setMinimumDate:[NSDate date]];
```

```
[datePicker setMaximumDate:[NSDate date]];
```

UIDatePicker.

```
enum UIDatePickerMode : Int {
 case Time
 case Date
 case DateAndTime
 case CountdownTimer
}
```

- Time - AM / PM.
- Date - .
- DateAndTime - AM / PM.
- CountdownTimer - [1 | 53]. .

datePickerMode

```
let datePicker = UIDatePicker(frame: CGRect(x: 0, y: 0, width: 320, height: 200))
datePicker.datePickerMode = .Date
```

minuteInterval. 130.

```
let datePicker = UIDatePicker(frame: CGRect(x: 0, y: 0, width: 320, height: 200))
datePicker.minuteInterval = 15
```

NSTimeInterval ◦ CountdownTimer ◦ **86,39923:59**

```
let datePicker = UIDatePicker(frame: CGRect(x: 0, y: 0, width: 320, height: 200))
datePicker.countDownDuration = 60 * 60
```

**UIDatePicker** <https://riptutorial.com/zh-TW/ios/topic/5643/uidatepicker>

# 174: UIDevice

	◦
systemNameString	◦
modelString	◦
systemVersionString	..

UIDeviceSingleton ◦ ◦

## Examples

### iOS

#### 2

```
import UIKit

extension UIDevice {

 var modelName: String {
 var systemInfo = utsname()
 uname(&systemInfo)
 let machineMirror = Mirror(reflecting: systemInfo.machine)
 let identifier = machineMirror.children.reduce("") { identifier, element in
 guard let value = element.value as? Int8 where value != 0 else { return identifier
 }

 return identifier + String(UnicodeScalar(UInt8(value)))
 }

 switch identifier {
 case "iPod5,1": return "iPod Touch 5"
 case "iPod7,1": return "iPod Touch 6"
 case "iPhone3,1", "iPhone3,2", "iPhone3,3": return "iPhone 4"
 case "iPhone4,1": return "iPhone 4s"
 case "iPhone5,1", "iPhone5,2": return "iPhone 5"
 case "iPhone5,3", "iPhone5,4": return "iPhone 5c"
 case "iPhone6,1", "iPhone6,2": return "iPhone 5s"
 case "iPhone7,2": return "iPhone 6"
 case "iPhone7,1": return "iPhone 6 Plus"
 case "iPhone8,1": return "iPhone 6s"
 case "iPhone8,2": return "iPhone 6s Plus"
 case "iPhone9,1", "iPhone9,3": return "iPhone 7"
 case "iPhone9,2", "iPhone9,4": return "iPhone 7 Plus"
 case "iPhone8,4": return "iPhone SE"
 case "iPad2,1", "iPad2,2", "iPad2,3", "iPad2,4": return "iPad 2"
 case "iPad3,1", "iPad3,2", "iPad3,3": return "iPad 3"
 case "iPad3,4", "iPad3,5", "iPad3,6": return "iPad 4"
 case "iPad4,1", "iPad4,2", "iPad4,3": return "iPad Air"
```

```

 case "iPad5,3", "iPad5,4": return "iPad Air 2"
 case "iPad2,5", "iPad2,6", "iPad2,7": return "iPad Mini"
 case "iPad4,4", "iPad4,5", "iPad4,6": return "iPad Mini 2"
 case "iPad4,7", "iPad4,8", "iPad4,9": return "iPad Mini 3"
 case "iPad5,1", "iPad5,2": return "iPad Mini 4"
 case "iPad6,3", "iPad6,4", "iPad6,7", "iPad6,8":return "iPad Pro"
 case "AppleTV5,3": return "Apple TV"
 case "i386", "x86_64": return "Simulator"
 default: return identifier
 }
}

if UIDevice.currentDevice().modelName == "iPhone 6 Plus" {
 // is an iPhone 6 Plus
}

```

### 3

```

import UIKit

public extension UIDevice {

 var modelName: String {
 var systemInfo = utsname()
 uname(&systemInfo)
 let machineMirror = Mirror(reflecting: systemInfo.machine)
 let identifier = machineMirror.children.reduce("") { identifier, element in
 guard let value = element.value as? Int8 , value != 0 else { return identifier
 }

 return identifier + String(UnicodeScalar(UInt8(value)))
 }

 switch identifier {
 case "iPod5,1": return "iPod Touch 5"
 case "iPod7,1": return "iPod Touch 6"
 case "iPhone3,1", "iPhone3,2", "iPhone3,3": return "iPhone 4"
 case "iPhone4,1": return "iPhone 4s"
 case "iPhone5,1", "iPhone5,2": return "iPhone 5"
 case "iPhone5,3", "iPhone5,4": return "iPhone 5c"
 case "iPhone6,1", "iPhone6,2": return "iPhone 5s"
 case "iPhone7,2": return "iPhone 6"
 case "iPhone7,1": return "iPhone 6 Plus"
 case "iPhone8,1": return "iPhone 6s"
 case "iPhone8,2": return "iPhone 6s Plus"
 case "iPhone9,1", "iPhone9,3": return "iPhone 7"
 case "iPhone9,2", "iPhone9,4": return "iPhone 7 Plus"
 case "iPhone8,4": return "iPhone SE"
 case "iPad2,1", "iPad2,2", "iPad2,3", "iPad2,4":return "iPad 2"
 case "iPad3,1", "iPad3,2", "iPad3,3": return "iPad 3"
 case "iPad3,4", "iPad3,5", "iPad3,6": return "iPad 4"
 case "iPad4,1", "iPad4,2", "iPad4,3": return "iPad Air"
 case "iPad5,3", "iPad5,4": return "iPad Air 2"
 case "iPad2,5", "iPad2,6", "iPad2,7": return "iPad Mini"
 case "iPad4,4", "iPad4,5", "iPad4,6": return "iPad Mini 2"
 case "iPad4,7", "iPad4,8", "iPad4,9": return "iPad Mini 3"
 case "iPad5,1", "iPad5,2": return "iPad Mini 4"
 case "iPad6,3", "iPad6,4", "iPad6,7", "iPad6,8":return "iPad Pro"
 case "AppleTV5,3": return "Apple TV"
 case "i386", "x86_64": return "Simulator"
 }
 }
}

```

```

 default:
 return identifier
 }
}

if UIDevice.current.modelName == "iPhone 7" {
 // is an iPhone 7
}

```

```

override func viewDidLoad() {
 super.viewDidLoad()
 NotificationCenter.default.addObserver(self, selector:
Selector(("batteryStateDidChange:")), name: NSNotification.Name.UIDeviceBatteryStateDidChange,
object: nil)
 NotificationCenter.default.addObserver(self, selector:
Selector(("batteryLevelDidChange:")), name: NSNotification.Name.UIDeviceBatteryLevelDidChange,
object: nil)

 // Stuff...
}

func batteryStateDidChange(notification: NSNotification){
 // The stage did change: plugged, unplugged, full charge...
}

func batteryLevelDidChange(notification: NSNotification){

 let batteryLevel = UIDevice.current.batteryLevel
 if batteryLevel < 0.0 {
 print("-1.0 means battery state is UIDeviceBatteryStateUnknown")
 return
 }

 print("Battery Level : \(batteryLevel * 100)%")
 // The battery's level did change (98%, 99%, ...)
}

```

```

UIDevice *deviceInfo = [UIDevice currentDevice];
NSLog(@"Device Name %@", deviceInfo.name);
//Ex: myIphone6s
NSLog(@"System Name %@", deviceInfo.systemName);
//Device Name iPhone OS
NSLog(@"System Version %@", deviceInfo.systemVersion);
//System Version 9.3
NSLog(@"Model %@", deviceInfo.model);
//Model iPhone
NSLog(@"Localized Model %@", deviceInfo.localizedModel);
//Localized Model iPhone
int device=deviceInfo.userInterfaceIdiom;
//UIUserInterfaceIdiomPhone=0
//UIUserInterfaceIdiomPad=1
//UIUserInterfaceIdiomTV=2
//UIUserInterfaceIdiomCarPlay=3
//UIUserInterfaceIdiomUnspecified=-1
NSLog(@"identifierForVendor %@", deviceInfo.identifierForVendor);
//identifierForVendor <__NSConcreteUUID 0x7a10ae20> 556395DC-0EB4-4FD5-BC7E-B16F612ECC6D

```

```

UIDevice *deviceInfo = [UIDevice currentDevice];

```

```
int d = deviceInfo.orientation;
```

deviceInfo.orientation **UIDeviceOrientation**

```
UIDeviceOrientationUnknown 0
UIDeviceOrientationPortrait 1
UIDeviceOrientationPortraitUpsideDown 2
UIDeviceOrientationLandscapeLeft 3
UIDeviceOrientationLandscapeRight 4
UIDeviceOrientationFaceUp 5
UIDeviceOrientationFaceDown 6
```

```
- (void)viewWillAppear:(BOOL)animated
{
 [super viewWillAppear:animated];
 [[UIDevice currentDevice] beginGeneratingDeviceOrientationNotifications];
 [[NSNotificationCenter defaultCenter] addObserver:self
 selector:@selector(deviceOrientationDidChange)
 name:UIDeviceOrientationDidChangeNotification
 object:nil];
}

-(void)deviceOrientationDidChange
{
 UIDeviceOrientation orientation = [[UIDevice currentDevice] orientation];
 if (orientation == UIDeviceOrientationPortrait || orientation ==
UIDeviceOrientationPortraitUpsideDown) {
 [self changedToPortrait];
 } else if (orientation == UIDeviceOrientationLandscapeLeft || orientation ==
UIDeviceOrientationLandscapeRight) {
 [self changedToLandscape];
 }
}

-(void)changedToPortrait
{
 // Function Body
}

-(void)changedToLandscape
{
 // Function Body
}
```

```
- (void)viewWillDisappear:(BOOL)animated {
 [super viewWillDisappear:animated];
 [[UIDevice currentDevice] endGeneratingDeviceOrientationNotifications];
}
```

```
//Get permission for Battery Monitoring
[[UIDevice currentDevice] setBatteryMonitoringEnabled:YES];
UIDevice *myDevice = [UIDevice currentDevice];

[myDevice setBatteryMonitoringEnabled:YES];
double batLeft = (float)[myDevice batteryLevel] * 100;
NSLog(@"%.f",batLeft);

int d = myDevice.batteryState;
```



```

//Returns an Integer Value
//UIDeviceBatteryStateUnknown 0
//UIDeviceBatteryStateUnplugged 1
//UIDeviceBatteryStateCharging 2
//UIDeviceBatteryStateFull 3

//Using notifications for Battery Monitoring
-(void)startMonitoringForBatteryChanges
{
// Enable monitoring of battery status
[[UIDevice currentDevice] setBatteryMonitoringEnabled:YES];
// Request to be notified when battery charge or state changes
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(checkBatteryStatus)
name:UIDeviceBatteryLevelDidChangeNotification object:nil];
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(checkBatteryStatus)
name:UIDeviceBatteryStateDidChangeNotification object:nil];
}
-(void) checkBatteryStatus
{
NSLog(@"Battery Level is %.f",[[UIDevice currentDevice] batteryLevel]*100);
int d=[[UIDevice currentDevice] batteryState];
if (d==0)
{
NSLog(@"Unknown");
}
else if (d==1)
{
NSLog(@"Unplugged");
}
else if (d==2)
{
NSLog(@"Charging");
}
else if (d==3)
{
NSLog(@"Battery Full");
}
}
}

```

```

//Enabling the proximity Sensor
- (void)viewWillAppear:(BOOL)animated {

[super viewWillAppear:animated];
[[UIDevice currentDevice] setProximityMonitoringEnabled:YES];
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(sensorStateMonitor:) name:@"UIDeviceProximityStateDidChangeNotification"
object:nil];
}

- (void)sensorStateMonitor:(NSNotificationCenter *)notification
{
if ([[UIDevice currentDevice] proximityState] == YES)
{
NSLog(@"Device is close to user.");
}

else
{
NSLog(@"Device is not closer to user.");
}
}

```

```
}
```

UIDevice <https://riptutorial.com/zh-TW/ios/topic/4878/uidevice>

---

# 175: UIImage

UIImage Apple

## Examples

### UIImage

---

```
let image = UIImage(named: "imageFromBundleOrAsset")
```

## Objective-C

```
UIImage *image = [UIImage imageNamed:@"imageFromBundleOrAsset"];
```

---

`imageNamed` ◦ UIImage `imageWithContentsOfFile` ◦

---

## NSData

```
let imageData = Data(base64Encoded: imageString, options: Data.Base64DecodingOptions.ignoreUnknownCharacters)
```

```
let image = UIImage(data: imageData!)
```

---

## UIColor

```
let color = UIColor.red
let size = CGSize(width: 200, height: 200)
```

```
UIGraphicsBeginImageContextWithOptions(size, false, 0.0)
UIGraphicsGetCurrentContext()!.setFillColor(color.cgColor)
UIGraphicsGetCurrentContext()!.fill(CGRect(origin: .zero, size: size))
let colorImage = UIGraphicsGetImageFromCurrentImageContext()
UIGraphicsEndImageContext()
```

## Objective-C

```
UIColor *color=[UIColor redColor];
CGRect frame = CGRectMake(0, 0, 80, 100);
UIGraphicsBeginImageContext(frame.size);
CGContextRef context = UIGraphicsGetCurrentContext();
CGContextSetFillColorWithColor(context, [color CGColor]);
```

```
CGContextFillRect(context, frame);
UIImage *image = UIGraphicsGetImageFromCurrentImageContext();
UIGraphicsEndImageContext();
```

---

---

## Objective-C

```
UIImage *image = [UIImage imageNamed:[NSBundle mainBundle]
pathForResource:@"cellCountry objectForKey:@"Country_Flag" ofType:nil];
```

```
NSMutableArray *imageArray = [[NSMutableArray alloc] init];

for (int imageNumber = 1; self.myPhoto != nil; imageNumber++) {
 NSString *fileName = [NSString stringWithFormat:@"%d.jpg", self.myPhoto];

 // check if a file exists
 if ([UIImage imageNamed:fileName]) {
 // if it exists, add it to the array
 [imageArray addObject:[UIImage imageNamed:[NSBundle
mainBundle]pathForResource:[NSString stringWithFormat:@"%d", fileName] ofType:@""]];
 } else {
 break;
 }
}
```

//

```
self.myImageView.animationImages = imageArray;
```

o

```
UIImage *image = [UIImage imageNamed:[NSBundle mainBundle]
pathForResource:@"cellCountry objectForKey:@"Country_Flag" ofType:nil];
```

```
NSMutableArray *imageArray = [[NSMutableArray alloc] init];

for (int imageNumber = 1; self.myPhoto != nil; imageNumber++) {
 NSString *fileName = [NSString stringWithFormat:@"%d.jpg", self.myPhoto];

 // check if a file exists
 if ([UIImage imageNamed:fileName]) {
 // if it exists, add it to the array
 [imageArray addObject:[UIImage imageNamed:[NSBundle
mainBundle]pathForResource:[NSString stringWithFormat:@"%d", fileName] ofType:@""]];
 } else {
 break;
 }
}

//Using image array for animations here
self.myImageView.animationImages = imageArray;
```

UIImageInsets◦



```
let insets = UIImageInsetsMake(12.0, 20.0, 22.0, 12.0)
let image = UIImage(named: "test")
image?.resizableImageWithCapInsets(insets, resizingMode: .Stretch)
```

isEqual:◦ ◦ isEqual:◦ 1◦

## Apple

```
// Load the same image twice.
let image1 = UIImage(named: "MyImage")
let image2 = UIImage(named: "MyImage")

// The image objects may be different, but the contents are still equal
if let image1 = image1, image1.isEqual(image2) {
 // Correct. This technique compares the image data correctly.
}

if image1 == image2 {
 // Incorrect! Direct object comparisons may not work.
}
```

## Objective-C

```
// Load the same image twice.
UIImage* image1 = [UIImage imageNamed:@"MyImage"];
UIImage* image2 = [UIImage imageNamed:@"MyImage"];

// The image objects may be different, but the contents are still equal
if ([image1 isEqual:image2]) {
 // Correct. This technique compares the image data correctly.
}

if (image1 == image2) {
 // Incorrect! Direct object comparisons may not work.
}
```

## UIColorUIImage

```
let color = UIColor.redColor()
```

```

let size = CGSize(width: 200, height: 200)

 UIGraphicsBeginImageContextWithOptions(size, false, 0.0)
 CGContextSetFillColorWithColor(UIGraphicsGetCurrentContext(), color.CGColor)
 CGContextFillRect(UIGraphicsGetCurrentContext(), CGRect(origin: .zero, size: size))
 let colorImage = UIGraphicsGetImageFromCurrentImageContext()
 UIGraphicsEndImageContext()

```

## 3

```

let color = UIColor.red()
let size = CGSize(width: 200, height: 200)

 UIGraphicsBeginImageContextWithOptions(size, false, 0.0)
 if let context = UIGraphicsGetCurrentContext() {
 context.setFillColor(color.cgColor)
 context.fill(CGRect(origin: .zero, size: size))
 let colorImage = UIGraphicsGetImageFromCurrentImageContext()
 }
 UIGraphicsEndImageContext()

```

# Objective-C

UIImage

```

+ (UIImage *)createImageWithColor: (UIColor *)color {
 CGRect rect=CGRectMake(0.0f, 0.0f, 1.0f, 1.0f);
 UIGraphicsBeginImageContext(rect.size);
 CGContextRef context = UIGraphicsGetCurrentContext();
 CGContextSetFillColorWithColor(context, [color CGColor]);
 CGContextFillRect(context, rect);

 UIImage *theImage = UIGraphicsGetImageFromCurrentImageContext();
 UIGraphicsEndImageContext();
 return theImage;
}

```

CGRectUIImage

```

extension UIImage {
 static func gradientImageWithBounds(bounds: CGRect, colors: [CGColor]) -> UIImage {
 let gradientLayer = CAGradientLayer()
 gradientLayer.frame = bounds
 gradientLayer.colors = colors

 UIGraphicsBeginImageContext(gradientLayer.bounds.size)
 gradientLayer.render(in: UIGraphicsGetCurrentContext()!)
 let image = UIGraphicsGetImageFromCurrentImageContext()
 UIGraphicsEndImageContext()
 return image!
 }
}

```

```
let image = UIImage.gradientImageWithBounds(CGRect(x: 0, y: 0, width: 200, height: 200),
colors: [UIColor.yellowColor().CGColor, UIColor.blueColor().CGColor])
```

## Objective-C

```
+ (UIImage *)gradientImageWithBounds:(CGRect)bounds colors:(NSArray *)colors {
 CAGradientLayer *gradientLayer = [CAGradientLayer layer];
 gradientLayer.frame = bounds;
 gradientLayer.colors = colors;

 UIGraphicsBeginImageContext(gradientLayer.bounds.size);
 [gradientLayer renderInContext:UIGraphicsGetCurrentContext()];
 UIImage *image = UIGraphicsGetImageFromCurrentImageContext();
 UIGraphicsEndImageContext();
 return image;
}
```

```
+ (CALayer *)gradientBGLayerForBounds:(CGRect)bounds colors:(NSArray *)colors
{
 CAGradientLayer * gradientBG = [CAGradientLayer layer];
 gradientBG.frame = bounds;
 gradientBG.colors = colors;
 return gradientBG;
}
```

## UIImagebase64

```
//convert the image to NSData first
let imageData:NSData = UIImagePNGRepresentation(image)!
// convert the NSData to base64 encoding
let strBase64:String =
imageData.base64EncodedStringWithOptions(.Encoding64CharacterLineLength)
```

```
let dataDecoded:NSData = NSData(base64EncodedString: strBase64, options:
NSDataBase64DecodingOptions(rawValue: 0))!
let decodedimage:UIImage = UIImage(data: dataDecoded)!
```

## UIView

```
//Here self.webView is the view whose screenshot I need to take
//The screenshot is saved in jpg format in the application directory to avoid any loss of
quality in retina display devices i.e. all current devices running iOS 10
UIGraphicsBeginImageContextWithOptions(self.webView.bounds.size, NO, [UIScreen
 mainScreen].scale);
[self.webView.layer renderInContext:UIGraphicsGetCurrentContext()];
UIImage *image = UIGraphicsGetImageFromCurrentImageContext();
UIGraphicsEndImageContext();
NSString *jpgPath = [NSHomeDirectory() stringByAppendingPathComponent:@"Documents/Test.jpg"];
[UIImageJPEGRepresentation(image, 1.0) writeToFile:jpgPath atomically:YES];
UIImage *pop=[[UIImage alloc] initWithContentsOfFile:jpgPath];
//pop is the final image in jpg format and high quality with the exact resolution of the view
you selected in pixels and not just points
```

## UIColorUIImage

### UIColorUIImageUIImage

```
// *** Create an UIImage instance with RenderingMode AlwaysTemplate ***
UIImage *imgMenu = [[UIImage imageNamed:@"iconMenu"]
imageWithRenderingMode:UIImageRenderingModeAlwaysTemplate];

// *** Now Apply `tintColor` to `UIImageView` of UIImageView or UIButton and convert image in
given color ***
[btn setImage:imgMenu forState:UIControlStateNormal]; // Set UIImage in UIButton.

[button.imageView setTintColor:[UIColor blueColor]]; // It changes image color of UIButton to
blue color
```

### UIImageView

```
[imageView setImage:imgMenu]; // Assign UIImage to UIImageView
[imageView setTintColor:[UIColor greenColor]]; // Change imageview image color to green.
[imageView setTintColor:[UIColor redColor]]; // Change imageview image color to red.
```

## UIImage

### SwiftUIImage

```
extension UIImage {
 func maskWithColor(color: UIColor) -> UIImage? {

 let maskImage = self.CGImage
 let width = self.size.width
 let height = self.size.height
 let bounds = CGRectMake(0, 0, width, height)

 let colorSpace = CGColorSpaceCreateDeviceRGB()
 let bitmapInfo = CGBitmapInfo(rawValue: CGImageAlphaInfo.PremultipliedLast.rawValue)
 let bitmapContext = CGContextCreate(nil, Int(width), Int(height), 8, 0,
colorSpace, bitmapInfo.rawValue) //needs rawValue of bitmapInfo

 CGContextClipToMask(bitmapContext, bounds, maskImage)
 CGContextSetFillColorWithColor(bitmapContext, color.CGColor)
 CGContextFillRect(bitmapContext, bounds)

 //is it nil?
 if let cImage = CGContextCreateImage(bitmapContext) {
 let coloredImage = UIImage(CGImage: cImage)

 return coloredImage
 } else {
 return nil
 }
 }
}
```

### UIImage



```
my_image.maskWithColor(UIColor.blueColor())
```

UIImage <https://riptutorial.com/zh-TW/ios/topic/1409/uiimage>

# 176: UIImagePickerController

UIImagePickerController ◦ UIImagePickerControllerDelegate didFinishPickingMediaWithInfo  
UIImagePickerControllerDidCancel ◦

## Examples

### UIImagePickerController

1

```
//Swift
class ImageUploadViewController: UIViewController, UIImagePickerControllerDelegate,
 UINavigationControllerDelegate {

 let imagePickerController = UIImagePickerController()

 override func viewDidLoad() {
 super.viewDidLoad()
 imagePickerController.delegate = self
 }
}

//Objective-C
@interface ImageUploadViewController : UIViewController
<UIImagePickerControllerDelegate, UINavigationControllerDelegate> {

 UIImagePickerController *imagePickerController;
}

@end

@implementation ImageUploadViewController

- (void)viewDidLoad {

 [super viewDidLoad];

 imagePickerController.delegate = self;
}

@end
```

UINavigationControllerDelegate UIImagePickerController UINavigationController  
UINavigationController ◦ UINavigationControllerDelegate ◦

### 2 UIImagePickerController

```
//Swift
self.imagePickerController.sourceType = .Camera // options: .Camera , .PhotoLibrary ,
.SavedPhotosAlbum
```

```

self.presentViewController(self.imagePickerController, animated: true, completion: nil)

//Objective-C
imagePickerController.sourceType = UIImagePickerControllerSourceTypeCamera; // options:
UIImagePickerControllerSourceTypeCamera, UIImagePickerControllerSourceTypePhotoLibrary,
UIImagePickerControllerSourceTypeSavedPhotosAlbum
[self presentViewController:imagePickerController animated:YES completion:nil];

```

### 3

```

//Swift
func imagePickerController(picker: UIImagePickerController, didFinishPickingMediaWithInfo
info: [String : AnyObject]) {
 if let pickedImage = info[UIImagePickerControllerOriginalImage] as? UIImage {
 // You have pickedImage now, do your logic here
 }
 self.dismissViewControllerAnimated(true, completion: nil)
}

func imagePickerControllerDidCancel(picker: UIImagePickerController) {
 self.dismissViewControllerAnimated(true, completion: nil)
}

//Objective-C
- (void)imagePickerController:(UIImagePickerController *)picker
didFinishPickingMediaWithInfo:(NSDictionary *)info {

 UIImage *pickedImage = info[UIImagePickerControllerOriginalImage];

 if (pickedImage) {

 //You have pickedImage now, do your logic here

 }

 [self dismissViewControllerAnimated:YES completion:nil];
}

- (void)imagePickerControllerDidCancel:(UIImagePickerController *)picker {

 [self dismissViewControllerAnimated:YES completion:nil];
}

```

**UIImagePickerController** <https://riptutorial.com/zh-TW/ios/topic/3023/uiimagepickercontroller>

# 177: UIImageView

## Examples

### UIImageView

UIImageView UIImageView

```
//Swift
let imageView = UIImageView()

//Objective-C
UIImageView *imageView = [[UIImageView alloc] init];
```

CGRectUIImageView

```
//Swift
imageView.frame = CGRect(x: 0, y: 0, width: 200, height: 200)

//Objective-C
imageView.frame = CGRectMake(0,0,200,200);
```

```
//Swift
UIImageView(frame: CGRect(x: 0, y: 0, width: 200, height: 200))

//Objective-C
UIImageView *imageView = [[UIImageView alloc] initWithFrame:CGRectMake(0,0,200,200);

//Alternative way of defining frame for UIImageView
UIImageView *imageView = [[UIImageView alloc] init];
CGRect imageViewFrame = imageView.frame;
imageViewFrame.size.width = 200;
imageViewFrame.size.height = 200;
imageViewFrame.origin.x = 0;
imageViewFrame.origin.y = 0;
imageView.frame = imageViewFrame;
```

UIKitUIImageView ◦

### UIImageView

UIImageView image

```
//Swift
UIImageView(image: UIImage(named: "image1"))

UIImageView(image: UIImage(named: "image1"), highlightedImage: UIImage(named: "image2"))

imageView.image = UIImage(named: "image1")

//Objective-C
[[UIImageView alloc] initWithImage:[UIImage imageNamed:@"image1"]];
```

```
[[UIImageView alloc] initWithImage:[UIImage imageNamed:@"image1"] highlightedImage:[UIImage
imageNamed:@"image2"]];

imageView.image = [UIImage imageNamed:@"image1"];
```

## UIImageView

UIImageViewUIImageView

```
imageView.animationImages = [UIImage(named: "image1")!,
 UIImage(named: "image2")!,
 UIImage(named: "image3")!,
 UIImage(named: "image4")!,
 UIImage(named: "image5")!,
 UIImage(named: "image6")!,
 UIImage(named: "image7")!,
 UIImage(named: "image8")!]

imageView.animationDuration = 0.3
imageView.animationRepeatCount = 1
```

animationImagesUIImages Array ◦

animationDurationDouble◦

animationRepeatCountInt ◦

```
imageView.startAnimating()
imageView.stopAnimating()
```

isAnimating() Boolean◦

◦

UIViewUIImageView



## Objective-C

```
someImageView.layer.cornerRadius = CGRectGetHeight(someImageView.frame) / 2;
someImageView.clipsToBounds = YES;
```

```
someImageView.layer.cornerRadius = someImageView.frame.height/2
// this should alleviate the performance hit that adding transparency may cause - see
http://stackoverflow.com/a/6254531/189804
// Be sure to check scrolling performance with Instruments if you take this approach.
someImageView.layer.shouldRasterize = true
someImageView.clipsToBounds = true // All parts of the image that are outside its bounds (the
frame) are cut out (makes the rounded corners visible)
```

**autolayout**`someImageView.layer.cornerRadius``viewDidLayoutSubviews` ◦ `cornerRadius`◦

```
override func viewDidLayoutSubviews() {
 super.viewDidLayoutSubviews()
 someImageView.layer.cornerRadius = someImageView.frame.size.width/2
 someImageView.layer.masksToBounds = true
}
```

## UILabelImage

## Objective-C

```
self.maskImage.layer.mask = self.maskLabel.layer;
self.maskImage.layer.masksToBounds = YES;
```

### 3

```
maskImageView.mask = maskLabel
maskImageView.masksToBounds = true
```



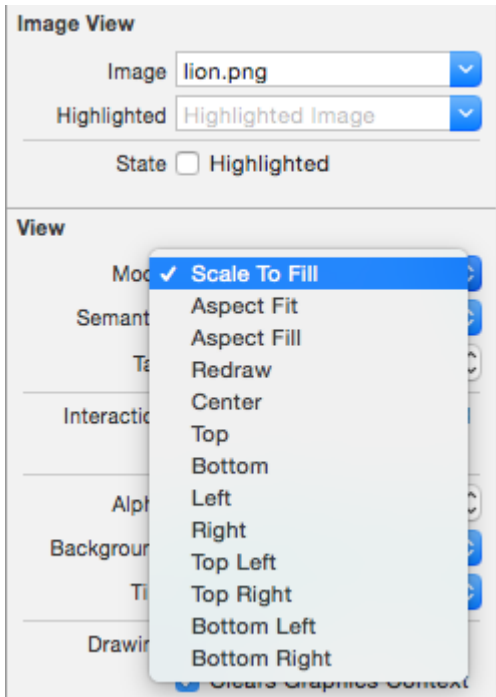
```
//Swift
imageView.tintColor = UIColor.redColor()
imageView.image = imageView.image?.imageWithRenderingMode(.AlwaysTemplate)

//Swift 3
imageView.tintColor = UIColor.red
imageView.image = imageView.image?.withRenderingMode(.alwaysTemplate)

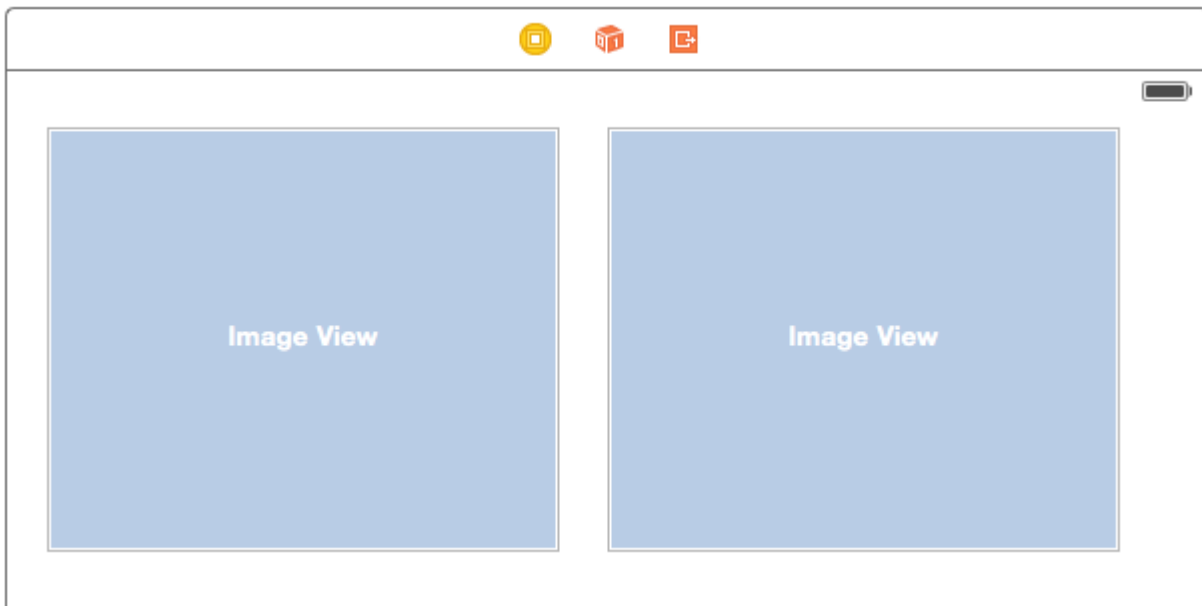
//Objective-C
imageView.tintColor = [UIColor redColor];
imageView.image = [imageView.image imageWithRenderingMode:UIImageRenderingModeAlwaysTemplate]
```

## Mode

- Interface BuilderAttributes Inspector◦



o



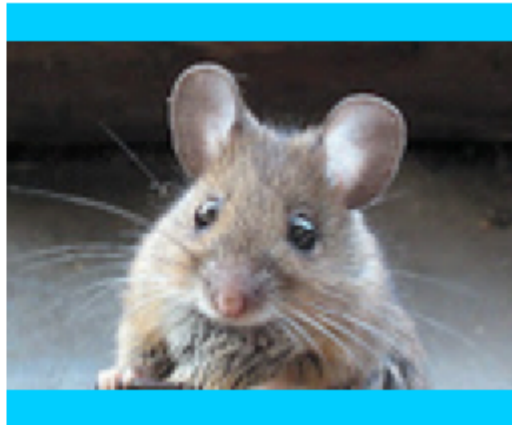




UIImageView◦

---

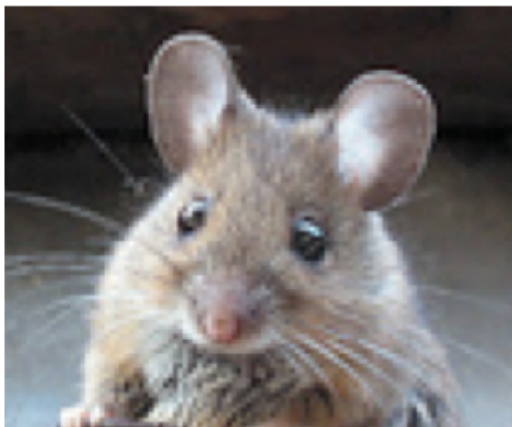
## Aspect Fit



◦ ◦ UIImageView◦

---

## Aspect



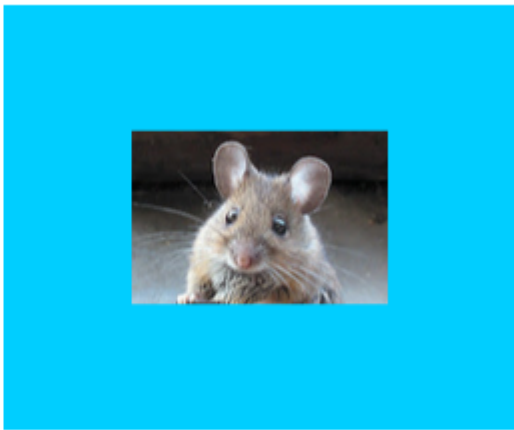
◦ “Aspect Fit”◦



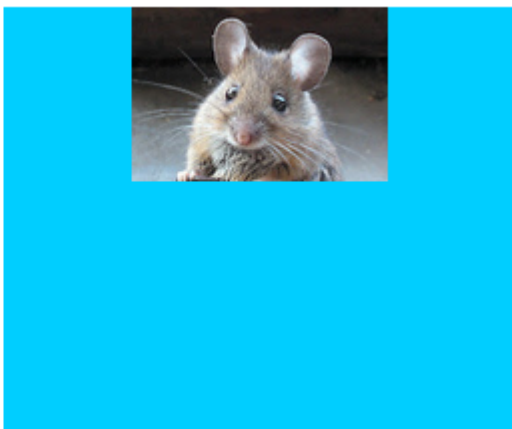
◦ Redraw ◦ UIImageViewScale to Fill◦

Redraw Apple

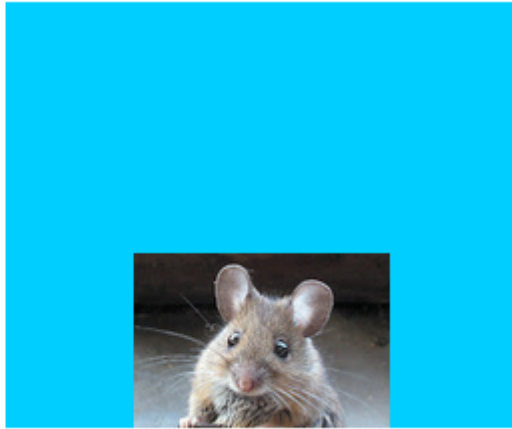
```
UIViewContentModeRedraw ◦ drawRect:◦ ◦
```



◦



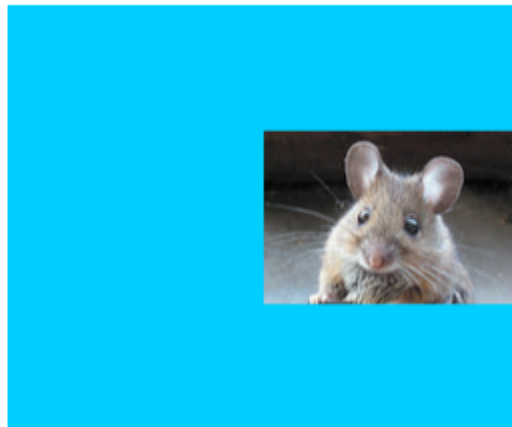
◦



o



o



o

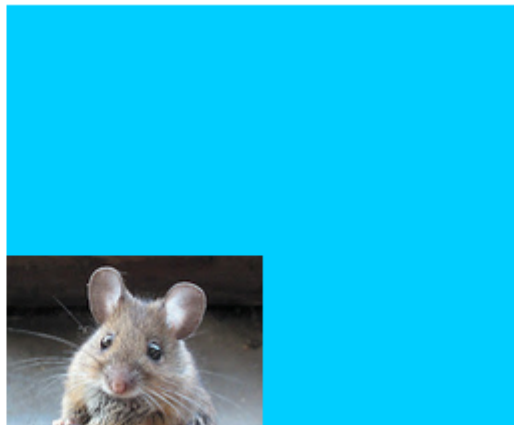




◦ ◦

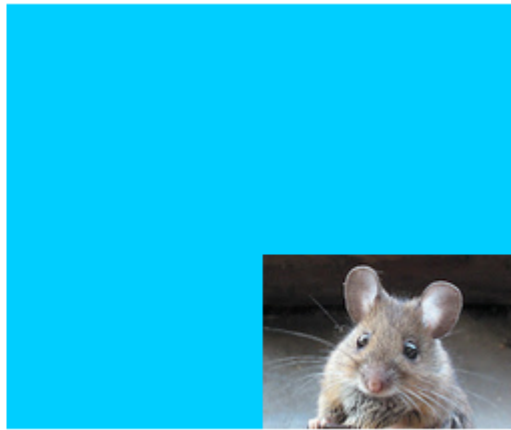


◦ ◦



◦ ◦





◦ ◦



- 
- UIImageView ◦
- UIImageView◦
- **Swift**

```
imageView.contentMode = UIViewContentMode.scaleToFill
imageView.contentMode = UIViewContentMode.scaleAspectFit
imageView.contentMode = UIViewContentMode.scaleAspectFill
imageView.contentMode = UIViewContentMode.redraw
imageView.contentMode = UIViewContentMode.center
imageView.contentMode = UIViewContentMode.top
imageView.contentMode = UIViewContentMode.bottom
imageView.contentMode = UIViewContentMode.left
imageView.contentMode = UIViewContentMode.right
imageView.contentMode = UIViewContentMode.topLeft
imageView.contentMode = UIViewContentMode.topRight
imageView.contentMode = UIViewContentMode.bottomLeft
imageView.contentMode = UIViewContentMode.bottomRight
```

**UIImageView** <https://riptutorial.com/zh-TW/ios/topic/695/uiimageview>

---

# 178: UILabel

UILabel ◦ ◦ UILabel ◦ ◦ ◦

- UILabel.numberOfLinesInt // ◦ 0
- UILabel.textString //
- UILabel.textColorUIColor //
- UILabel.tintColorUIColor //
- UILabel.attributedStringNSAttributedString //
- UILabel.fontUIFont //
- UILabel.textAlignmentNSTextAlignment //

UILabels ◦ ◦

UILabelsAttributed Strings+ ◦

UILabelUIAppearanceUIAppearanceUILabel ◦

Apple Developer

## Examples

UILabeltextUILabeltext ◦ String ◦

---

### String

```
label.text = "the new text"
```

### Objective-C

```
// Dot Notation
label.text = @"the new text";

// Message Pattern
[label setText:@"the new text"];
```

---

```
let stringVar = "basic String var"
label.text = stringVar
```

### Objective-C

```
NSString * stringVar = @"basic String var";

// Dot Notation
label.text = stringVar;
```

```
// Message Pattern
[label setText: stringWith];
```

textColor°

```
label.textColor = UIColor.redColor()
label.textColor = UIColor(red: 64.0/255.0, green: 88.0/255.0, blue: 41.0/225.0, alpha: 1)
```

### 3

```
label.textColor = UIColor.red
label.textColor = UIColor(red: 64.0/255.0, green: 88.0/255.0, blue: 41.0/225.0, alpha: 1)
```

## Objective-C

```
label.textColor = [UIColor redColor];
label.textColor = [UIColor colorWithRed:64.0f/255.0f green:88.0f/255.0f blue:41.0f/255.0f
alpha:1.0f];
```

---

[NSAttributedString](#)

## Objective-C

```
attributedString = [[NSMutableAttributedString alloc] initWithString:@"The grass is green; the
sky is blue."];
[attributedString addAttribute: NSForegroundColorAttributeName value:[UIColor greenColor]
range:NSMakeRange(13, 5)];
[attributedString addAttribute: NSForegroundColorAttributeName value:[UIColor blueColor]
range:NSMakeRange(31, 4)];
label.attributedString = attributedString;
```

```
let attributedString = NSMutableAttributedString(string: "The grass is green; the sky is
blue.")
attributedString.addAttribute(NSForegroundColorAttributeName, value: UIColor.green(), range:
NSRange(location: 13, length: 5))
attributedString.addAttribute(NSForegroundColorAttributeName, value: UIColor.blue(), range:
NSRange(location: 31, length: 4))
label.attributedString = attributedString
```

```
label.textAlignment = NSTextAlignment.left
//or the shorter
label.textAlignment = .left
```

[NSTextAlignment](#) .left .center .right .justified .natural

## Objective-C

```
label.textAlignment = NSTextAlignmentLeft;
```

[NSTextAlignment](#) NSTextAlignmentLeft NSTextAlignmentCenter NSTextAlignmentRight

NSTextAlignmentJustified NSTextAlignmentNatural

UILabelUILabel

## UILabel

---

CGRectUILabel ◦

```
let frame = CGRect(x: 0, y: 0, width: 200, height: 21)
let label = UILabel(frame: frame)
view.addSubview(label)
```

## Objective-C

---

```
CGRect frame = CGRectMake(0, 0, 200, 21);
UILabel *label = [[UILabel alloc] initWithFrame:frame];
[view addSubview:label];
```

iOSUILabel ◦

```
let label = UILabel()
label.backgroundColor = .red
label.translatesAutoresizingMaskIntoConstraints = false
view.addSubview(label)

NSLayoutConstraint.activate([
 //stick the top of the label to the top of its superview:
 label.topAnchor.constraint(equalTo: view.topAnchor)

 //stick the left of the label to the left of its superview
 //if the alphabet is left-to-right, or to the right of its
 //superview if the alphabet is right-to-left:
 label.leadingAnchor.constraint(equalTo: view.leadingAnchor)

 //stick the label's bottom to the bottom of its superview:
 label.bottomAnchor.constraint(equalTo: view.bottomAnchor)

 //the label's width should be equal to 100 points:
 label.widthAnchor.constraint(equalToConstant: 100)
])
```

## Objective-C

---

```
UILabel *label = [[UILabel alloc] init];
```

## Objective-c +VFL



```

UILabel *label = [UILabel new];
label.translatesAutoresizingMaskIntoConstraints = NO;
[self.view addSubview label];
// add horizontal constraints with 5 left and right padding from the leading and trailing

[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"V:|-5-
[labelName]-5-|"
 options:0
 metrics:nil

views:@{@"labelName":label}}];
// vertical constraints that will use the height of the superView with no padding on top and
bottom
[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"H:|[labelName]|"
 options:0
 metrics:nil

views:@{@"labelName":label}}];

```

VFL

“”。 Xcode。

## Interface Builder

Interface Builder UILabelStoryboard.xib “Label

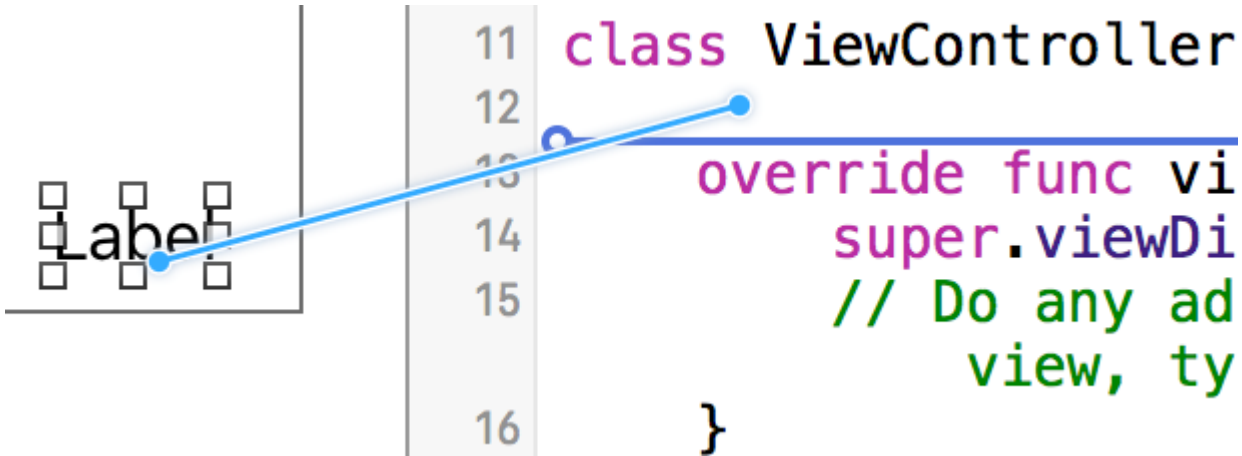


UILabel Storyboard.xib “ ”。

storyboardxibIBOutlet。

## Interface BuilderView Controller

UILabelStoryboard.xibControl ^UILabelViewController。



UILabelstrongweak ◦ strongweak

```
@IBOutlet weak var nameLabel : UILabel!
```

## Objective-C

```
@property (nonatomic, weak) IBOutlet UILabel *nameLabel;
```

```
let label = UILabel()
```

## Objective-C

```
UILabel *label = [[UILabel alloc] init];
or
UILabel *label = [UILabel new]; // convenience method for calling alloc-init
```

```
label.font = UIFont.systemFontOfSize(17)
```

## 3

```
label.font = UIFont.systemFont(ofSize: 17)
```

## Objective-C

```
label.font = [UIFont systemFontOfSize:17];
```

iOS 8.2

```
label.font = UIFont.systemFont(ofSize:17, weight: UIFontWeightBold)
```

## Swift3

```
label.font = UIFont.systemFont(ofSize: 17, weight: UIFontWeightBold)
```

## Objective-C

```
label.font = [UIFont systemFontOfSize:17 weight:UIFontWeightBold];
```

## iOS 8.2

```
label.font = UIFont.boldSystemFont(ofSize:17)
```

## Swift3

```
label.font = UIFont.boldSystemFont(ofSize: 17)
```

## Objective-C

```
label.font = [UIFont boldSystemFontOfSize:17];
```

---

—

○

◦

```
label.font = UIFont.preferredFontForTextStyle(UIFontTextStyleBody)
```

## 3

```
label.font = UIFont.preferredFont(forTextStyle: .body)
```

## Objective-C

```
label.font = [UIFont preferredFontForTextStyle:UIFontTextStyleBody];
```

---

—

```
label.font = UIFont(name: "Avenir", size: 15)
```

## Objective-C

```
label.font = [UIFont fontWithName:@"Avenir" size:15];
```

UILabel **font**

```
label.font = label.font.fontWithSize(15)
```

## 3

```
label.font = label.font.withSize(15)
```

## Objective-C

```
label.font = [label.font fontWithSize:15];
```

## Swift

... `numberOfLines` 1. UILabel bug UILabel 51,2,3,4,5

```
label.numberOfLines = 2
```

## Objective-C

```
label.numberOfLines = 2;
```

0. "Infinity"

```
label.numberOfLines = 0
```

## Objective-C

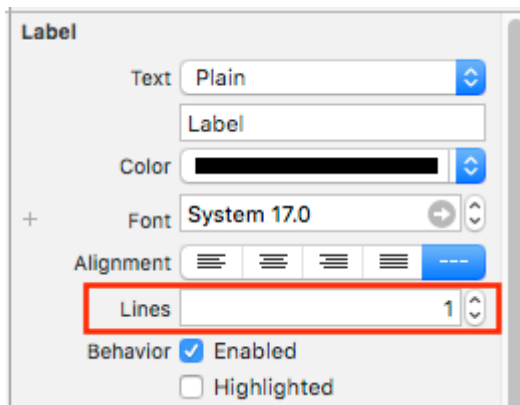
```
label.numberOfLines = 0;
```

◦ label.numberOfLines = 0◦

UIView◦ \*

## Interface Builder

numberOfLinesStoryboard.xibnumberOfLines◦ ◦



storyboardUILabel ViewController.swift / ViewController.mIBOutlet labelOne◦

viewDidLoadlabelOne.backgroundColor.textColor

sizeToFit◦

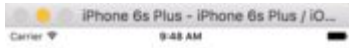
```
labelOne.backgroundColor = UIColor.blueColor()
labelOne.textColor = UIColor.whiteColor()
labelOne.text = "Hello, World!"
labelOne.sizeToFit()
```

3

```
labelOne.backgroundColor = UIColor.blue
labelOne.textColor = UIColor.white
labelOne.text = "Hello, World!"
labelOne.sizeToFit()
```

### Objective-C

```
labelOne.backgroundColor = [UIColor blueColor];
labelOne.textColor = [UIColor whiteColor];
labelOne.text = @"Hello, World!";
[labelOne sizeToFit];
```



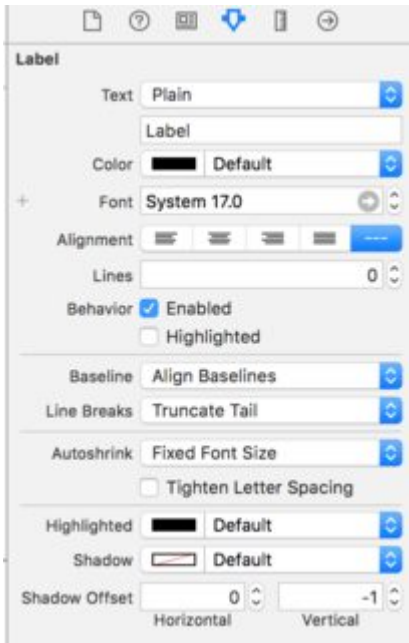
labelOne.sizeToFit()

```
labelOne.text = "Hello, World! I'm glad to be alive!"
```

labelOne



sizeToFit UILabelnumberOfLines1.



labelOne



numberOfLinesViewController

```
// Objective-C
labelOne.numberOfLines = 0;

// Swift
labelOne.numberOfLines = 0
```

```
label.backgroundColor = UIColor.redColor()

label.backgroundColor = .redColor()
```

3

```
label.backgroundColor = UIColor.red
```

## Objective-C

```
label.backgroundColor = [UIColor redColor];
```

```
label1.layer.shadowOffset = CGSize(width: 3, height: 3)
label1.layer.shadowOpacity = 0.7
label1.layer.shadowRadius = 2
```

## 3

```
label1.layer.shadowOffset = CGSize(width: 3, height: 3)
label1.layer.shadowOpacity = 0.7
label1.layer.shadowRadius = 2
```

## Objective-C

```
label1.layer.shadowOffset = CGSizeMake(3, 3);
label1.layer.shadowOpacity = 0.7;
label1.layer.shadowRadius = 2;
```

I Like My Cat

UILabel ◦

numberOfLines0.Height.GreaterThanOrEqualTo

## iOS 6

```
label.numberOfLines = 0

let heightConstraint = NSLayoutConstraint(
 item: label,
 attribute: .Height,
 relatedBy: .GreaterThanOrEqualTo,
 toItem: nil,
 attribute: .NotAnAttribute,
 multiplier: 0,
 constant: 20
)

label.addConstraint(heightConstraint)
```

## iOS 9

```
label.numberOfLines = 0
```



```
label.translatesAutoresizingMaskIntoConstraints = false
label.heightAnchor.constraintGreaterThanOrEqualToConstant(20).active = true
```

## LineBreakMode

---

```
UILabel.lineBreakMode: NSLineBreakMode
```

---

```
label.lineBreakMode = .ByTruncatingTail
```

- .ByWordWrapping
- .ByCharWrapping
- .ByClipping
- .ByTruncatingHead
- .ByTruncatingTail
- .ByTruncatingMiddle

## 3

---

```
label.lineBreakMode = .byTruncatingTail
```

- .byWordWrapping
- .byCharWrapping
- .byClipping
- .byTruncatingHead
- .byTruncatingTail
- .byTruncatingMiddle

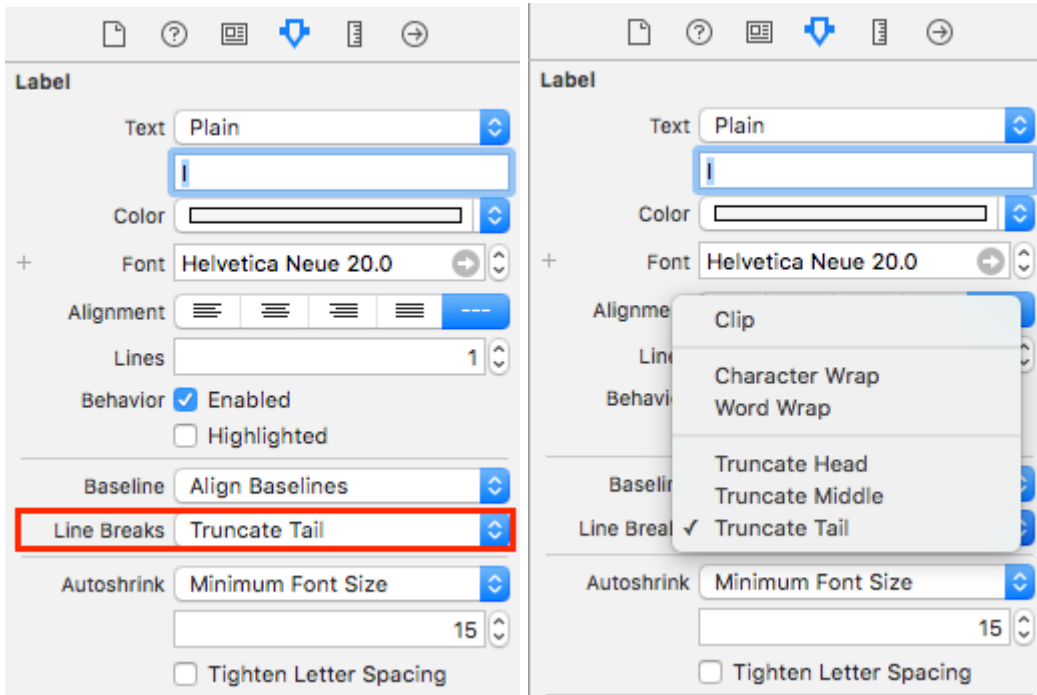
## Objective-C

---

```
[label setLineBreakMode:NSLineBreakByTruncatingTail];
```

- NSLineBreakByWordWrapping
- NSLineBreakByCharWrapping
- NSLineBreakByClipping
- NSLineBreakByTruncatingHead
- NSLineBreakByTruncatingTail
- NSLineBreakByTruncatingMiddle

## UILabel



—————

- -
- -
- -
- -
- -
- -
- -

◦ NSStringboundingRectWithSize:options:attributes:context: .

options

- NSStringDrawingUsesLineFragmentOrigin
- |NSStringDrawingTruncatesLastVisibleLine

attributesNSDictionary [Apple Docs](#)

- **NSFontAttributeName** ◦
- **NSParagraphStyleAttributeName** ◦ ◦ IB◦

contextnilNSStringDrawingContextrect◦

### C.

```

- (CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath:(NSIndexPath *)indexPath
{
 UITableViewCell *cell = [tableView cellForRowAtIndexPath:indexPath];

 NSString *labelContent = cell.textLabel.text;
 // you may choose to get the content directly from the data source if you have done

```

```

minimal customizations to the font or are comfortable with hardcoding a few values
// NSString *labelContent = [self.dataSource objectAtIndex:indexPath:indexPath];

// value may be hardcoded if retrieved from data source
NSFont *labelFont = [cell.theLabel font];

// The NSParagraphStyle, even if you did not code any changes these values may have been
altered in IB
NSMutableParagraphStyle *paragraphStyle = [NSMutableParagraphStyle new];
paragraphStyle.lineBreakMode = NSLineBreakByWordWrapping;
paragraphStyle.alignment = NSTextAlignmentCenter;

NSDictionary *attributes = @{@"NSFontAttributeName: labelFont,
 NSParagraphStyleAttributeName: paragraphStyle};

// The width is also important to the height
CGFloat labelWidth = CGRectGetWidth(cell.theLabel.frame);
// If you have been hardcoding up to this point you will be able to get this value by
subtracting the padding on left and right from tableView.bounds.size.width
// CGFloat labelWidth = CGRectGetWidth(tableView.frame) - 20.0f - 20.0f;

CGRect bodyBounds = [labelContent boundingRectWithSize:CGSizeMake(width, CGFLOAT_MAX)
options:NSStringDrawingUsesLineFragmentOrigin attributes:attributes context:nil];

return CGRectGetHeight(bodyBounds) + heightForObjectsOnTopOfLabel +
heightForObjectBelowLabel;
}

```

### Swift 3

```

override func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) ->
CGFloat {
 var cell = tableView.cellForRow(atIndexPath: indexPath)!
 var labelContent = cell.theLabel.text
 var labelFont = cell.theLabel.font
 var paragraphStyle = NSMutableParagraphStyle()

 paragraphStyle.lineBreakMode = .byWordWrapping
 paragraphStyle.alignment = .center

 var attributes = [NSFontAttributeName: labelFont, NSParagraphStyleAttributeName:
paragraphStyle]

 var labelWidth: CGFloat = cell.theLabel.frame.width

 var bodyBounds = labelContent.boundingRect(withSize: CGSize(width: width, height:
CGFLOAT_MAX), options: .usesLineFragmentOrigin, attributes: attributes, context: nil)

 return bodyBounds.height + heightForObjectsOnTopOfLabel + heightForObjectBelowLabel
}

```

```

// We calculate the height of a line by omitting the NSStringDrawingUsesLineFragmentOrigin
option, which will assume an infinitely wide label
CGRect singleLineRect = [labelContent boundingRectWithSize:CGSizeMake(CGFLOAT_MAX,
CGFLOAT_MAX)

options:NSStringDrawingTruncatesLastVisibleLine
context:nil];
CGFloat lineHeight = CGRectGetHeight(singleLineRect);

```

```

CGFloat maxHeight = lineHeight * cell.theLabel.numberOfLines;

// Now you can call the method appropriately
CGRect bodyBounds = [labelContent boundingRectWithSize:CGSizeMake(width, maxHeight)
options:(NSStringDrawingUsesLineFragmentOrigin|NSStringDrawingTruncatesLastVisibleLine)
attributes:attributes context:nil];

return CGRectGetHeight(bodyBounds) + heightForObjectsOnTopOfLabel +
heightForObjectBelowLabel;

```

UIButton UILabel ◦ UIButton ◦

- 1.
- 2.
3. UITapGestureRecognizer

UILabel ◦

```

let label = UILabel()
label.userInteractionEnabled = true

let gesture = UITapGestureRecognizer(target: self, action: #selector(labelClicked(_:)))
label.addGestureRecognizer(gesture)

```

## Objective-C

```

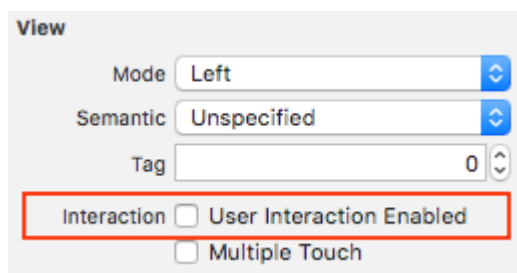
UILabel *label = [[UILabel alloc] init];
[label setUserInteractionEnabled:YES];

UITapGestureRecognizer* gesture = [[UITapGestureRecognizer alloc] initWithTarget:self
action:@selector(labelClicked:)];
[label addGestureRecognizer:gesture];

```

# storyboard “userInteractionEnabled”

UILabel



UILabel ◦ UILabel2809999.maximumLabelSize ◦

## Objective-C

```

UILabel * label = [[UILabel alloc] init];

NSString *message = @"Some dynamic text for label";

//set the text and style if any.
label.text = message;

label.numberOfLines = 0;

CGSize maximumLabelSize = CGSizeMake(280, 9999); //280:max width of label and 9999-max height
of label.

// use font information from the UILabel to calculate the size
CGSize expectedLabelSize = [label sizeThatFits:maximumLabelSize];

//Deprecated in iOS 7.0
//CGSize expectedLabelSize = [message sizeWithFont:label.font
constrainedToSize:maximumLabelSize lineBreakMode:NSLineBreakByWordWrapping];

// create a frame that is filled with the UILabel frame data
CGRect newFrame = label.frame;

// resizing the frame to calculated size
newFrame.size.height = expectedLabelSize.height;

// put calculated frame into UILabel frame
label.frame = newFrame;

```

```

var message: String = "Some dynamic text for label"
//set the text and style if any.
label.text = message
label.numberOfLines = 0
var maximumLabelSize: CGSize = CGSize(width: 280, height: 9999)
var expectedLabelSize: CGSize = label.sizeThatFits(maximumLabelSize)
// create a frame that is filled with the UILabel frame data
var newFrame: CGRect = label.frame
// resizing the frame to calculated size
newFrame.size.height = expectedLabelSize.height
// put calculated frame into UILabel frame
label.frame = newFrame

```

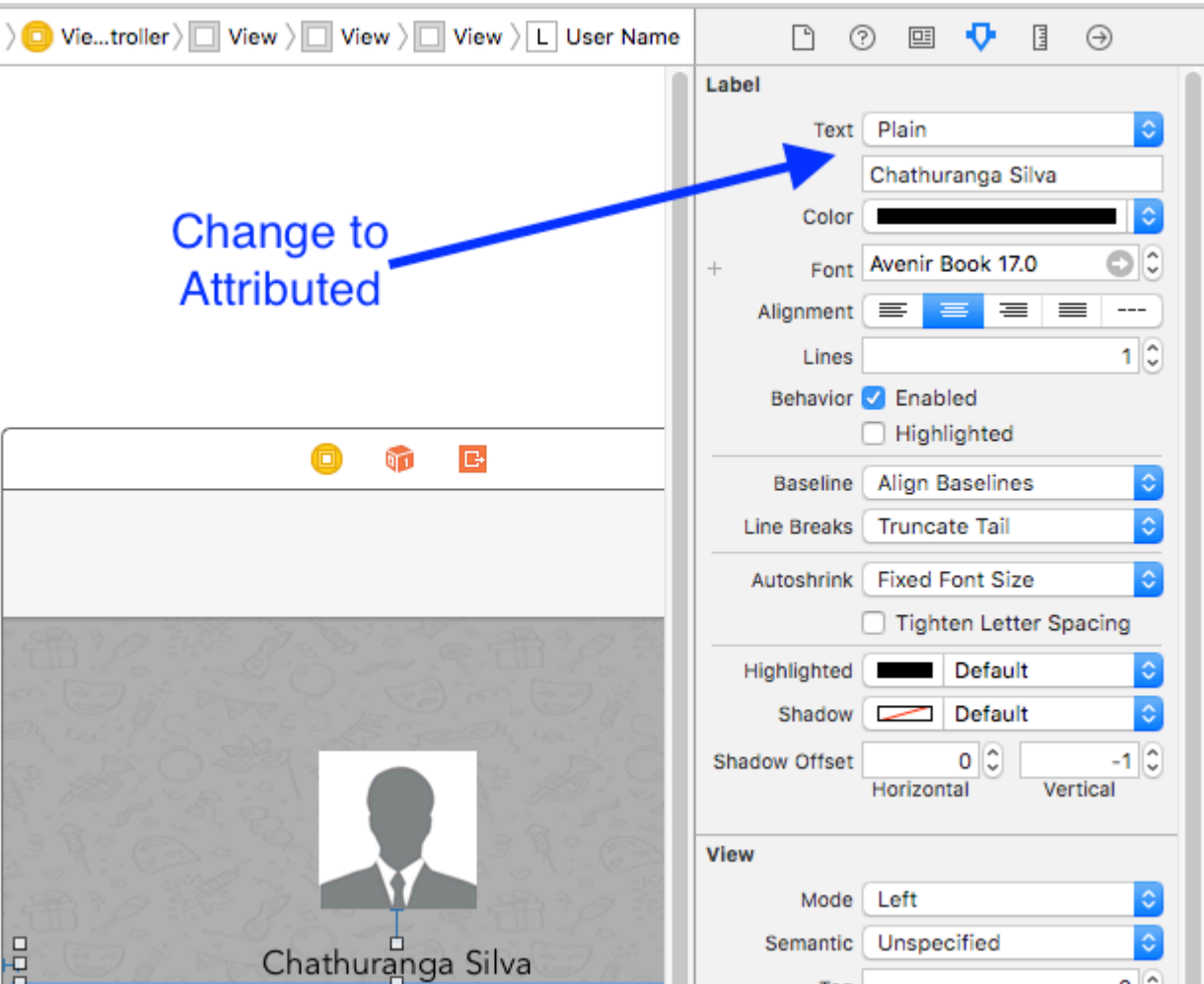
**01. - / - /**

**1**

**LabelPlainAttributed**

e) > Vie...cene > Vie...troller > View > View > View > L User Name

Change to Attributed



**Label**

Text Plain

Chaturanga Silva

Color

Font Avenir Book 17.0

Alignment

Lines 1

Behavior  Enabled  
 Highlighted

Baseline Align Baselines

Line Breaks Truncate Tail

Autoshrink Fixed Font Size

Tighten Letter Spacing

Highlighted Default

Shadow Default

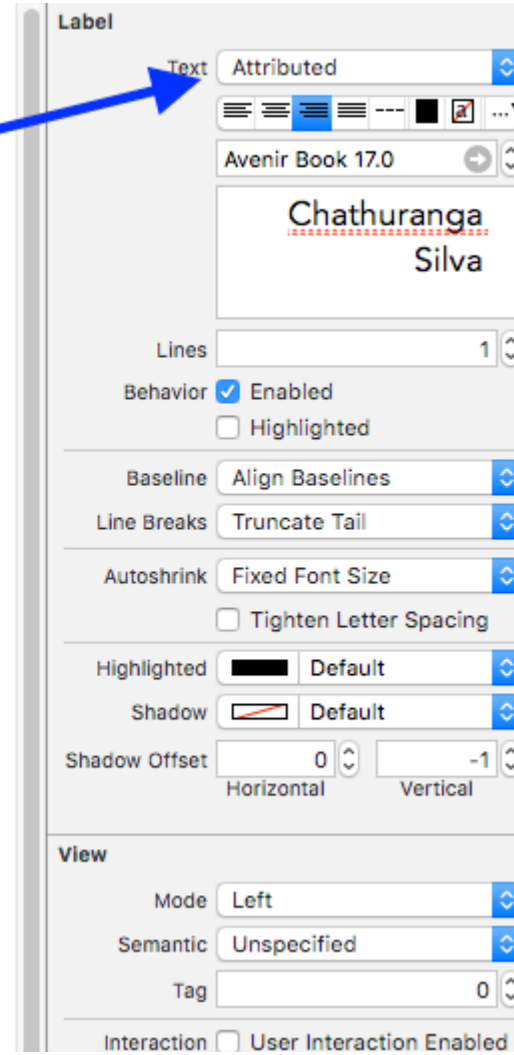
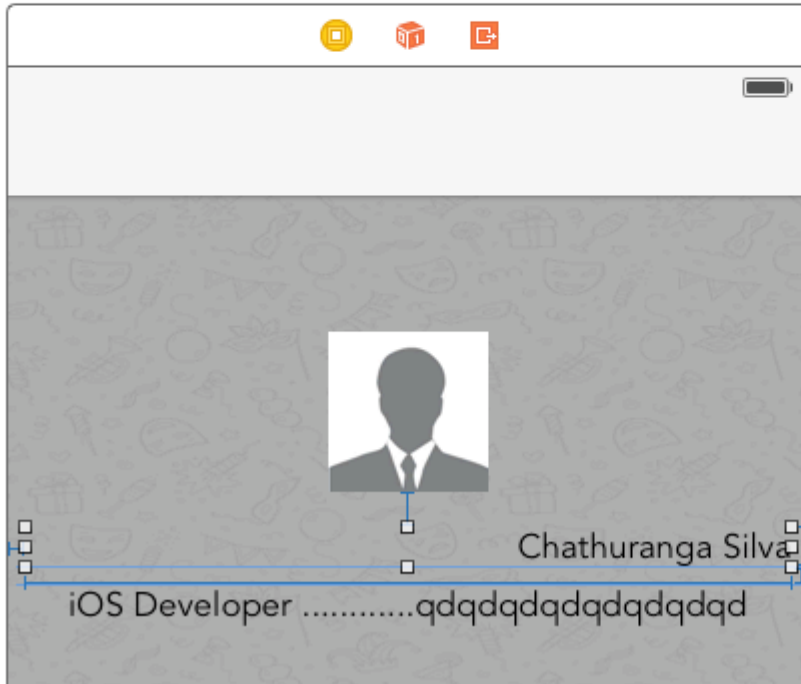
Shadow Offset 0 -1  
Horizontal Vertical

**View**

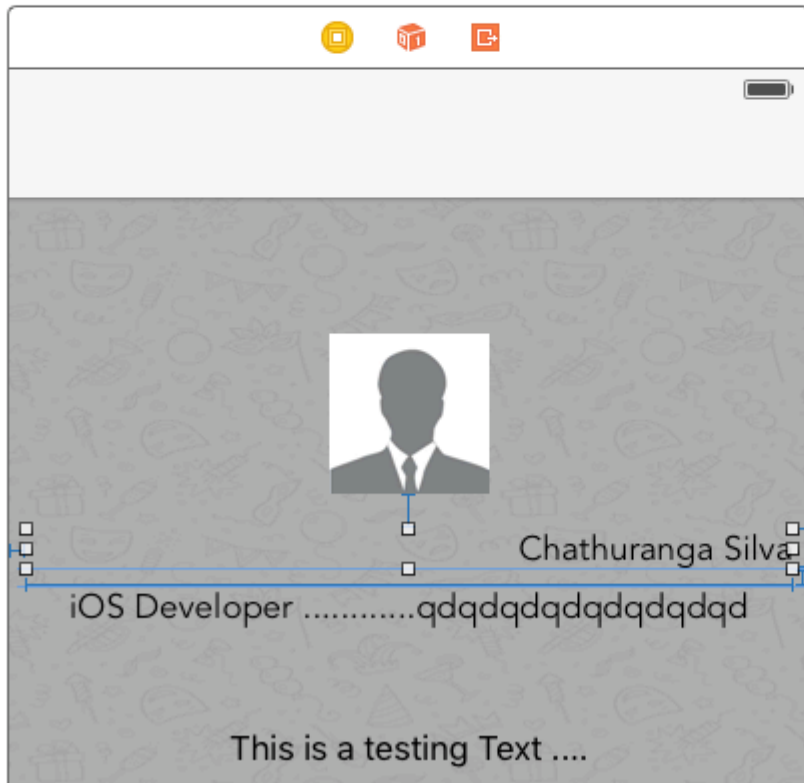
Mode Left

Semantic Unspecified

Changed  
To  
Attributed  
text



2



**Label**

Text: Attributed

Avenir Book 17.0

Chathuranga  
Silva

Lines: 1

Behavior:  Enabled  
 Highlighted

Baseline: Align Baselines

Line Breaks: Truncate Tail

Autoshrink: Fixed Font Size  
 Tighten Letter Spacing

Highlighted:  Default

Shadow:  Default

Shadow Offset: Horizontal: 0, Vertical: -1

**View**

Mode: Left

Semantic: Unspecified

Tag: 0

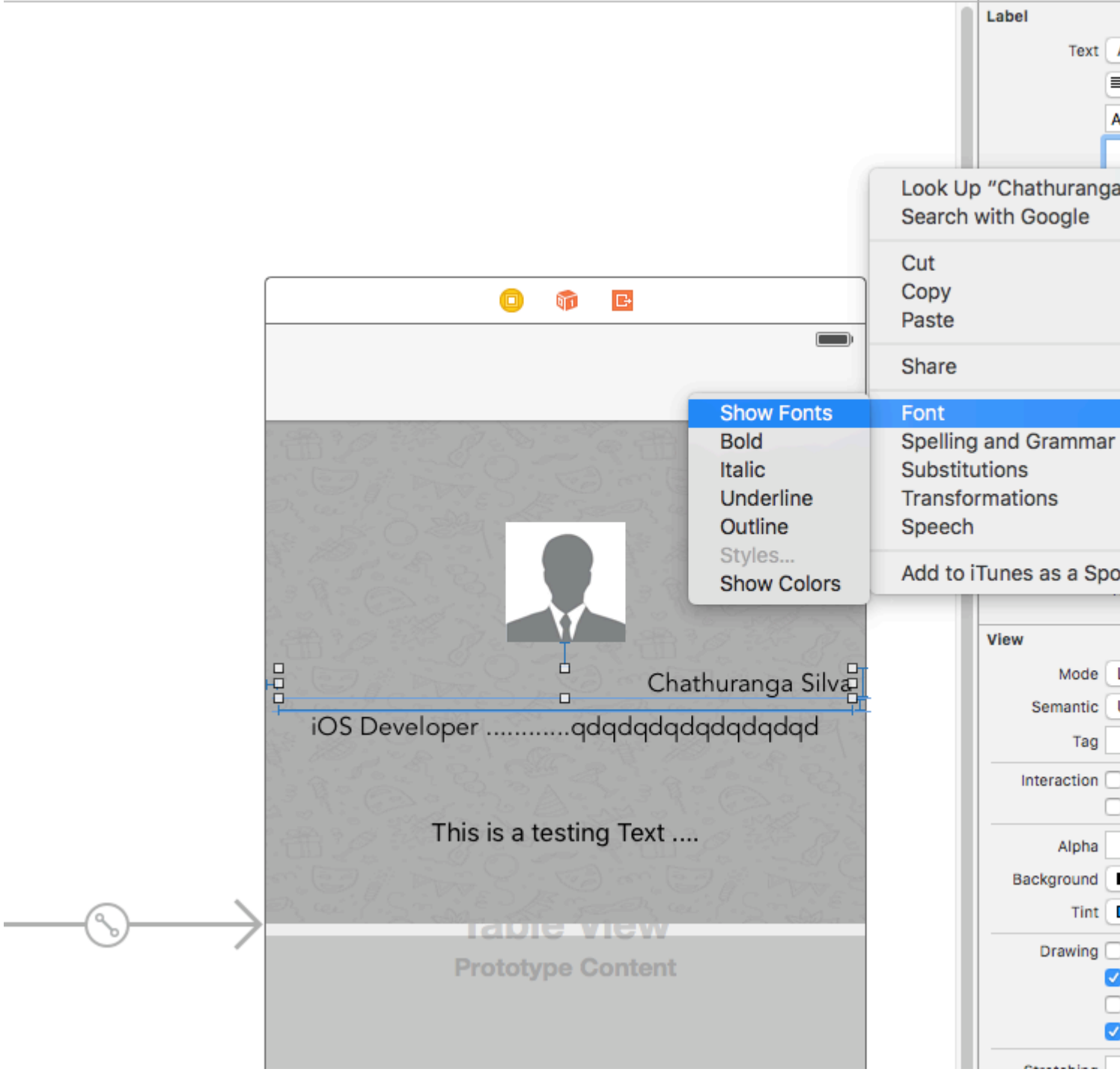
Interaction:  User Interaction Enabled  
 Multiple Touch

Alpha: 1

3

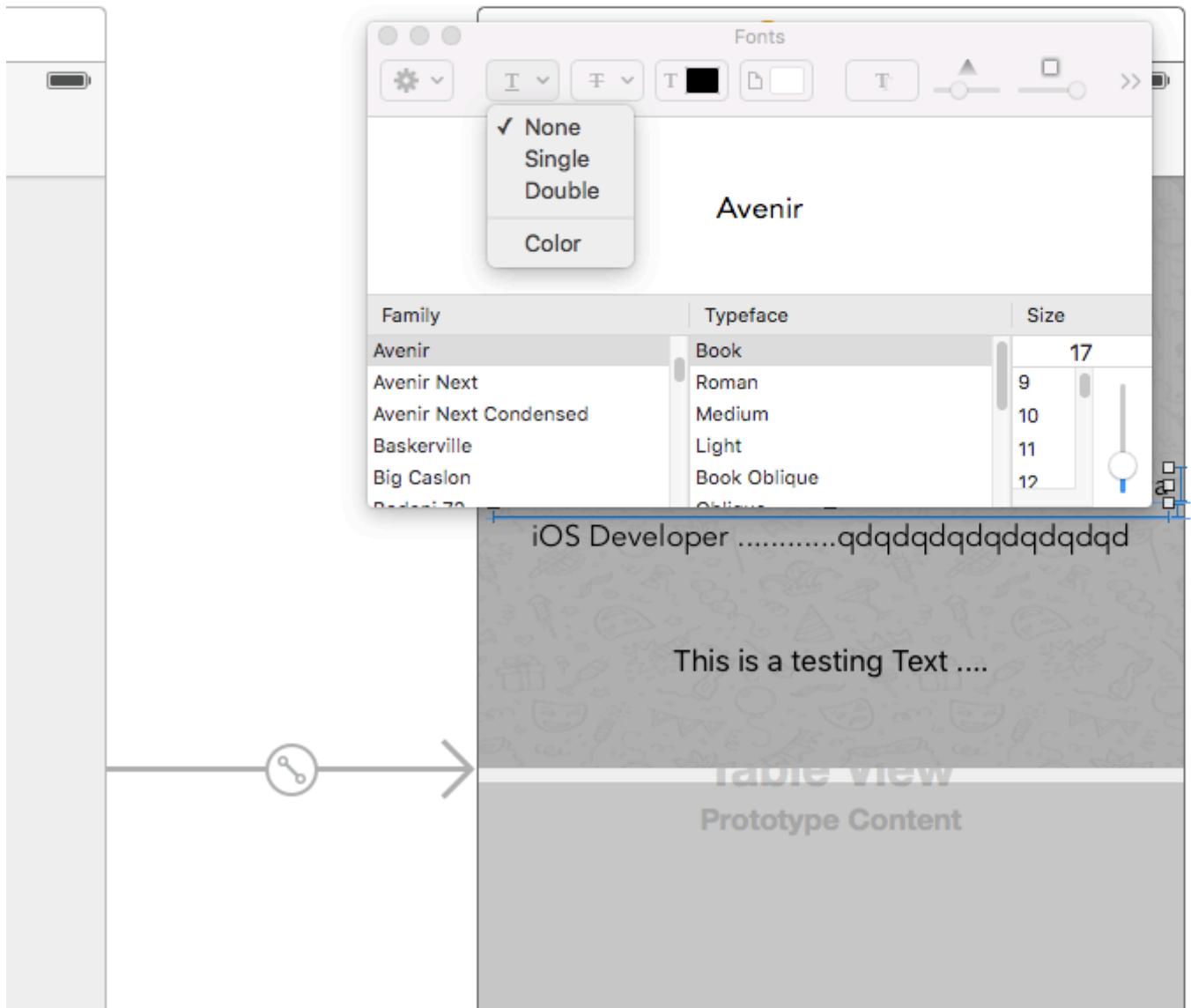
- >

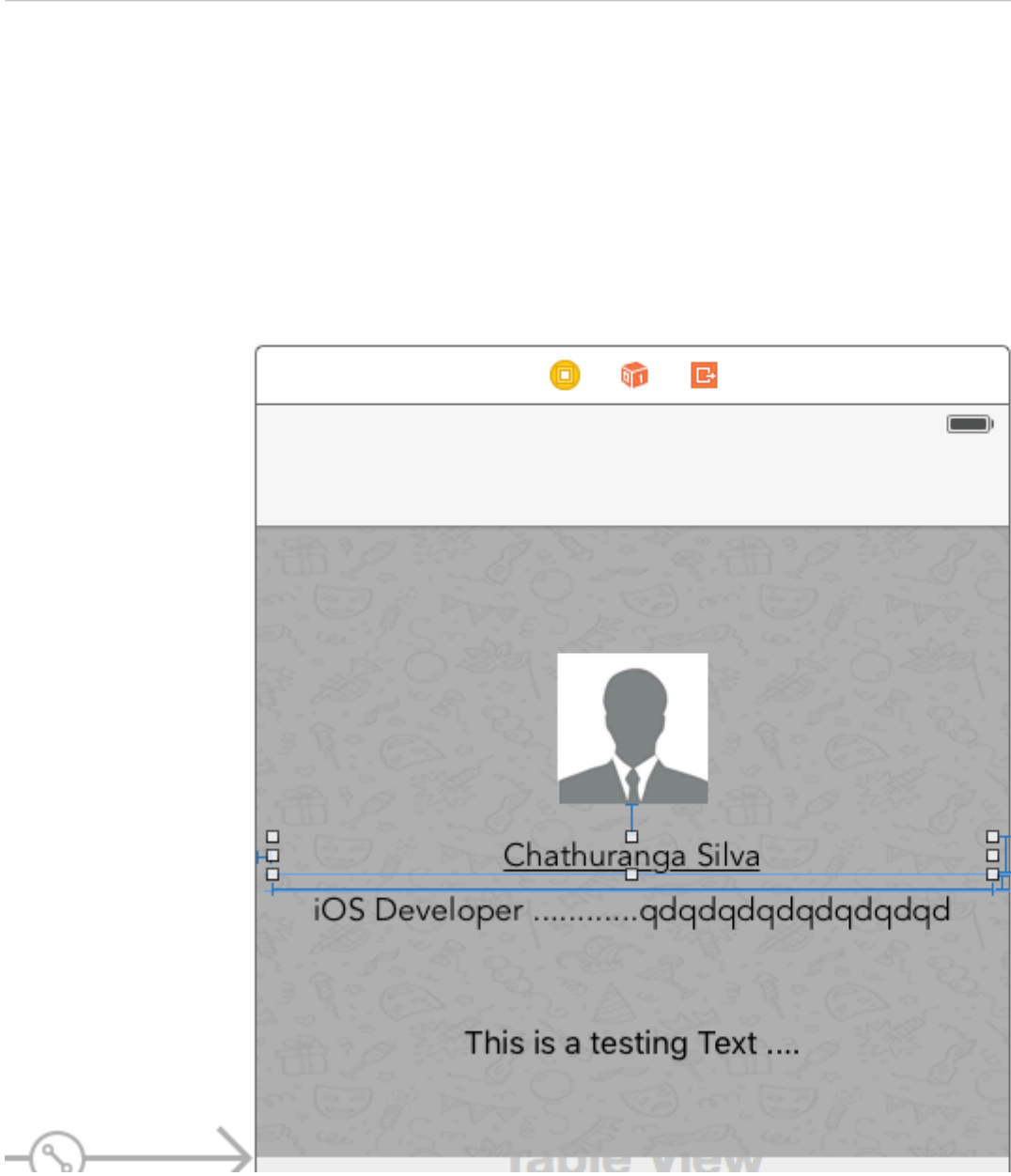




4

o o





**Label**

Text Attribute

Avenir Bo

Cha

Lines

Behavior  Enable  Highlig

Baseline Align Ba

Line Breaks Truncate

Autoshrink Fixed Fo

Tighte

Highlighted  D

Shadow  D

Shadow Offset

Horizontal

**View**

Mode Left

Semantic Unspeci

Tag

Interaction  User In  Multip

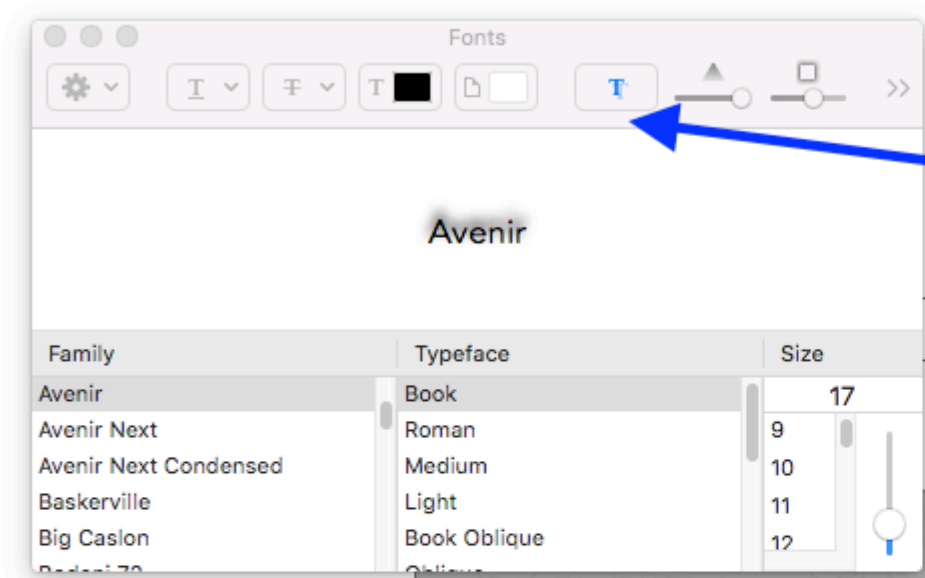
Alpha

Background  C

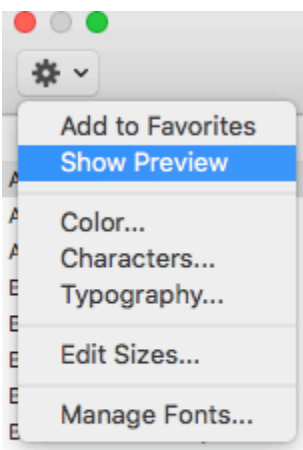
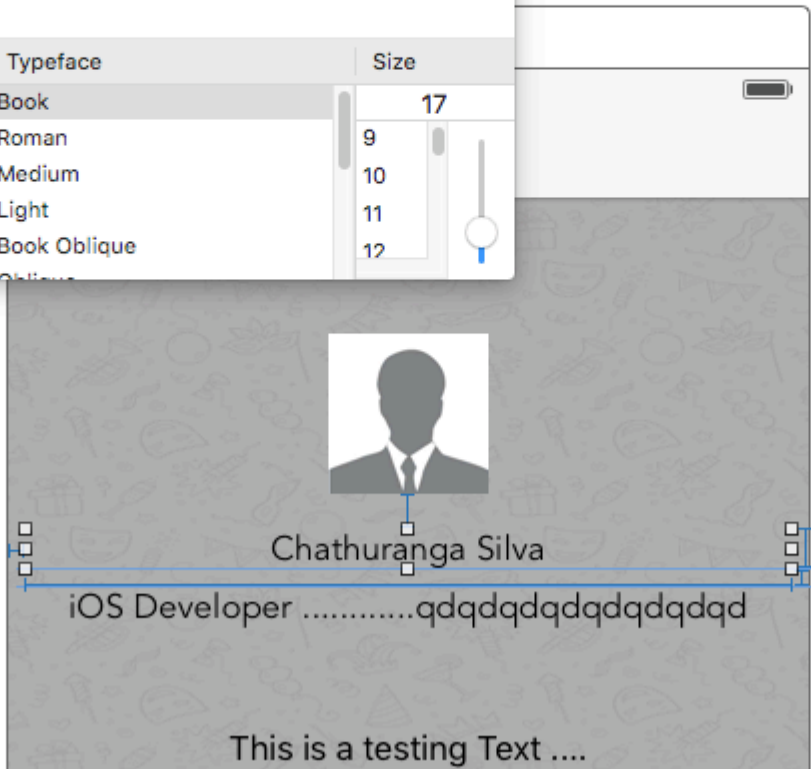
Tint  D

02./

o



Effects Button





## Avenir

```

let sampleText = "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint
occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est
laborum."

// Create label
let label = UILabel(frame: CGRectMake(0, 0, view.frame.size.width, 400))
label.numberOfLines = 0
label.lineBreakMode = NSLineBreakMode.ByWordWrapping

// Justify text through paragraph style
let paragraphStyle = NSMutableParagraphStyle()
paragraphStyle.alignment = NSTextAlignment.Justified
let attributes = [NSParagraphStyleAttributeName: paragraphStyle,
NSBaselineOffsetAttributeName: NSNumber(float: 0)]
let attributedString = NSAttributedString(string: sampleText, attributes: attributes)
label.attributedText = attributedString
view.addSubview(label)

```

## Objective-C

```

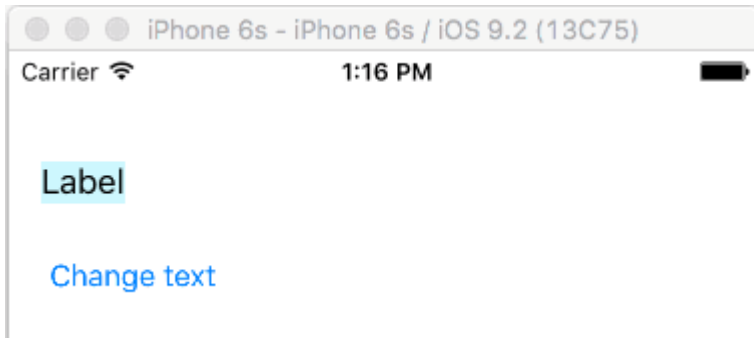
NSString *sampleText = @"Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure
dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim
id est laborum.";

// Create label
UILabel *label = [[UILabel alloc] initWithFrame:CGRectMake(0, 0,
self.view.frame.size.width, 400)];
label.numberOfLines = 0;
label.lineBreakMode = NSLineBreakByWordWrapping;

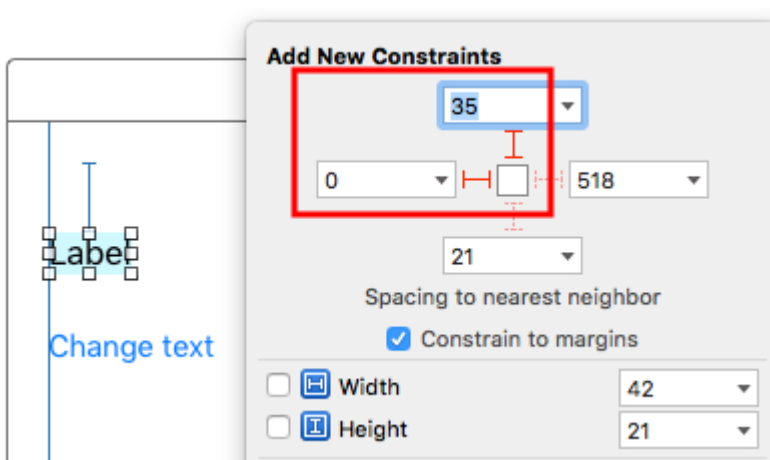
// Justify text through paragraph style
NSMutableParagraphStyle *paragraphStyle = [[NSMutableParagraphStyle alloc] init];
paragraphStyle.alignment = NSTextAlignment.Justified;
NSAttributedString *attributedString = [[NSAttributedString alloc]
initWithString:sampleText attributes:@{
 NSParagraphStyleAttributeName : paragraphStyle,
 NSBaselineOffsetAttributeName : [NSNumber numberWithInt:0]
}];
label.attributedText = attributedString;
[self.view addSubview:label];

```

◦



◦



◦



- [Stack Overflow](#) ◦
- ◦ ◦
- `sizeToFit` ◦

```
import UIKit
class ViewController: UIViewController {

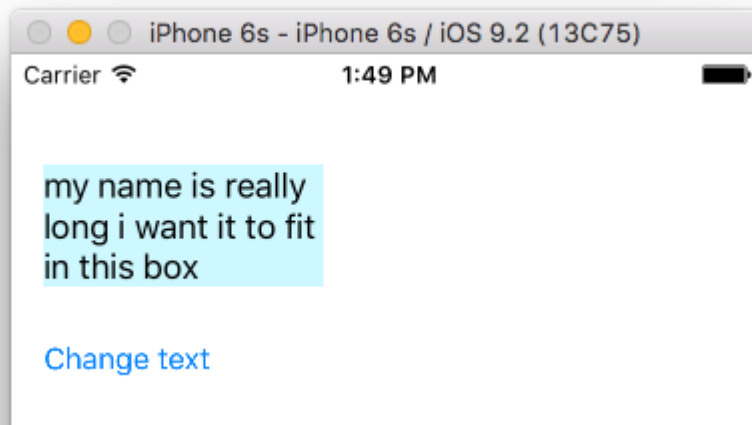
 @IBOutlet weak var myLabel: UILabel!

 @IBAction func changeTextButtonTapped(sender: UIButton) {
 myLabel.text = "my name is really long i want it to fit in this box"
 }
}
```

- ◦



- `IBOutlet myLabel.preferredMaxLayoutWidth = 150 // or whatever` ◦ ◦



## UILabel

`NSStringboundingRectWithSize UILabelCGSizeUILabel`

## Objective-C

```
[[text boundingRectWithSize: maxSize options: (NSStringDrawingTruncatesLastVisibleLine |
NSStringDrawingUsesLineFragmentOrigin) attributes: @{NSFontAttributeName: fontName}
context: nil] size];
```

```
let nsText = text as NSString?
nsText?.boundingRectWithSize(maxSize, options: [.TruncatesLastVisibleLine,
.UsesLineFragmentOrigin], attributes: [NSFontAttributeName: fontName], context: nil).size
```

◦ ◦

```
@IBOutlet var lblDescriptionHeightConstration: NSLayoutConstraint!
@IBOutlet weak var lblDescription: UILabel!

let maxWidth = UIScreen.mainScreen().bounds.size.width - 40
let sizeOfLabel = self.lblDesc.sizeThatFits(CGSize(width: maxWidth, height: CGFloat.max))
self.lblDescriptionHeightConstration.constant = sizeOfLabel.height
```

“40”。

## Objective-C

```
UILabel *label = [[UILabel alloc] init];
label.highlighted = YES;
label.highlightedTextColor = [UIColor redColor];
```

```
let label = UILabel()
label.highlighted = true
label.highlightedTextColor = UIColor.redColor()
```

### 3

```
let label = UILabel()
label.isHighlighted = true
label.highlightedTextColor = UIColor.red
```

**UILabel** <https://riptutorial.com/zh-TW/ios/topic/246/UILabel>



# 179: UITabBarController


## Examples

iOS。

Controller。

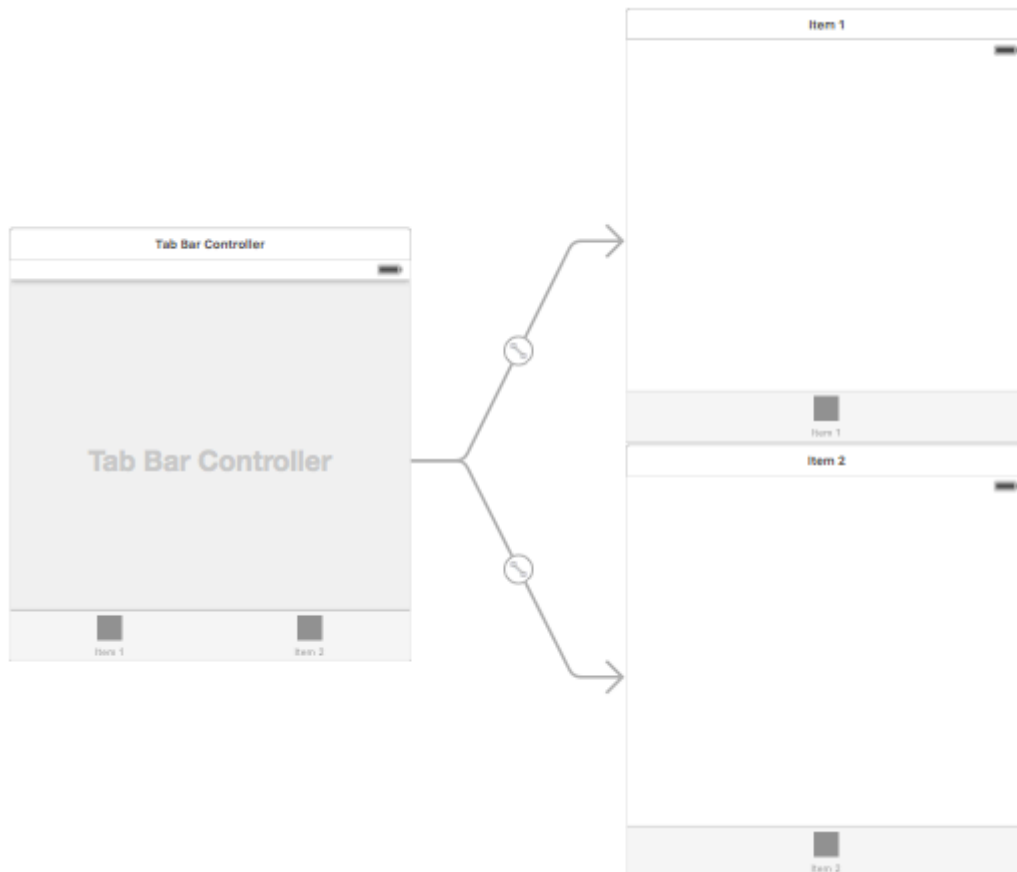
---

📄 {} 🔍 📄



**Tab Bar Controller** - A controller that manages a set of view controllers that represent tab bar items.

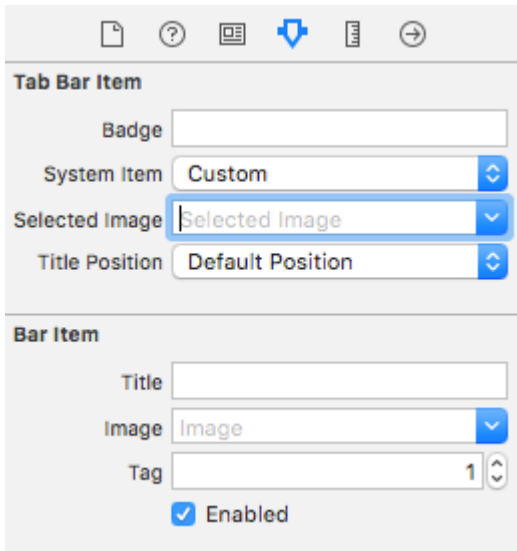
◦ segue-down。



。

assets。

。



```
viewDidLoad()
```

## Objective-C

```
self.title = @"item";

self.tabBarItem.image = [UIImage imageNamed:@"item"];
self.tabBarItem.selectedImage = [UIImage imageNamed:@"item_selected"];
```

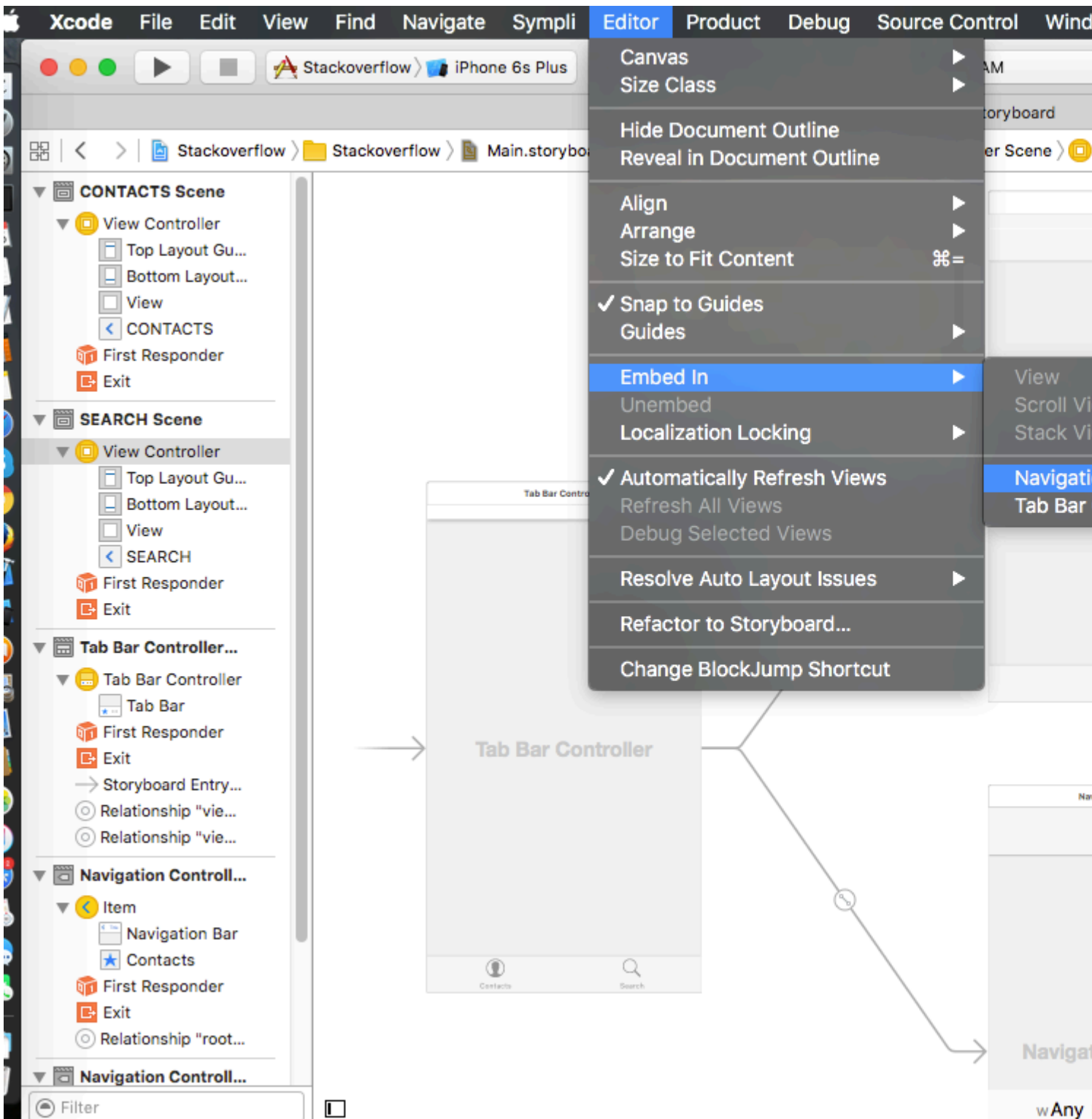
```
self.title = "item"
self.tabBarItem.image = UIImage(named: "item")
self.tabBarItem.selectedImage = UIImage(named: "item_selected")
```

## TabBar

◦ ◦

### Tab Bar ControllerView Controller

- ◦ ◦
- **Xcode Editor Embed In - > Navigation Controller**



```

[[UITabBar appearance] setTintColor:[UIColor whiteColor]];
[[UITabBar appearance] setBarTintColor:[UIColor tabBarBackgroundColor]];
[[UITabBar appearance] setBackgroundColor:[UIColor tabBarInactiveColor]];
[[UINavigationBar appearance] setBarTintColor:[UIColor appBlueColor]];
[[UINavigationBar appearance] setTintColor:[UIColor whiteColor]];
[[UINavigationBar appearance] setBarStyle:UIBarStyleBlack];

```

## UITabBarController

Swift 3UITabBarController.

```

import UIKit

class TabBarController: UITabBarController {

 override func viewDidLoad() {
 super.viewDidLoad()

 self.navigationController?.isNavigationBarHidden = true

 UITabBar.appearance().tintColor = UIColor.purple

 // set red as selected background color
 let numberOfItems = CGFloat(tabBar.items!.count)
 let tabBarItemSize = CGSize(width: tabBar.frame.width / numberOfItems, height:
tabBar.frame.height)
 tabBar.selectionIndicatorImage =
UIImage.imageWithColor(UIColor.lightText.withAlphaComponent(0.5), size:
tabBarItemSize).resizableImage(withCapInsets: UIEdgeInsets.zero)

 // remove default border
 tabBar.frame.size.width = self.view.frame.width + 4
 tabBar.frame.origin.x = -2
 }

 override func viewWillAppear(_ animated: Bool) {
 // For Images
 let firstViewController:UIViewController = NotificationVC()
 // The following statement is what you need
 let customTabBarItem:UITabBarItem = UITabBarItem(title: nil, image: UIImage(named:
"notification@2x")?.withRenderingMode(UIImageRenderingMode.alwaysOriginal), selectedImage:
UIImage(named: "notification_sel@2x"))
 firstViewController.tabBarItem = customTabBarItem

 for item in self.tabBar.items! {
 let unselectedItem = [NSForegroundColorAttributeName: UIColor.white]
 let selectedItem = [NSForegroundColorAttributeName: UIColor.purple]

 item.setTitleTextAttributes(unselectedItem, for: .normal)
 item.setTitleTextAttributes(selectedItem, for: .selected)
 }
 }
}

extension UIImage {
 class func imageWithColor(_ color: UIColor, size: CGSize) -> UIImage {
 let rect: CGRect = CGRect(origin: CGPoint(x: 0,y :0), size: CGSize(width: size.width,
height: size.height))
 UIGraphicsBeginImageContextWithOptions(size, false, 0)
 color.setFill()
 UIRectFill(rect)
 let image: UIImage = UIGraphicsGetImageFromCurrentImageContext()!
 UIGraphicsEndImageContext()
 return image
 }
}
}

```

Tab Bar Item Hide

Badge

System Item

Selected Image

Title Position

---

Bar Item

Title

Image

Tag

Enabled





## Tab BarStoryboard

```
class AppDelegate: UIResponder, UIApplicationDelegate {

 var window: UIWindow?

 var firstTabNavigationController : UINavigationController!
 var secondTabNavigationControoller : UINavigationController!
 var thirdTabNavigationController : UINavigationController!
 var fourthTabNavigationControoller : UINavigationController!
 var fifthTabNavigationController : UINavigationController!

 func application(_ application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
 // Override point for customization after application launch.
 Fabric.with([Crashlytics.self])

 window = UIWindow(frame: UIScreen.main.bounds)

 window?.backgroundColor = UIColor.black

 let tabBarController = UITabBarController()

 firstTabNavigationController = UINavigationController.init(rootViewController:
FirstViewController())
 secondTabNavigationControoller = UINavigationController.init(rootViewController:
SecondViewController())
 thirdTabNavigationController = UINavigationController.init(rootViewController:
ThirdViewController())
 fourthTabNavigationControoller = UINavigationController.init(rootViewController:
FourthViewController())
 fifthTabNavigationController = UINavigationController.init(rootViewController:
FifthViewController())

 tabBarController.viewControllers = [firstTabNavigationController,
```

```

secondTabNavigationControoller, thirdTabNavigationController, fourthTabNavigationControoller,
fifthTabNavigationController]

 let item1 = UITabBarItem(title: "Home", image: UIImage(named: "ico-home"), tag: 0)
 let item2 = UITabBarItem(title: "Contest", image: UIImage(named: "ico-contest"), tag:
1)
 let item3 = UITabBarItem(title: "Post a Picture", image: UIImage(named: "ico-photo"),
tag: 2)
 let item4 = UITabBarItem(title: "Prizes", image: UIImage(named: "ico-prizes"), tag:
3)
 let item5 = UITabBarItem(title: "Profile", image: UIImage(named: "ico-profile"), tag:
4)

 firstTabNavigationController.tabBarItem = item1
 secondTabNavigationControoller.tabBarItem = item2
 thirdTabNavigationController.tabBarItem = item3
 fourthTabNavigationControoller.tabBarItem = item4
 fifthTabNavigationController.tabBarItem = item5

 UITabBar.appearance().tintColor = UIColor(red: 0/255.0, green: 146/255.0, blue:
248/255.0, alpha: 1.0)

 self.window?.rootViewController = tabBarController

 window?.makeKeyAndVisible()

 return true
}

```

**UITabBarController** <https://riptutorial.com/zh-TW/ios/topic/2763/uitabBarController>

---

# 180: UITableViewCell

xibnib

## Examples

### UITableViewCellXib

UITableViewCell°

### UITableViewCell + RRCell.h

```
#import <UIKit/UIKit.h>

@interface UITableViewCell (RRCell)

-(id)initWithOwner:(id)owner;

@end
```

### UITableViewCell + RRCell.m

```
#import "UITableViewCell+RRCell.h"

@implementation UITableViewCell (RRCell)

#pragma clang diagnostic push
#pragma clang diagnostic ignored "-Wobjc-designated-initializers"

-(id)initWithOwner:(id)owner {

 if (self = [super init]) {

 NSArray *nib = [[NSBundle mainBundle]loadNibNamed:NSStringFromClass([self class])
owner:self options:nil];
 self = [nib objectAtIndex:0];
 }
 return self;
}

#pragma clang diagnostic pop

@end
```

cellForRowAtIndexPath

```
-(UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
 //Creted custom cell xib file to load by cell category class
 CustomCell *cell = [[CustomCell alloc]initWithOwner:self];
}
```



```
return cell;
}
```

[UITableViewCell](https://riptutorial.com/zh-TW/ios/topic/10101/uitableviewcell) <https://riptutorial.com/zh-TW/ios/topic/10101/uitableviewcell>

---

# 181: UITableViewController

UITableViewController。 UITableViewControllerUITextfield。

## Examples

tableViewCellStyleTableView。

```
override func numberOfSections(in tableView: UITableView) -> Int {
 // You need to return minimum one to show the cell inside the tableView
 return 1
}

override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
 // return the number of rows inside the tableView.
 return 3
}

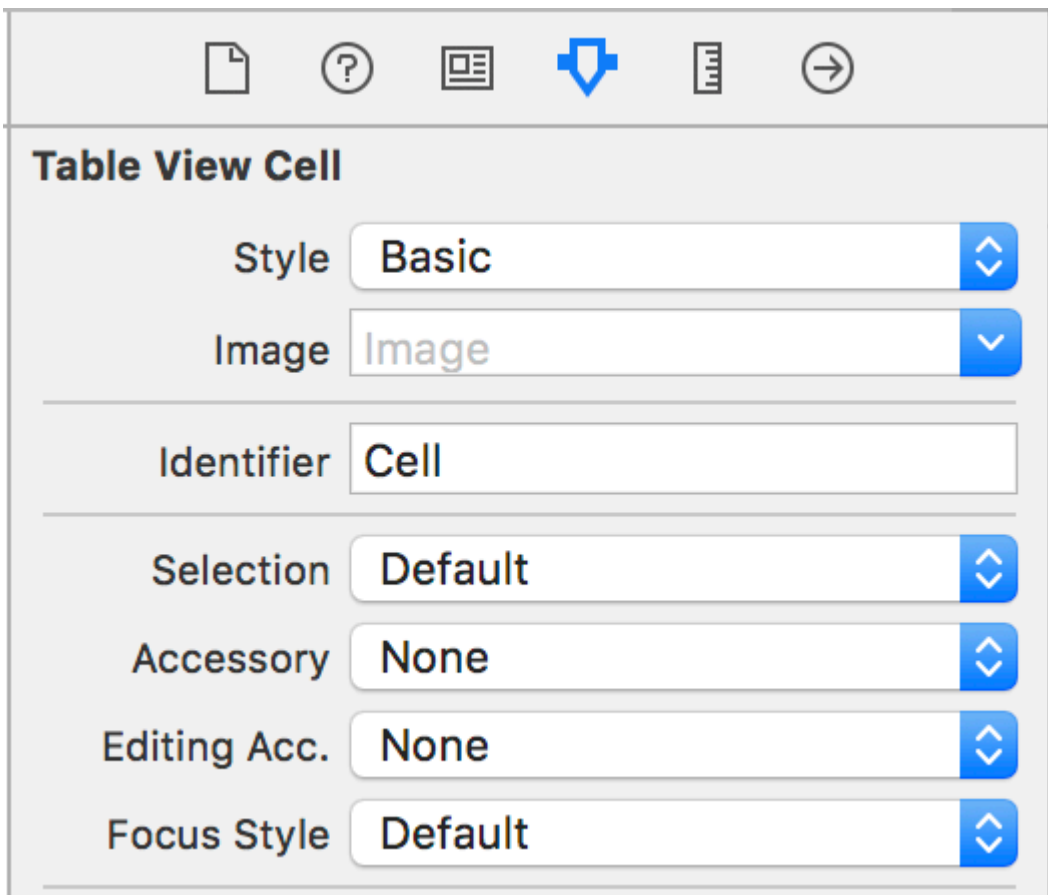
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
UITableViewCell {

 let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)
 // identifier string should be same as what you have entered in the cell Attribute inspector -
 > identifier (see the image).

 // Configure the cell...
 cell.textLabel?.text = "Cell \(indexPath.row) :" + "Hello"
 //cell have different style Custom, basic, right detail, left detail, subtitle.
 //For custom you can use your own objects and constrains, for other styles all
 //is ready just select according to your design. (see the image for changing the style)

 return cell
}

override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
 // this delegate method will trigger when you click a cell
}
```



## TableView

tableViewUITableViewController◦

```
class TableViewCell: UITableViewController {

 @IBOutlet weak var lblTitle: UILabel!

 override func awakeFromNib() {
 super.awakeFromNib()
 // Initialization code
 }

 override func setSelected(_ selected: Bool, animated: Bool) {
 super.setSelected(selected, animated: animated)

 // Configure the view for the selected state
 }

}
```

tableView

```
override func numberOfSections(in tableView: UITableView) -> Int {
 // You need to return minimum one to show the cell inside the tableView
 return 1
}

override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
```

```
// return the number of rows inside the tableview.
return 3
}

override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
UITableViewCell {

 let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath) as!
 TableViewCell
 // identifier string should be same as what you have entered in the cell Attribute
 inspector -> identifier.

 // Configure the cell...
 cell.lblTitle.text = "Cell \(indexPath.row) :" + "Hello"

 return cell
}

override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
 // this delegate method will trigger when you click a cell
}
```

**UITableViewController** <https://riptutorial.com/zh-TW/ios/topic/10953/uitableviewController>

# 182: UITextField

## UITextFieldUIKit

- UITextField.textString //。
- UITextField.attributedStringNSAttributedString //。
- UITextField.textColorUIColor //
- UITextField.fontUIFont //
- UITextField.textAlignmentNSTextAlignment //NSLeftTextAlignment
- UITextField.borderStyleUITextBorderStyle //UITextBorderStyleNone。  
UITextBorderStyleRoundedRect。
- UITextField.placeholderString //nil。 70
- UITextField.attributedStringNSAttributedString //
- UITextField.clearsOnBeginEditingBool //NO。
- UITextField.adjustsFontSizeToFitWidthBool //NO。 minFontSize
- UITextField.minimumFontSizeCGFloat //0.0。 。 adjustsFontSizeToFitWidthYES
- UITextField.delegateUITextFieldDelegate //nil。
- UITextField.clearButtonModeUITextFieldViewMode //。 UITextFieldViewModeNever
- UITextField.leftViewUIView //
- UITextField.leftViewModeUITextFieldViewMode //。 UITextFieldViewModeNever
- UITextField.rightViewUIView //
- UITextField.rightViewModeUITextFieldViewMode //。 UITextFieldViewModeNever
- UITextField.inputViewUIView //。 nil。 reloadInputViews。
- UITextField.inputAccessoryViewUIView
- UITextField.isSecureTextEntryBool //

## Examples

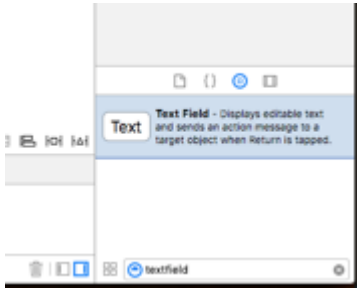
```
let frame = CGRect(x: 0, y: 0, width: 100, height: 100)
let textField = UITextField(frame: frame)
```

## Objective-C

```
CGRect *frame = CGRectMake(0, 0, 100, 100);
UITextField *textField = [[UITextField alloc] initWithFrame:frame];
```

## Interface Builder

UITextField。



◦ `////`◦

```
let textField = UITextField() // initialized however

let toolbar = UIToolbar(frame: CGRect(x: 0, y: 0, width: view.frame.size.width, height: 0)

let flexibleSpace = UIBarButtonItem(barButtonItemSystemItem: .FlexibleSpace, target: nil, action:
nil)

let doneButton = UIBarButtonItem(barButtonItemSystemItem: .Done, target: self, action:
Selector("done"))

let items = [flexibleSpace, doneButton] // pushes done button to right side

toolbar.setItems(items, animated: false) // or toolbar.items = ...
toolbar.sizeToFit()

textField.inputAccessoryView = toolbar
```

## Objective-C

```
UITextField *textField = [[UITextField alloc] init];

UIToolbar *toolbar = [[UIToolbar alloc] initWithFrame:CGRectMake(0, 0,
self.view.frame.size.width, 0)];

UIBarButtonItem *flexibleSpace = [[UIBarButtonItem alloc]
initWithBarButtonSystemItem:UIBarButtonSystemItemFlexibleSpace target:nil action:nil];
UIBarButtonItem *doneButton = [[UIBarButtonItem alloc]
initWithBarButtonSystemItem:UIBarButtonSystemItemDone target:self action:@selector(done)];
NSArray *items = @[
 flexibleSpace,
 doneButton
];

[toolbar setItems:items];
[toolbar sizeToFit];

textField.inputAccessoryView = toolbar;
```

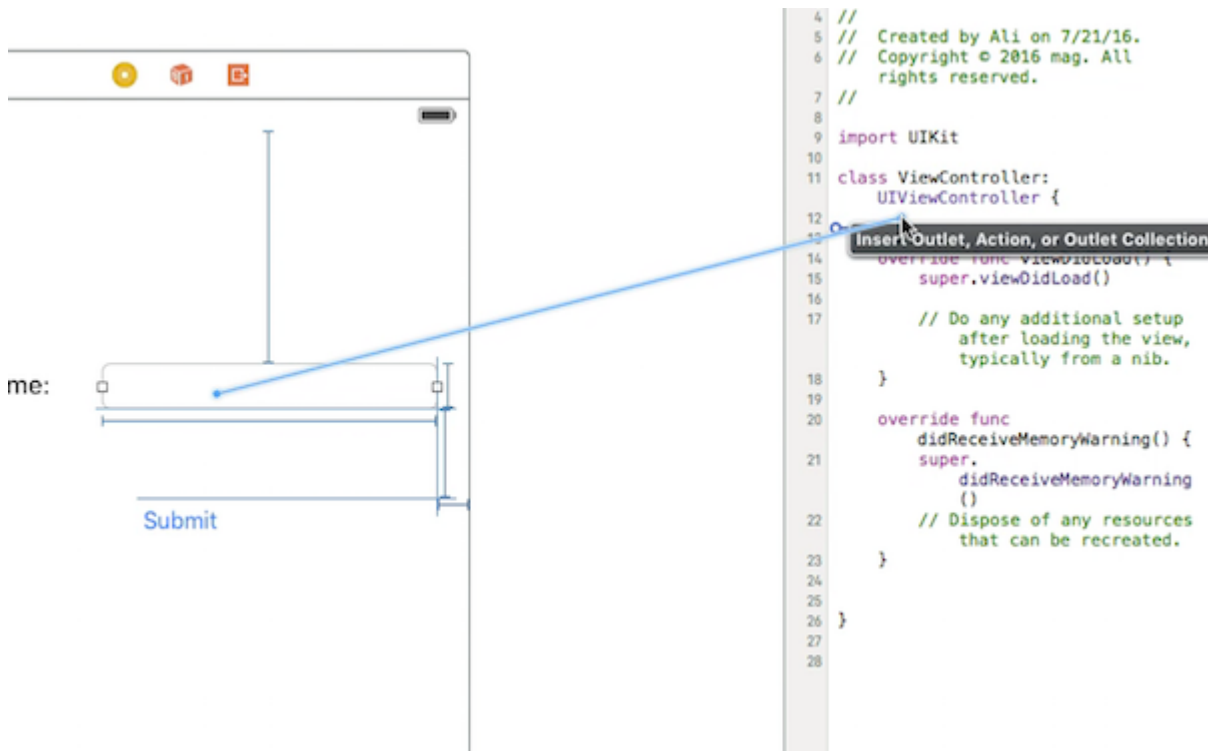
```
textField.autocapitalizationType = .None
```

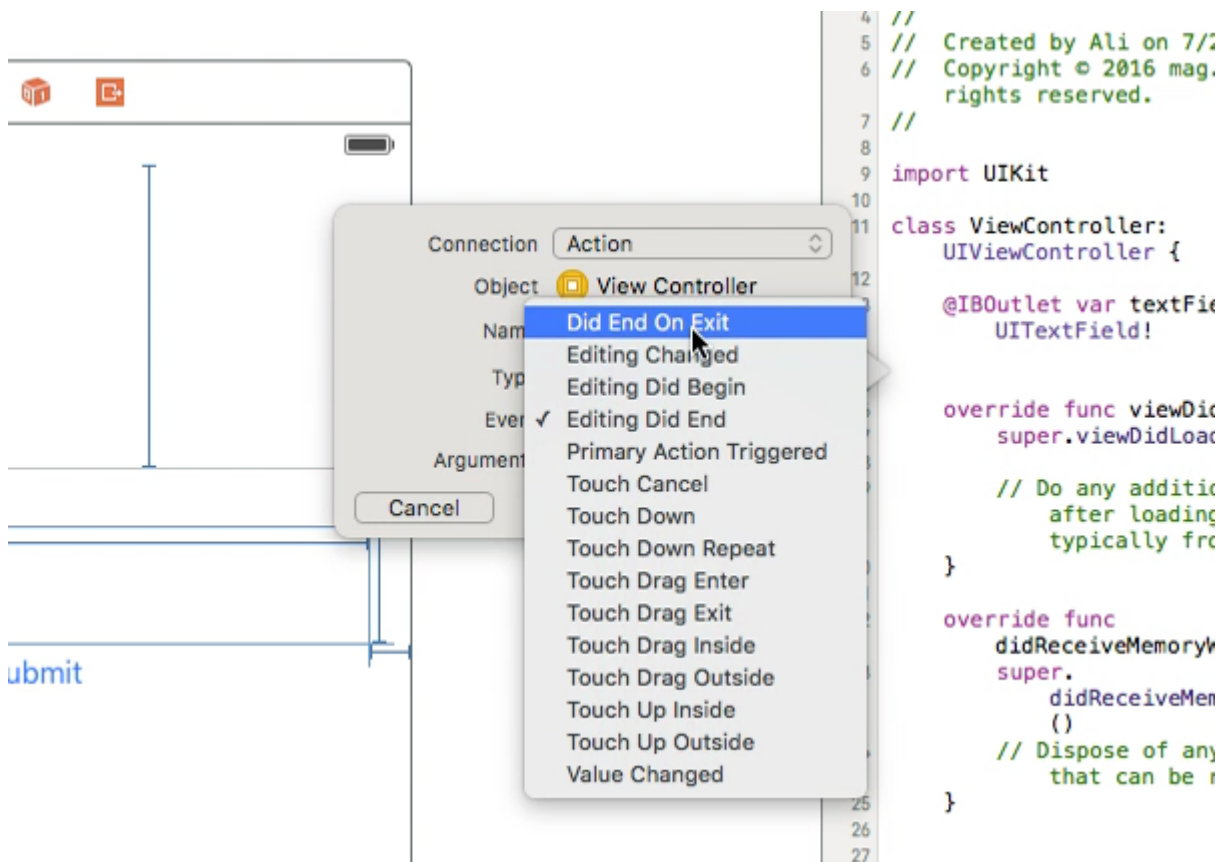
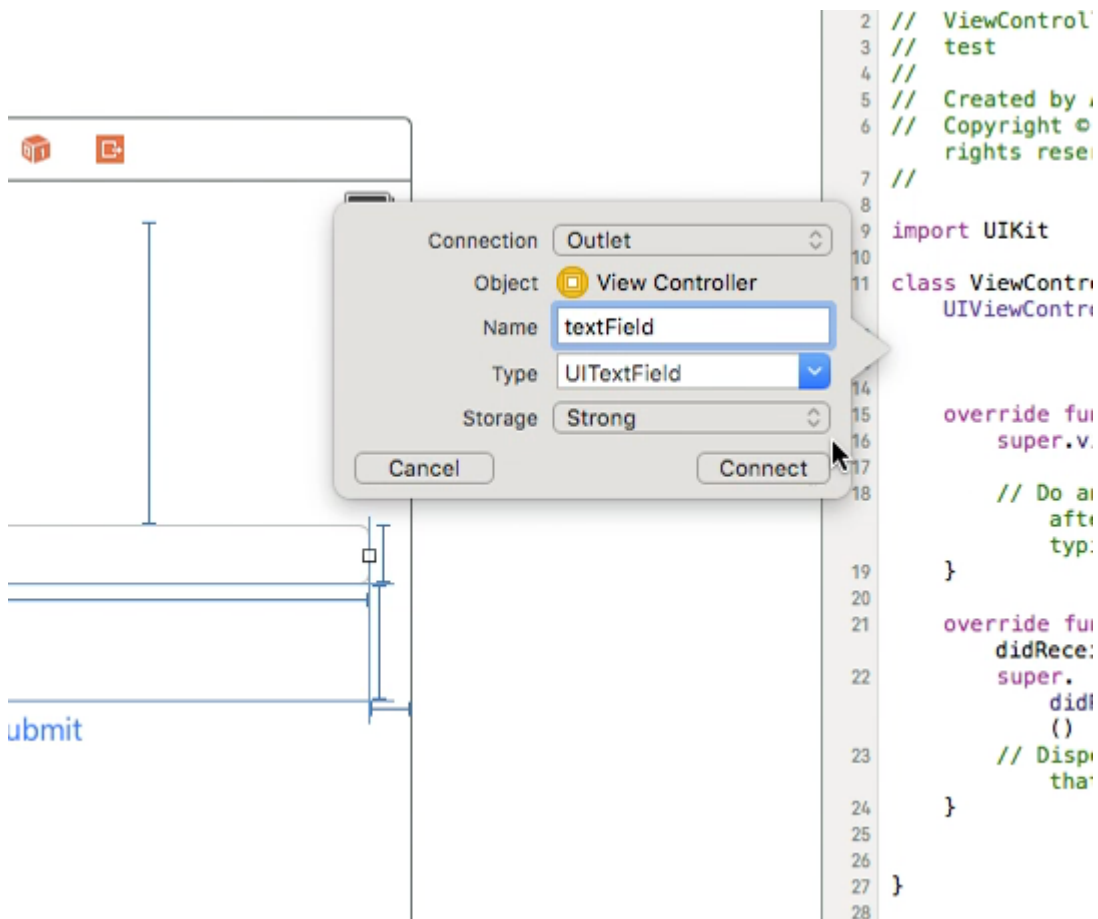
## Objective-C

```
textField.autocapitalizationType = UITextAutocapitalizationTypeNone;
```

- .None \ UITextAutocapitalizationTypeNone
- .Words \ UITextAutocapitalizationTypeWords **Autocapitalize**
- .Sentences \ UITextAutocapitalizationTypeSentences **autocapitalize the first word in a**
- .AllCharacters \ UITextAutocapitalizationTypeAllCharacters **Autocapitalize**

### Ctrl +MainStoryboardUitextfieldViewControllerUITextField





UITextFieldViewControllerCtrl +ActionDid End On Exitconnect.

UITextField.resignFirstResponder()



```
@IBAction func textFieldResign(sender: AnyObject) {
 yourTextFieldName.resignFirstResponder()
}
```

◦

UIViewControllerUITextFieldDelegate

**viewDidLoad** self.yourTextFieldName.delegate = self

```
func textFieldShouldReturn(textField: UITextField) -> Bool {
 yourTextFieldName.resignFirstResponder()
 return true
}
```

```
class ViewController: UIViewController, UITextFieldDelegate {

 @IBOutlet var textField: UITextField!

 func textFieldShouldReturn(textField: UITextField) -> Bool {
 textField.resignFirstResponder()
 return true
 }

 override func touchesBegan(touches: Set<UITouch>, withEvent event: UIEvent?) {
 view.endEditing(true)
 super.touchesBegan(touches, withEvent: event)
 }

 override func viewDidLoad() {
 super.viewDidLoad()
 self.textField.delegate = self
 }

}
```

## Objective-C

```
[textField resignFirstResponder];
```

```
textField.textAlignment = .Center
```

## Objective-C

```
[textField setTextAlignment: NSTextAlignmentCenter];
```

NSTextAlignment **center** ◦ .Left ◦ .Right ◦ .Justified ◦ .Natural ◦

.Natural ◦ .Left ; .Right ◦

UITextFields keyboardType

```
typedef NS_ENUM(NSInteger, UIKeyboardType) {
 UIKeyboardTypeDefault, // Default type for the current input method.
 UIKeyboardTypeASCIICapable, // Displays a keyboard which can enter ASCII
characters, non-ASCII keyboards remain active
 UIKeyboardTypeNumbersAndPunctuation, // Numbers and assorted punctuation.
 UIKeyboardTypeURL, // A type optimized for URL entry (shows . / .com
prominently).
 UIKeyboardTypeNumberPad, // A number pad (0-9). Suitable for PIN entry.
 UIKeyboardTypePhonePad, // A phone pad (1-9, *, 0, #, with letters under the
numbers).
 UIKeyboardTypeNamePhonePad, // A type optimized for entering a person's name or
phone number.
 UIKeyboardTypeEmailAddress, // A type optimized for multiple email address entry
(shows space @ . prominently).
 UIKeyboardTypeDecimalPad NS_ENUM_AVAILABLE_IOS(4_1), // A number pad with a decimal
point.
 UIKeyboardTypeTwitter NS_ENUM_AVAILABLE_IOS(5_0), // A type optimized for twitter
text entry (easy access to @ #)
 UIKeyboardTypeWebSearch NS_ENUM_AVAILABLE_IOS(7_0), // A default keyboard type with
URL-oriented addition (shows space . prominently).

 UIKeyboardTypeAlphabet = UIKeyboardTypeASCIICapable, // Deprecated
};
```

## UITextView

UIKeyboardWillShowNotificationUIKeyboardWillHideNotification scrollView

```
- (void)viewDidLoad
{
 [super viewDidLoad];

 // register for keyboard notifications
 [[NSNotificationCenter defaultCenter] addObserver:self
 selector:@selector(keyboardWillShow:)
 name:UIKeyboardWillShowNotification
 object:self.view.window];

 // register for keyboard notifications
 [[NSNotificationCenter defaultCenter] addObserver:self
 selector:@selector(keyboardWillHide:)
 name:UIKeyboardWillHideNotification
 object:self.view.window];
}

// Called when UIKeyboardWillShowNotification is sent
- (void)keyboardWillShow:(NSNotification*)notification
{
 // if we have no view or are not visible in any window, we don't care
 if (!self.isViewLoaded || !self.view.window) {
 return;
 }

 NSDictionary *userInfo = [notification userInfo];

 CGRect keyboardFrameInWindow;
 [[userInfo objectForKey:UIKeyboardFrameEndUserInfoKey] getValue:&keyboardFrameInWindow];
```

```

// the keyboard frame is specified in window-level coordinates. this calculates the frame
as if it were a subview of our view, making it a sibling of the scroll view
CGRect keyboardFrameInView = [self.view convertRect:keyboardFrameInWindow fromView:nil];

CGRect scrollViewKeyboardIntersection = CGRectIntersection(_scrollView.frame,
keyboardFrameInView);
UIEdgeInsets newContentInsets = UIEdgeInsetsMake(0, 0,
scrollViewKeyboardIntersection.size.height, 0);

// this is an old animation method, but the only one that retains compaitibility between
parameters (duration, curve) and the values contained in the userInfo-Dictionary.
[UIView beginAnimations:nil context:NULL];
[UIView setAnimationDuration:[userInfo
objectForKey:UIKeyboardAnimationDurationUserInfoKey] doubleValue]];
[UIView setAnimationCurve:[userInfo objectForKey:UIKeyboardAnimationCurveUserInfoKey]
intValue]];

_scrollView.contentInset = newContentInsets;
_scrollView.scrollIndicatorInsets = newContentInsets;

/*
 * Depending on visual layout, _focusedControl should either be the input field
(UITextField,..) or another element
 * that should be visible, e.g. a purchase button below an amount text field
 * it makes sense to set _focusedControl in delegates like -textFieldShouldBeginEditing:
if you have multiple input fields
 */
if (_focusedControl) {
 CGRect controlFrameInScrollView = [_scrollView convertRect:_focusedControl.bounds
fromView:_focusedControl]; // if the control is a deep in the hierarchy below the scroll view,
this will calculate the frame as if it were a direct subview
 CGRect controlInset = CGRectInset(controlFrameInScrollView, 0, -10); // replace
10 with any nice visual offset between control and keyboard or control and top of the scroll
view.

 CGFloat controlVisualOffsetToTopOfScrollview = controlFrameInScrollView.origin.y -
_scrollView.contentOffset.y;
 CGFloat controlVisualBottom = controlVisualOffsetToTopOfScrollview +
controlFrameInScrollView.size.height;

 // this is the visible part of the scroll view that is not hidden by the keyboard
 CGFloat scrollViewVisibleHeight = _scrollView.frame.size.height -
scrollViewKeyboardIntersection.size.height;

 if (controlVisualBottom > scrollViewVisibleHeight) { // check if the keyboard will
hide the control in question
 // scroll up until the control is in place
 CGPoint newContentOffset = _scrollView.contentOffset;
 newContentOffset.y += (controlVisualBottom - scrollViewVisibleHeight);

 // make sure we don't set an impossible offset caused by the "nice visual offset"
// if a control is at the bottom of the scroll view, it will end up just above the
keyboard to eliminate scrolling inconsistencies
 newContentOffset.y = MIN(newContentOffset.y, _scrollView.contentSize.height -
scrollViewVisibleHeight);

 [_scrollView setContentOffset:newContentOffset animated:NO]; // animated:NO
because we have created our own animation context around this code
 } else if (controlFrameInScrollView.origin.y < _scrollView.contentOffset.y) {
 // if the control is not fully visible, make it so (useful if the user taps on a

```

```

partially visible input field
 CGPoint newContentOffset = _scrollView.contentOffset;
 newContentOffset.y = controlFrameInScrollView.origin.y;

 [_scrollView setContentOffset:newContentOffset animated:NO]; // animated:NO
because we have created our own animation context around this code
 }
}

[UIView commitAnimations];
}

// Called when the UIKeyboardWillHideNotification is sent
- (void)keyboardWillHide:(NSNotification*)notification
{
 // if we have no view or are not visible in any window, we don't care
 if (!self.isViewLoaded || !self.view.window) {
 return;
 }

 NSDictionary *userInfo = notification.userInfo;

 [UIView beginAnimations:nil context:NULL];
 [UIView setAnimationDuration:[userInfo
valueForKey:UIKeyboardAnimationDurationUserInfoKey] doubleValue]];
 [UIView setAnimationCurve:[userInfo valueForKey:UIKeyboardAnimationCurveUserInfoKey]
intValue]];

 // undo all that keyboardWillShow-magic
 // the scroll view will adjust its contentOffset appropriately
 _scrollView.contentInset = UIEdgeInsetsZero;
 _scrollView.scrollIndicatorInsets = UIEdgeInsetsZero;

 [UIView commitAnimations];
}

```

```
textField.becomeFirstResponder()
```

## Objective-C

```
[textField becomeFirstResponder];
```

```
textField.resignFirstResponder()
```

## Objective-C

```
[textField resignFirstResponder];
```

## UIPickerView

[UIPickerViewUITextField](#)

## UIPickerView

UIPickerView UIPickerViewDataSource UIPickerViewDelegate ◦

```
class MyPickerView: UIPickerView, UIPickerViewDataSource, UIPickerViewDelegate
```

## DataSourceDelegate

```
public func pickerView(_ pickerView: UIPickerView, numberOfRowsInComponent component: Int) -> Int {
 if data != nil {
 return data!.count
 } else {
 return 0
 }
}

public func numberOfComponents(in pickerView: UIPickerView) -> Int {
 return 1
}

public func pickerView(_ pickerView: UIPickerView, titleForRow row: Int, forComponent component: Int) -> String? {
 if data != nil {
 return data![row]
 } else {
 return ""
 }
}
```

MyPickerViewdata selectedValuetextFieldBeingEdited

```
/**
 * The data for the `UIPickerViewDelegate`
 *
 * Always needs to be an array of `String`! The `UIPickerView` can ONLY display Strings
 */
public var data: [String]? {
 didSet {
 super.delegate = self
 super.dataSource = self
 self.reloadAllComponents()
 }
}

/**
 * Stores the UITextField that is being edited at the moment
 */
public var textFieldBeingEdited: UITextField?

/**
 * Get the selected Value of the picker
 */
public var selectedValue: String {
 get {
 if data != nil {
 return data![selectedRow(inComponent: 0)]
 } else {

```

```

 return ""
 }
}
}

```

## ViewController

`textField` UIPickerView ◦ `textField` @IBOutlet

```

/**
 * The picker view to present as keyboard
 */
var picker: UIPickerView?

```

`viewDidLoad()` picker

```

picker = UIPickerView()
picker?.autoresizingMask = [.flexibleHeight, .flexibleWidth]
picker?.backgroundColor = UIColor.white()

picker?.data = ["One", "Two", "Three", "Four", "Five"] //The data shown in the picker

```

`MyPickerUITextField` inputView

```

textField.inputView = picker

```

`UIPickerView` ◦ `.inputAccessoryView`

`pickerAccessoryViewController` ◦

```

/**
 * A toolbar to add to the keyboard when the `picker` is presented.
 */
var pickerAccessory: UIToolbar?

```

`viewDidLoad()` `inputAccessoryViewUIToolbar`

```

pickerAccessory = UIToolbar()
pickerAccessory?.autoresizingMask = .flexibleHeight

//this customization is optional
pickerAccessory?.barStyle = .default
pickerAccessory?.barTintColor = UIColor.red()
pickerAccessory?.backgroundColor = UIColor.red()
pickerAccessory?.isTranslucent = false

```

◦ `iOS44.0`

```

var frame = pickerAccessory?.frame
frame?.size.height = 44.0
pickerAccessory?.frame = frame!

```

“”。

```
let cancelButton = UIBarButtonItem(barButtonItemSystemItem: .cancel, target: self, action:
#selector(ViewController.cancelBtnClicked(_:)))
cancelButton.tintColor = UIColor.white()
let flexSpace = UIBarButtonItem(barButtonItemSystemItem: .flexibleSpace, target: nil, action: nil)
//a flexible space between the two buttons
let doneButton = UIBarButtonItem(barButtonItemSystemItem: .done, target: self, action:
#selector(ViewController.doneBtnClicked(_:)))
doneButton.tintColor = UIColor.white()

//Add the items to the toolbar
pickerAccessory?.items = [cancelButton, flexSpace, doneButton]
```

inputAccessoryView

```
textField.inputAccessoryView = pickerAccessory
```

```
/**
 Called when the cancel button of the `pickerAccessory` was clicked. Dismisses the picker
 */
func cancelBtnClicked(_ button: UIBarButtonItem?) {
 textField?.resignFirstResponder()
}

/**
 Called when the done button of the `pickerAccessory` was clicked. Dismisses the picker and
 puts the selected value into the textField
 */
func doneBtnClicked(_ button: UIBarButtonItem?) {
 textField?.resignFirstResponder()
 textField.text = picker?.selectedValue
}
```

textField

Cancel

Done

One

Two

Three

Four

UIPickerView

```
picker?.selectRow(3, inComponent: 0, animated: false) //Will select the row at index 3
```

◦

```
class MyViewController: UITextFieldDelegate {

 override viewDidLoad() {
 super.viewDidLoad()

 textField.delegate = self
 }

}
```

textFieldShouldReturn ◦

```
func textFieldShouldReturn(textField: UITextField) -> Bool {
 textField.resignFirstResponder()
 return true;
}
```

## Objective-C

```
- (BOOL)textFieldShouldReturn:(UITextField *)textField {
 [textField resignFirstResponder];
 return true;
}
```



```
let startPosition: UITextPosition = textField.beginningOfDocument
```

```
let endPosition: UITextPosition = textField.endOfDocument
```

```
let selectedRange: UITextRange? = textField.selectedTextRange
```

---

```
if let selectedRange = textField.selectedTextRange {
 let cursorPosition = textField.offsetFromPosition(textField.beginningOfDocument,
toPosition: selectedRange.start)
 print("\(cursorPosition)")
}
```

o

```
let newPosition = textField.beginningOfDocument
textField.selectedTextRange = textField.textRangeFromPosition(newPosition, toPosition:
newPosition)
```

```
let newPosition = textField.endOfDocument
textField.selectedTextRange = textField.textRangeFromPosition(newPosition, toPosition:
newPosition)
```

```
// only if there is a currently selected range
if let selectedRange = textField.selectedTextRange {
 // and only if the new position is valid
 if let newPosition = textField.positionFromPosition(selectedRange.start, inDirection:
UITextLayoutDirection.Left, offset: 1) {
 // set the new position
 textField.selectedTextRange = textField.textRangeFromPosition(newPosition, toPosition:
newPosition)
 }
}
```

5.

```
let arbitraryValue: Int = 5
if let newPosition = textField.positionFromPosition(textField.beginningOfDocument,
inDirection: UITextLayoutDirection.Right, offset: arbitraryValue) {
 textField.selectedTextRange = textField.textRangeFromPosition(newPosition, toPosition:
newPosition)
}
```

```
textField.selectedTextRange = textField.textRangeFromPosition(textField.beginningOfDocument,
toPosition: textField.endOfDocument)
```

```
// Range: 3 to 7
let startPosition = textField.positionFromPosition(textField.beginningOfDocument, inDirection:
UITextLayoutDirection.Right, offset: 3)
let endPosition = textField.positionFromPosition(textField.beginningOfDocument, inDirection:
UITextLayoutDirection.Right, offset: 7)

if startPosition != nil && endPosition != nil {
 textField.selectedTextRange = textField.textRangeFromPosition(startPosition!, toPosition:
endPosition!)
}
```

```
textField.insertText("Hello")
```

- 
- [Stack Overflow](#) ◦
  - UITextView ◦
  - textField.becomeFirstResponder() ◦
  - ◦

- 
- [Swift](#) selectedTextRange.selectedRange

UITextField.caretRectForPosition: CGRectZero ◦

## Swift 2.3 <

```
public override func caretRectForPosition(position: UITextPosition) -> CGRect {
 return CGRectZero
}
```

## 3

```
override func caretRect(for position: UITextPosition) -> CGRect {
 return CGRect.zero
}
```

## Objective-C

```
- (CGRect) caretRectForPosition:(UITextPosition*) position{
return CGRectZero;
}
```

attributedString NSAttributedString。

```
var placeholderAttributes = [String: AnyObject]()
placeholderAttributes[NSForegroundColorAttributeName] = color
placeholderAttributes[NSFontAttributeName] = font

if let placeholder = textField.placeholder {
 let newAttributedString = NSAttributedString(string: placeholder, attributes:
placeholderAttributes)
 textField.attributedString = newAttributedString
}
```

colorfont。 NSAttributedString。

## UITextField

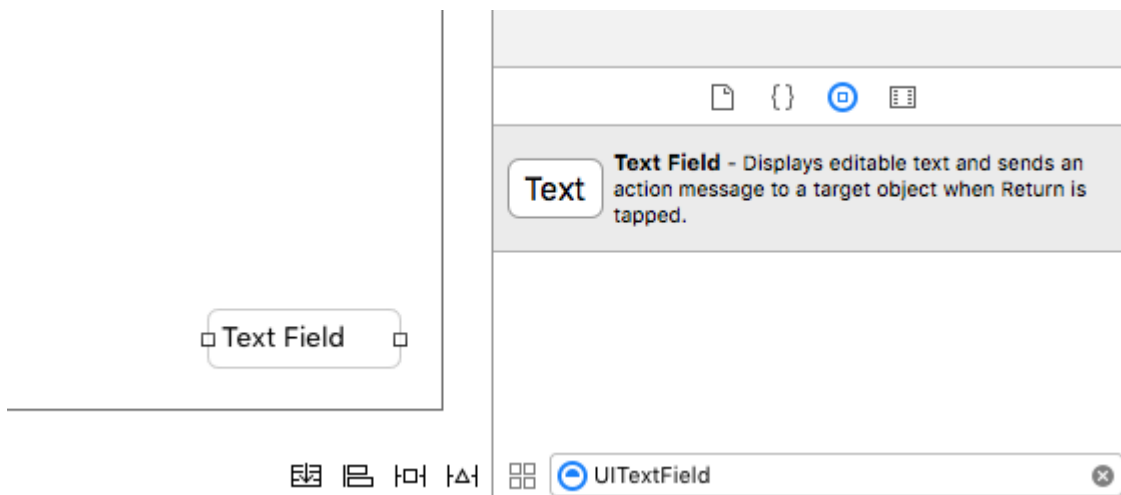
CGRectUITextField

```
let textField = UITextField(frame: CGRect(x: 0, y: 0, width: 200, height: 21))
```

# Objective-C

```
UITextField *textField = [[UITextField alloc] initWithFrame:CGRectMake(0, 0, 200, 21)];
```

Interface BuilderUITextField



UITextField <https://riptutorial.com/zh-TW/ios/topic/1630/uitextfield>

---

# 183: UITextView

## Examples

```
textView.text = "Hello, world!"
```

### Objective-C

```
textView.text = @"Hello, world!";
```

```
// Modify some of the attributes of the attributed string.
let attributedText = NSMutableAttributedString(attributedString: textView.attributedText!)

// Use NSString so the result of rangeOfString is an NSRange.
let text = textView.text! as NSString

// Find the range of each element to modify.
let tintedRange = text.range(of: NSLocalizedString("tinted", comment: ""))
let highlightedRange = text.range(of: NSLocalizedString("highlighted", comment: ""))

// Add tint.
attributedText.addAttribute(NSForegroundColorAttributeName, value: UIColor.blue, range:
tintedRange)

// Add highlight.
attributedText.addAttribute(NSBackgroundColorAttributeName, value: UIColor.yellow, range:
highlightedRange)

textView.attributedText = attributedText
```

```
textView.textAlignment = .left
```

### Objective-C

```
textView.textAlignment = NSTextAlignmentLeft;
```

## UITextViewDelegate

- textViewShouldBeginEditing(\_:)
- textViewDidBeginEditing(\_:)
- textViewShouldEndEditing(\_:)
- textViewDidEndEditing(\_:)
  
- textView(\_:shouldChangeTextIn:replacementText:)
- textViewDidChange(\_:)

## URL

- textView(\_: UITextView, shouldInteractWithURL: NSURL, inRange: NSRange) -> Bool

```
//System Font
textView.font = UIFont.systemFont(ofSize: 12)

//Font of your choosing
textView.font = UIFont(name: "Font Name", size: 12)
```

## Objective-C

```
//System Font
textView.font = [UIFont systemFontOfSize:12];

//Font of your choosing
textView.font = [UIFont fontWithName:@"Font Name" size:12];
```

```
textView.textColor = UIColor.red
```

## Objective-C

```
textView.textColor = [UIColor redColor];
```

## HTMLUITextView

```
NSString *htmlString = @"<p> This is an HTML text</p>";
NSAttributedString *attributedString = [[NSMutableAttributedString alloc]
 initWithData: [htmlString
 dataUsingEncoding:NSUTF8StringEncoding]
 options: @{
NSDocumentTypeDocumentAttribute: NSHTMLTextDocumentType }
 documentAttributes: nil
 error: nil
];
_yourTextView.attributedString = attributedString;
// If you want to modify the font
field.font = [UIFont fontWithName:@"Raleway-Regular" size:15];
```

UITextView°

```
enum {
 UIDataDetectorTypePhoneNumber = 1 << 0,
 UIDataDetectorTypeLink = 1 << 1,
 UIDataDetectorTypeAddress = 1 << 2,
 UIDataDetectorTypeCalendarEvent = 1 << 3,
 UIDataDetectorTypeNone = 0,
 UIDataDetectorTypeAll = NSUIntegerMax
};
```

```
// you may add as many as you like by using the `|` operator between options
textView.dataDetectorTypes = (UIDataDetectorTypeLink | UIDataDetectorTypePhoneNumber);
```

UITextView

UITextView

```
textView.editable = NO;
textView.selectable = YES;
textView.userInteractionEnabled = YES; // YES by default
```

```
if let text = self.textView.text where !text.isEmpty {
 // Do stuff for text
} else {
 // Do stuff for nil text or empty string
}
```

## Objective-C

```
if (self.textView.text.length > 0){
 // Do stuff for text
} else {
 // Do stuff for nil text or empty string
}
```

---

```
let startPosition: UITextPosition = textView.beginningOfDocument
```

```
let endPosition: UITextPosition = textView.endOfDocument
```

```
let selectedRange: UITextRange? = textView.selectedTextRange
```

---

```
if let selectedRange = textView.selectedTextRange {

 let cursorPosition = textView.offsetFromPosition(textView.beginningOfDocument, toPosition:
selectedRange.start)

 print("\(cursorPosition)")
}
```

o

```
let newPosition = textView.beginningOfDocument
textView.selectedTextRange = textView.textRangeFromPosition(newPosition, toPosition:
newPosition)
```

```
let newPosition = textView.endOfDocument
textView.selectedTextRange = textView.textRangeFromPosition(newPosition, toPosition:
newPosition)
```

```
// only if there is a currently selected range
if let selectedRange = textView.selectedTextRange {
```

```

// and only if the new position is valid
if let newPosition = textView.positionFromPosition(selectedRange.start, inDirection:
UITextLayoutDirection.Left, offset: 1) {

 // set the new position
 textView.selectedTextRange = textView.textRangeFromPosition(newPosition, toPosition:
newPosition)
}
}

```

5.

```

let arbitraryValue: Int = 5
if let newPosition = textView.positionFromPosition(textView.beginningOfDocument, inDirection:
UITextLayoutDirection.Right, offset: arbitraryValue) {

 textView.selectedTextRange = textView.textRangeFromPosition(newPosition, toPosition:
newPosition)
}

```

---

```

textView.selectedTextRange = textView.textRangeFromPosition(textView.beginningOfDocument,
toPosition: textView.endOfDocument)

```

```

// Range: 3 to 7
let startPosition = textView.positionFromPosition(textView.beginningOfDocument, inDirection:
UITextLayoutDirection.Right, offset: 3)
let endPosition = textView.positionFromPosition(textView.beginningOfDocument, inDirection:
UITextLayoutDirection.Right, offset: 7)

if startPosition != nil && endPosition != nil {
 textView.selectedTextRange = textView.textRangeFromPosition(startPosition!, toPosition:
endPosition!)
}

```

```

textView.insertText("Hello")

```

- 
- [Stack Overflow](#) ◦
  - UITextView ◦
  - textView.becomeFirstResponder() ◦
  - ◦

- 
- [Swift](#) selectedTextRange.selectedRange

◦

UITextView° °

°

```
messageTextView.textContainerInset = UIEdgeInsetsZero
messageTextView.textContainer.lineFragmentPadding = 0
```

NSAttributedString.boundingBoxWithSize(...)UITextView°

```
let budget = getSomeCGSizeBudget()
let text = getSomeAttributedString()
let textSize = text.boundingBoxWithSize(budget, options: [.UsesLineFragmentOrigin,
.UsesFontLeading], context: nil).size
messageTextView.frame.size = textSize // Just fits.
```

**UITextView** <https://riptutorial.com/zh-TW/ios/topic/1043/uitextView>



---

# 184: UIView

1. // Objective-C
2. [UIView new] //
3. [[UIView alloc] initWithFrame :(CGRect) //
4. [[UIView alloc] init] //
5. //
6. UIView//CGRect.zeroUIView
7. UIViewframeCGRect//UIView
8. UIView.addSubviewUIView//UIView
9. UIView.hidden //
10. UIView.alpha //
11. UIView.setNeedsLayout//

**UIView** ◦ ◦

## Examples

### UIView

#### Objective-C

```
CGRect myFrame = CGRectMake(0, 0, 320, 35)
UIView *view = [[UIView alloc] initWithFrame:myFrame];

//Alternative way of defining the frame
UIView *view = [[UIView alloc] init];
CGRect myFrame = view.frame;
myFrame.size.width = 320;
myFrame.size.height = 35;
myFrame.origin.x = 0;
myFrame.origin.y = 0;
view.frame = myFrame;
```

```
let myFrame = CGRect(x: 0, y: 0, width: 320, height: 35)
let view = UIView(frame: myFrame)
```

**UIView** layercornerRadius ◦

## SWIFT

```
someImageView.layoutIfNeeded()
someImageView.clipsToBounds = true
someImageView.layer.cornerRadius = 10
```

## Objective-C

```
[someImageView layoutIfNeeded];
someImageView.clipsToBounds = YES;
someImageView.layer.cornerRadius = 10;
```

```
//Swift code
topImageView.layoutIfNeeded()
bottomImageView.layoutIfNeeded()
topImageView.clipsToBounds = true
topImageView.layer.cornerRadius = 10
bottomImageView.clipsToBounds = true
bottomImageView.layer.cornerRadius = bottomImageView.frame.width / 2

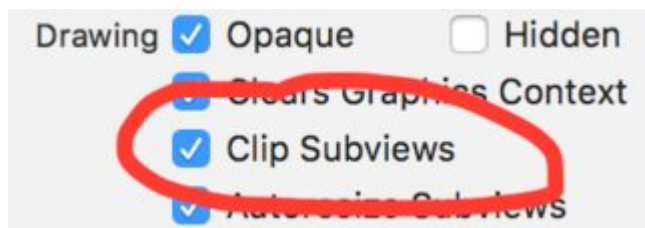
//Objective-C code
[topImageView layoutIfNeeded]
[bottomImageView layoutIfNeeded];
topImageView.clipsToBounds = YES;
topImageView.layer.cornerRadius = 10;
bottomImageView.clipsToBounds = YES;
bottomImageView.cornerRadius = CGRectGetWidth(bottomImageView.frame) / 2;
```



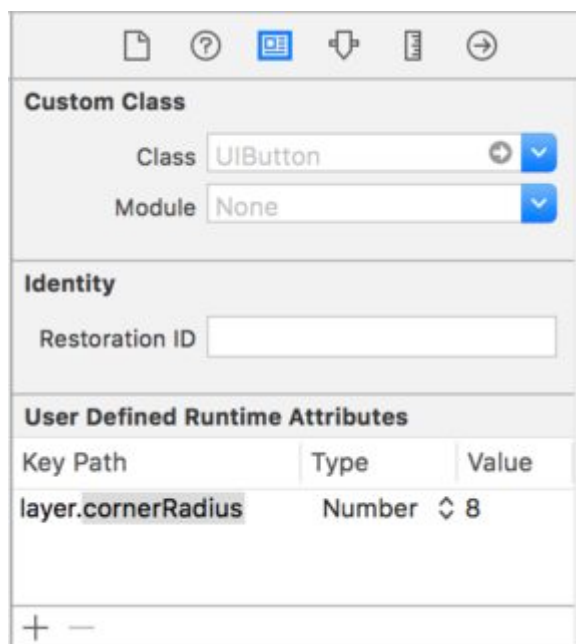
## QuartzCore◦

```
#import <QuartzCore/QuartzCore.h>
```

## Storyboard non-programmatically



`layer.cornerRadius`



.

```
extension UIView {
 @discardableResult
 public func setAsCircle() -> Self {
 self.clipsToBounds = true
 let frameSize = self.frame.size
 self.layer.cornerRadius = min(frameSize.width, frameSize.height) / 2.0
 return self
 }
}
```

```
yourView.setAsCircle()
```

UIView

```
let snapshot = view.snapshotView(afterScreenUpdates: true)
```

## Objective-C

```
UIView *snapshot = [view snapshotViewAfterScreenUpdates: YES];
```

# IBInspectableIBDesignable

Xcode IBInspectableIBDesignable UIView ◦ Xcode ◦ iOS ◦ CustomView UIView ◦ ◦ ◦

```
var textColor: UIColor = UIColor.blackColor()
var text: String?
var showText: Bool = true
```

drawRect

```
if showText {
 if let text = text {
 let s = NSString(string: text)
 s.drawInRect(rect,
 withAttributes: [
 NSForegroundColorAttributeName: textColor,
 NSFontAttributeName: UIFont(name: "Helvetica Neue", size: 18)!
])
 }
}
```

text ◦ ◦ IBInspectableIBDesignable IBInspectable Xcode ◦ IBDesignable ◦

```
@IBDesignable
class CustomView: UIView {
 @IBInspectable var textColor: UIColor = UIColor.blackColor()
 @IBInspectable var text: String?
 @IBInspectable var showText: Bool = true

 override func drawRect(rect: CGRect) {
 // ...
 }
}
```

## C

```
IB_DESIGNABLE
@interface CustomView: UIView

@property (nonatomic, strong) IBInspectable UIColor* textColor;
@property (nonatomic, strong) IBInspectable NSString* text;
@property (nonatomic, assign) IBInspectable BOOL showText;

@end

@implementation CustomView

- (instancetype)init {
 if(self = [super init]) {
 self.textColor = [UIColor blackColor];
 self.showText = YES;
 }
 return self;
}

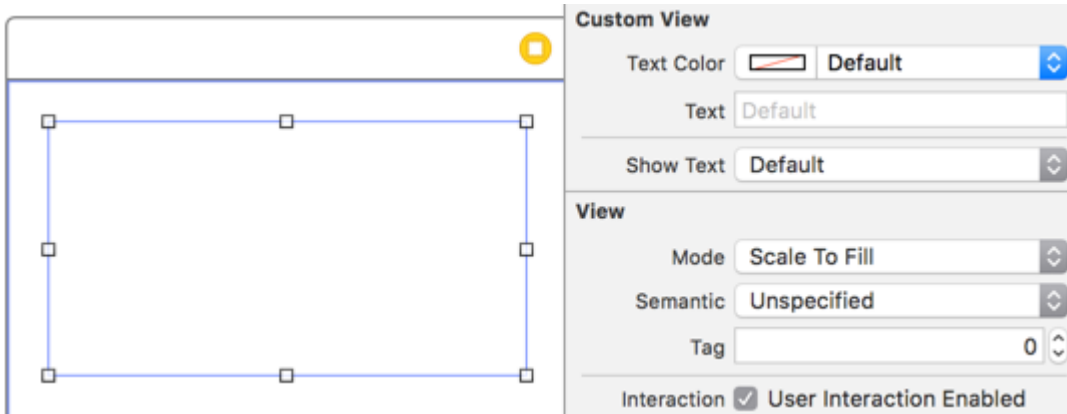
- (void)drawRect:(CGRect)rect {
```

```

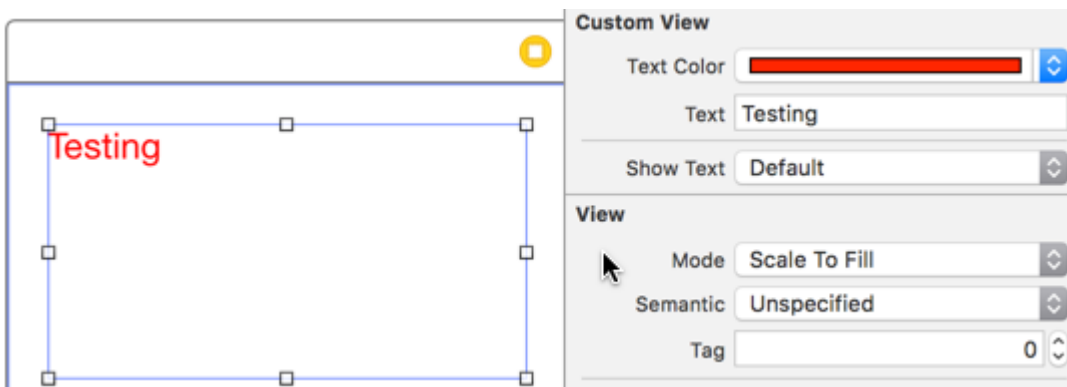
//...
}
@end

```

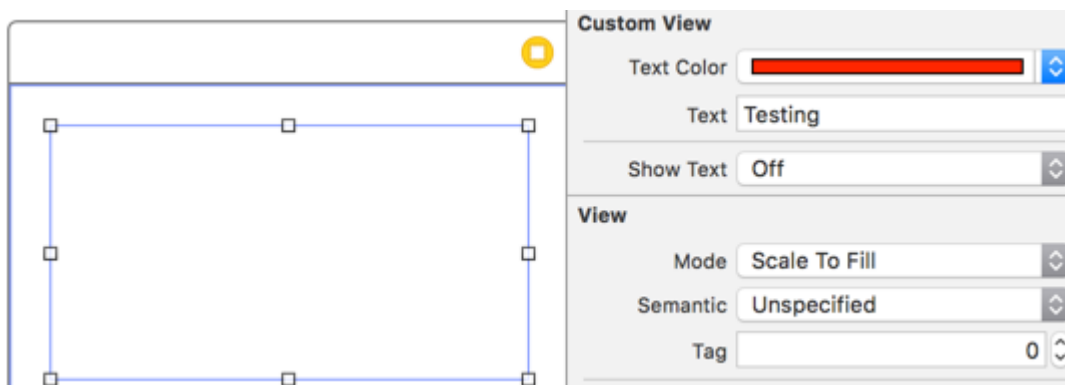
Xcode ◦ ◦ UI ◦ OffOnfalsetrue ◦



◦ drawRect ◦ Interface Builder ◦



“ Off “ Off ”Interface Builder ◦



UIViewStoryboard ◦ IBDesignableStoryboard ExtensionUIViewUIView ◦ UIView ◦

```

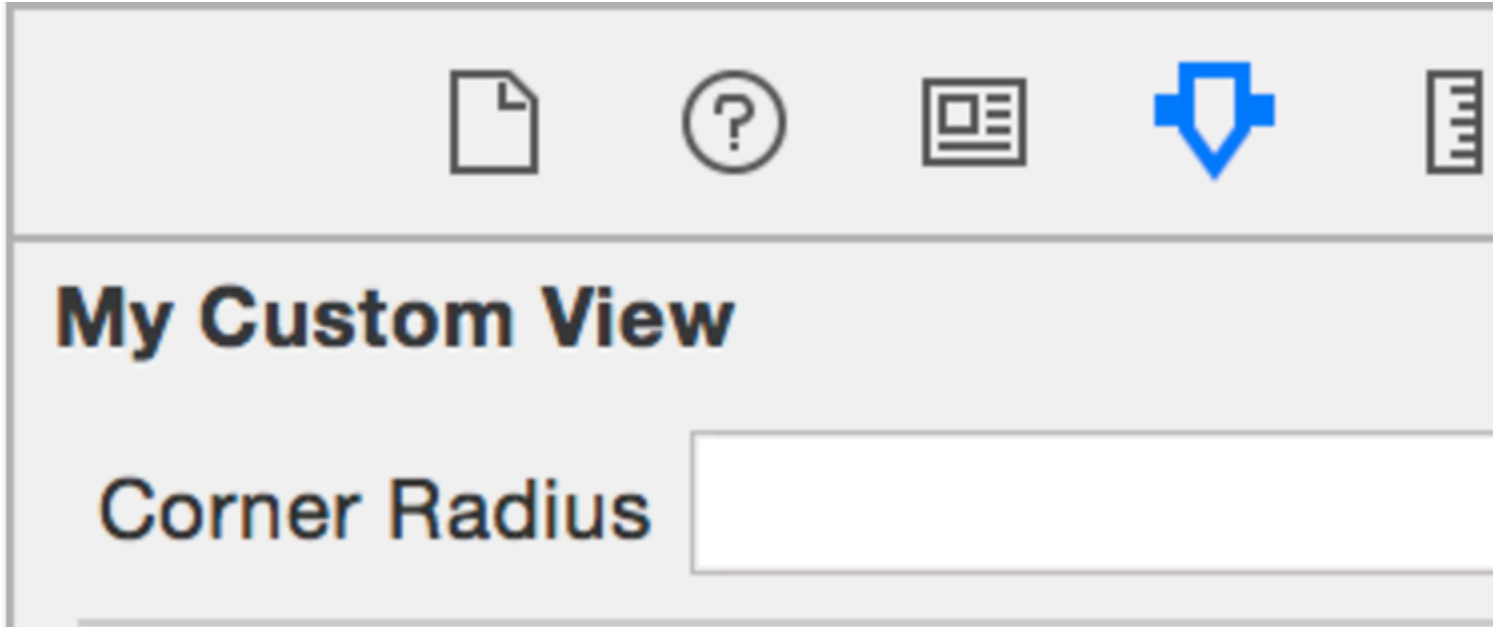
extension UIView {
 @IBInspectable var cornerRadius:CGFloat {
 set {
 layer.cornerRadius = newValue
 clipsToBounds = newValue > 0
 }
 }
}

```

```

 }
 get {
 return layer.cornerRadius
 }
}
}

```



## UIView

```

let view = UIView(frame: CGRect(x: 0, y: 0, width: 100, height: 100))
view.backgroundColor = UIColor.orange
self.view.addSubview(view)
UIView.animate(withDuration: 0.75, delay: 0.5, options: .curveEaseIn, animations: {
 //This will cause view to go from (0,0) to
 // (self.view.frame.origin.x,self.view.frame.origin.y)
 view.frame.origin.x = self.view.frame.origin.x
 view.frame.origin.y = self.view.frame.origin.y
}) { (finished) in
 view.backgroundColor = UIColor.blueColor()
}

```

## UIView

### x-cordinate

```
view.frame.origin.x
```

```
view.frame.size.width
```

### UIView

```

view.x
view.y
view.width

```

```
view.height
```

```
view.x = 10
view.y = 10
view.width = 100
view.height = 200
```

```
extension UIView {

 var x: CGFloat {
 get {
 return self.frame.origin.x
 }
 set {
 self.frame = CGRect(x: newValue, y: self.frame.origin.y, width:
self.frame.size.width, height: self.frame.size.height)
 }
 }

 var y: CGFloat {
 get {
 return self.frame.origin.y
 }
 set {
 self.frame = CGRect(x: self.frame.origin.x, y: newValue, width:
self.frame.size.width, height: self.frame.size.height)
 }
 }

 var width: CGFloat {
 get {
 return self.frame.size.width
 }
 set {
 self.frame = CGRect(x: self.frame.origin.x, y: self.frame.origin.y, width:
newValue, height: self.frame.size.height)
 }
 }

 var height: CGFloat {
 get {
 return self.frame.height
 }
 set {
 self.frame = CGRect(x: self.frame.origin.x, y: self.frame.origin.y, width:
self.frame.size.width, height: newValue)
 }
 }
}
```

## UIViewUIView

```
parentView addSubview:UIImageViewUIViewController°
```

### Objective-C

```
[parentView addSubview:subView];
```

```
parentView.addSubview(subView)
```

subView2 parentView

## Objective-C

```
[parentView insertSubview:subView belowSubview:subView2];
```

```
parentView.insertSubview(subView, belowSubview: subView2)
```

subView2

## Objective-C

```
[parentView insertSubview:subView aboveSubview:subView2];
```

```
parentView.insertSubview(subView, aboveSubview: subView2)
```

subViewparentView

## Objective-C

```
[parentView bringSubviewToFront:subView];
```

```
parentView.bringSubviewToFront(subView)
```

subViewparentView

## Objective-C

```
[subView removeFromSuperview];
```

```
subView.removeFromSuperview()
```

## AutolayoutUIView

```
UIView *view = [[UIView alloc] init];
```

```
[self.view addSubview:view];
```

```
//Use the function if you want to use height as constraint
[self addSubview:view onParentView:self.view withHeight:200.f];
```

```
//Use this function if you want to add view with respect to parent and should resize with it
[self addFullResizeConstraintForSubview:view addedOnParentView:self.view];
```

```
-(void)addSubview:(UIView*)subView onParentView:(UIView*)parentView withHeight:(CGFloat)height{
```



```

subView.translatesAutoresizingMaskIntoConstraintsMaskIntoConstraints = NO;

NSLayoutConstraint *trailing = [NSLayoutConstraint
 constraintWithItem:subView
 attribute:NSLayoutAttributeTrailing
 relatedBy:NSLayoutRelationEqual
 toItem:parent
 attribute:NSLayoutAttributeTrailing
 multiplier:1.0
 constant:10.f];

NSLayoutConstraint *top = [NSLayoutConstraint
 constraintWithItem:subView
 attribute:NSLayoutAttributeTop
 relatedBy:NSLayoutRelationEqual
 toItem:parent
 attribute:NSLayoutAttributeTop
 multiplier:1.0
 constant:10.f];

NSLayoutConstraint *leading = [NSLayoutConstraint
 constraintWithItem:subView
 attribute:NSLayoutAttributeLeading
 relatedBy:NSLayoutRelationEqual
 toItem:parent
 attribute:NSLayoutAttributeLeading
 multiplier:1.0
 constant:10.f];

[parent addConstraint:trailing];
[parent addConstraint:top];
[parent addConstraint:leading];

NSLayoutConstraint *heightConstraint = [NSLayoutConstraint
 constraintWithItem:subView
 attribute:NSLayoutAttributeHeight
 relatedBy:NSLayoutRelationEqual
 toItem:nil
 attribute:0
 multiplier:0.0
 constant:height];

[subView addConstraint:heightConstraint];
}

```

## UIViewresize。

```

- (void) addFullResizeConstraintForSubview: (UIView*) subView
addedOnParentView: (UIView*) parentView{

subView.translatesAutoresizingMaskIntoConstraintsMaskIntoConstraints = NO;

NSLayoutConstraint *trailing = [NSLayoutConstraint
 constraintWithItem:subView
 attribute:NSLayoutAttributeTrailing
 relatedBy:NSLayoutRelationEqual
 toItem:parent
 attribute:NSLayoutAttributeTrailing

```

```

 multiplier:1.0
 constant:10.f];

NSLayoutConstraint *top = [NSLayoutConstraint
 constraintWithItem:subView
 attribute:NSLayoutAttributeTop
 relatedBy:NSLayoutRelationEqual
 toItem:parent
 attribute:NSLayoutAttributeTop
 multiplier:1.0
 constant:10.f];

NSLayoutConstraint *leading = [NSLayoutConstraint
 constraintWithItem:subView
 attribute:NSLayoutAttributeLeading
 relatedBy:NSLayoutRelationEqual
 toItem:parent
 attribute:NSLayoutAttributeLeading
 multiplier:1.0
 constant:10.f];

NSLayoutConstraint *bottom = [NSLayoutConstraint
 constraintWithItem:subView
 attribute:NSLayoutAttributeBottom
 relatedBy:NSLayoutRelationEqual
 toItem:parent
 attribute:NSLayoutAttributeBottom
 multiplier:1.0
 constant:0.f];

[parent addConstraint:trailing];
[parent addConstraint:top];
[parent addConstraint:leading];
[parent addConstraint:bottom];
}

```

## UIView

```

class ImageView: UIView {
 var image: UIImage {
 didSet {
 invalidateIntrinsicContentSize()
 }
 }
 // omitting initializers
 // convenience init(image: UIImage)

 override func intrinsicContentSize() -> CGSize {
 return CGSize(width: image.size.width, height: image.size.height)
 }
}

```

UIViewNoIntrinsicMetric ◦

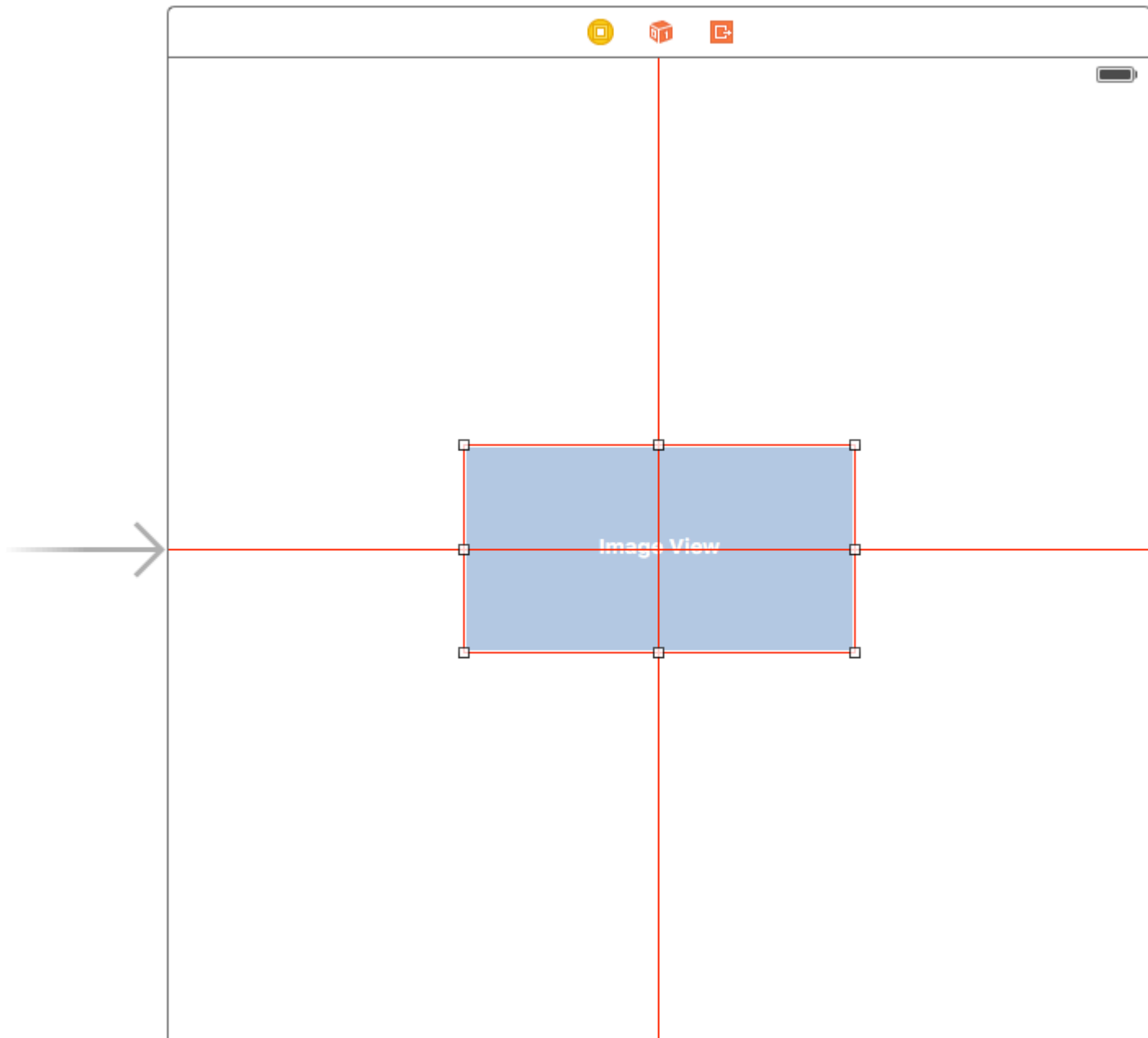
```

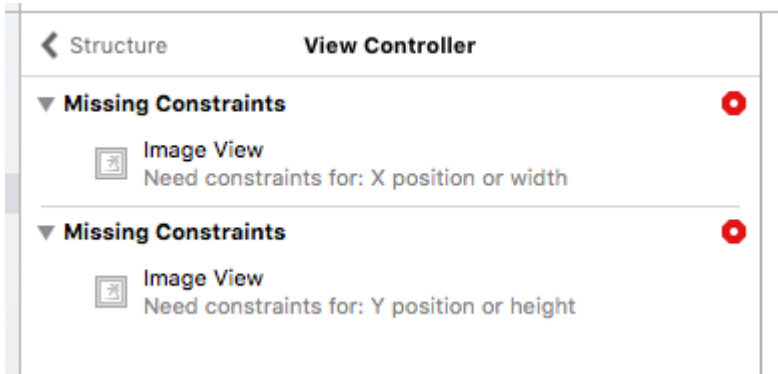
override func intrinsicContentSize() -> CGSize {
 return CGSize(width: UIViewNoIntrinsicMetric, height: image.size.width)
}

```

## AutoLayoutInterface Builder

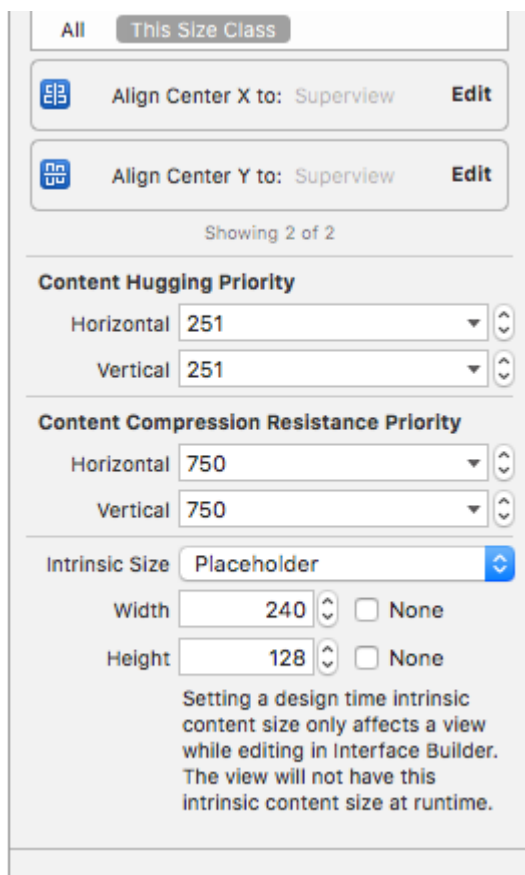
UIImageViewUIImageViewXY.





Placeholder Intrinsic Size

◦



◦

```
extension UIView {
 func shake() {
 let animation = CAKeyframeAnimation(keyPath: "transform.translation.x")
 animation.timingFunction = CAMediaTimingFunction(name: kCAMediaTimingFunctionLinear)
 animation.duration = 0.6
 animation.values = [-10.0, 10.0, -7.0, 7.0, -5.0, 5.0, 0.0]
 layer.add(animation, forKey: "shake")
 }
}
```

◦



# 185: UIViewController

## Examples

UIControl

- `beginTrackingWithTouch` ◦
- `continueTrackingWithTouch` ◦
- `endTrackingWithTouch` ◦

### *MyCustomControl.swift*

```
import UIKit

// These are our self-defined rules for how we will communicate with other classes
protocol ViewControllerCommunicationDelegate: class {
 func myTrackingBegan()
 func myTrackingContinuing(location: CGPoint)
 func myTrackingEnded()
}

class MyCustomControl: UIControl {

 // whichever class wants to be notified of the touch events must set the delegate to
 // itself
 weak var delegate: ViewControllerCommunicationDelegate?

 override func beginTrackingWithTouch(touch: UITouch, withEvent event: UIEvent?) -> Bool {

 // notify the delegate (i.e. the view controller)
 delegate?.myTrackingBegan()

 // returning true means that future events (like continueTrackingWithTouch and
 // endTrackingWithTouch) will continue to be fired
 return true
 }

 override func continueTrackingWithTouch(touch: UITouch, withEvent event: UIEvent?) -> Bool
 {

 // get the touch location in our custom control's own coordinate system
 let point = touch.locationInView(self)

 // Update the delegate (i.e. the view controller) with the new coordinate point
 delegate?.myTrackingContinuing(point)

 // returning true means that future events will continue to be fired
 return true
 }

 override func endTrackingWithTouch(touch: UITouch?, withEvent event: UIEvent?) {

 // notify the delegate (i.e. the view controller)
 delegate?.myTrackingEnded()
 }
}
```

## ViewController.swift

◦

```
import UIKit
class ViewController: UIViewController, ViewControllerCommunicationDelegate {

 @IBOutlet weak var myCustomControl: MyCustomControl!
 @IBOutlet weak var trackingBeganLabel: UILabel!
 @IBOutlet weak var trackingEndedLabel: UILabel!
 @IBOutlet weak var xLabel: UILabel!
 @IBOutlet weak var yLabel: UILabel!

 override func viewDidLoad() {
 super.viewDidLoad()
 myCustomControl.delegate = self
 }

 func myTrackingBegan() {
 trackingBeganLabel.text = "Tracking began"
 }

 func myTrackingContinuing(location: CGPoint) {
 xLabel.text = "x: \(location.x)"
 yLabel.text = "y: \(location.y)"
 }

 func myTrackingEnded() {
 trackingEndedLabel.text = "Tracking ended"
 }
}
```

- 
- print◦

```
import UIKit
class MyCustomControl: UIControl {

 override func beginTrackingWithTouch(touch: UITouch, withEvent event: UIEvent?) -> Bool {
 print("Began tracking")
 return true
 }

 override func continueTrackingWithTouch(touch: UITouch, withEvent event: UIEvent?) -> Bool {
 let point = touch.locationInView(self)
 print("x: \(point.x), y: \(point.y)")
 return true
 }

 override func endTrackingWithTouch(touch: UITouch?, withEvent event: UIEvent?) {
 print("Ended tracking")
 }
}
```

```
let viewController = UIViewController()
```

## Objective-C

```
UIViewController *viewController = [UIViewController new];
```

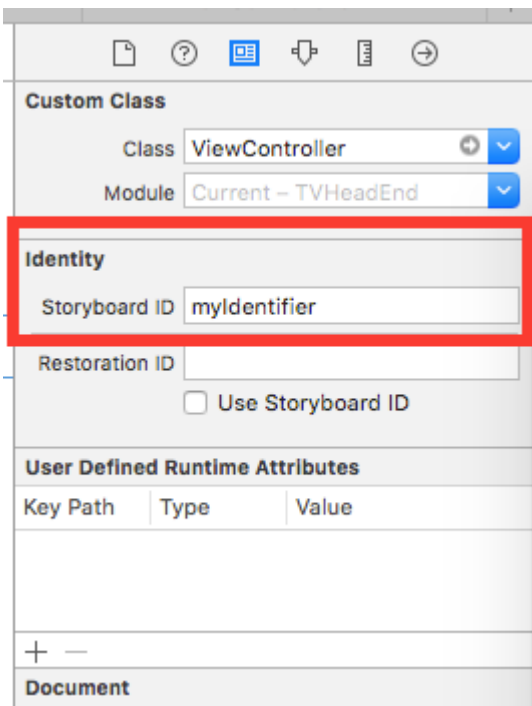
```
class FooViewController: UIViewController {

 override func loadView() {
 view = FooView()
 }

}
```

```
UINavigationController *storyboard = [UINavigationController storyboardWithName:@"Main" bundle:nil];
```

## Storyboard ID。

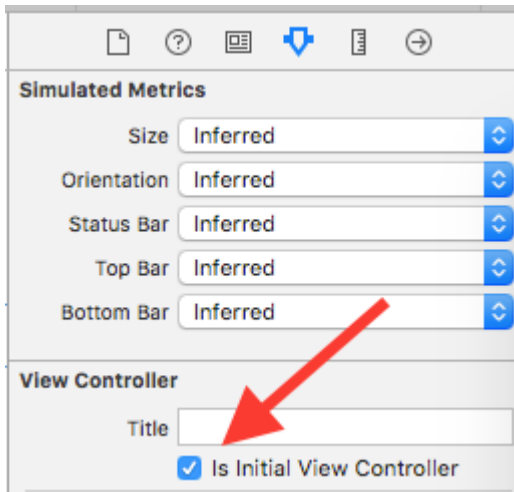


```
UIViewController *controller = [storyboard
instantiateViewControllerWithIdentifier:@"myIdentifier"];
```

## viewController

“”。





```
UIStoryboard *storyboard = [UIStoryboard storyboardWithName:@"Main" bundle:nil];
UIViewController *controller = [storyboard instantiateInitialViewController];
```

```
let tabBarController = viewController.tabBarController
```

## Objective-C

```
UITabBarController *tabBarController = self.tabBarController;
```

```
let navigationController = viewController.navigationController
```

## Objective-C

```
UINavigationController *navigationController = self.navigationController;
```

/

```
- (void)displayContentController:(UIViewController *)vc {
 [self addChildViewController:vc];
 vc.view.frame = self.view.frame;
 [self.view addSubview:vc.view];
 [vc didMoveToParentViewController:self];
}
```

```
- (void)hideContentController:(UIViewController *)vc {
 [vc willMoveToParentViewController:nil];
 [vc.view removeFromSuperview];
 [vc removeFromParentViewController];
}
```

**UIViewController** <https://riptutorial.com/zh-TW/ios/topic/1956/uiviewcontroller>

# 186:

## Examples

### Objective-C

UIImageViewControllerAirdrop ◦ UIImageViewControllerMessagesAirDrop ◦

UIImageViewController

```
UIImage *hatImage = [UIImage imageNamed:@"logo.png"];
if (hatImage)//checks if the image file is not nil
{
//Initialise a UIImageViewController
UIImageViewController *controller = [[UIImageViewController alloc]
initWithActivityItems:@[hatImage] applicationActivities:nil];
//Excludes following options from the UIImageViewController menu
NSArray *excludeActivities = @[UIActivityTypePostToWeibo,UIActivityTypePrint,
UIActivityTypeMail,UIActivityTypeMessage,UIActivityTypePostToTwitter,UIActivityTypePostToFacebook,

UIActivityTypeCopyToPasteboard,UIActivityTypeAssignToContact,

UIActivityTypeSaveToCameraRoll,UIActivityTypeAddToReadingList,

UIActivityTypePostToFlickr,UIActivityTypePostToVimeo,

UIActivityTypePostToTencentWeibo];
controller.excludedActivityTypes = excludeActivities;
[self presentViewController:controller animated:YES completion:nil];
}
```

```
if ((newImage) != nil)
{
 let activityVC = UIImageViewController(activityItems: [newImage],
applicationActivities: nil)
 activityVC.excludedActivityTypes =[UIActivityTypeAddToReadingList]
 self.presentViewController(activityVC, animated: true, completion: nil)
}
```

<https://riptutorial.com/zh-TW/ios/topic/7360/>

# 187:

## Examples

### UIScrollView / UITableView

1.

```
//Swift 2.0+
override func viewDidLoad() {
 super.viewDidLoad()

 NotificationCenter.defaultCenter().addObserver(self, selector:
#selector(YourVCClassName.keyboardWillShow(_:)), name: UIKeyboardWillShowNotification, object:
nil)
 NotificationCenter.defaultCenter().addObserver(self, selector:
#selector(YourVCClassName.keyboardWillHide(_:)), name: UIKeyboardWillHideNotification, object:
nil)
}

func keyboardWillShow(notification: NSNotification) {
 if let userInfo = notification.userInfo {
 if let keyboardHeight =
userInfo[UIKeyboardFrameEndUserInfoKey]?.CGRectValue.size.height {
 tableView.contentInset = UIEdgeInsetsMake(0, 0, keyboardHeight, 0)
 }
 }
}

func keyboardWillHide(notification: NSNotification) {
 tableView.contentInset = UIEdgeInsetsMake(0, 0, 0, 0)
}

//Objective-C
- (void)viewDidLoad {

 [super viewDidLoad];

 [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(keyboardWillShow:) name:UIKeyboardWillShowNotification object:nil];
 [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(keyboardWillHide:) name:UIKeyboardWillHideNotification object:nil];
}

- (void)keyboardWillShow:(NSNotification *)notification {

 NSDictionary *userInfo = [notification userInfo];

 if (userInfo) {

 CGRect keyboardEndFrame;
 [[userInfo objectForKey:UIKeyboardFrameEndUserInfoKey] getValue:&keyboardEndFrame];
 tableView.contentInset = UIEdgeInsetsMake(0, 0, keyboardEndFrame.size.height, 0);
 }
}
```

```

}

- (void)keyboardWillHide:(NSNotification *)notification {

 tableView.contentInset = UIEdgeInsetsMake(0, 0, 0, 0);

}

```

## 2. TPKeyboardAvoidingTableViewTPKeyboardAvoidingScrollView

<https://github.com/michaeltyson/TPKeyboardAvoiding>

### hackyObjective-C

```

- (void)viewDidLoad {
 [super viewDidLoad];

 // dismiss keyboard when tap outside a text field
 UITapGestureRecognizer *tapGestureRecognizer = [[UITapGestureRecognizer alloc]
initWithTarget:self.view action:@selector(endEditing:)];
 [tapGestureRecognizer setCancelsTouchesInView:NO];
 [self.view addGestureRecognizer:tapGestureRecognizer];
}

```

### Swift

```

override func viewDidLoad() {
 super.viewDidLoad()

 // dismiss keyboard when tap outside a text field
 let tapGestureRecognizer: UITapGestureRecognizer = UITapGestureRecognizer(target: self,
action: #selector(YourVCName.dismissKeyboard))
 view.addGestureRecognizer(tapGestureRecognizer)
}

//Calls this function when the tap is recognized.
func dismissKeyboard() {
 //Causes the view (or one of its embedded text fields) to resign the first responder
status.
 view.endEditing(true)
}

```

### Swift 3 / iOS 10

```

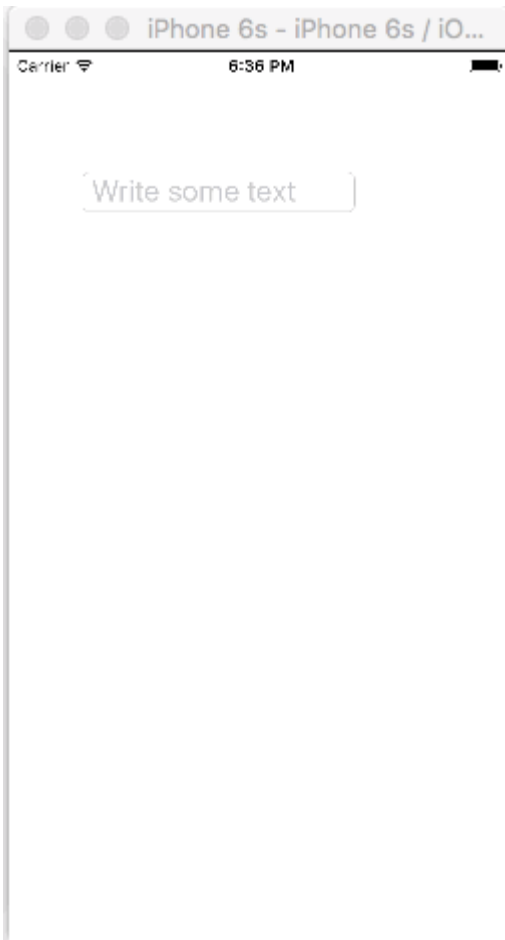
class vc: UIViewController {
 override func viewDidLoad() {
 super.viewDidLoad()
 // Do any additional setup after loading the view, typically from a nib.

 txtSomeField.delegate = self
 }
}

extension vc: UITextFieldDelegate {
 //Hide the keyboard for any text field when the UI is touched outside of the keyboard.
 override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?)

```

```
{
 self.view.endEditing(true) //Hide the keyboard
}
```



◦ ◦

- `.xibSwiftObjective-C` `UIView`
- `UITextField`
- ◦

---

## **.xib**

- Xcode>>...> **iOS**>>.xib
- Keyboard.xib
- ◦
- ◦
- Keyboard ◦ ◦ ◦

---

## **.swift UIView**

- XcodeFile> New> File ...> iOS> Source> Cocoa Touch ClassSwiftObjective-C ◦ UIView
- Keyboard.swift Objective-CKeyboard
- Swift

```
import UIKit

// The view controller will adopt this protocol (delegate)
// and thus must contain the keyWasTapped method
protocol KeyboardDelegate: class {
 func keyWasTapped(character: String)
}

class Keyboard: UIView {

 // This variable will be set as the view controller so that
 // the keyboard can send messages to the view controller.
 weak var delegate: KeyboardDelegate?

 // MARK:- keyboard initialization

 required init?(coder aDecoder: NSCoder) {
 super.init(coder: aDecoder)
 initializeSubviews()
 }

 override init(frame: CGRect) {
 super.init(frame: frame)
 initializeSubviews()
 }

 func initializeSubviews() {
 let xibName = "Keyboard" // xib extension not included
 let view = NSBundle.mainBundle().loadNibNamed(xibName, owner: self,
options: nil)[0] as! UIView
 self.addSubview(view)
 view.frame = self.bounds
 }

 // MARK:- Button actions from .xib file

 @IBAction func keyTapped(sender: UIButton) {
 // When a button is tapped, send that information to the
 // delegate (ie, the view controller)
 self.delegate?.keyWasTapped(sender.titleLabel!.text!) // could alternatively
send a tag value
 }

}
```

- Objective-C

## Keyboard.h

```
#import <UIKit/UIKit.h>

// The view controller will adopt this protocol (delegate)
```

```

// and thus must contain the keyWasTapped method
@protocol KeyboardDelegate<NSObject>
- (void)keyWasTapped:(NSString *)character;
@end

@interface Keyboard : UIView
@property (nonatomic, weak) id<KeyboardDelegate> delegate;
@end

```

## Keyboard.m

```

#import "Keyboard.h"

@implementation Keyboard

- (id)initWithCoder:(NSCoder *)aDecoder {
 self = [super initWithCoder:aDecoder];
 [self initializeSubviews];
 return self;
}

- (id)initWithFrame:(CGRect)frame {
 self = [super initWithFrame:frame];
 [self initializeSubviews];
 return self;
}

- (void)initializeSubviews {
 NSString *xibName = @"Keyboard"; // xib extension not included
 UIView *view = [[[NSBundle mainBundle] loadNibNamed:xibName owner:self
options:nil] firstObject];
 [self addSubview:view];
 view.frame = self.bounds;
}

// MARK:- Button actions from .xib file

- (IBAction)keyTapped:(UIButton *)sender {
 // When a button is tapped, send that information to the
 // delegate (ie, the view controller)
 [self.delegate keyWasTapped:sender.titleLabel.text]; // could alternatively send a
tag value
}

@end

```

- @IBAction **SwiftObjective-C** @IBAction°
- ° °

- 
- UITextFieldIBOutlet° textField°

- **SwiftView Controller**

```
import UIKit
```

```

class ViewController: UIViewController, KeyboardDelegate {

 @IBOutlet weak var textField: UITextField!

 override func viewDidLoad() {
 super.viewDidLoad()

 // initialize custom keyboard
 let keyboardView = Keyboard(frame: CGRect(x: 0, y: 0, width: 0, height: 300))
 keyboardView.delegate = self // the view controller will be notified by the
 keyboard whenever a key is tapped

 // replace system keyboard with custom keyboard
 textField.inputView = keyboardView
 }

 // required method for keyboard delegate protocol
 func keyWasTapped(character: String) {
 textField.insertText(character)
 }
}

```

- Objective-C

### .h

```

#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

@end

```

### .m

```

#import "ViewController.h"
#import "Keyboard.h"

@interface ViewController ()<KeyboardDelegate>

@property (nonatomic, weak) IBOutlet UITextField *textField;

@end

@implementation ViewController

- (void)viewDidLoad {
 [super viewDidLoad];
 // Do any additional setup after loading the view, typically from a nib.

 // initialize custom keyboard
 Keyboard *keyboardView = [[Keyboard alloc] initWithFrame:CGRectMake(0, 0, 0, 300)];
 keyboardView.delegate = self; // the view controller will be notified by the keyboard
 whenever a key is tapped

 // replace system keyboard with custom keyboard
 self.textField.inputView = keyboardView;
}

```



```

- (void)keyWasTapped:(NSString *)character {
 [self.textField insertText:character];
}

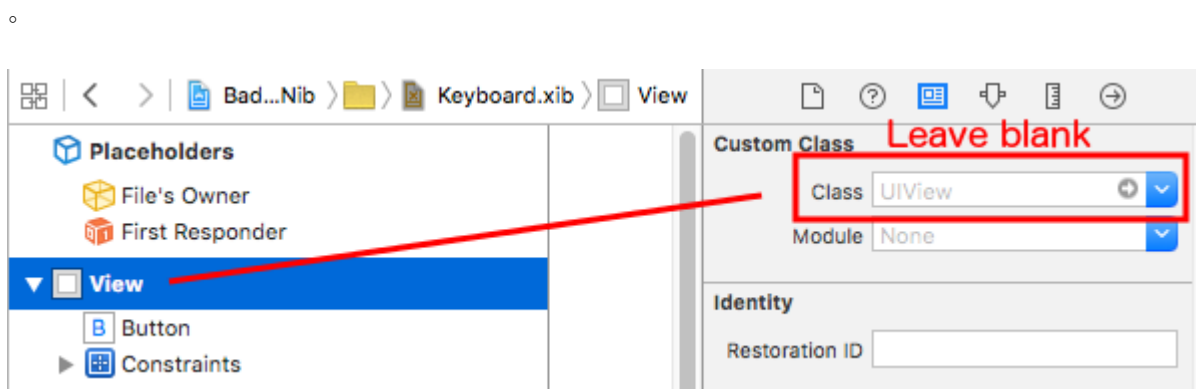
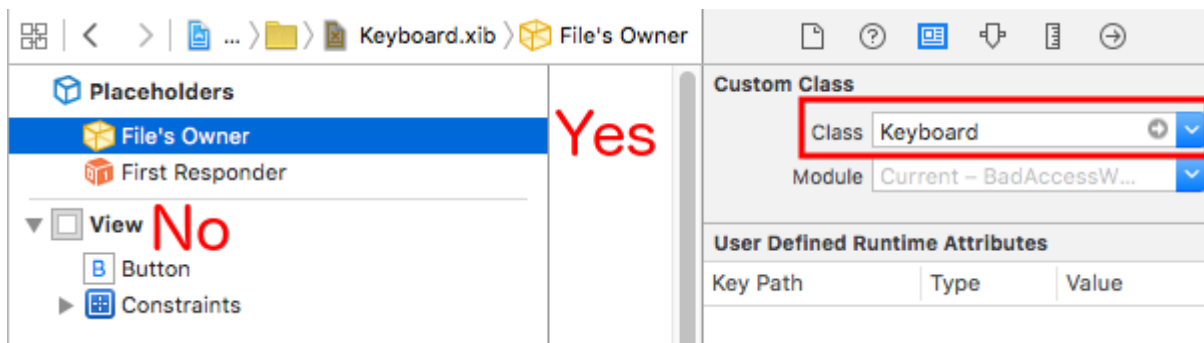
@end

```

- KeyboardDelegate

EXC\_BAD\_ACCESSKeyboardnib

Keyboard.nib File's Owner



[Stack Overflow](#)

## Singleton + Delegate

ViewControllerNotifications

NSNotification

```

class ViewController: UIViewController {
 override func viewDidLoad() {
 super.viewDidLoad()
 NotificationCenter.defaultCenter().addObserver(self, selector:
#selector(ViewController.keyboardNotification(_:)), name:
UIKeyboardWillChangeFrameNotification, object: nil)
 }
}

```

```

func keyboardNotification(notification: NSNotification) {
 guard let userInfo = notification.userInfo else { return }

 let endFrame = (userInfo[UIKeyboardFrameEndUserInfoKey] as? NSValue)?.CGRectValue()
 let duration: NSTimeInterval = (userInfo[UIKeyboardAnimationDurationUserInfoKey] as?
NSNumber)?.doubleValue ?? 0
 let animationCurveRawNSN = userInfo[UIKeyboardAnimationCurveUserInfoKey] as? NSNumber
 let animationCurveRaw = animationCurveRawNSN?.unsignedLongValue ??
UIViewAnimationOptions.CurveEaseOut.rawValue
 let animationCurve: UIViewAnimationOptions = UIViewAnimationOptions(rawValue:
animationCurveRaw)

 if endFrame?.origin.y >= UIScreen.mainScreen().bounds.size.height {
 lowerViewBottomConstraint.constant = 0
 } else {
 lowerViewBottomConstraint.constant = endFrame?.size.height ?? 0.0
 }
 view.animateConstraintWithDuration(duration, delay: NSTimeInterval(0), options:
animationCurve, completion: nil)
}
}

```

## ViewController。 Singleton + Delegate

+

```

protocol KeyboardManagerDelegate: class {
 func keyboardWillChangeFrame(endFrame: CGRect?, duration: NSTimeInterval, animationCurve:
UIViewAnimationOptions)
}

class KeyboardManager {

 weak var delegate: KeyboardManagerDelegate?

 class var sharedInstance: KeyboardManager {
 struct Singleton {
 static let instance = KeyboardManager()
 }
 return Singleton.instance
 }

 init() {
 NotificationCenter.defaultCenter().addObserver(self, selector:
#selector(KeyboardManager.keyboardWillChangeFrameNotification(_:)), name:
UIKeyboardWillChangeFrameNotification, object: nil)
 }

 @objc func keyboardWillChangeFrameNotification(notification: NSNotification) {
 guard let userInfo = notification.userInfo else { return }

 let endFrame = (userInfo[UIKeyboardFrameEndUserInfoKey] as? NSValue)?.CGRectValue()
 let duration: NSTimeInterval = (userInfo[UIKeyboardAnimationDurationUserInfoKey] as?
NSNumber)?.doubleValue ?? 0
 let animationCurveRawNSN = userInfo[UIKeyboardAnimationCurveUserInfoKey] as? NSNumber
 let animationCurveRaw = animationCurveRawNSN?.unsignedLongValue ??
UIViewAnimationOptions.CurveEaseOut.rawValue
 let animationCurve: UIViewAnimationOptions = UIViewAnimationOptions(rawValue:
animationCurveRaw)
 }
}

```

```

 delegate?.keyboardWillChangeFrame(endFrame, duration: duration, animationCurve:
animationCurve)
 }
}

```

## ViewControllerViewController。

```

class ViewController: UIViewController {
 override func viewWillAppear(animated: Bool) {
 super.viewWillAppear(animated)
 KeyboardManager.sharedInstance.delegate = self
 }
}

// MARK: - Keyboard Manager

extension ViewController: KeyboardManagerDelegate {
 func keyboardWillChangeFrame(endFrame: CGRect?, duration: NSTimeInterval, animationCurve:
UIViewAnimationOptions) {
 if endFrame?.origin.y >= UIScreen.mainScreen().bounds.size.height {
 lowerViewBottomConstraint.constant = 0
 } else {
 lowerViewBottomConstraint.constant = (endFrame?.size.height ?? 0.0)
 }
 view.animateConstraintWithDuration(duration, delay: NSTimeInterval(0), options:
animationCurve, completion: nil)
 }
}

```

UIKeyboardWillHideNotification ◦ KeyboardManagerDelegate ◦

KeyboardManagerDelegateUIKeyboardWillHideNotification

```

protocol KeyboardManagerDelegate: class {
 func keyboardWillChangeFrame(endFrame: CGRect?, duration: NSTimeInterval, animationCurve:
UIViewAnimationOptions)
 func keyboardWillHide(notificationUserInfo: [NSObject: AnyObject])
}

class KeyboardManager {
 init() {
 NotificationCenter.defaultCenter().addObserver(self, selector:
#selector(KeyboardManager.keyboardWillChangeFrameNotification(_:)), name:
UIKeyboardWillChangeFrameNotification, object: nil)
 NotificationCenter.defaultCenter().addObserver(self, selector:
#selector(KeyboardManager.keyboardWillHide(_:)), name: UIKeyboardWillHideNotification, object:
nil)
 }

 func keyboardWillHide(notification: NSNotification) {
 guard let userInfo = notification.userInfo else { return }
 delegate?.keyboardWillHide(userInfo)
 }
}

```

ViewControllerfunc keyboardWillHide(notificationUserInfo: [NSObject: AnyObject]) ◦ ◦

```

typealias KeyboardManagerDelegate = protocol<KeyboardManagerModel,
KeyboardManagerConfigurable>

protocol KeyboardManagerModel: class {
 func keyboardWillChangeFrame(endFrame: CGRect?, duration: NSTimeInterval, animationCurve:
UIViewAnimationOptions)
}

@objc protocol KeyboardManagerConfigurable {
 optional func keyboardWillHide(userInfo: [NSObject: AnyObject])
}

```

\*@objc ◦ <http://www.jessesquires.com/avoiding-objc-in-swift/>

Singleton + DelegateNotifications

## iOS

### UIViewController

```

func keyboardWillShow(notification: NSNotification) {

 if let keyboardSize = (notification.userInfo?[UIKeyboardFrameBeginUserInfoKey] as?
NSValue)?.cgRectValue {
 if self.view.frame.origin.y == 0{
 self.view.frame.origin.y -= keyboardSize.height
 }
 }
}

func keyboardWillHide(notification: NSNotification) {
 if let keyboardSize = (notification.userInfo?[UIKeyboardFrameBeginUserInfoKey] as?
NSValue)?.cgRectValue {
 if self.view.frame.origin.y != 0{
 self.view.frame.origin.y += keyboardSize.height
 }
 }
}

```

viewDidLoad()

```

NotificationCenter.default.addObserver(self, selector: #selector(Login.keyboardWillShow),
name: NSNotification.Name.UIKeyboardWillShow, object: nil)
NotificationCenter.default.addObserver(self, selector: #selector(Login.keyboardWillHide),
name: NSNotification.Name.UIKeyboardWillHide, object: nil)

```

height◦

## Objective-C

Objective-C

```

- (void) viewWillAppear:(BOOL)animated {
 [super viewWillAppear:animated];
 [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(keyboardWillShow:) name:UIKeyboardWillShowNotification object:nil];
 [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(keyboardWillHide:) name:UIKeyboardWillHideNotification object:nil];
}

- (void) viewWillDisappear:(BOOL)animated {
 [super viewWillDisappear:animated];
 [[NSNotificationCenter defaultCenter] removeObserver:self
name:UIKeyboardWillShowNotification object:nil];
 [[NSNotificationCenter defaultCenter] removeObserver:self
name:UIKeyboardWillHideNotification object:nil];
}

- (void) keyboardWillShow:(NSNotification *)notification
{
 CGSize keyboardSize = [[[notification userInfo]
objectForKey:UIKeyboardFrameBeginUserInfoKey] CGRectValue].size;

 [UIView animateWithDuration:0.3 animations:^(
 CGRect f = self.view.frame;
 f.origin.y = -keyboardSize.height;
 self.view.frame = f;
)];
}

- (void) keyboardWillHide:(NSNotification *)notification
{
 [UIView animateWithDuration:0.3 animations:^(
 CGRect f = self.view.frame;
 f.origin.y = 0.0f;
 self.view.frame = f;
)];
}

```

<https://riptutorial.com/zh-TW/ios/topic/436/>

# 188:

## Examples

### AlamofireImage

AlamofireImage° SwiftAlamofire° cocoapodsAlamofireImage

```
pod 'AlamofireImage', '~> 3.1'
```

1. AlamofireImageAlamofire
2. **SetUp** let imageCache = AutoPurgingImageCache( memoryCapacity: 111\_111\_111, preferredMemoryUsageAfterPurge: 90\_000\_000)
- 3.

```
Alamofire.request(self.nameUrl[i]).responseImage { response in
 if response.result.value != nil {
 let image = UIImage(data: response.data!, scale: 1.0)!
 imageCache.add(image, withIdentifier: self.nameUrl[i])
 }
}
```

- 4.

```
if let image = imageCache.image(withIdentifier: self.nameUrl[self.a])
{
 self.localImageView.image = image
}
```

<https://riptutorial.com/zh-TW/ios/topic/9450/>

---

# 189:

---

- [Apple](#)
- [Stack Overflow](#)
- [WWDC15](#)

## Examples

---

```
import Contacts
```

## Objective-C

```
#import <Contacts/Contacts.h>
```

```
switch (CNContactStore.authorizationStatusForEntityType (CNEntityType.Contacts)) {
 case .Authorized: //access contacts
 case .Denied, .NotDetermined: //request permission
 default: break
}
```

## Objective-C

```
switch ([CNContactStore authorizationStatusForEntityType:CNEntityType.Contacts]) {
 case CNAuthorizationStatus.Authorized:
 //access contacts
 break;
 case CNAuthorizationStatus.Denied:
 //request permission
 break;
 case CNAuthorizationStatus.NotDetermined:
 //request permission
 break;
}
```

```
var contactStore = CKContactStore()
contactStore.requestAccessForEntityType (CKEntityType.Contacts, completionHandler: { (ok, _) ->
Void in
 if access{
 //access contacts
 }
})
```

---

NSPredicate @"contactStore"

```
let predicate = CNContact.predicateForContactsMatchingName("Some Name")
```

## Objective-C

```
NSPredicate *predicate = [CNContact predicateForContactsMatchingName:@"Some Name"];
```

---

```
let keys = [CNContactGivenNameKey, CNContactFamilyNameKey, CNContactImageDataKey]
```

---

```
do {
 let contacts = try contactStore.unifiedContactsMatchingPredicate(predicate, keysToFetch:
keys)
} catch let error as NSError {
 //...
}
```

---

```
print(contacts[0].givenName)
print(contacts[1].familyName)
let image = contacts[2].imageData
```

---

```
import Contacts

// Creating a mutable object to add to the contact
let contact = CNMutableContact()

contact.imageData = NSData() // The profile picture as a NSData object

contact.givenName = "John"
contact.familyName = "Appleseed"

let homeEmail = CNLabeledValue(label:CNLabelHome, value:"john@example.com")
let workEmail = CNLabeledValue(label:CNLabelWork, value:"j.appleseed@icloud.com")
contact.emailAddresses = [homeEmail, workEmail]

contact.phoneNumbers = [CNLabeledValue(
 label:CNLabelPhoneNumberiPhone,
 value:CNPhoneNumber(stringValue:"(408) 555-0126"))]

let homeAddress = CNMutablePostalAddress()
homeAddress.street = "1 Infinite Loop"
homeAddress.city = "Cupertino"
homeAddress.state = "CA"
homeAddress.postalCode = "95014"
contact.postalAddresses = [CNLabeledValue(label:CNLabelHome, value:homeAddress)]
```



```
let birthday = NSDateComponents()
birthday.day = 1
birthday.month = 4
birthday.year = 1988 // You can omit the year value for a yearless birthday
contact.birthday = birthday

// Saving the newly created contact
let store = CNContactStore()
let saveRequest = CNSaveRequest()
saveRequest.addContact(contact, toContainerWithIdentifier:nil)
try! store.executeSaveRequest(saveRequest)
```

<https://riptutorial.com/zh-TW/ios/topic/5872/>



# 190:

◦ ◦

## Examples


1. Xcode◦
2. ◦
3. ◦

---

  O. ⚡

General      **Capabilities**      Resource Tags

---

▼  **Background Modes**

Modes:  Audio, AirPlay, and Picture in Picture  
 Location updates  
 Voice over IP  
 Newsstand downloads  
 External accessory communication  
 Uses Bluetooth LE accessories  
 Acts as a Bluetooth LE accessory  
 Background fetch  
 Remote notifications

---

Steps: ✓ Add the Required Background Modes

---

◦ Feed◦

1-Xcode◦

2-application(\_:didFinishLaunchingWithOptions:) AppDelegate

---

```
UIApplication.shared.setMinimumBackgroundFetchInterval(UINavigationControllerBackgroundFetchIntervalMinimum)
```

---

## Objective-C

```
[[UIApplication shared]
 setMinimumBackgroundFetchInterval:UINavigationControllerBackgroundFetchIntervalMinimum]
```

CGFloatUINavigationControllerBackgroundFetchIntervalMinimum ◦

3-application(\_:performFetchWithCompletionHandler:) ◦ AppDelegate

---

```
func application(_ application: UINavigationController, performFetchWithCompletionHandler
 completionHandler: @escaping (UINavigationControllerBackgroundFetchResult) -> Void) {
 // your code here
}
```

1-Xcode ◦

2-“” ”

Pause

⌘ Y

Continue To Current Line

⌘ C

Step Over

F6

Step Into

F7

Step Out

F8

Step Over Instruction

⌘ F6

Step Over Thread

⌘ ↑ F6

Step Into Instruction

⌘ F7

Step Into Thread

⌘ ↑ F7

Capture GPU Frame

GPU Overrides



Simulate Location



Simulate Background Fetch

Simulate UI Snapshot

iCloud



View Debugging



Deactivate Breakpoints

⌘ Y

Breakpoints



Debug Workflow



# 191:

## Examples

.plistRequired background modes.

Key	Type	Value
▼ Information Property List	Dictionary	(16 items)
Localization native development r...	String	en
Bundle display name	String	
Executable file	String	\${EXECUTABLE_NAME}
▶ Icon files	Array	(14 items)
Bundle identifier	String	
InfoDictionary version	String	6.0
Bundle name	String	\${PRODUCT_NAME}
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.1
Bundle creator OS Type code	String	????
Bundle version	String	1.1
Application requires iPhone envir...	Boolean	YES
▼ Required background modes	Array	(1 item)
Item 0	String	App plays audio or streams audio/video using AirPlay
Icon already includes gloss effects	Boolean	YES
▶ Required device capabilities	Array	(1 item)
▶ Supported interface orientations	Array	(1 item)

## AppDelegate.h

```
#import <AVFoundation/AVFoundation.h>
#import <AudioToolbox/AudioToolbox.h>
```

## AppDelegate.m

### didFinishLaunchingWithOptions

```
[[AVAudioSession sharedInstance] setDelegate:self];
[[AVAudioSession sharedInstance] setCategory:AVAudioSessionCategoryPlayback error:nil];
[[AVAudioSession sharedInstance] setActive:YES error:nil];
[[UIApplication sharedApplication] beginReceivingRemoteControlEvents];

UInt32 size = sizeof(CFStringRef);
CFStringRef route;
AudioSessionGetProperty(kAudioSessionProperty_AudioRoute, &size, &route);
NSLog(@"route = %@", route);
```

## AppDelegate.m

```
- (void)remoteControlReceivedWithEvent:(UIEvent *)theEvent {

 if (theEvent.type == UIEventTypeRemoteControl) {
 switch(theEvent.subtype) {
 case UIEventSubtypeRemoteControlPlay:
 [[NSNotificationCenter defaultCenter] postNotificationName:@"TogglePlayPause"
object:nil];
 break;
 case UIEventSubtypeRemoteControlPause:
 [[NSNotificationCenter defaultCenter] postNotificationName:@"TogglePlayPause"
object:nil];
 break;
 case UIEventSubtypeRemoteControlStop:
 break;
 case UIEventSubtypeRemoteControlTogglePlayPause:
 [[NSNotificationCenter defaultCenter] postNotificationName:@"TogglePlayPause"
object:nil];
 break;
 default:
 return;
 }
 }
}
```

..

<https://riptutorial.com/zh-TW/ios/topic/3515/>

# 192:

“”。

- NSLayoutConstraintItemAnyNSLayoutConstraintAttributeRelatedByNSLayoutConstraintRelationToItemAnyAttribute  
NSLayoutConstraintAttributeMultiplierCGFloatConstantCGFloat//

## Examples

### Boilerplate

```
override func viewDidLoad() {
 super.viewDidLoad()

 let myView = UIView()
 myView.backgroundColor = UIColor.blueColor()
 myView.translatesAutoresizingMaskIntoConstraints = false
 view.addSubview(myView)

 // Add constraints code here
 // ...
}
```

### Anchor Style NSLayoutConstraint Style iOS 9 iOS 8 NSLayoutConstraint Style。

```
let margins = view.layoutMarginsGuide
myView.leadingAnchor.constraintEqualToAnchor(margins.leadingAnchor, constant: 20).active = true
```

- `leadingAnchor trailingAnchor topAnchor bottomAnchor`。

### *NSLayoutConstraint*

```
NSLayoutConstraint(item: myView, attribute: NSLayoutConstraintAttribute.Leading, relatedBy:
NSLayoutConstraintRelation.Equal, toItem: view, attribute: NSLayoutConstraintAttribute.LeadingMargin, multiplier:
1.0, constant: 20.0).active = true
```

- `.Leading.Trailing .Top.Bottom`。
- `.LeadingMargin.TrailingMargin .TopMargin.BottomMargin`。

```
NSLayoutConstraint.constraintsWithVisualFormat("H:|-20-[myViewKey]", options: [], metrics:
nil, views: ["myViewKey": myView])
```

```
myView.widthAnchor.constraintEqualToAnchor(nil, constant: 200).active = true
myView.heightAnchor.constraintEqualToAnchor(nil, constant: 100).active = true
```

## NSLayoutConstraint

```
NSLayoutConstraint(item: myView, attribute: NSLayoutConstraint.Width, relatedBy:
NSLayoutConstraint.Relation.Equal, toItem: nil, attribute: NSLayoutConstraint.NotAnAttribute, multiplier:
1, constant: 200).active = true
NSLayoutConstraint(item: myView, attribute: NSLayoutConstraint.Height, relatedBy:
NSLayoutConstraint.Relation.Equal, toItem: nil, attribute: NSLayoutConstraint.NotAnAttribute, multiplier:
1, constant: 100).active = true
```

```
NSLayoutConstraint.constraintsWithVisualFormat("H:[myViewKey(200)]", options: [], metrics:
nil, views: ["myViewKey": myView])
NSLayoutConstraint.constraintsWithVisualFormat("V:[myViewKey(100)]", options: [], metrics:
nil, views: ["myViewKey": myView])
```

```
myView.centerXAnchor.constraintEqualToAnchor(view.centerXAnchor).active = true
myView.centerYAnchor.constraintEqualToAnchor(view.centerYAnchor).active = true
```

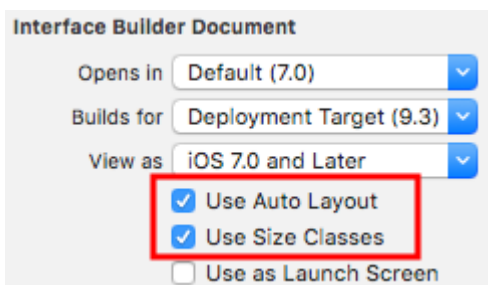
## NSLayoutConstraint

```
NSLayoutConstraint(item: myView, attribute: NSLayoutConstraint.CenterX, relatedBy:
NSLayoutConstraint.Relation.Equal, toItem: view, attribute: NSLayoutConstraint.CenterX, multiplier: 1,
constant: 0).active = true
NSLayoutConstraint(item: myView, attribute: NSLayoutConstraint.CenterY, relatedBy:
NSLayoutConstraint.Relation.Equal, toItem: view, attribute: NSLayoutConstraint.CenterY, multiplier: 1,
constant: 0).active = true
```

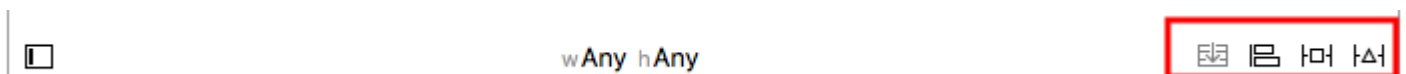
```
NSLayoutConstraint.constraintsWithVisualFormat("V:[viewKey]-(<=0)-[myViewKey]", options:
NSLayoutConstraint.FormatOptions.AlignAllCenterX, metrics: nil, views: ["myViewKey": myView, "viewKey":
view])
NSLayoutConstraint.constraintsWithVisualFormat("H:[viewKey]-(<=0)-[myViewKey]", options:
NSLayoutConstraint.FormatOptions.AlignAllCenterY, metrics: nil, views: ["myViewKey": myView, "viewKey":
view])
```

◦ ◦ ◦

◦ Project Navigator *Main.storyboard* ◦ “”””



Interface Builder ◦ Interface Builder ◦ ◦

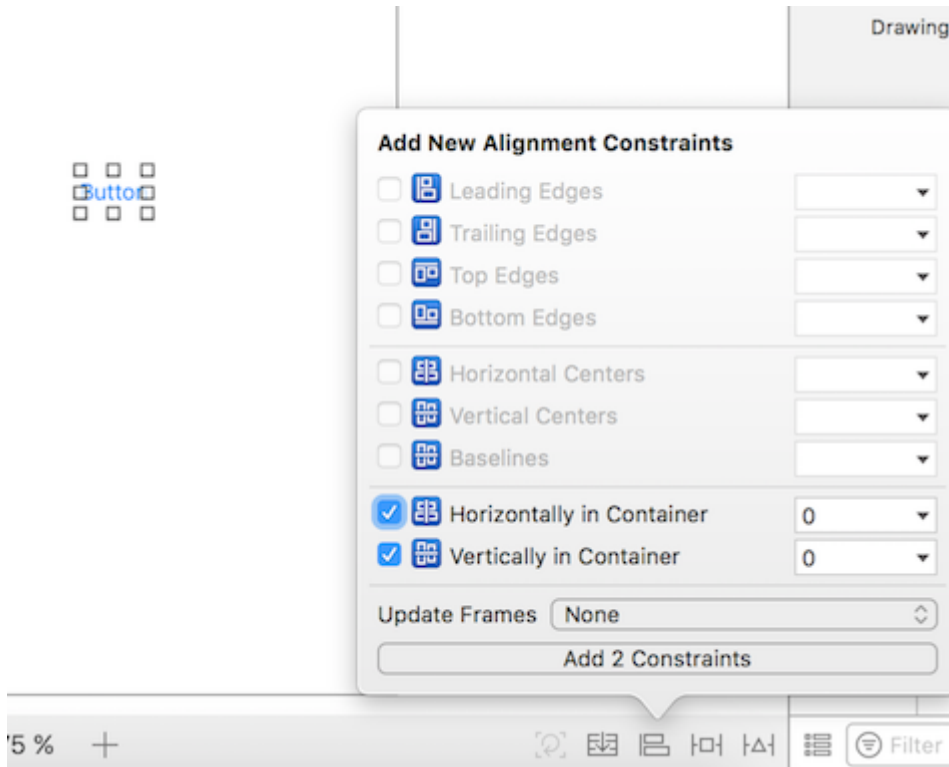


wAny hAny Size Class ◦

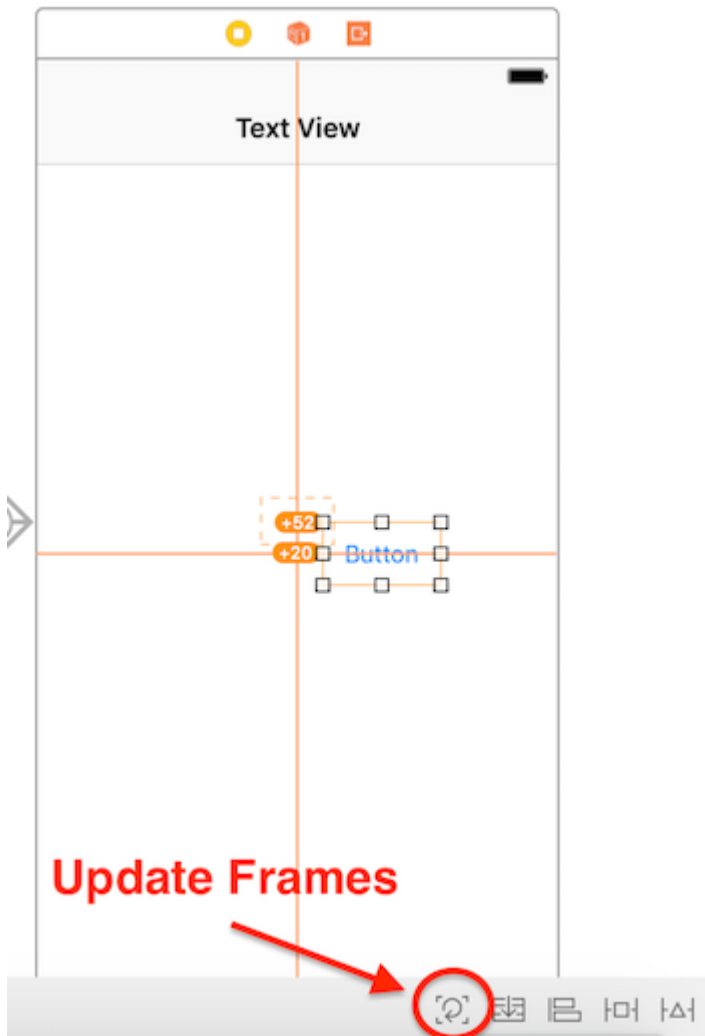




- ◦ Horizontally in Container Horizontally in Container Vertically in Container ◦ “2”◦



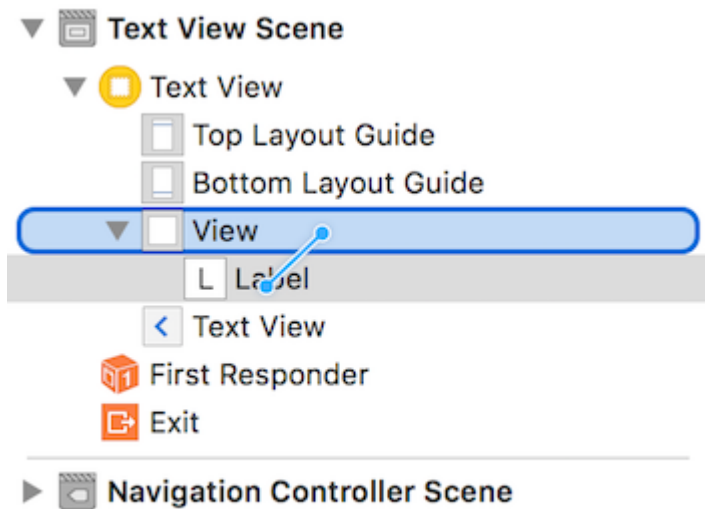
- “”◦

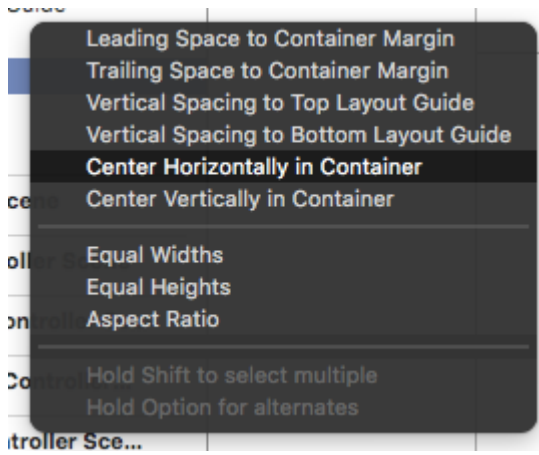


⌘ + ⌘ + = Command + Option+equals""。

。

Interface Builder ◦ UILabel ◦ Document Outline ◦ ctrl





“”。

。 UI

## Objective-C

```

- (void)viewDidLoad
{
 [super viewDidLoad];
 UIButton *yourButton = [[UIButton alloc] initWithFrame:CGRectMake(0, 0, 100, 18)];
 [yourButton setTitle:@"Button" forState:UIControlStateNormal];

 [self.view addConstraint:[NSLayoutConstraint constraintWithItem:yourButton
 attribute:NSLayoutAttributeCenterY relatedBy:NSLayoutRelationEqual toItem:self.view
 attribute:NSLayoutAttributeCenterY multiplier:1 constant:0]]; //Align vertically center to
 superView

 [self.view addConstraint:[NSLayoutConstraint constraintWithItem:yourButton
 attribute:NSLayoutAttributeCenterX relatedBy:NSLayoutRelationEqual toItem:self.view
 attribute:NSLayoutAttributeCenterX multiplier:1 constant:0]]; //Align horizontally center to
 superView

 [self.view addSubview:yourButton]; //Add button to superView
}

```

```

override func viewDidLoad()
{
 super.viewDidLoad()
 let yourButton: UIButton = UIButton(frame: CGRect(x: 0, y: 0, width: 100, height: 18))
 yourButton.setTitle("Button", forState: .Normal)

 let centerVertically = NSLayoutConstraint(item: yourButton,
 attribute: .CenterX,
 relatedBy: .Equal,
 toItem: view,
 attribute: .CenterX,
 multiplier: 1.0,
 constant: 0.0)

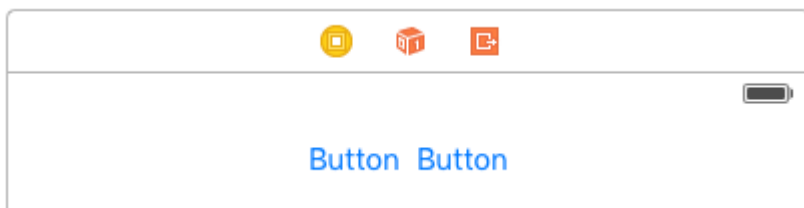
 let centerHorizontally = NSLayoutConstraint(item: yourButton,
 attribute: .CenterY,
 relatedBy: .Equal,
 toItem: view,
 attribute: .CenterY,

```

```

 multiplier: 1.0,
 constant: 0.0)
 NSLayoutConstraint.activateConstraints([centerVertically, centerHorizontally])
}

```



◦ [Stack Overflow](#) [UIView](#) [UIView](#) ◦ ◦ [UILayoutGuide](#)

◦ ◦ ◦ ◦

[UILayoutGuide](#) [UIView](#) ◦ [UILayoutGuide](#) ◦ [UILayoutGuide](#) ◦ ◦

```

view.addSubview(button1)
view.addSubview(button2)

let leftSpace = UILayoutGuide()
view.addLayoutGuide(leftSpace)

let rightSpace = UILayoutGuide()
view.addLayoutGuide(rightSpace)

let views = [
 "leftSpace" : leftSpace,
 "button1" : button1,
 "button2" : button2,
 "rightSpace" : rightSpace
]

// Lay the buttons and layout guides out horizontally in a line.
// Put the layout guides on each end.
NSLayoutConstraint.activateConstraints(NSLayoutConstraint.constraintsWithVisualFormat("H:|[leftSpace][button1][button2][rightSpace]|", options: [], metrics: nil, views: views))

// Now set the layout guides widths equal, so that the space on the
// left and the right of the buttons will be equal
leftSpace.widthAnchor.constraintEqualToAnchor(rightSpace.widthAnchor).active = true

```

```

let leadingSpace = UILayoutGuide()
let trailingSpace = UILayoutGuide()
view.addLayoutGuide(leadingSpace)
view.addLayoutGuide(trailingSpace)

leadingSpace.widthAnchor.constraintEqualToAnchor(trailingSpace.widthAnchor).active = true

leadingSpace.leadingAnchor.constraintEqualToAnchor(view.leadingAnchor).active = true
leadingSpace.trailingAnchor.constraintEqualToAnchor(button1.leadingAnchor).active = true

trailingSpace.leadingAnchor.constraintEqualToAnchor(button2.trailingAnchor).active = true
trailingSpace.trailingAnchor.constraintEqualToAnchor(view.trailingAnchor).active = true

```

“”CPU。 ◦

iOS 8。 iOS 9。

Interface BuilderXcode 7.2.1。 ◦ ◦

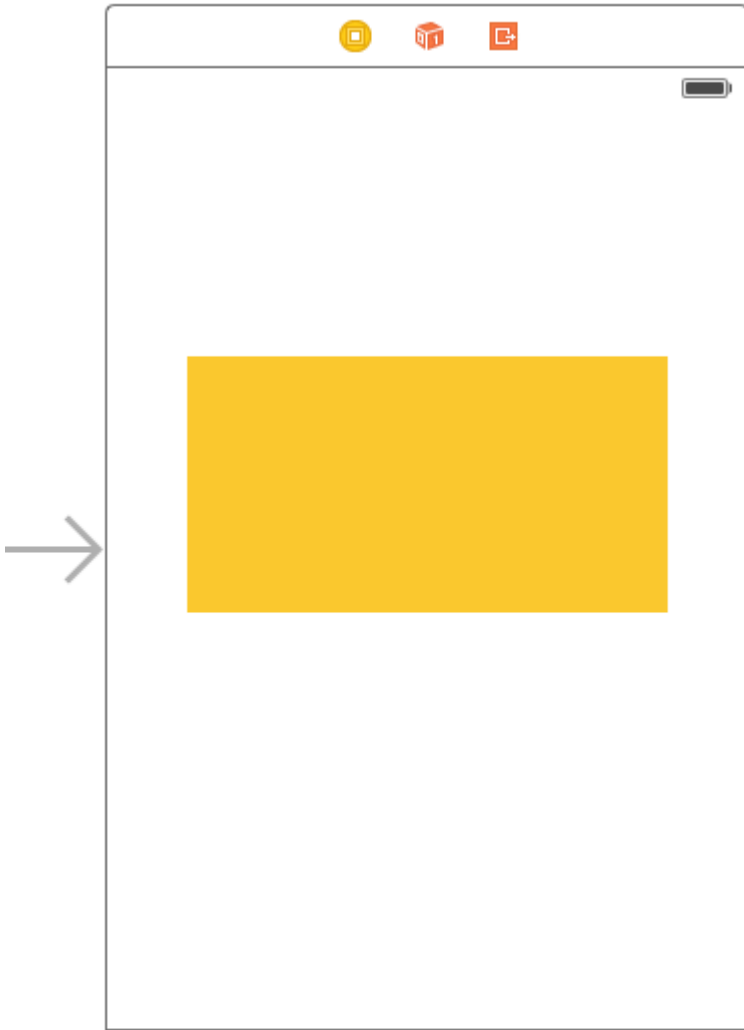
## **UILabel**

◦ text。 Image + text。

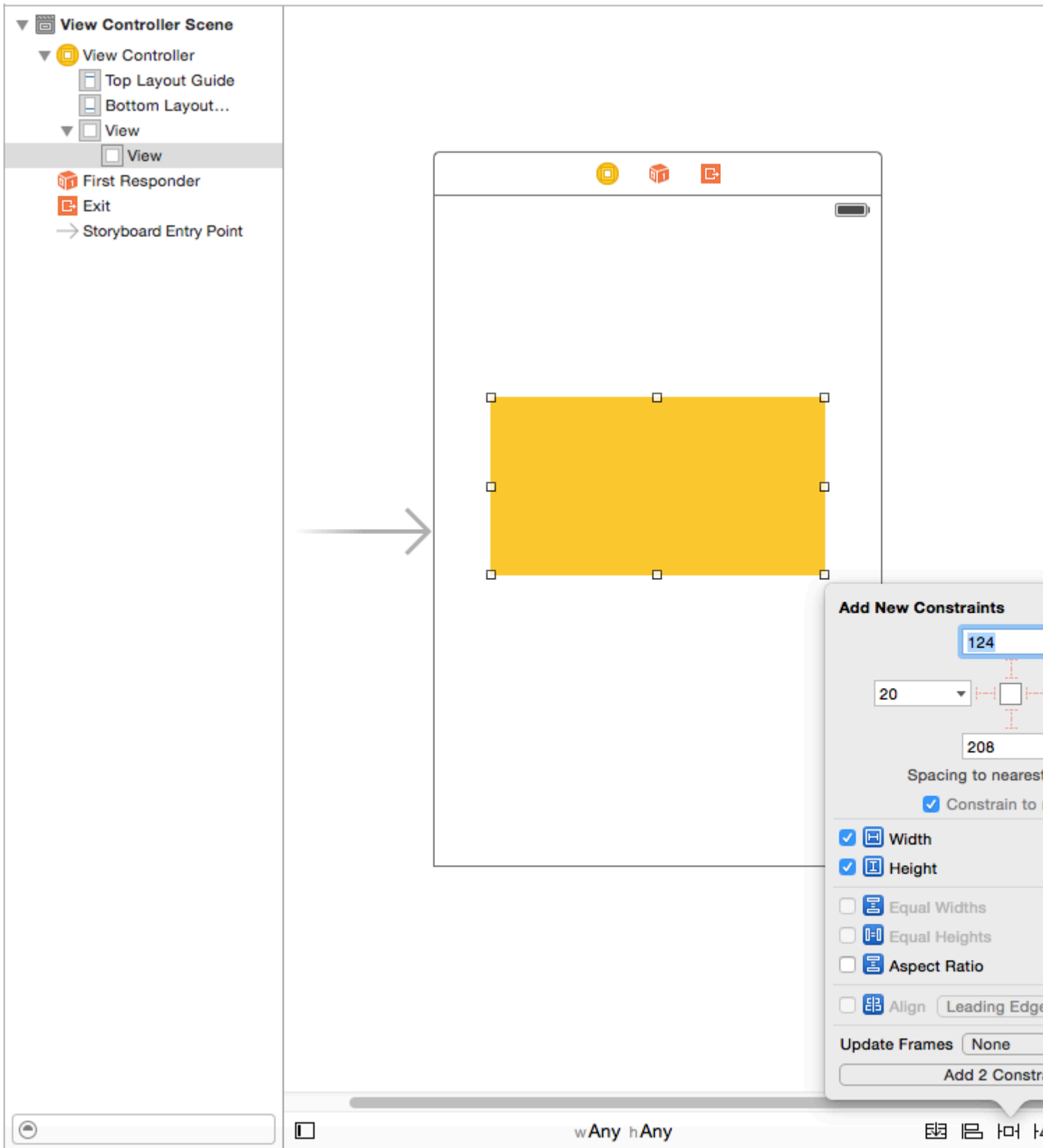


---

1. ◦ 3.5" ◦ view

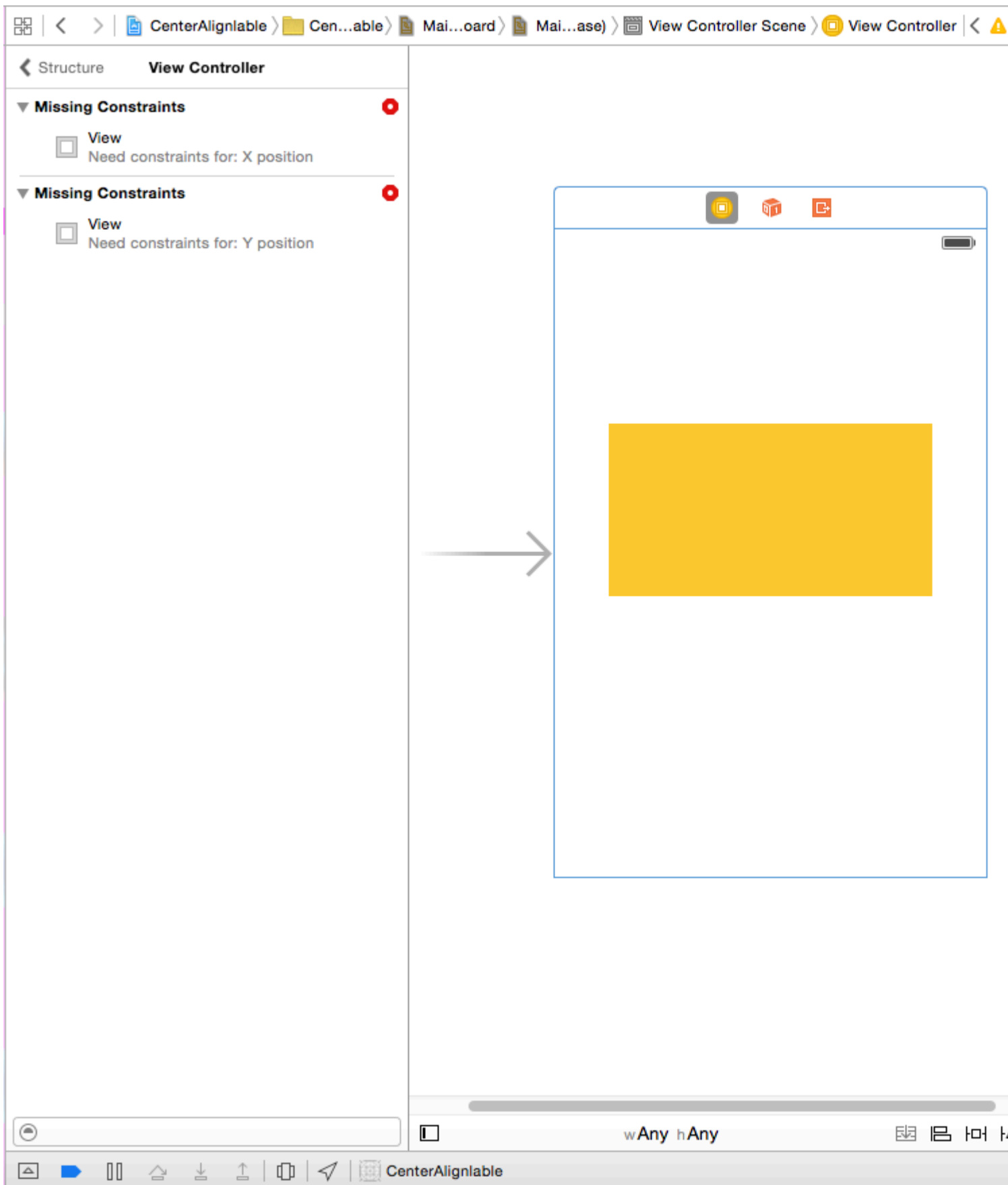


2. ◦



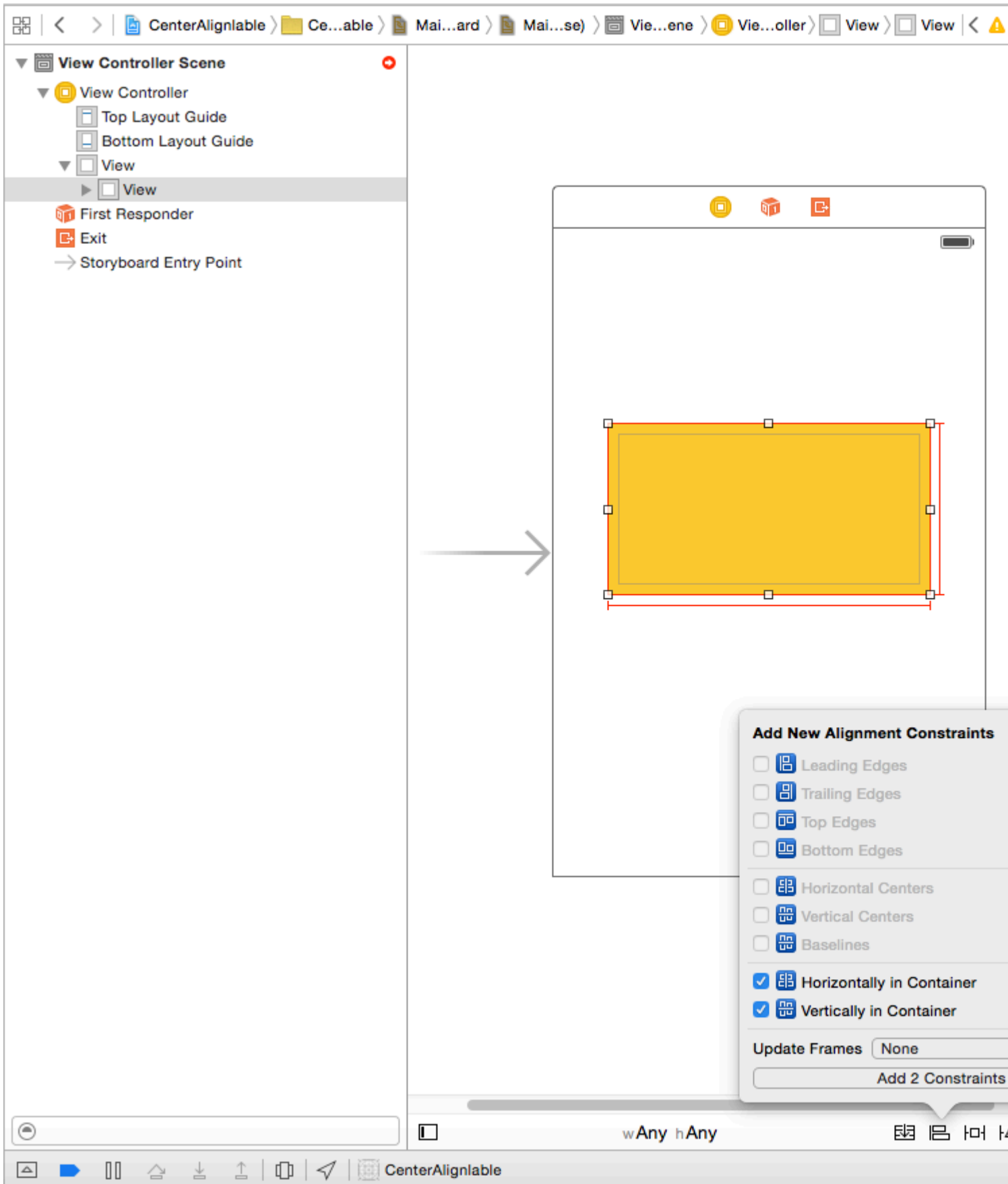
XCode.





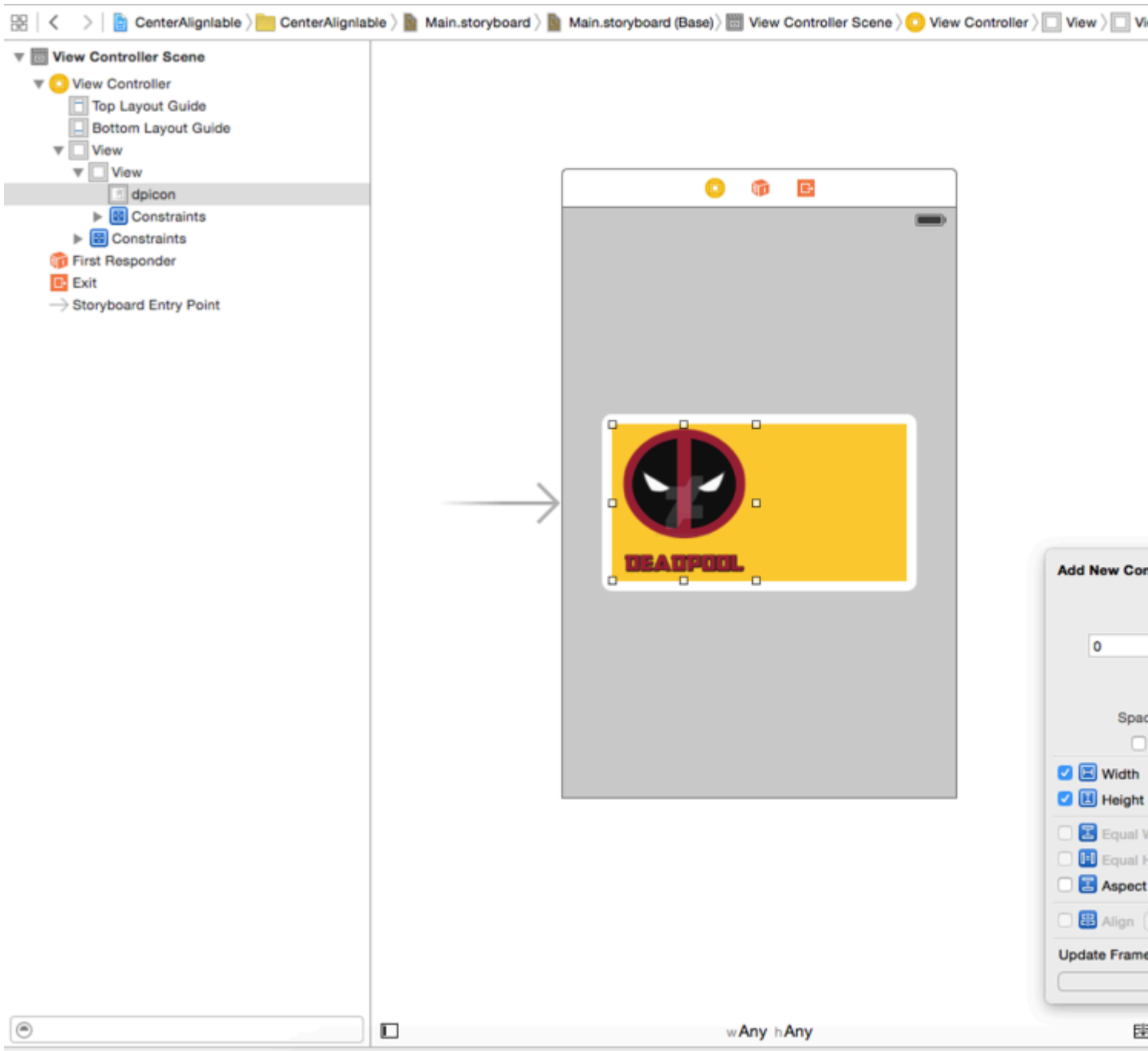
X - "BOUNDS" "FRAME". AutolayoutX

Y - "BOUNDS" "FRAME". AutolayoutY autolayoutXY.



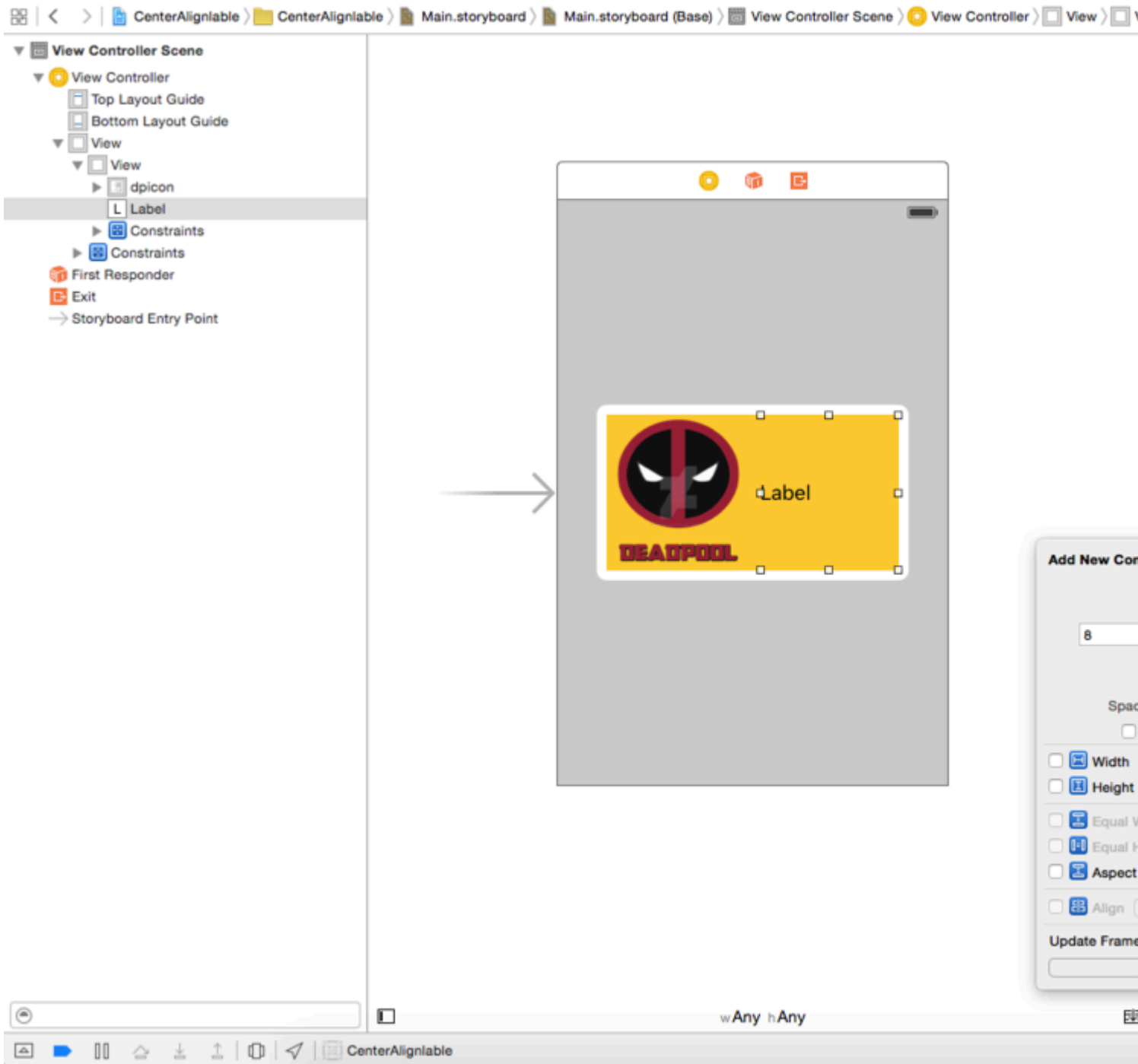
“”“”autolayoutXY.。

3. . .



imageView。

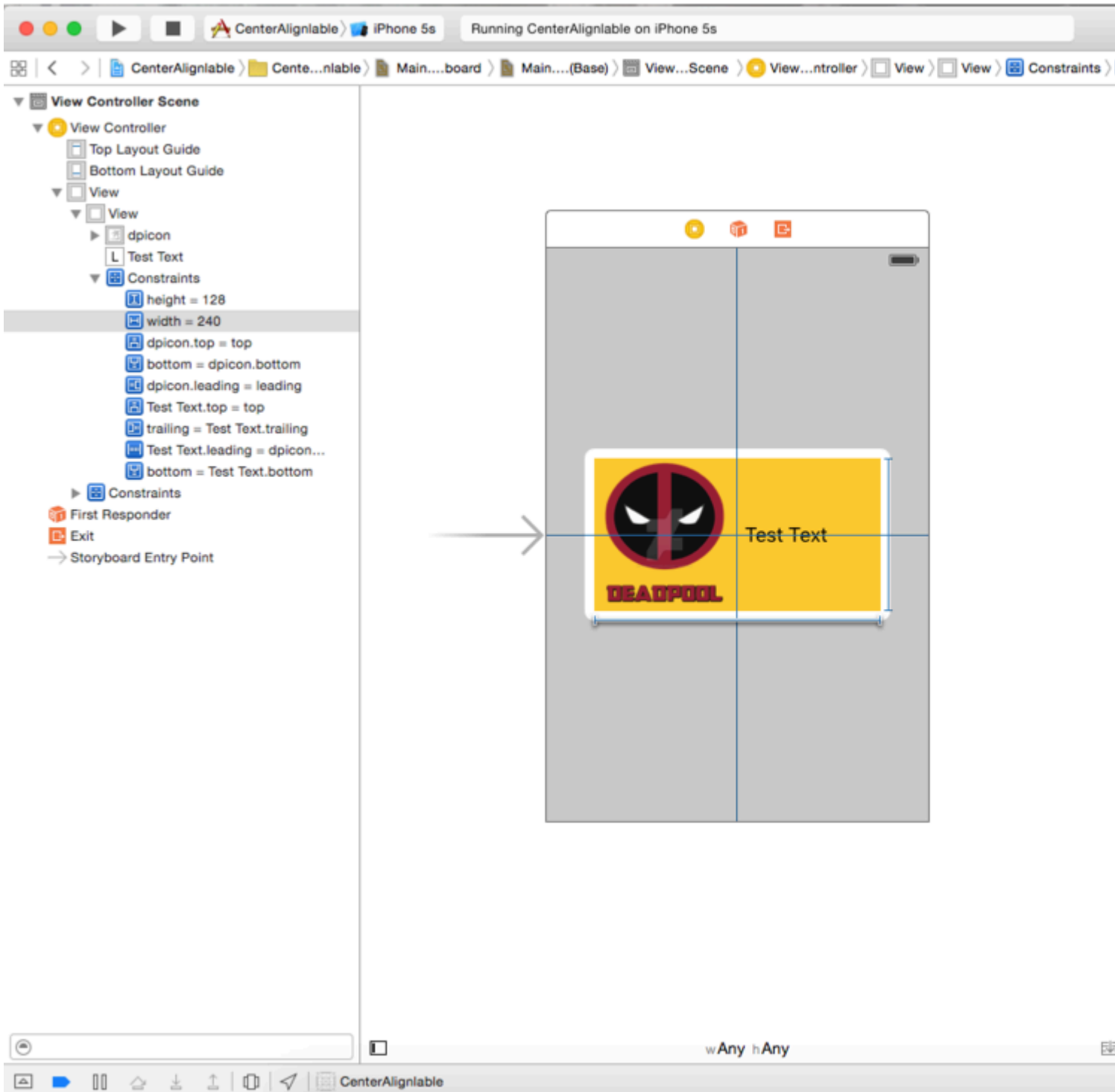
4UILabel



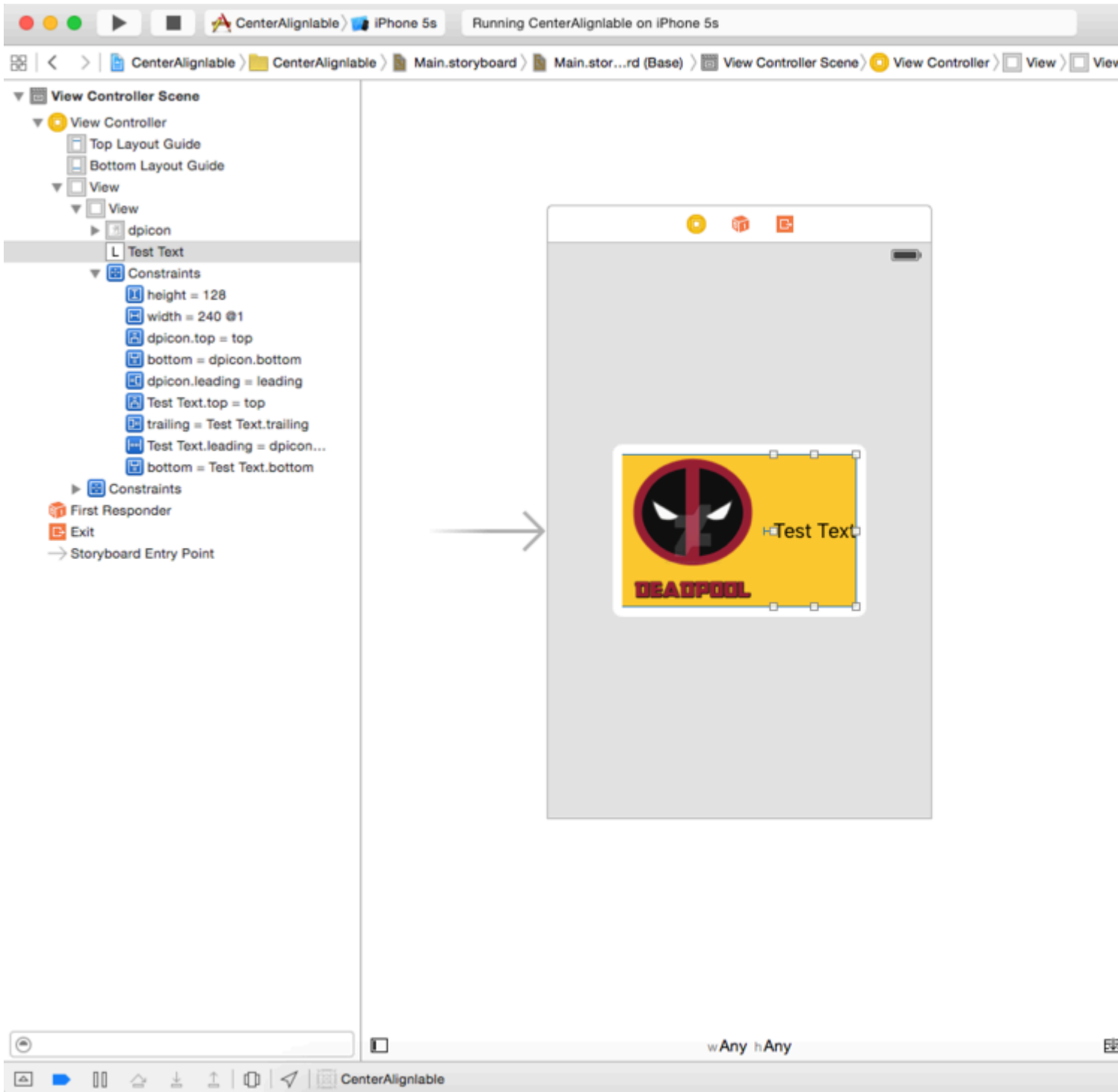
UILabel. 80,0,0. .

Ans - . . autolayoutLabel.

5. . UILabel. ""1. .

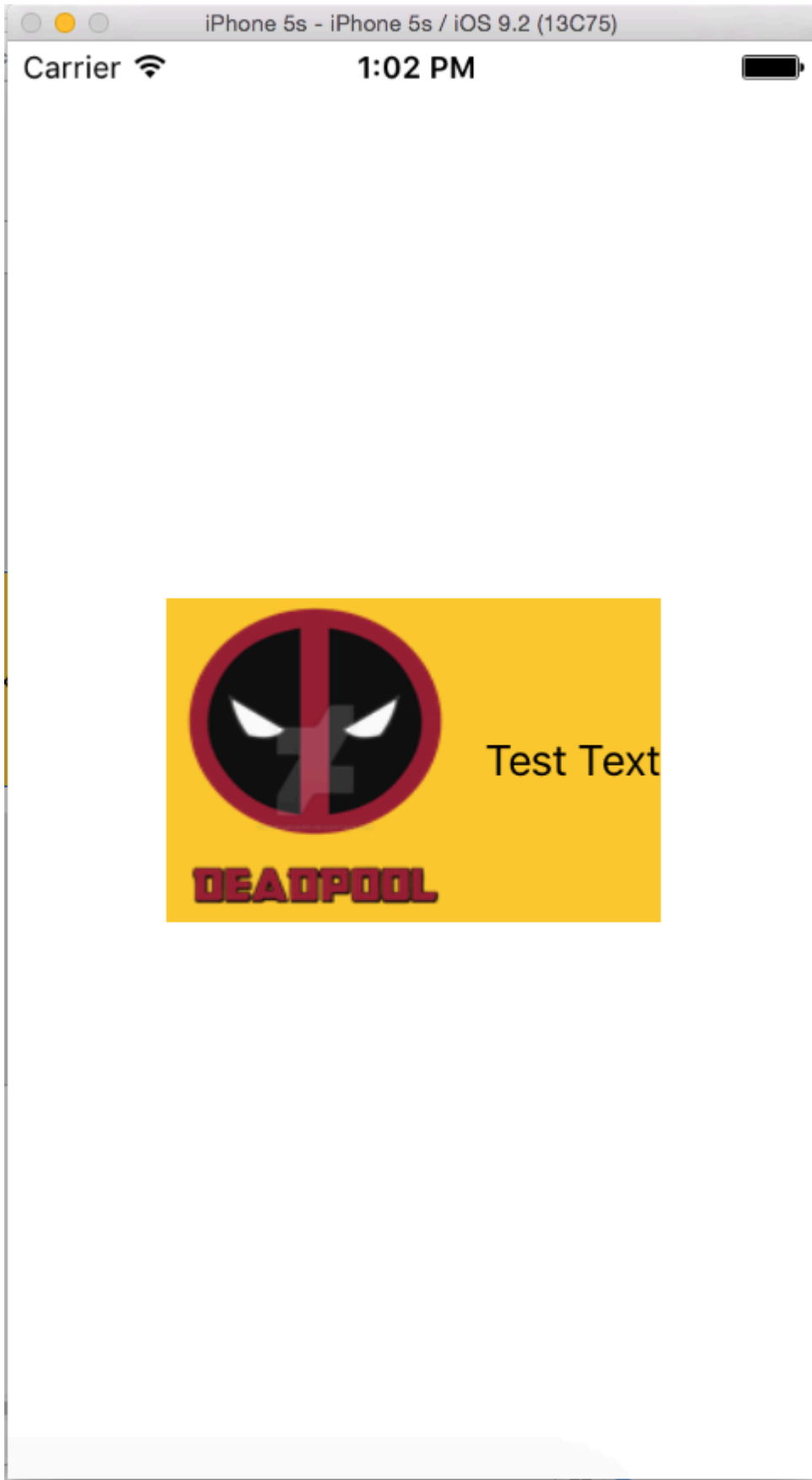


6UILabel. ◦ UILabel. UILabel.



5007511。

。



。 “” 。 。

1. `CADisplayLink dispatch_source_t dispatch_after NSTimer 。` `layoutIfNeeded`

## Objective-C

```
self.someConstraint.constant = 10.0;
[UIView animateWithDuration:0.25 animations:^(
 [self.view layoutIfNeeded];
)];
```

```
self.someConstraint.constant = 10.0
UIView.animate(withDuration: 0.25, animations: self.view.layoutIfNeeded)
```

2. [view layoutIfNeeded] ◦ ◦

```
[UIView animateWithDuration:0.5 animations:^(
 [view layoutIfNeeded];
)]
```

3. ◦ CPU◦

4. ◦ view.translatesAutoresizingMaskIntoConstraints = YES ◦

5. ◦

6. ◦ ◦ ◦ **Superview**◦

7. ◦ ◦

```
CABasicAnimation* ba = [CABasicAnimation animationWithKeyPath:@"transform"];
ba.autoreverses = YES;
ba.duration = 0.3;
ba.toValue = [NSValue valueWithCATransform3D:CATransform3DMakeScale(1.1, 1.1, 1)];
[v.layer addAnimation:ba forKey:nil];
```

8. **layoutSubviews** ◦ [super layoutSubviews]◦

9. **viewDidLayoutSubviews** ◦ layoutSubviews viewDidLayoutSubviews◦

10. ◦ layoutSubviews / layout ◦

layoutIfNeeded() ◦ ◦

## UILabel



Structure View Controller

Content Priority Ambiguity

- 3rd Label  
Set vertical hugging priority to 252
- 3rd Label  
Set vertical compression resistan...
- 4th Label  
Set vertical hugging priority to 250
- Lbl Right  
Set vertical hugging priority to 252
- Lbl Right  
Set vertical compression resistan...
- Lbl Left  
Set vertical hugging priority to 252
- Lbl Left  
Set vertical compression resistan...



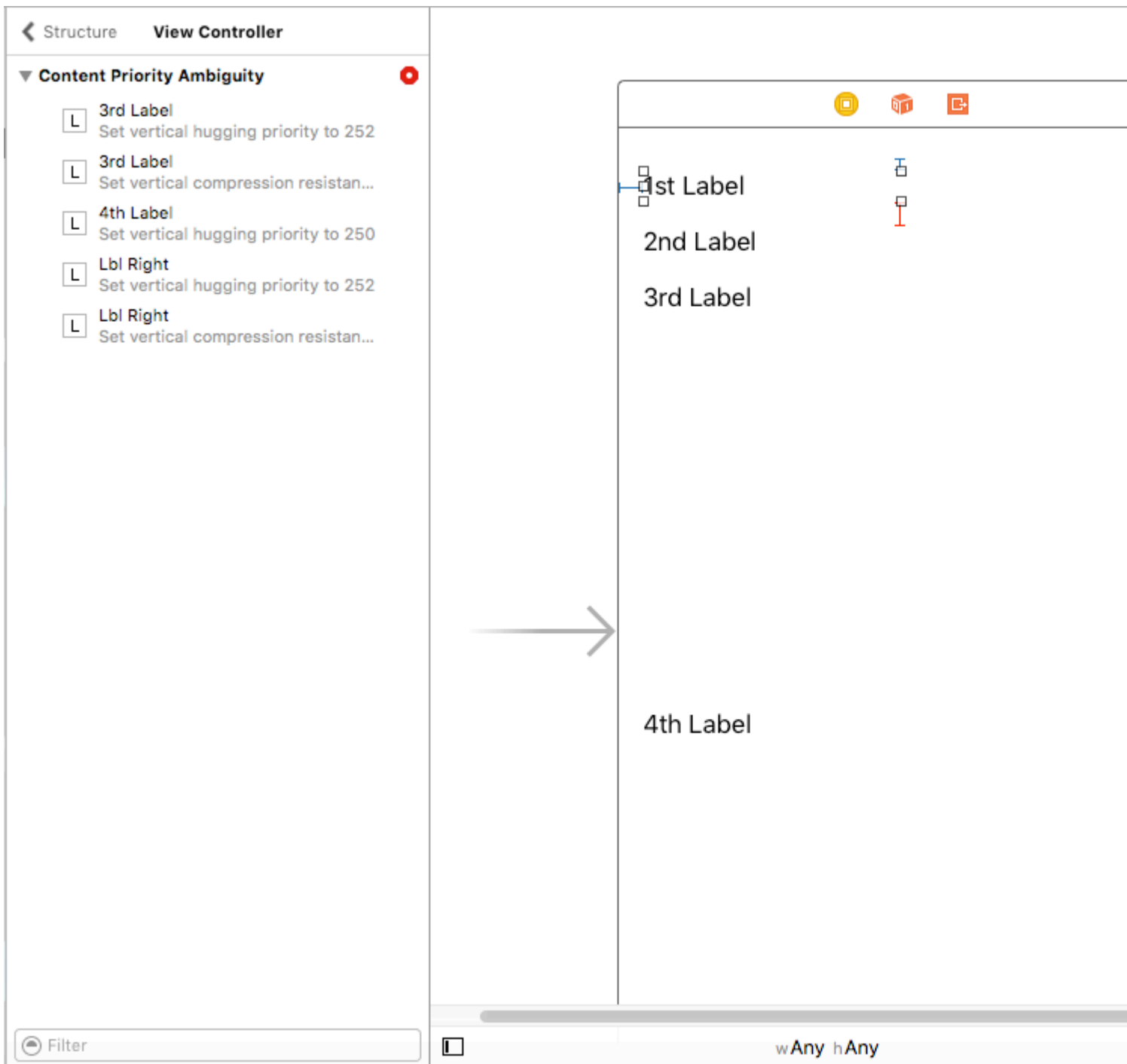
1st Label

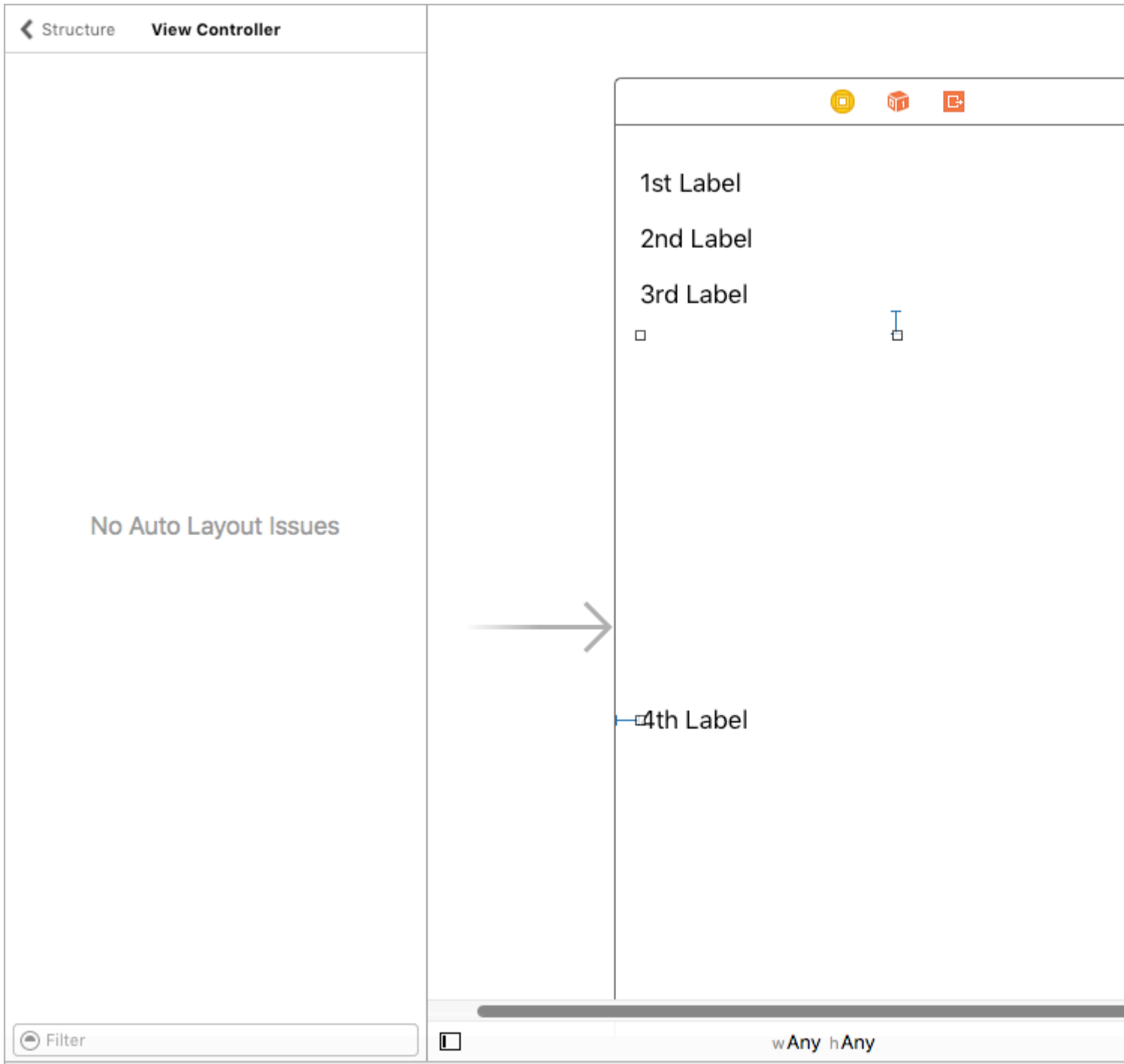
2nd Label

3rd Label

4th Label

1st4th





ViewController. UITableViewCell +.

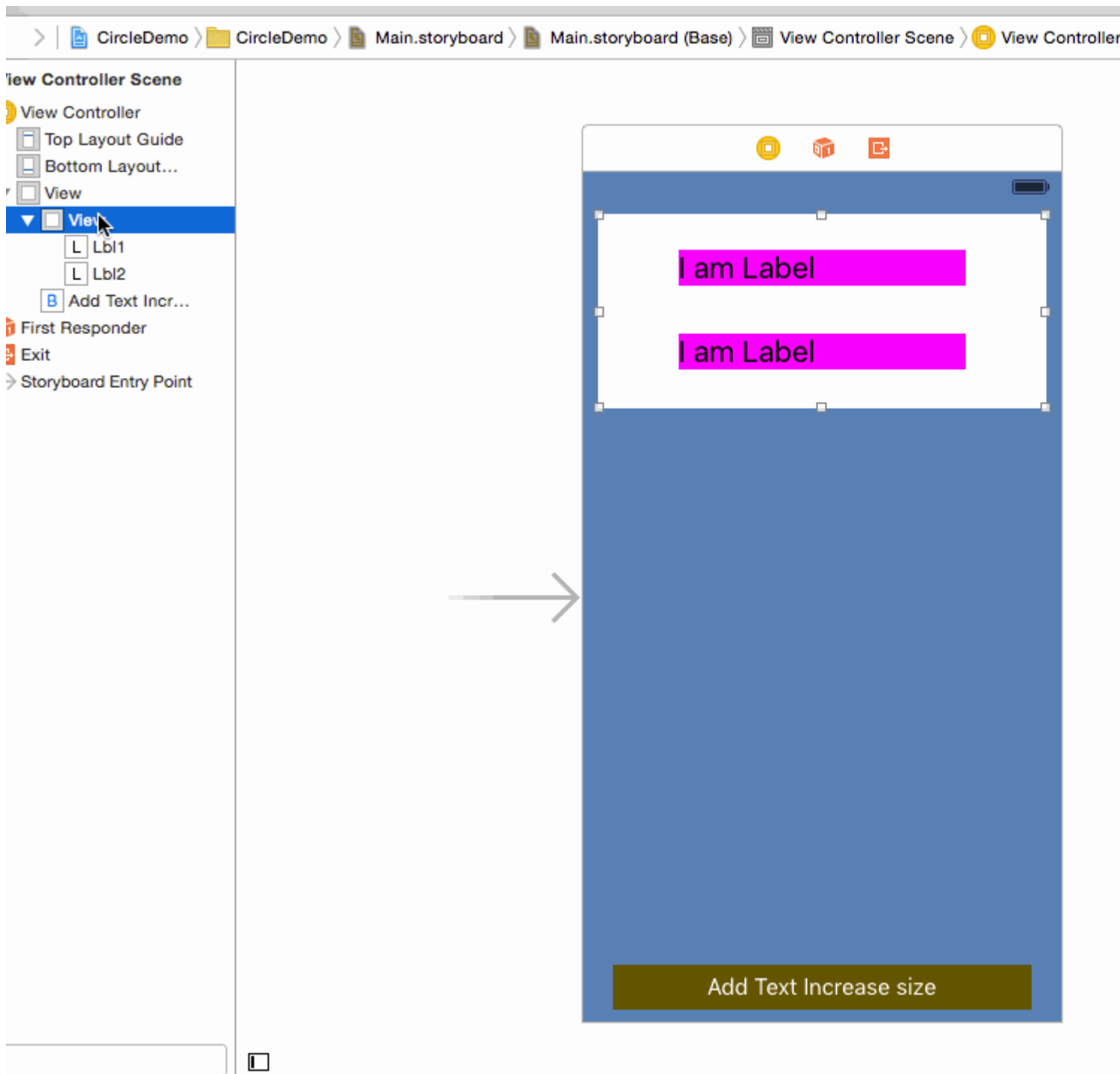
o

UILabelParentviewUILabelText

-

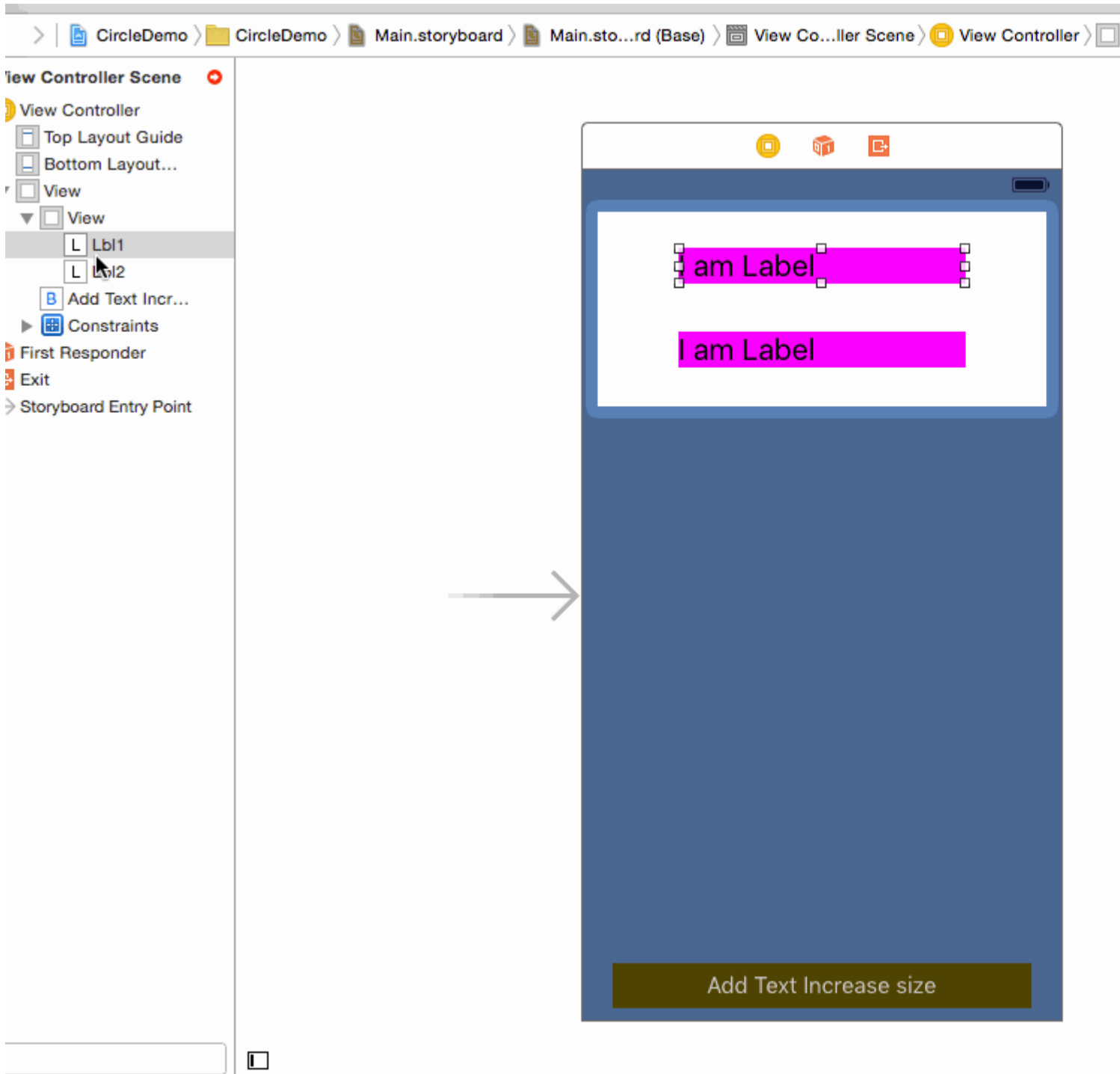
1 -UIView

1. o 2. 3.



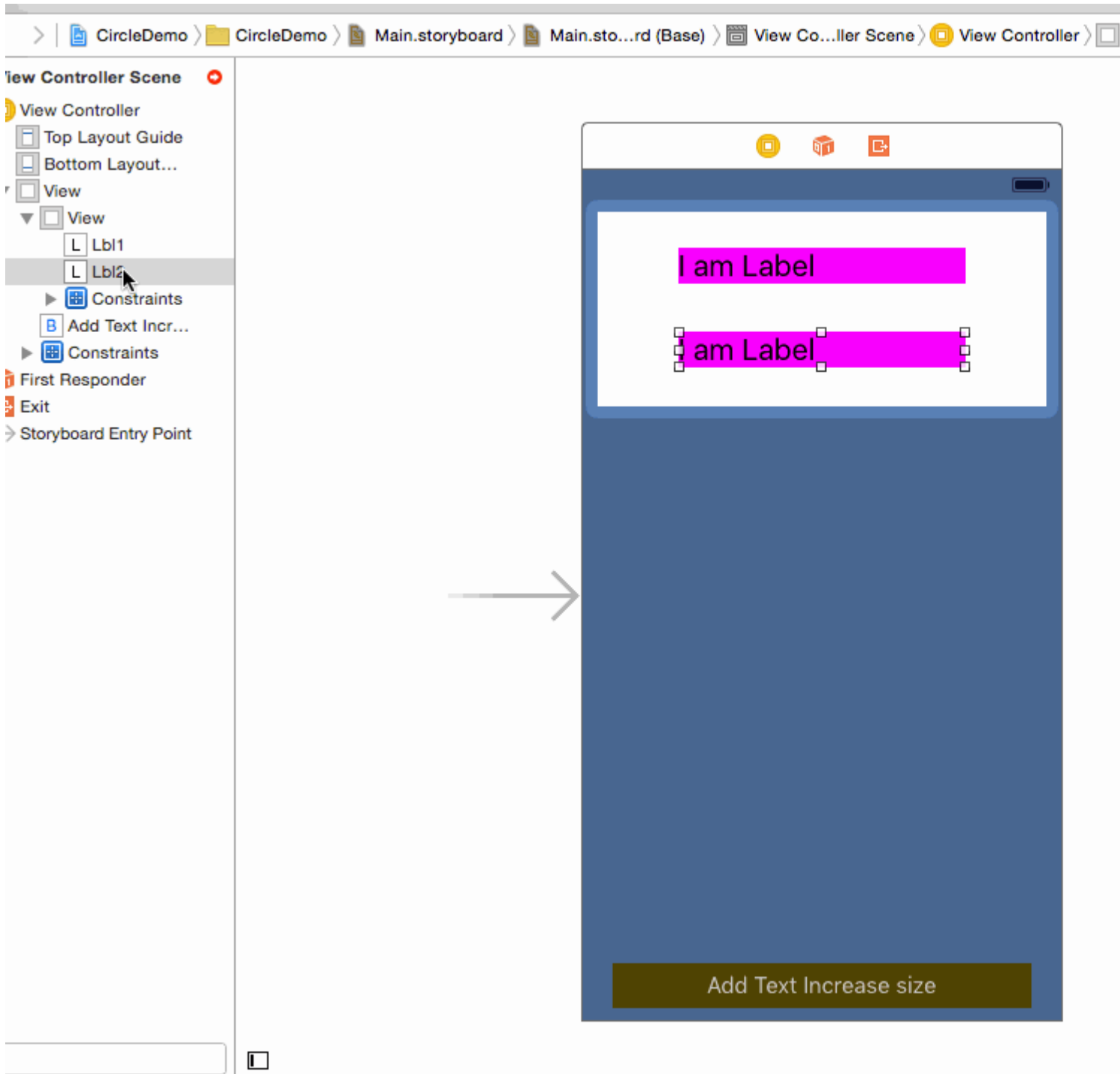
2 -1

1. 23

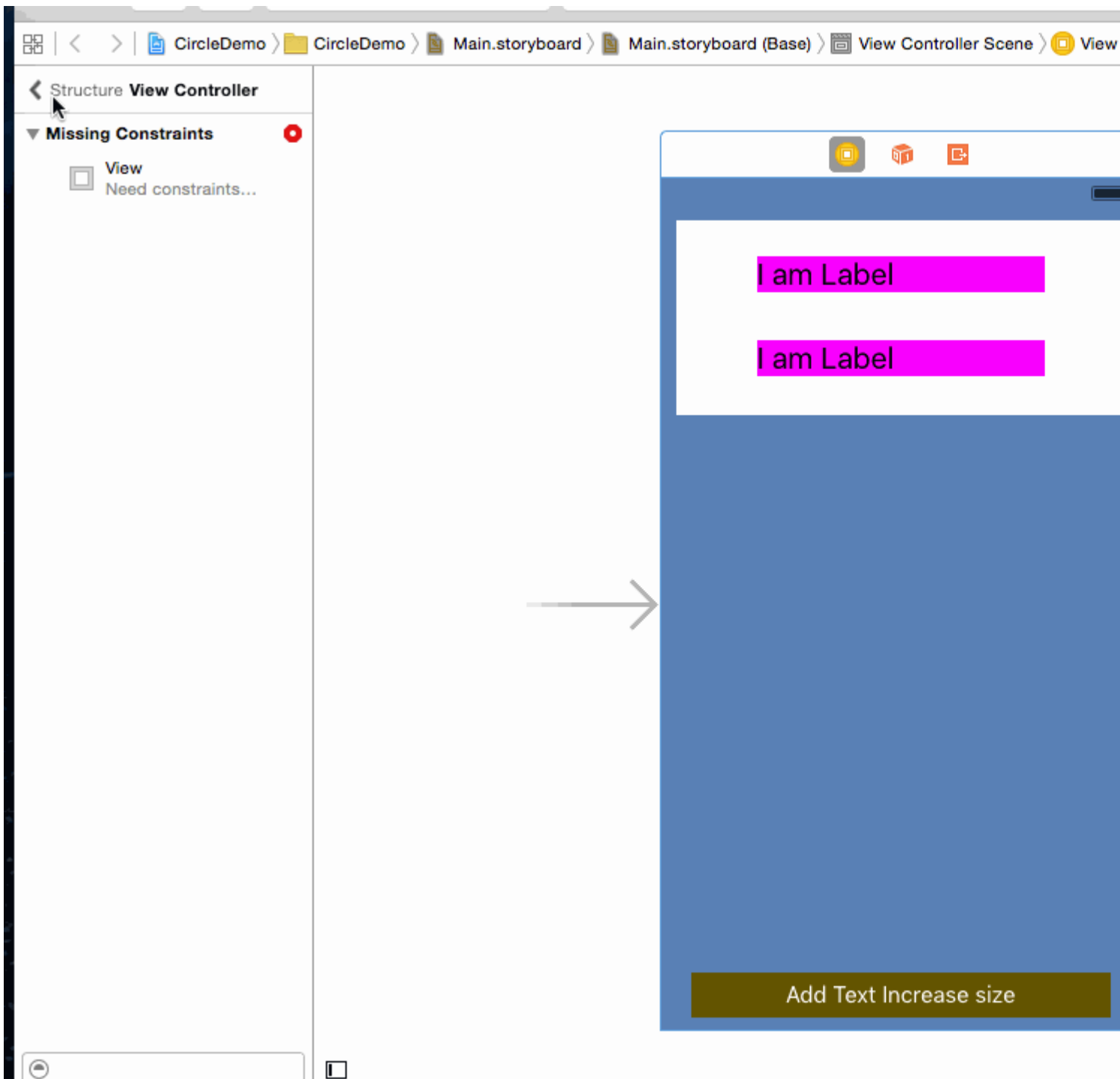


3 -2

1. 23

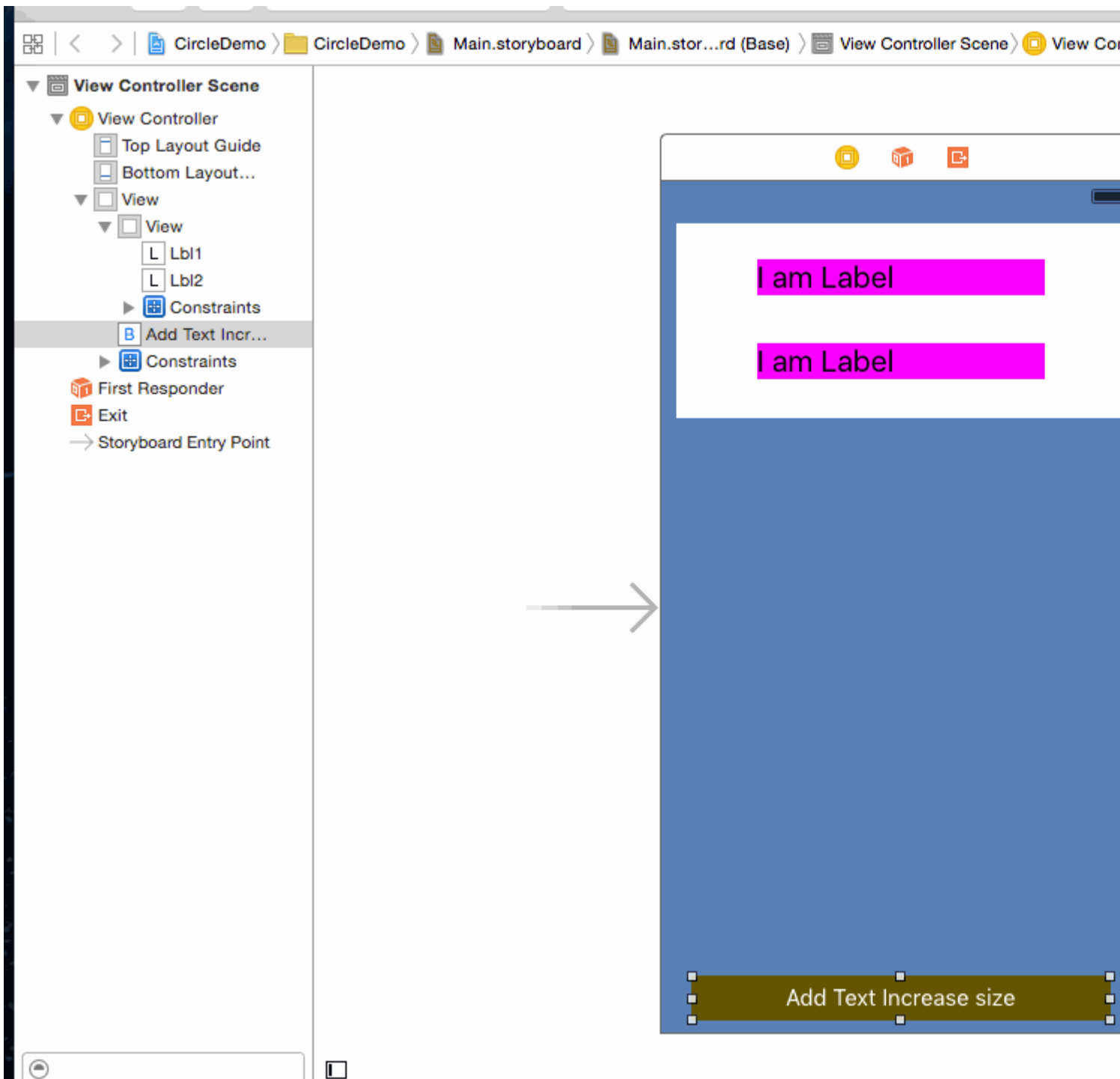


#### 4 - UIViewUILabel.



## 5 - UIButton

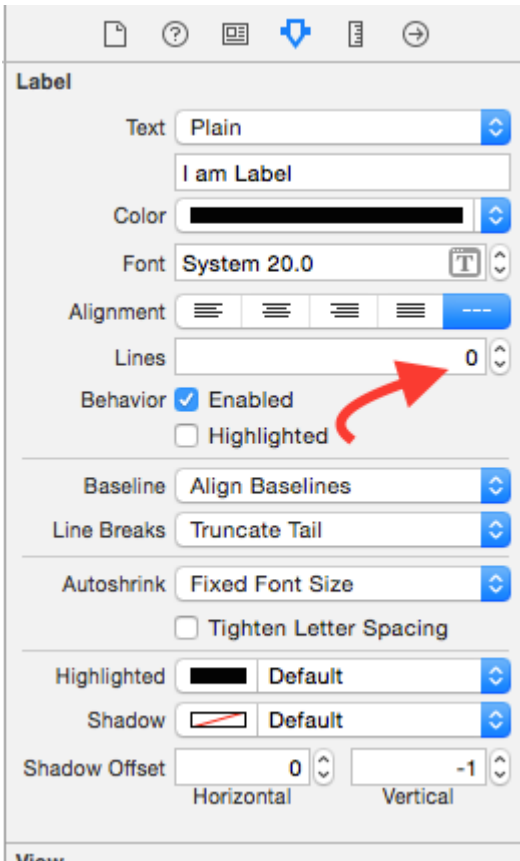
1. 234







-LabelNumber of lines = 0.



Autosize UIViewUILabelAutosize UILabel。

HVFLUI。 VFLInterface BuilderUI。

VFLsuperView.width aGradeViewaGradeView

```
"H:|[bgView][aGradeView(40)][bGradeView(40)]|"
```

UI。

VFLH:V:。 H:

- 。 |。 VFL。

- 。

bgView。 UIUI

- 。

[aGradeView(50)][aGradeView(50)]/50。

[bgView]。

- 。



```
// 1. create views
UIView *blueView = [[UIView alloc] init];
blueView.backgroundColor = [UIColor blueColor];
[self.view addSubview:blueView];

UIView *redView = [[UIView alloc] init];
redView.backgroundColor = [UIColor redColor];
[self.view addSubview:redView];

// 2. forbid Autoresizing
blueView.translatesAutoresizingMaskIntoConstraints = NO;
redView.translatesAutoresizingMaskIntoConstraints = NO;

// 3. make constraints
// horizontal
NSArray *blueH = [NSLayoutConstraint constraintsWithVisualFormat:@"H:|-20-[blueView]-20-|"
options:NSLayoutFormatAlignAllLeft metrics:nil views:@{@"blueView" : blueView}];
[self.view addConstraints:blueH];

// vertical
NSArray *blueVandRedV = [NSLayoutConstraint constraintsWithVisualFormat:@"V:|-20-
[blueView(50)]-20-[redView(==blueView)]" options:NSLayoutFormatAlignAllTrailing metrics:nil
views:@{@"blueView" : blueView, @"redView" : redView}];
[self.view addConstraints:blueVandRedV];

NSLayoutConstraint *redW = [NSLayoutConstraint constraintWithItem:redView
attribute:NSLayoutAttributeWidth relatedBy:NSLayoutRelationEqual toItem:blueView
attribute:NSLayoutAttributeWidth multiplier:0.5 constant:0];
[self.view addConstraint:redW];
```

UIKit

maskLayerUIViewUIViewframemaskLayer

```
// CustomView.m
- (void)layoutSubviews {
 [super layoutSubviews];
```

```

// now you can assume Auto Layout did its job
// you can use view's frame in your calculations
CALayer maskLayer = self.maskLayer;
maskLayer.bounds = self.bounds;
...
}

```

ViewController

```

- (void)viewDidLayoutSubviews {
 [super viewDidLayoutSubviews];
 // now you can assume all your subviews are positioned/resized correctly
 self.customView.frame = self.containerView.frame;
}

```

```

NSLayoutConstraint(item: myView, attribute: NSLayoutConstraint.Leading, relatedBy:
NSLayoutConstraint.Equal, toItem: view, attribute: NSLayoutConstraint.LeadingMargin, multiplier:
1.0, constant: 20.0)

```

```

view.attribute * multiplier + constant (1)

```

◦

V111.1

V2V1◦ = 60 = 1.125 = 0

◦





- Objective-C

```
// 1. create views
UIView *blueView = [[UIView alloc] init];
blueView.backgroundColor = [UIColor blueColor];
[self.view addSubview:blueView];

UIView *redView = [[UIView alloc] init];
redView.backgroundColor = [UIColor redColor];
[self.view addSubview:redView];

// 2. forbid Autoresizing
blueView.translatesAutoresizingMaskIntoConstraints = NO;
redView.translatesAutoresizingMaskIntoConstraints = NO;

// 3. make constraints
// 3.1 blueView
NSLayoutConstraint *blueLeft = [NSLayoutConstraint constraintWithItem:blueView
attribute:NSLayoutAttributeLeft relatedBy:NSLayoutRelationEqual toItem:self.view
attribute:NSLayoutAttributeLeft multiplier:1 constant:20];
[self.view addConstraint:blueLeft];

NSLayoutConstraint *blueTop = [NSLayoutConstraint constraintWithItem:blueView
attribute:NSLayoutAttributeTop relatedBy:NSLayoutRelationEqual toItem:self.view
attribute:NSLayoutAttributeTop multiplier:1 constant:20];
[self.view addConstraint:blueTop];

NSLayoutConstraint *blueRight = [NSLayoutConstraint constraintWithItem:blueView
attribute:NSLayoutAttributeRight relatedBy:NSLayoutRelationEqual toItem:self.view
attribute:NSLayoutAttributeRight multiplier:1 constant:-20];
[self.view addConstraint:blueRight];

NSLayoutConstraint *blueHeight = [NSLayoutConstraint constraintWithItem:blueView
attribute:NSLayoutAttributeHeight relatedBy:NSLayoutRelationEqual toItem:nil
attribute:NSLayoutAttributeNotAnAttribute multiplier:1 constant:50];
[self.view addConstraint:blueHeight];

// 3.2 redView
NSLayoutConstraint *redTop = [NSLayoutConstraint constraintWithItem:redView
attribute:NSLayoutAttributeTop relatedBy:NSLayoutRelationEqual toItem:blueView
attribute:NSLayoutAttributeBottom multiplier:1 constant:20];
[self.view addConstraint:redTop];
```

```
NSLayoutConstraint *redRight = [NSLayoutConstraint constraintWithItem:redView
attribute:NSLayoutAttributeRight relatedBy:NSLayoutRelationEqual toItem:self.view
attribute:NSLayoutAttributeRight multiplier:1 constant:-20];
[self.view addConstraint:redRight];

NSLayoutConstraint *redHeight = [NSLayoutConstraint constraintWithItem:redView
attribute:NSLayoutAttributeHeight relatedBy:NSLayoutRelationEqual toItem:blueView
attribute:NSLayoutAttributeHeight multiplier:1 constant:0];
[self.view addConstraint:redHeight];

NSLayoutConstraint *redWidth = [NSLayoutConstraint constraintWithItem:redView
attribute:NSLayoutAttributeWidth relatedBy:NSLayoutRelationEqual toItem:blueView
attribute:NSLayoutAttributeWidth multiplier:0.5 constant:0];
[self.view addConstraint:redWidth];
```

<https://riptutorial.com/zh-TW/ios/topic/792/>

---

# 193: UITableViewCells

UITableViewCell ◦ didSelect...UITableViewDelegate ◦

## Examples

◦

UITableViewCell ◦

```
override func viewDidLoad() {
 viewDidLoad()

 let doubleTapGestureRecognizer = UITapGestureRecognizer(target: self, action:
#selector(handleDoubleTap(sender:)))
 doubleTapGestureRecognizer.numberOfTapsRequired = 2
 tableView.addGestureRecognizer(doubleTapGestureRecognizer)

 let tapGestureRecognizer = UITapGestureRecognizer(target: self, action:
#selector(handleTapGesture(sender:)))
 tapGestureRecognizer.numberOfTapsRequired = 1
 tapGestureRecognizer.require(toFail: doubleTapGestureRecognizer)
 tableView.addGestureRecognizer(tapGestureRecognizer)
}

func handleTapGesture(sender: UITapGestureRecognizer) {
 let touchPoint = sender.location(in: tableView)
 if let indexPath = tableView.indexPathForRow(at: touchPoint) {
 print(indexPath)
 }
}

func handleDoubleTap(sender: UITapGestureRecognizer) {
 let touchPoint = sender.location(in: tableView)
 if let indexPath = tableView.indexPathForRow(at: touchPoint) {
 print(indexPath)
 }
}
```

UITableViewCells <https://riptutorial.com/zh-TW/ios/topic/9961/uitableviewcells>



---

# 194: UITableViewCells

## Examples

◦

### UITableView◦

```
override func viewDidLoad() {
 viewDidLoad()

 let doubleTapGestureRecognizer = UITapGestureRecognizer(target: self, action:
#selector(handleDoubleTap(sender:)))
 doubleTapGestureRecognizer.numberOfTapsRequired = 2
 tableView.addGestureRecognizer(doubleTapGestureRecognizer)

 let tapGestureRecognizer = UITapGestureRecognizer(target: self, action:
#selector(handleTapGesture(sender:)))
 tapGestureRecognizer.numberOfTapsRequired = 1
 tapGestureRecognizer.require(toFail: doubleTapGestureRecognizer)
 tableView.addGestureRecognizer(tapGestureRecognizer)
}

func handleTapGesture(sender: UITapGestureRecognizer) {
 let touchPoint = sender.location(in: tableView)
 if let indexPath = tableView.indexPathForRow(at: touchPoint) {
 print(indexPath)
 }
}

func handleDoubleTap(sender: UITapGestureRecognizer) {
 let touchPoint = sender.location(in: tableView)
 if let indexPath = tableView.indexPathForRow(at: touchPoint) {
 print(indexPath)
 }
}
```

[UITableViewCells](https://riptutorial.com/zh-TW/ios/topic/9962/uitableviewcells) <https://riptutorial.com/zh-TW/ios/topic/9962/uitableviewcells>

---

# 195: UITextField

UITextField

## Examples

### UITextField

UITextField

iPhoneiPad

```
class NumberTextField: UITextField {

 required init(coder aDecoder: NSCoder) {
 super.init(coder: aDecoder)
 registerForTextFieldNotifications()
 }

 override init(frame: CGRect) {
 super.init(frame: frame)
 }

 override func awakeFromNib() {
 super.awakeFromNib()
 keyboardType = .numberPad//useful for iPhone only
 }

 private func registerForTextFieldNotifications() {
 NotificationCenter.default.addObserver(self, selector:
#selector(NumberTextField.textDidChange), name: NSNotification.Name(rawValue:
"UITextFieldTextDidChangeNotification"), object: self)
 }

 deinit {
 NotificationCenter.default.removeObserver(self)
 }

 func textDidChange() {
 text = filteredText()
 }
 private func filteredText() -> String {
 let inverseSet = CharacterSet(charactersIn:"0123456789").inverted
 let components = text!.components(separatedBy: inverseSet)
 return components.joined(separator: "")
 }
}
```

UITextField

### UITextField

UITextField

```
class CustomTextField: UITextField {

 var enableLongPressActions = false

 required init(coder aDecoder: NSCoder) {
 super.init(coder: aDecoder)!
 }

 override init(frame: CGRect) {
 super.init(frame: frame)
 }

 override func canPerformAction(_ action: Selector, withSender sender: Any?) -> Bool {
 return enableLongPressActions
 }
}
```

enableLongPressActions°

**UITextField** <https://riptutorial.com/zh-TW/ios/topic/9997/uitextfield>

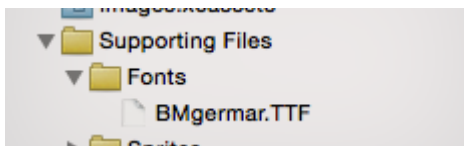
# 196:

## Examples

Info.plistUIAppFonts. . .

Info.plistIB.

1. Xcode Supporting Files. "" . IB.



2. Info.plistFonts provided by application key UIAppFonts Fonts provided by application key . Item 0. .

▼ Fonts provided by application	Array	(1 item)
Item 0	String	BMgermar.TTF

- 3.

[ 3 ]

```
for family in UIFont.familyNames {
 print("\(family)")

 for name in UIFont.fontNames(forFamilyName: family) {
 print(" \(name)")
 }
}
```

[ - C ]

```
for (NSString *familyName in [UIFont familyNames]){
 NSLog(@"Family name: %@", familyName);
 for (NSString *fontName in [UIFont fontNamesForFamilyName:familyName]) {
 NSLog(@"--Font name: %@", fontName);
 }
}
```

UI.

- ui
- .

1. .TTF Info.plist.

2. **UIKit + IBExtensions**UIUILabelUIButton. say **fontName** . 4.

---

# UIKit+ IBExtensions.h

```
#import <UIKit/UIKit.h>

//Category extension for UILabel
@interface UILabel (IBExtensions)

@property (nonatomic, copy) NSString *fontName;
@end

// Category extension for UITextField
@interface UITextField (IBExtensions)

@property (nonatomic, copy) NSString *fontName;
@end

// Category extension for UIButton
@interface UIButton (IBExtensions)

@property (nonatomic, copy) NSString *fontName;
@end
```

## 3. GettersSetterfontNamegettersetter

---

# UIKit+ IBExtensions.m

```
#import "UIKit+IBExtensions.h"

@implementation UILabel (IBExtensions)

- (NSString *)fontName {
 return self.font.fontName;
}

- (void)setFontName:(NSString *)fontName {
 self.font = [UIFont fontWithName:fontName size:self.font.pointSize];
}
@end

@implementation UITextField (IBExtensions)

- (NSString *)fontName {
 return self.font.fontName;
}

- (void)setFontName:(NSString *)fontName {
 self.font = [UIFont fontWithName:fontName size:self.font.pointSize];
}
@end

@implementation UIButton (IBExtensions)

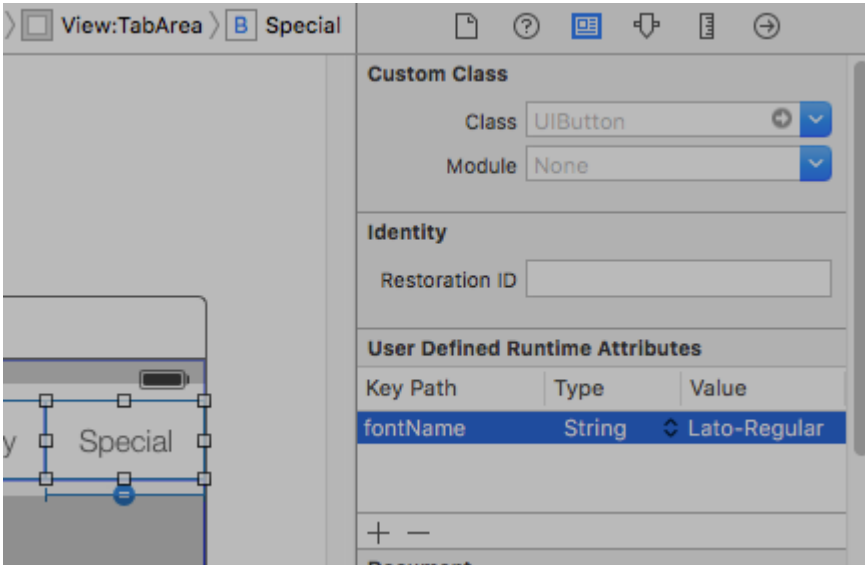
- (NSString *)fontName {
 return self.titleLabel.font.fontName;
}
}
```

```

- (void)setFontName:(NSString *)fontName{
 self.titleLabel.font = [UIFont fontWithName:fontName size:self.titleLabel.font.pointSize];
}
@end

```

#### 4. **fontName**keyPath String。

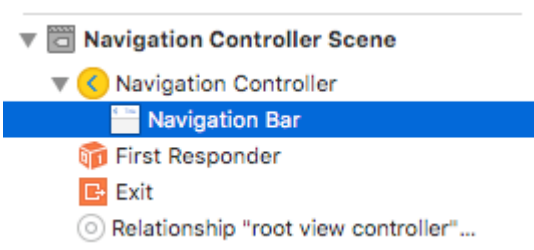


- Lato-Regular。
- bundle.**.tff**storyboard。
- UI。

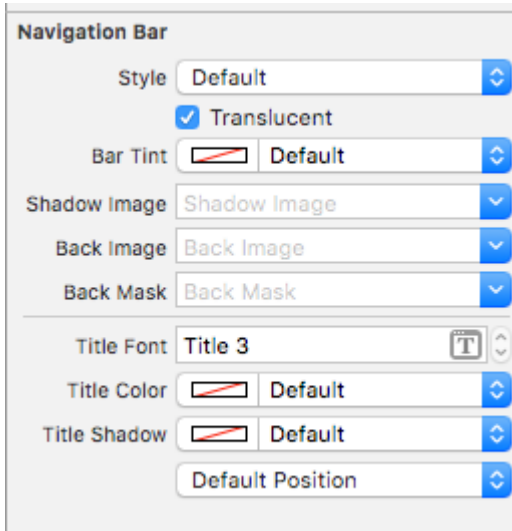
## Storyboard

Xcode。 **UIControl UILabels UIButtons。** 。

1.



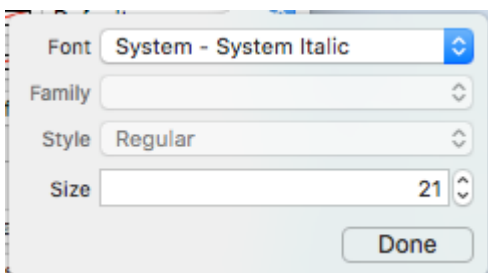
2. """"



*XcodeBar Tint*

Xcode 7.1.1+.

1. Xcode;
2. 0.0



3. NavBar. NavBars. navBarnavBar.

## deux

◦

1. xml. Storyboardstoryboard> Open As ...
2. navigationItemxml. ◦
3. ◦
- 4.
5. ◦ ◦

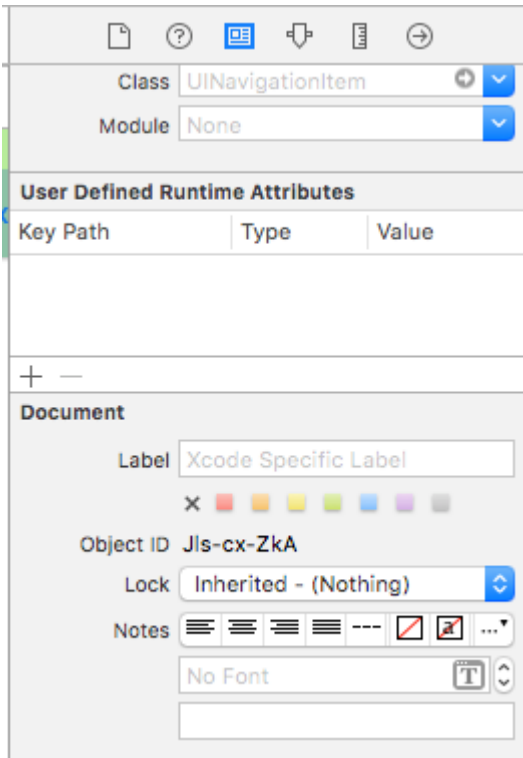
- 
- 
- [NavBar](#)
- 

*Code With Chris* ◦

[SO](#) ◦

1. .ttf.ttc
2. Xcode
3. app-info.plistFontsapplication.It.
4. UINavigationController。

XcodeUINavigationController。



UIViewUIButtonUILabelUIView... Xcode。 View Controller。 UINavigationController.titleView  
。 。 23600285。

```
@IBOutlet var customFontTitleView: UIButton!

//Sometime later...
self.navigationController.titleView = customFontTitleView
```

- SO 。

<https://riptutorial.com/zh-TW/ios/topic/1504/>



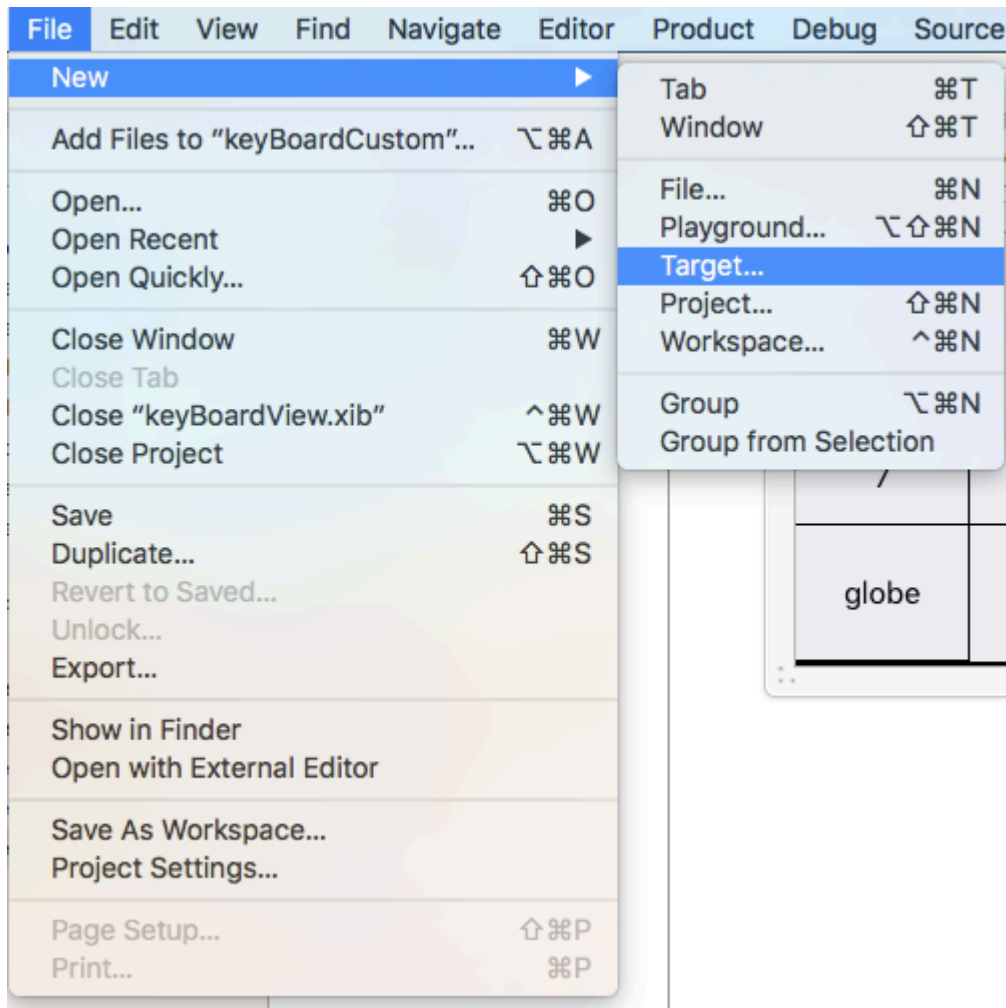
# 197:

## Examples

### KeyBoard

### Objective-CXib

### XCode



### Custom KeyBoard

Choose a template for your new target:

iOS

Application

Framework & Library

Application Extensi...

Test

watchOS

Application

Framework & Library

tvOS

Application

Framework & Library

Application Extensi...

Test

OS X

Application

Framework & Library

Application Extensi...



Action  
Extension



Audio Unit  
Extension



Content Blocker  
Extension



Custom  
Keyboard



Document  
Provider



Photo Editing  
Extension



Share Extension



Shared Links  
Extension



Spotlight Index  
Extension



Today Extension

Custom Keyboard Extension

This template builds a custom keyboard that users can choose in place of the system keyboard.

Cancel

Previous

Next

Choose options for your new target:

Product Name:

Organization Name:

Organization Identifier:

Bundle Identifier:

Language:

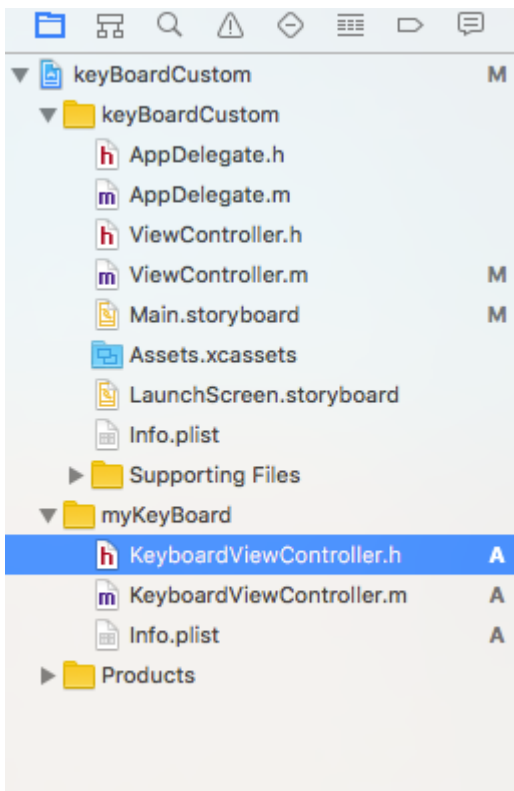
Project:

Embed in Application:

Cancel

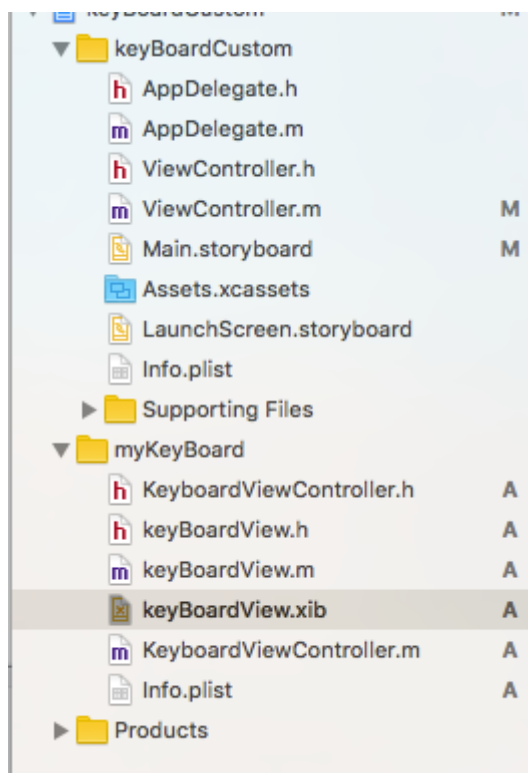
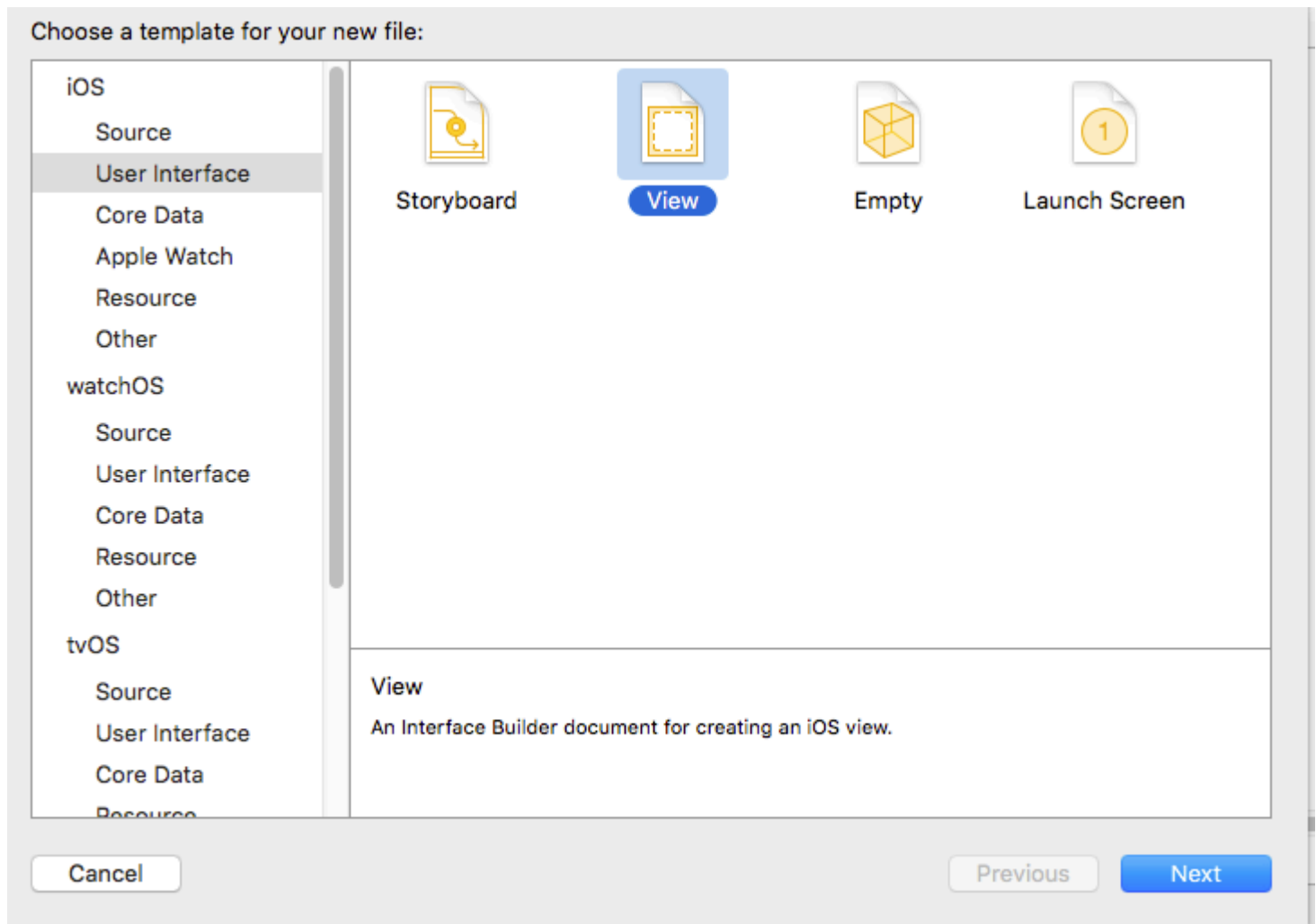
Previous

Finish

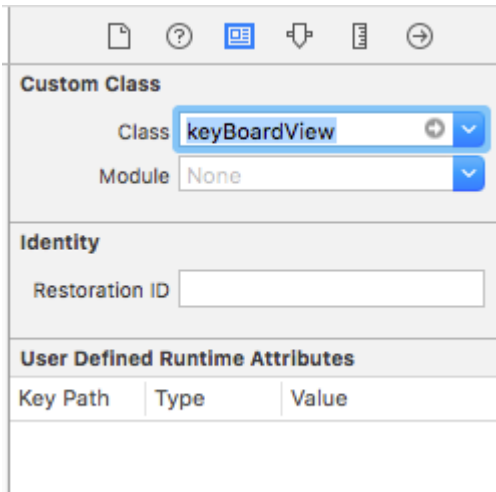


myKeyboardTarget

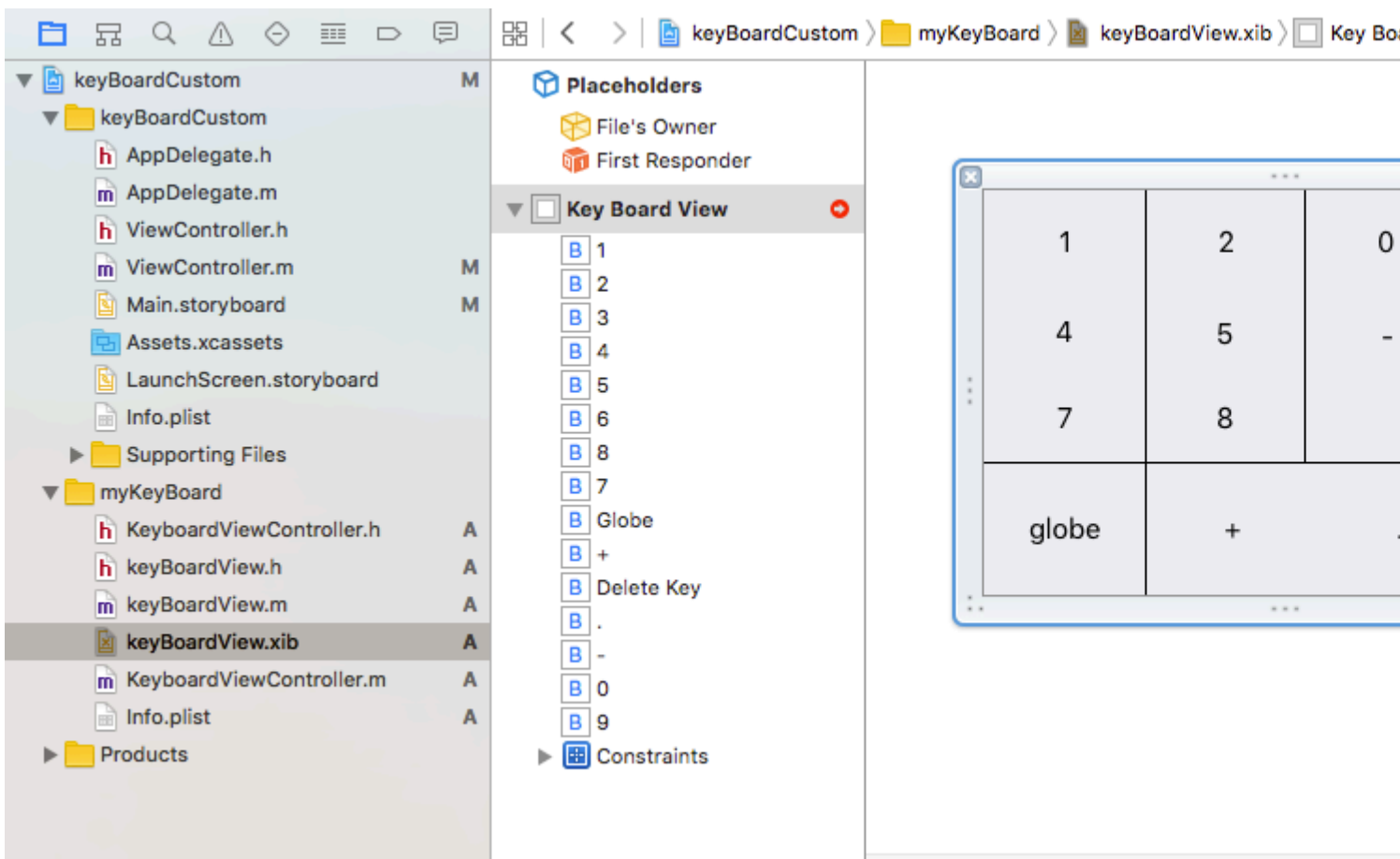
# UIViewCocoatouch



keyBoardView.xibkeyBoardView



keyBoardView.xib



keyBoardView.xibkeyBoardView.h

keyBoardView.h

```
#import <UIKit/UIKit.h>

@interface keyBoardView : UIView

@property (weak, nonatomic) IBOutlet UIButton *deleteKey;
//IBOutlet for the delete Key
@property (weak, nonatomic) IBOutlet UIButton *globe;
//Outlet for the key with title globe which changes the keyboard type
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *keys;
```

```
//Contains a collection of all the keys '0 to 9' '+' '-' and '.'
```

```
@end
```

```
keyBoardViewController.h#import "keyBoardView.h"
```

```
@property (strong, nonatomic)keyBoardView *keyboard;@property (strong, nonatomic)keyBoardView *keyboard;
```

```
@property (nonatomic, strong) UIButton *nextKeyboardButton and all the code associated with it
```

## KeyboardViewController.mviewDidLoad

```
- (void)viewDidLoad {
 [super viewDidLoad];
 self.keyboard=[[NSBundle mainBundle]loadNibNamed:@"keyBoardView" owner:nil
options:nil]objectAtIndex:0];
 self.inputView=self.keyboard;
 [self addGestureToKeyboard];

 // Perform custom UI setup here
 // self.nextKeyboardButton = [UIButton buttonWithType:UIButtonTypeSystem];
 //
 // [self.nextKeyboardButton setTitle:NSString(@"Next Keyboard", @"Title for 'Next
Keyboard' button") forState:UIControlStateNormal];
 // [self.nextKeyboardButton sizeToFit];
 // self.nextKeyboardButton.translatesAutoresizingMaskIntoConstraints = NO;
 //
 // [self.nextKeyboardButton addTarget:self action:@selector(advanceToNextInputMode)
forControlEvents:UIControlEventTouchUpInside];
 //
 // [self.view addSubview:self.nextKeyboardButton];
 //
 // [self.nextKeyboardButton.leftAnchor constraintEqualToAnchor:self.view.leftAnchor].active
= YES;
 // [self.nextKeyboardButton.bottomAnchor
constraintEqualToAnchor:self.view.bottomAnchor].active = YES;
}
```

```
addGestureToKeyboard pressDeleteKey keyPressed
```

```
-(void) addGestureToKeyboard
{
 [self.keyboard.deleteKey addTarget:self action:@selector(pressDeleteKey)
forControlEvents:UIControlEventTouchUpInside];
 [self.keyboard.globe addTarget:self action:@selector(advanceToNextInputMode)
forControlEvents:UIControlEventTouchUpInside];

 for (UIButton *key in self.keyboard.keys)
 {
 [key addTarget:self action:@selector(keyPressed:)
forControlEvents:UIControlEventTouchUpInside];
 }
}
-(void) pressDeleteKey
{
```

```

 [self.textDocumentProxy deleteBackward];
}

-(void)keyPressed:(UIButton *)key
{
 [self.textDocumentProxy insertText:[key currentTitle]];
}

```

-> -> -> -> keyboardNamekeyBoardCustom

Bundle display name Value String Value Project

key	type	value
Information Property List	Dictionary	(15 items)
Localization native development re...	String	en
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFI...
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle display name	String	keyBoardMi
Bundle creator OS Type code	String	????
Bundle version	String	1
Application requires iPhone enviro...	Boolean	YES
Launch screen interface file base...	String	LaunchScreen
Main storyboard file base name	String	Main
Required device capabilities	Array	(1 item)
Supported interface orientations	Array	(3 items)

Youtube

<https://riptutorial.com/zh-TW/ios/topic/7358/>

# 198: URL

1. // **canOpenURL**URL。

2. //

```
UIApplication.sharedApplication.canOpenURL_ aUrINSURL
```

3. // Objective-C

```
[[UIApplication sharedApplication] canOpenURLNSURL *aUrl];
```

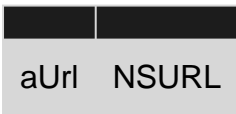
4. // **openURL**URL。 //。

5. //

```
UIApplication.sharedApplication.openURL_ aUrINSURL
```

6. // Objective-C

```
[[UIApplication sharedApplication] openURLNSURL *aUrl];
```



iOS9。 `LSApplicationQueriesSchemesInfo.plist`

---

iOS `tel http / https sms mailto facetime Youtube Maps iTunes httpURL。`

**URL**

`tel://123456890 tel:123456890`

**HTTP** `http://www.google.com`

**facetime** `facetime://azimov@demo.com`

**mailto** `mailto://azimov@demo.com`

`sms://123456890 sms:123456890`

**Youtube** `https://www.youtube.com/watch?v=-eCaif2QKfA https://www.youtube.com/watch?v=-eCaif2QKfA`

- `http://maps.apple.com/?address=1, Infinite+Loop, Cupertino, California`  
`http://maps.apple.com/?address=1, Infinite+Loop, Cupertino, California address = 1`  
`http://maps.apple.com/?address=1, Infinite+Loop, Cupertino, California`
- `http://maps.apple.com/?ll=46.683155557, 6.683155557`



<http://maps.apple.com/?ll=46.683155557,6.683155557>

**iTunes** <https://itunes.apple.com/us/artist/randy-newman/id200900>

<https://itunes.apple.com/us/artist/randy-newman/id200900>

tel\*#° Phone°

## Examples

### URLMail

---

```
if let url = URL(string: "mailto://azimov@demo.com") {
 if UIApplication.shared.canOpenURL(url) {
 UIApplication.shared.openURL(url)
 } else {
 print("Cannot open URL")
 }
}
```

## Objective-C

```
NSURL *url = [NSURL URLWithString:@"mailto://azimov@demo.com"];
if ([[UIApplication sharedApplication] canOpenURL:url]) {
 [[UIApplication sharedApplication] openURL:url];
} else {
 NSLog(@"Cannot open URL");
}
```

### Apple URL

iOSOS XwatchOS 2URL°

### Safari

### Objective-C

```
NSString *stringURL = @"http://stackoverflow.com/";
NSURL *url = [NSURL URLWithString:stringURL];
[[UIApplication sharedApplication] openURL:url];
```

```
let stringURL = "http://stackoverflow.com/"
if let url = URL(string: stringURL) {
 UIApplication.shared.openURL(url)
}
```

### Objective-C

```
NSString *stringURL = @"tel:1-408-555-5555";
```

```
NSURL *url = [NSURL URLWithString:stringURL];
[[UIApplication sharedApplication] openURL:url];
```

```
let urlString = "tel:1-408-555-5555"
if let url = URL(string: urlString) {
 UIApplication.shared.openURL(url)
}
```

## HTML

```
1-408-555-5555
```

## FaceTime

### Objective-C

```
NSString *stringURL = @"facetime:14085551234";
NSURL *url = [NSURL URLWithString:stringURL];
[[UIApplication sharedApplication] openURL:url];
```

```
let urlString = "facetime:14085551234"
if let url = URL(string: urlString) {
 UIApplication.shared.openURL(url)
}
```

## HTML

```
Connect using FaceTime
Connect using FaceTime
```

### Objective-C

```
NSString *stringURL = @"sms:1-408-555-1212";
NSURL *url = [NSURL URLWithString:stringURL];
[[UIApplication sharedApplication] openURL:url];
```

```
let urlString = "sms:1-408-555-1212"
if let url = URL(string: urlString) {
 UIApplication.shared.openURL(url)
}
```

## HTML

```
Launch Messages App
New SMS Message
```

### Objective-C

```
NSString *stringURL = @"mailto:foo@example.com";
NSURL *url = [NSURL URLWithString:stringURL];
```

```
[[UIApplication sharedApplication] openURL:url];
```

```
let urlString = "mailto:foo@example.com"
if let url = URL(string: urlString) {
 UIApplication.shared.openURL(url)
}
```

## HTML

```
John Frank
```

“””。 iOSfrom。 mailto URL

```
mailto:foo@example.com?cc=bar@example.com&subject=Greetings%20from%20Cupertino!&body=Wish%20you%20were
```

MFMailComposeViewController

URL <https://riptutorial.com/zh-TW/ios/topic/3646/url>

# 199:

## Examples

### C

```
view.backgroundColor = [UIColor redColor];
```

```
view.backgroundColor! = UIColor.redColor()
```

### 3

```
view.backgroundColor = UIColor.redColor
```

## UIView

### Objective-C

```
UIGraphicsBeginImageContext(self.view.frame.size);
[[UIImage imageNamed:@"image.png"] drawInRect:self.view.bounds];
UIImage *image = UIGraphicsGetImageFromCurrentImageContext();
UIGraphicsEndImageContext();
self.view.backgroundColor = [UIColor colorWithPatternImage:image];
```

```
self.view.backgroundColor = [UIColor colorWithPatternImage:[UIImage
imageNamed:@"Background.png"]];
```

## CAGradientLayer

### Swift 3.1

```
func createGradient() {
 let caLayer = CAGradientLayer()
 caLayer.colors = [UIColor.white, UIColor.green, UIColor.blue]
 caLayer.locations = [0, 0.5, 1]
 caLayer.bounds = self.bounds
 self.layer.addSublayer(caLayer)
}
```

### viewDidLoad

```
override func viewDidLoad() {
 super.viewDidLoad()
 createGradient()
}
```

## CAGradientLayer

o

<https://riptutorial.com/zh-TW/ios/topic/6854/>

---

**200:**

---

## Siri

- MyApp
- MyAppJohn
- MyApp
- MyAppJohn20
- VoIPMyAppMike
- MyApp
- CarPlay72

## Examples

### Siri ExtensionApp

Siri iOS 10 Widget。

---

1-iOS

2-Siri

---

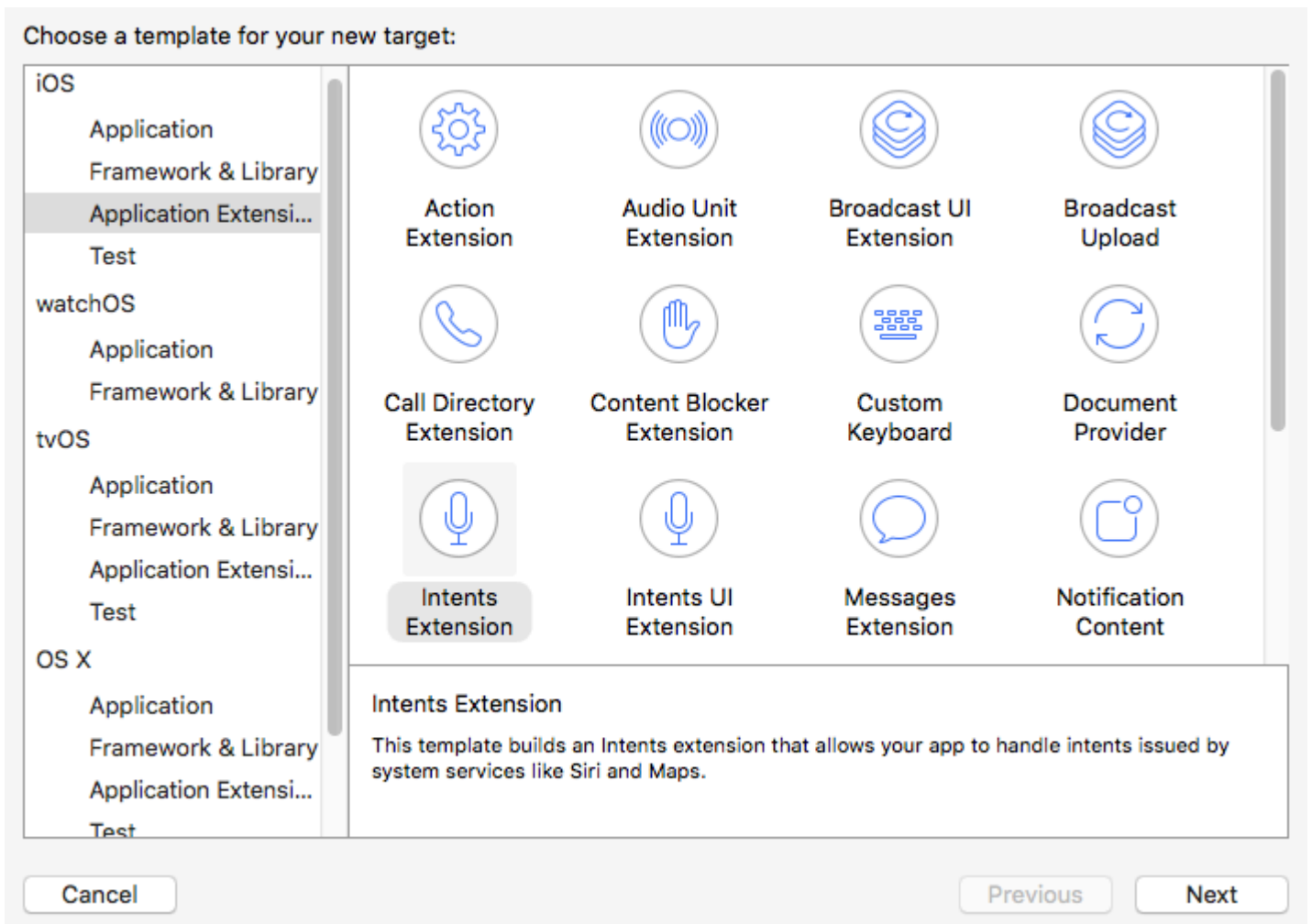
1- - > - >...

2-iOS - > Application Extension

3-Intents Extension

## Apple

IntentsIntentsSiriMaps。



#### 4-“UI”

Language:

Include UI Extension

Intents Extension UI Extension Workout Intent Siri

\_\_\_\_\_

Intent

\_\_\_\_\_

SiriKit

<https://riptutorial.com/zh-TW/ios/topic/5869/>

# 201: UIImage

## CGInterpolationQuality

```
typedef enum CGInterpolationQuality CGInterpolationQuality;
```

## Examples

```
- (UIImage *)drawImageBySize:(CGSize)size quality:(CGInterpolationQuality)quality
{
 UIGraphicsBeginImageContextWithOptions(size, NO, 0.0);
 CGContextRef context = UIGraphicsGetCurrentContext();
 CGContextSetInterpolationQuality(context, quality);
 [self drawInRect: CGRectMake (0, 0, size.width, size.height)];
 UIImage *resizedImage = UIGraphicsGetImageFromCurrentImageContext();
 UIGraphicsEndImageContext();
 return resizedImage;
}
```

UIImage <https://riptutorial.com/zh-TW/ios/topic/6422/uiimage>



---

**202:**

## Examples

Xcode

test > My Mac

test PID 12093

- CPU 0%
- Memory 1.3 MB
- Energy Impact Zero
- Disk Zero KB/s
- Network Zero KB/s

Thread 1 Queue: com....hread (serial)

- 0 \_\_pthread\_kill
- 10 +[NSArray arrayWithObject:]
- 11 main**
- 12 start
- 13 start

```

1 //
2 // main.m
3 // test
4 //
5 // Created
6 // Copyright
7 //
8
9 #import <Fou
10
11 int main(int
12 {
13 @autorel
14 id o
15 NSLo
16 }
17 return 0
18 }
19
20

```

- NSRangeException NSArrayNSString
- NSInternalInconsistencyException
- NSUnknownKeyException XIB

## EXC\_BAD\_ACCESS

EXC\_BAD\_ACCESS NULL / Swift

EXC\_BAD\_ACCESS NULL 0x0 printf / NSLog NULL

EXC\_BAD\_ACCESS

- 
- C
- 

Scheme EditorDiagnosticsXcode



test > My Mac



test > My Mac

**Build**  
1 target

**Run**  
Debug

**Test**  
Debug

**Profile**  
Release

**Analyze**  
Debug

**Archive**  
Release

No Debug Session

---

## 203:

Rich Notifications.RichUNNotificationServiceExtensionUNNotificationContentExtension

### Examples


#### UNNotificationContentExtension

1


。 “ ” “ ”




PROJECT

 TestApplication

TARGETS

 **TestApplication**


 TestAppNotifConten...




 Filter

 **Background Modes**

 **Inter-App Audio**

 **Keychain Sharing**

 **Associated Domains**

 **App Groups**



## PROJECT



TestApplication




## TARGETS



TestApplication



TestAppNotifConten...

  **iCloud**  **Push Notifications**  **Game Center**  **Wallet**  **Siri**  **Apple Pay**  **In-App Purchase**  **Maps**

Filter

## 2UNNotificationContentExtension

+ - > - > - >

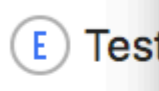




PROJECT



TARGETS



Choose a template for your new target:

ios watchOS tvOS macOS Cr

Application Extension



Action Extension



Audio Unit Extension



Content Blocker Extension



Custom Keyboard Extension



Intents UI Extension



Message Filter Extension



Click this + icon

- `UNNotificationDefaultContentHidden`
- `UNNotificationCategory` `UNUserNotificationCenter` ◦ `UI` ◦
- `UNNotificationExtensionInitialContentSizeRatioContentExtension` ◦ 1

#### 4 `UNNotificationAction` `UNNotificationCategory`

`AppDelegate.swift` `didFinishLaunchingWithOptions`

```

let userNotificationAction:UNNotificationAction = UNNotificationAction.init(identifier:
"ID1", title: "வணக்கம்", options: .destructive)
let userNotificationAction2:UNNotificationAction = UNNotificationAction.init(identifier:
"ID2", title: "Success", options: .destructive)

let notifCategory:UNNotificationCategory = UNNotificationCategory.init(identifier:
"CATID1", actions: [userNotificationAction,userNotificationAction2], intentIdentifiers:
["ID1","ID2"] , options:.customDismissAction)

UNUserNotificationCenter.current().delegate = self
UNUserNotificationCenter.current().setNotificationCategories([notifCategory])
UIApplication.shared.registerForRemoteNotifications()

```

`UNNotificationAction` `ID1` `ID2` `UNNotificationCategory` `CATID1` `categoryID` `ContentExtensionInfo.plist` `plist`

◦ `UNUserNotificationCenter` `didRegisterForRemoteNotificationsWithDeviceToken`

`AppDelegate.swift` `import UserNotifications` `UNUserNotificationCenterDelegate`

#### 5 `NotificationContent`

```

'aps': {
 'badge': 0,
 'alert': {
 'title': "Rich Notification",
 'body': "Body of RICH NOTIFICATION",
 },
 'sound' : "default",
 'category': "CATID1",
 'mutable-content':"1",
},
'attachment': "2"

```

#### 6 `ContentExtension`

◦

```

import UIKit
import UserNotifications
import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

@IBOutlet var imageView: UIImageView?
override func viewDidLoad() {
super.viewDidLoad()
}
}

```

```
func didReceive(_ notification: UNNotification) {
 self.title = "Koushik"
 imageView?.backgroundColor = UIColor.clear
 imageView?.image = #imageLiteral(resourceName: "welcome.jpeg")
}

func didReceive(_ response: UNNotificationResponse, completionHandler completion: @escaping
(UNNotificationContentExtensionResponseOption) -> Void) {

 self.title = "Koushik"
 imageView?.image = UIImage.init(named: "Success.jpeg")

 if(response.actionIdentifier == "ID1")
 {
 imageView?.image = UIImage.init(named: "Success.jpeg")
 }
 else
 {
 imageView?.image = UIImage.init(named: "welcome.jpeg")
 }

}
}
```

**7**

/



Koushik



**Rich Notification**  
Body of RICH NOTIFICATION

வணக்கம்

Success



Koushik



**Rich Notification**  
Body of RICH NOTIFICATION

வணக்கம்

Success



Koushik



Koushik



UNNotificationExtensionDefaultContentHiddenNO34。

<https://riptutorial.com/zh-TW/ios/topic/10769/>

---

# 204: iOS

## Examples

Mac

Carthage

**Carthage.pkg**

pkg

carthage0.18-19-g743fa0f

iOS <https://riptutorial.com/zh-TW/ios/topic/7404/ios>

# 205:

1. iOS 9 Safari. Safari.
2. Safari UIWebView / WKWebView.
3. iOS 9.2. iOS 9.3.
4. iOS Universal Links. Safari Safari. "".

## Examples

◦ iOS Apple apple-app-site-association. JSON.

apple-app-site-association HTTPS *https:// {domain} / apple-app-site-association* .

```
{
 "applinks": {
 "apps": [],
 "details": [
 {
 "appID": "{app_prefix}.{app_identifier}",
 "paths": ["/path/to/content", "/path/to/other/*", "NOT /path/to/exclude"]
 },
 {
 "appID": "TeamID.BundleID2",
 "paths": ["*"]
 }
]
 }
}
```

- .json apple-app-site-association

apps ◦ Apple.

details iOS. ID.

Static / static / terms

Wildcards A \*/ books / \*. books / 1ID1.

Exclusions NOT.

◦ ◦ ◦ ◦

---

#Website

<https://github.com/vineetchoudhary/iOS-Universal-Links/tree/gh-pages> gh-pages

apple-app-site-association. ◦ ◦

# App-Site-Association

HTTPS

iOS 9 HTTPS iOS 8 Handoff SSL

Apple App Store HTTPS

.txt

```
cat <unsigned_file>.txt | openssl smime -sign -inkey example.com.key -signer example.com.pem -certfile intermediate.pem -noattr -nodetach -outform DER > apple-app-site-association
```

example.com.key example.com.pem intermediate.pem

Content-Type of application/json application/pkcs7-mime

## Apple App

iOS 9 API Applebot <https://search.developer.apple.com/appsearch-validation-tool/>

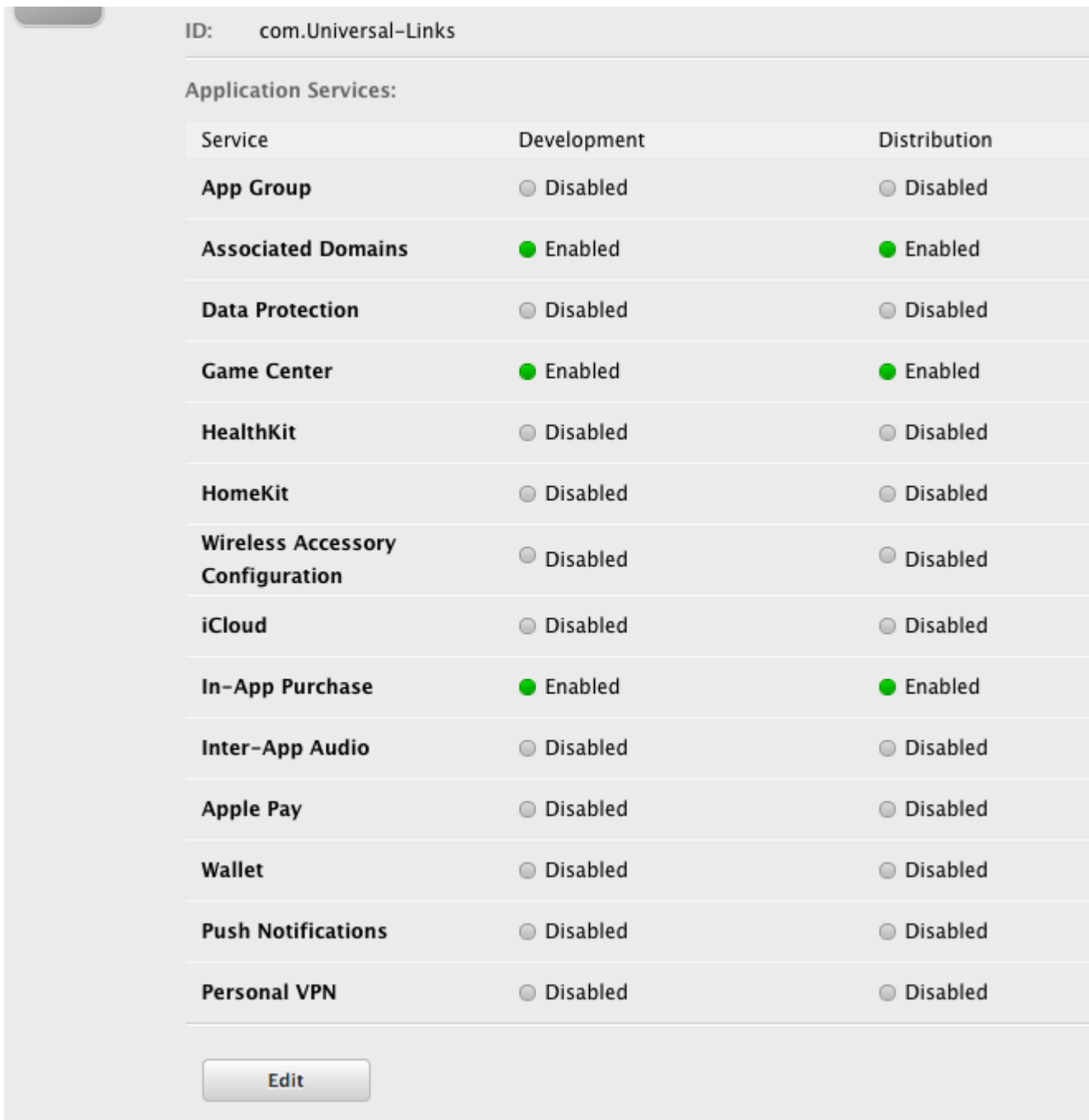
## iOS

- 1.
2. AppDelegate

1.

ID Apple Developer Member Center ID





App IDApp ID。

App IDapple-app-site-association。

Xcode ◦ **app**

**applinksYourCustomDomainName.com**


General Capabilities Resource Tags Info


**PROJECT**


- Universal Links


**TARGETS**


- Universal Links


▶  **Apple Pay**


▶  **In-App Purchase**


▶  **Personal VPN**

▶  **Maps**

▶  **Keychain Sharing**

▶  **Background Modes**


▶  **Inter-App Audio**


▼  **Associated Domains**

Domains:

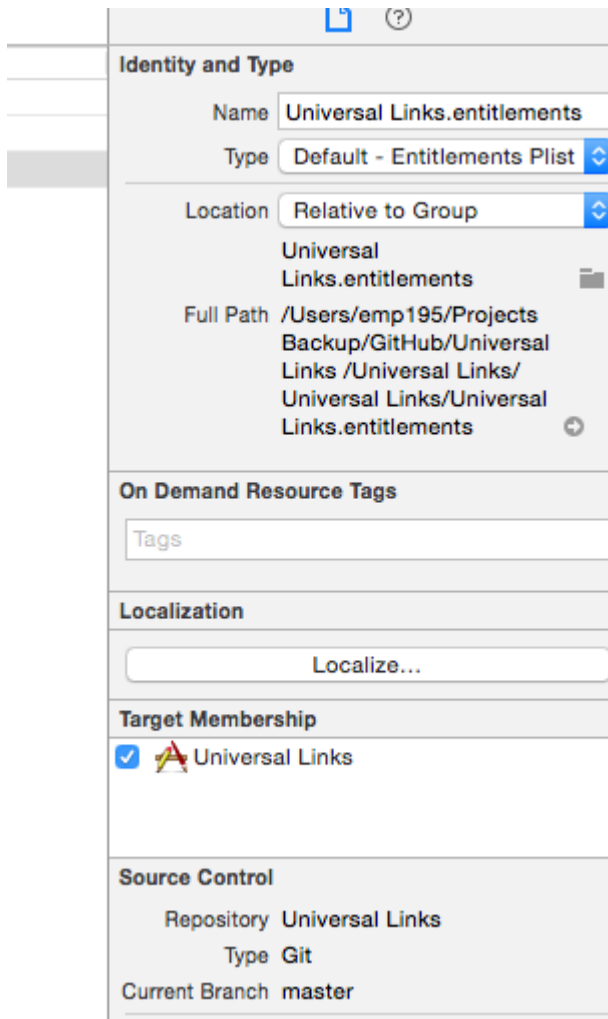
+ -

Steps:  Add the "Associated Domain"  Add the "Associated Domain"

▶  **App Groups**

▶  **Data Protection**

App IDBundle ID. Xcode.



## 2.AppDelegate

SafariApplicationAppDelegate。 URL。

```
[UIApplicationDelegate application: continueUserActivity: restorationHandler:]
```

## Objective-C

```
-(BOOL)application:(UIApplication *)application continueUserActivity:(NSUserActivity
*)userActivity restorationHandler:(void (^)(NSArray * _Nullable))restorationHandler{
 ///Checking whether the activity was from a web page redirect to the app.
 if ([userActivity.activityType isEqualToString: NSUserActivityTypeBrowsingWeb]) {
 ///Getting the URL from the UserActivity Object.
 NSURL *url = userActivity.webpageURL;
 UIStoryboard *storyBoard = [UIStoryboard storyboardWithName:@"Main" bundle:nil];
 UINavigationController *navigationController = (UINavigationController
*)_window.rootViewController;
 if ([url.pathComponents containsObject:@"home"]) {
 [navigationController pushViewController:[storyBoard
instantiateViewControllerWithIdentifier:@"HomeScreenId"] animated:YES];
 }else if ([url.pathComponents containsObject:@"about"]){
 [navigationController pushViewController:[storyBoard
instantiateViewControllerWithIdentifier:@"AboutScreenId"] animated:YES];
 }
 }
}
```

```
}
return YES;
}
```

```
func application(application: UIApplication, continueUserActivity userActivity:
NSUserActivity, restorationHandler: ([AnyObject]?) -> Void) -> Bool {
 if userActivity.activityType == NSUserActivityTypeBrowsingWeb {
 let url = userActivity.webpageURL!
 //handle url
 }
 return true
}
```

---

# ios

master。

<https://riptutorial.com/zh-TW/ios/topic/2362/>

# 206: iOS

MVCMVPMVVMVIPER

## Examples

### MVC

```
import UIKit

struct Person { // Model
 let firstName: String
 let lastName: String
}

class GreetingViewController : UIViewController { // View + Controller
 var person: Person!
 let showGreetingButton = UIButton()
 let greetingLabel = UILabel()

 override func viewDidLoad() {
 super.viewDidLoad()
 self.showGreetingButton.addTarget(self, action: "didTapButton:", forControlEvents:
.TouchUpInside)
 }

 func didTapButton(button: UIButton) {
 let greeting = "Hello" + " " + self.person.firstName + " " + self.person.lastName
 self.greetingLabel.text = greeting
 }

 // layout code goes here
}

// Assembling of MVC
let model = Person(firstName: "David", lastName: "Blaine")
let view = GreetingViewController()
view.person = model
```

### MVP

```
import UIKit

struct Person { // Model
 let firstName: String
 let lastName: String
}

protocol GreetingView: class {
 func setGreeting(greeting: String)
}

protocol GreetingViewPresenter {
 init(view: GreetingView, person: Person)
 func showGreeting()
}
```

```

}

class GreetingPresenter : GreetingViewPresenter {
 unowned let view: GreetingView
 let person: Person
 required init(view: GreetingView, person: Person) {
 self.view = view
 self.person = person
 }
 func showGreeting() {
 let greeting = "Hello" + " " + self.person.firstName + " " + self.person.lastName
 self.view.setGreeting(greeting)
 }
}

class GreetingViewController : UIViewController, GreetingView {
 var presenter: GreetingViewPresenter!
 let showGreetingButton = UIButton()
 let greetingLabel = UILabel()

 override func viewDidLoad() {
 super.viewDidLoad()
 self.showGreetingButton.addTarget(self, action: "didTapButton:", forControlEvents:
.TouchUpInside)
 }

 func didTapButton(button: UIButton) {
 self.presenter.showGreeting()
 }

 func setGreeting(greeting: String) {
 self.greetingLabel.text = greeting
 }

 // layout code goes here
}
// Assembling of MVP
let model = Person(firstName: "David", lastName: "Blaine")
let view = GreetingViewController()
let presenter = GreetingPresenter(view: view, person: model)
view.presenter = presenter

```

## MVVM

```

import UIKit

struct Person { // Model
 let firstName: String
 let lastName: String
}

protocol GreetingViewModelProtocol: class {
 var greeting: String? { get }
 var greetingDidChange: ((GreetingViewModelProtocol) -> ())? { get set } // function to
call when greeting did change
 init(person: Person)
 func showGreeting()
}

```

```

class GreetingViewModel : GreetingViewModelProtocol {
 let person: Person
 var greeting: String? {
 didSet {
 self.greetingDidChange?(self)
 }
 }
 var greetingDidChange: ((GreetingViewModelProtocol) -> ())?
 required init(person: Person) {
 self.person = person
 }
 func showGreeting() {
 self.greeting = "Hello" + " " + self.person.firstName + " " + self.person.lastName
 }
}

class GreetingViewController : UIViewController {
 var viewModel: GreetingViewModelProtocol! {
 didSet {
 self.viewModel.greetingDidChange = { [unowned self] viewModel in
 self.greetingLabel.text = viewModel.greeting
 }
 }
 }
 let showGreetingButton = UIButton()
 let greetingLabel = UILabel()

 override func viewDidLoad() {
 super.viewDidLoad()
 self.showGreetingButton.addTarget(self.viewModel, action: "showGreeting",
forControlEvents: .TouchUpInside)
 }
 // layout code goes here
}
// Assembling of MVVM
let model = Person(firstName: "David", lastName: "Blaine")
let viewModel = GreetingViewModel(person: model)
let view = GreetingViewController()
view.viewModel = viewModel

```

## VIPER

```

import UIKit

struct Person { // Entity (usually more complex e.g. NSManagedObject)
 let firstName: String
 let lastName: String
}

struct GreetingData { // Transport data structure (not Entity)
 let greeting: String
 let subject: String
}

protocol GreetingProvider {
 func provideGreetingData()
}

protocol GreetingOutput: class {

```

```

 func receiveGreetingData(greetingData: GreetingData)
}

class GreetingInteractor : GreetingProvider {
 weak var output: GreetingOutput!

 func provideGreetingData() {
 let person = Person(firstName: "David", lastName: "Blaine") // usually comes from data
 access layer
 let subject = person.firstName + " " + person.lastName
 let greeting = GreetingData(greeting: "Hello", subject: subject)
 self.output.receiveGreetingData(greeting)
 }
}

protocol GreetingViewEventHandler {
 func didTapShowGreetingButton()
}

protocol GreetingView: class {
 func setGreeting(greeting: String)
}

class GreetingPresenter : GreetingOutput, GreetingViewEventHandler {
 weak var view: GreetingView!
 var greetingProvider: GreetingProvider!

 func didTapShowGreetingButton() {
 self.greetingProvider.provideGreetingData()
 }

 func receiveGreetingData(greetingData: GreetingData) {
 let greeting = greetingData.greeting + " " + greetingData.subject
 self.view.setGreeting(greeting)
 }
}

class GreetingViewController : UIViewController, GreetingView {
 var eventHandler: GreetingViewEventHandler!
 let showGreetingButton = UIButton()
 let greetingLabel = UILabel()

 override func viewDidLoad() {
 super.viewDidLoad()
 self.showGreetingButton.addTarget(self, action: "didTapButton:", forControlEvents:
.TouchUpInside)
 }

 func didTapButton(button: UIButton) {
 self.eventHandler.didTapShowGreetingButton()
 }

 func setGreeting(greeting: String) {
 self.greetingLabel.text = greeting
 }

 // layout code goes here
}

// Assembling of VIPER module, without Router
let view = GreetingViewController()
let presenter = GreetingPresenter()

```



```
let interactor = GreetingInteractor()
view.eventHandler = presenter
presenter.view = view
presenter.greetingProvider = interactor
interactor.output = presenter
```

iOS <https://riptutorial.com/zh-TW/ios/topic/10029/ios>

---

## 207:

- kSecClassGenericPassword //Internet
- kSecClassInternetPassword //
- kSecClassCertificate //
- kSecClassCertificate //
- kSecClassIdentity //

iOSKeychain. Secure Enclave. iOS.

Xcode.

Keychain Servicesac. [AppleKeychain Services Programming Guide](#) .

Keychain ServicesFoundationCoreFoundation. CFDictionaryCFStringCoreFoundation.

Keychain ServicesSecurityFoundation.

KeychainApple[Generic Keychain](#) SwiftKeychain ServicesSwift.

## Examples

CFDictionary. **Objective-C**NSDictionary**Swift**DictionaryCFDictionary.

---

```
var dict = [String : AnyObject]()
```

/Keychain. String CFStringString**3**. CFString.

---

```
dict[kSecClass as String] = kSecClassGenericPassword
```

◦ ◦

---

```
// The password will only be accessible when the device is unlocked
dict[kSecAttrAccessible as String] = kSecAttrAccessibleWhenUnlocked
// Label may help you find it later
dict[kSecAttrLabel as String] = "com.me.myapp.myaccountpassword" as CFString
// Username
dict[kSecAttrAccount as String] = "My Name" as CFString
// Service name
dict[kSecAttrService as String] = "MyService" as CFString
```

◦ ◦ CFData ◦

```
dict[kSecValueData as String] = "my_password!!".data(using: .utf8) as! CFData
```

## Keychain Services addkeychain◦

---

```
dict[kSecReturnAttributes as String] = kCFBooleanTrue
```

◦

```
var result: AnyObject?
let status = withUnsafeMutablePointer(to: &result) {
 SecItemAdd(dict as CFDictionary, UnsafeMutablePointer($0))
}
let newAttributes = result as! Dictionary<String, AnyObject>
```

**dict**◦ SecItemAdd◦ OSStatus◦ ◦

CFDictionary◦ **Objective-C**NSDictionary**Swift**DictionaryCFDictionary◦

```
var dict = [String : AnyObject]()
dict[kSecClass as String] = kSecClassGenericPassword
```

```
// Label
dict[kSecAttrLabel as String] = "com.me.myapp.myaccountpassword" as CFString
// Username
dict[kSecAttrAccount as String] = "My Name" as CFString
// Service name
dict[kSecAttrService as String] = "MyService" as CFString
```

◦

◦ CFData

```
dict[kSecReturnData as String] = kCFBooleanTrue
```

```
var queryResult: AnyObject?
let status = withUnsafeMutablePointer(to: &queryResult) {
 SecItemCopyMatching(dict as CFDictionary, UnsafeMutablePointer($0))
}
// Don't keep this in memory for long!!
let password = String(data: queryResult as! Data, encoding: .utf8)!
```

SecItemCopyMatching ◦ OSStatus ◦ ◦

CFDictionary ◦ ◦ CFDictionary ◦

◦ ◦

---

```
var dict = [String : AnyObject]()
dict[kSecClass as String] = kSecClassGenericPassword
// Label
dict[kSecAttrLabel as String] = "com.me.myapp.myaccountpassword" as CFString
// Username
dict[kSecAttrAccount as String] = "My Name" as CFString
```

---

```
dict[kSecValueData as String] = "my_password!".data(using: .utf8) as! CFData
```

---

```
var newDict = [String : AnyObject]()
newDict[kSecClass as String] = kSecClassGenericPassword
// Label
newDict[kSecAttrLabel as String] = "com.me.myapp.myaccountpassword" as CFString
// Username
newDict[kSecAttrAccount as String] = "My Name" as CFString
// New password
newDict[kSecValueData as String] = "new_password!".data(using: .utf8) as! CFData
```

## Keychain Services

---

```
let status = SecItemUpdate(dict as CFDictionary, newDict as CFDictionary)
```

SecItemUpdate ◦ ◦

## KeychainCFDictionary ◦ Objective-CSwiftNSDictionary DictionaryCFDictionary ◦

◦ kSecMatchCaseInsensitive ◦

---

```
var dict = [String : AnyObject]()
dict[kSecClass as String] = kSecClassGenericPassword
// Label
dict[kSecAttrLabel as String] = "com.me.myapp.myaccountpassword" as CFString
// Username
dict[kSecAttrAccount as String] = "My Name" as CFString
```

---

```
let status = SecItemDelete(dict as CFDictionary)
```

SecItemDeleteOSStatus ◦ ◦

## Keychain◦

### Keychain.h

```
#import <Foundation/Foundation.h>
typedef void (^KeychainOperationBlock)(BOOL successfulOperation, NSData *data, OSStatus
status);

@interface Keychain : NSObject

-(id) initWithService:(NSString *) service_ withGroup:(NSString*)group_;

-(void)insertKey:(NSString *)key withData:(NSData *)data
withCompletion:(KeychainOperationBlock)completionBlock;
-(void)updateKey:(NSString*)key withData:(NSData*) data
withCompletion:(KeychainOperationBlock)completionBlock;
-(void)removeDataForKey:(NSString*)key
withCompletionBlock:(KeychainOperationBlock)completionBlock;
-(void)findDataForKey:(NSString*)key
withCompletionBlock:(KeychainOperationBlock)completionBlock;

@end
```

### Keychain.m

```
#import "Keychain.h"
#import <Security/Security.h>

@implementation Keychain

{
 NSString * keychainService;
 NSString * keychainGroup;
}

-(id) initWithService:(NSString *)service withGroup:(NSString*)group
{
 self =[super init];
 if(self) {
 keychainService = [NSString stringWithString:service];
 if(group) {
 keychainGroup = [NSString stringWithString:group];
 }
 }

 return self;
}

-(void)insertKey:(NSString *)key
 withData:(NSData *)data
 withCompletion:(KeychainOperationBlock)completionBlock
{
 NSMutableDictionary * dict =[self prepareDict:key];
 [dict setObject:data forKey:(__bridge id)kSecValueData];
 [dict setObject:keychainService forKey:(id)kSecAttrService];
```

```

OSStatus status = SecItemAdd((__bridge CFDictionaryRef)dict, NULL);
if(errSecSuccess != status) {
 DLog(@"Unable add item with key =%@ error:%d",key, (int)status);
 if (completionBlock) {
 completionBlock(errSecSuccess == status, nil, status);
 }
}
if (status == errSecDuplicateItem) {
 [self updateKey:key withData:data withCompletion:^(BOOL successfulOperation, NSData
*updateData, OSStatus updateStatus) {
 if (completionBlock) {
 completionBlock(successfulOperation, updateData, updateStatus);
 }
 DLog(@"Found duplication item -- updating key with data");
 }];
}
}

-(void)findDataForKey:(NSString *)key
withCompletionBlock:(KeychainOperationBlock)completionBlock
{
 NSMutableDictionary *dict = [self prepareDict:key];
 [dict setObject:(__bridge id)kSecMatchLimitOne forKey:(__bridge id)kSecMatchLimit];
 [dict setObject:keychainService forKey:(id)kSecAttrService];
 [dict setObject:(id)kCFBooleanTrue forKey:(__bridge id)kSecReturnData];
 CFTypeRef result = NULL;
 OSStatus status = SecItemCopyMatching((__bridge CFDictionaryRef)dict,&result);

 if(status != errSecSuccess) {
 DLog(@"Unable to fetch item for key %@ with error:%d",key, (int)status);
 if (completionBlock) {
 completionBlock(errSecSuccess == status, nil, status);
 }
 } else {
 if (completionBlock) {
 completionBlock(errSecSuccess == status, (__bridge NSData *)result, status);
 }
 }
}

-(void)updateKey:(NSString *)key
withData:(NSData *)data
withCompletion:(KeychainOperationBlock)completionBlock
{
 NSMutableDictionary * dictKey =[self prepareDict:key];

 NSMutableDictionary * dictUpdate =[NSMutableDictionary alloc] init];
 [dictUpdate setObject:data forKey:(__bridge id)kSecValueData];
 [dictUpdate setObject:keychainService forKey:(id)kSecAttrService];
 OSStatus status = SecItemUpdate((__bridge CFDictionaryRef)dictKey, (__bridge
CFDictionaryRef)dictUpdate);
 if(status != errSecSuccess) {
 DLog(@"Unable to remove item for key %@ with error:%d",key, (int)status);
 }
 if (completionBlock) {
 completionBlock(errSecSuccess == status, nil, status);
 }
}

-(void)removeDataForKey:(NSString *)key
withCompletionBlock:(KeychainOperationBlock)completionBlock {

```

```

NSMutableDictionary *dict = [self prepareDict:key];
OSStatus status = SecItemDelete((__bridge CFDictionaryRef)dict);
if(status != errSecSuccess) {
 DLog(@"Unable to remove item for key %@ with error:%d",key,(int)status);
}
if (completionBlock) {
 completionBlock(errSecSuccess == status, nil, status);
}
}

#pragma mark Internal methods

-(NSMutableDictionary*) prepareDict:(NSString *) key {

 NSMutableDictionary *dict = [[NSMutableDictionary alloc] init];
 [dict setObject:(__bridge id)kSecClassGenericPassword forKey:(__bridge id)kSecClass];

 NSData *encodedKey = [key dataUsingEncoding:NSUTF8StringEncoding];
 [dict setObject:encodedKey forKey:(__bridge id)kSecAttrGeneric];
 [dict setObject:encodedKey forKey:(__bridge id)kSecAttrAccount];
 [dict setObject:keychainService forKey:(__bridge id)kSecAttrService];
 [dict setObject:(__bridge id)kSecAttrAccessibleAlwaysThisDeviceOnly forKey:(__bridge
id)kSecAttrAccessible];

 //This is for sharing data across apps
 if(keychainGroup != nil) {
 [dict setObject:keychainGroup forKey:(__bridge id)kSecAttrAccessGroup];
 }

 return dict;
}

@end

```

## TouchID

KeychainSecAccessControlTouch IDKeychain。 UIiOS。

### SecAccessControl

---

```

let error: Unmanaged<CFError>?

guard let accessControl = SecAccessControlCreateWithFlags(kCFAllocatorDefault,
kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly, .userPresence, &error) else {
 fatalError("Something went wrong")
}

```

### kSecAttrAccessControlkSecAttrAccessible

---

```

var dictionary = [String : Any]()

dictionary[kSecClass as String] = kSecClassGenericPassword
dictionary[kSecAttrLabel as String] = "com.me.myapp.myaccountpassword" as CFString
dictionary[kSecAttrAccount as String] = "My Name" as CFString

```

```
dictionary[kSecValueData as String] = "new_password!!".data(using: .utf8) as! CFData
dictionary[kSecAttrAccessControl as String] = accessControl
```

```
let lastResultCode = SecItemAdd(query as CFDictionary, nil)
```

## Keychain。 Keychain Servicesnil。

```
var query = [String: Any]()

query[kSecClass as String] = kSecClassGenericPassword
query[kSecReturnData as String] = kCFBooleanTrue
query[kSecAttrAccount as String] = "My Name" as CFString
query[kSecAttrLabel as String] = "com.me.myapp.myaccountpassword" as CFString
query[kSecUseOperationPrompt as String] = "Please put your fingers on that button" as CFString

var queryResult: AnyObject?
let status = withUnsafeMutablePointer(to: &queryResult) {
 SecItemCopyMatching(query as CFDictionary, UnsafeMutablePointer($0))
}
```

statuserr。

```
if status == noErr {
 let password = String(data: queryResult as! Data, encoding: .utf8)!
 print("Password: \(password)")
} else {
 print("Authorization not passed")
}
```

<https://riptutorial.com/zh-TW/ios/topic/6839/>



## 208: -

### KVC -

KVC。 “age”。

```
For example, you have employee class with "age" property. Normally we access like this.
emp.age = @"20";
NSString age = emp.age;
```

```
But KVC works like this:
[emp valueForKey:@"age"];
[emp setValue:@"25" forKey:@"age"];
```

### KVO -

KVO。

```
personBankAccountaccountBalance。 Person ObjectaddObserverforKeyPathoptions
contextmessageBankAccountaccountBalance。
```

## Examples

### KVO

```
-(void)observeValueForKeyPath:(NSString *)keyPath ofObject:(id)object
change:(NSDictionary<NSString *,id> *)change context:(void *)context
```

.Context。

- 
- 
- 

### NSObject

KVOKVCNSObject。

personObjectfirstName

```
[personObject addObserver:self
 forKeyPath:@"firstName"
 options:NSKeyValueObservingOptionNew
 context:nil];
```

selfobserveValueForKeyPath:ofObject:change:context: **message**。

```
- (void)observeValueForKeyPath:(NSString *)keyPath
 ofObject:(id)object
 change:(NSDictionary<NSString *,id> *)change
 context:(void *)context
{
 NSLog(@"new value of %@ is: %@", keyPath, change[NSKeyValueChangeNewKey]);
}
```

“KVC” NSObject KVC

\_firstName@"firstName"

@"firstName"\_firstNamesetFirstName **setter**firstName **getter**

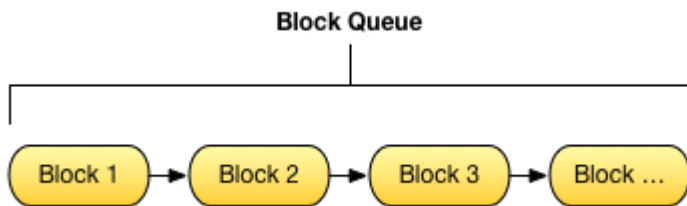
- <https://riptutorial.com/zh-TW/ios/topic/3493/--->

# 209: MKBlockQueue

MKBlockQueue。 NSOperationMKBlockQueue。 ◦

<https://github.com/MKGitHub/MKBlockQueue>

## Examples



```
// create the dictionary that will be sent to the blocks
var myDictionary:Dictionary<String, Any> = Dictionary<String, Any>()
myDictionary["InitialKey"] = "InitialValue"

// create block queue
let myBlockQueue:MKBlockQueue = MKBlockQueue()

// block 1
let b1:MKBlockQueueBlockType =
{
 (blockQueueObserver:MKBlockQueueObserver, dictionary:inout Dictionary<String, Any>) in

 print("Block 1 started with dictionary: \(dictionary)")
 dictionary["Block1Key"] = "Block1Value"

 // tell this block is now completed
 blockQueueObserver.blockCompleted(with:&dictionary)
}

// block 2
let b2:MKBlockQueueBlockType =
{
 (blockQueueObserver:MKBlockQueueObserver, dictionary:inout Dictionary<String, Any>) in

 var copyOfDictionary:Dictionary<String, Any> = dictionary

 // test calling on main thread, async, with delay
 DispatchQueue.main.asyncAfter(deadline:(.now() + .seconds(1)), execute:
 {
 print("Block 2 started with dictionary: \(copyOfDictionary)")

 copyOfDictionary["Block2Key"] = "Block2Value"

 // tell this block is now completed
 blockQueueObserver.blockCompleted(with:©OfDictionary)
 })
}

// block 3
```

```

let b3:MKBlockQueueBlockType =
{
 (blockQueueObserver:MKBlockQueueObserver, dictionary:inout Dictionary<String, Any>) in

 var copyOfDictionary:Dictionary<String, Any> = dictionary

 // test calling on global background queue, async, with delay
 DispatchQueue.global(qos:.background).asyncAfter(deadline:(.now() + .seconds(1)), execute:
 {
 print("Block 3 started with dictionary: \(copyOfDictionary)")

 copyOfDictionary["Block3Key"] = "Block3Value"

 // tell this block is now completed
 blockQueueObserver.blockCompleted(with:©OfDictionary)
 })
}

// add blocks to the queue
myBlockQueue.addBlock(b1)
myBlockQueue.addBlock(b2)
myBlockQueue.addBlock(b3)

// add queue completion block for the queue
myBlockQueue.queueCompletedBlock(
{
 (dictionary:Dictionary<String, Any>) in
 print("Queue completed with dictionary: \(dictionary)")
})

// run queue
print("Queue starting with dictionary: \(myDictionary)")
myBlockQueue.run(with:&myDictionary)

```

**MKBlockQueue** <https://riptutorial.com/zh-TW/ios/topic/9122/-mkblockqueue->

# 210:

## RLMObjectRealm - SchemaMigrations

Realm;◦

## Examples

### RLMObject - Objective-C

RLMObject◦ ◦

```
@interface BaseModel : RLMObject

@property NSString *uuid;
@property NSString *metadata;

@end

@implementation BaseModel

+ (NSString *)primaryKey
{
 return @"uuid";
}

+ (NSDictionary *)defaultPropertyValues
{
 NSMutableDictionary *defaultPropertyValues = [NSMutableDictionary
dictionaryWithDictionary:[super defaultPropertyValues]];
 NSString *uuid = [[NSUUID UUID] UUIDString];
 [defaultPropertyValues setValue:@"" forKey:@"metadata"];
 [defaultPropertyValues setValue:uuid forKey:@"uuid"];
 return defaultPropertyValues;
}

+ (NSArray *)ignoredProperties
{
 return @[];
}

@end
```

<https://riptutorial.com/zh-TW/ios/topic/4084/>

S. No		Contributors
1	iOS	<a href="#">Ali Beadle</a> , <a href="#">Allan Burlison</a> , <a href="#">Anand Nimje</a> , <a href="#">Anatoliy</a> , <a href="#">Ashutosh Dave</a> , <a href="#">Bhadresh Kathiriya</a> , <a href="#">bjtitus</a> , <a href="#">Blachshma</a> , <a href="#">bmike</a> , <a href="#">Charlie H</a> , <a href="#">Cin316</a> , <a href="#">Community</a> , <a href="#">Dair</a> , <a href="#">dan</a> , <a href="#">deyanm</a> , <a href="#">Efraim Weiss</a> , <a href="#">Erik Godard</a> , <a href="#">FelixSFD</a> , <a href="#">Fogmeister</a> , <a href="#">Hudson Taylor</a> , <a href="#">Irfan</a> , <a href="#">J F</a> , <a href="#">Jack Ngai</a> , <a href="#">James</a> , <a href="#">Josh Brown</a> , <a href="#">jrf</a> , <a href="#">Kampai</a> , <a href="#">Kevin</a> , <a href="#">Losiowaty</a> , <a href="#">M. Galban</a> , <a href="#">Maddy</a> , <a href="#">Matthew Cawley</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">Midhun MP</a> , <a href="#">Miguel Cabezas</a> , <a href="#">Muhammad Zohaib Ehsan</a> , <a href="#">Pro Q</a> , <a href="#">PSN</a> , <a href="#">RamenChef</a> , <a href="#">Sam Fischer</a> , <a href="#">Seyyed Parsa Neshaei</a> , <a href="#">shim</a> , <a href="#">Skeleton Bow</a> , <a href="#">Stephen Leppik</a> , <a href="#">Steve Moser</a> , <a href="#">Suragch</a> , <a href="#">SuzGupta</a> , <a href="#">The_Curry_Man</a> , <a href="#">ThrowingSpoon</a> , <a href="#">Undo</a> , <a href="#">user3480295</a> , <a href="#">user6939352</a> , <a href="#">Vignan</a>
2	3D	<a href="#">4444</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">LinusGeffarth</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">Onur Tuna</a> , <a href="#">Stephen Leppik</a> , <a href="#">tobeiosdeveloper</a>
3	AFNetworking	<a href="#">4444</a> , <a href="#">Mayank Patel</a> , <a href="#">OhadM</a> , <a href="#">Ruby</a>
4	Alamofire	<a href="#">Alex Koshy</a> , <a href="#">Josh Caswell</a> , <a href="#">Sour LeangChhean</a> , <a href="#">yogesh wadhwa</a>
5	AppDelegate	<a href="#">CodeChanger</a> , <a href="#">Oleh Zayats</a> , <a href="#">Saumil Shah</a>
6	ARC	<a href="#">4444</a> , <a href="#">Irfan</a> , <a href="#">John Militer</a> , <a href="#">Ketan P</a> , <a href="#">Tricertops</a>
7	Autolayout/	<a href="#">Mehul Chuahan</a>
8	AVPlayer AVPlayerViewController	<a href="#">Bonnie</a> , <a href="#">Chirag Desai</a> , <a href="#">Gazi Alankus</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Konda Yadav</a> , <a href="#">Stephen Leppik</a>
9	AVSpeechSynthesizer	<a href="#">Ali Beadle</a> , <a href="#">Bhumit Mehta</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Midhun MP</a> , <a href="#">Stephen Leppik</a>
10	AWSSDK	<a href="#">OhadM</a>
11	CAAnimation	<a href="#">Bhavin Ramani</a> , <a href="#">James P</a> , <a href="#">Mr. Xcoder</a> , <a href="#">Narendra Pandey</a> , <a href="#">Rahul</a> , <a href="#">Rob</a> , <a href="#">Undo</a>
12	CAGradientLayer	<a href="#">Bhavin Ramani</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Sam Fischer</a> , <a href="#">Stephen Leppik</a> , <a href="#">Undo</a>
13	CALayer	<a href="#">Alistra</a> , <a href="#">Dunja Lalic</a> , <a href="#">HariKrishnan.P</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">ignotusverum</a> , <a href="#">iOS BadBoy</a> , <a href="#">Kamil Harasimowicz</a> , <a href="#">Luiz Henrique Guimaraes</a> , <a href="#">Stephen Leppik</a> , <a href="#">Suragch</a> , <a href="#">Viktor</a>

		<a href="#">Simkó</a> , <a href="#">william205</a>
14	CAShapeLayer	<a href="#">Filip Radelic</a> , <a href="#">HariKrishnan.P</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Narendra Pandey</a> , <a href="#">Stephen Leppik</a>
15	CGContext	<a href="#">4444</a> , <a href="#">Narendra Pandey</a>
16	CLLocation	<a href="#">amar</a> , <a href="#">Duly Kinsky</a> , <a href="#">FelixSFD</a> , <a href="#">Siddharth Sunil</a> , <a href="#">Sujania</a> , <a href="#">That lazy iOS Guy</a> , <a href="#">void</a> , <a href="#">Zee</a>
17	CloudKit	<a href="#">Seyyed Parsa Neshaei</a>
18	CoreImage	<a href="#">Md. Ibrahim Hassan</a>
19	CTCallCenter	<a href="#">MANI</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">OhadM</a>
20	CydiaSubstrate	<a href="#">gkpln3</a>
21	DispatchGroup	<a href="#">Brandon</a> , <a href="#">Fonix</a>
22	EventKit	<a href="#">Seyyed Parsa Neshaei</a>
23	FacebookSDK	<a href="#">Brian</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Irfan</a> , <a href="#">Mehul Chuahan</a> , <a href="#">OhadM</a> , <a href="#">Ravi Prakash Verma</a> , <a href="#">Stephen Leppik</a>
24	GameCenter	<a href="#">4444</a> , <a href="#">Cyril Ivar Garcia</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Stephen Leppik</a>
25	GameplayKit	<a href="#">BennX</a> , <a href="#">Seyyed Parsa Neshaei</a>
26	GCDGrand Central Dispatch	<a href="#">Andrea Antonioni</a> , <a href="#">DS Dharma</a> , <a href="#">Fonix</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">skyline75489</a>
27	Healthkit	<a href="#">Md. Ibrahim Hassan</a>
28	iBeacon	<a href="#">amar</a> , <a href="#">Arefly</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Stephen Leppik</a>
29	IBOutlets	<a href="#">Fabio</a> , <a href="#">SharkbaitWhohaha</a>
30	iOS - Robbie Hanson XMPP	<a href="#">Saheb Roy</a>
31	iOS 10API	<a href="#">rohit90</a> , <a href="#">Stephen Leppik</a>
32	iOS Google Places API	<a href="#">Cyril Ivar Garcia</a> , <a href="#">Vignan</a>
33	iOS TTS	<a href="#">Ali Abbas</a> , <a href="#">Stephen Leppik</a>
34	iOSAirPrint	<a href="#">Md. Ibrahim Hassan</a>
35	iOSSpotLight	<a href="#">Md. Ibrahim Hassan</a>
36	iOS	<a href="#">bryanjclark</a> , <a href="#">Dunja Lalic</a> , <a href="#">FelixSFD</a> , <a href="#">sanman</a>

37	MKDistanceFormatter	<a href="#">Harshal Bhavsar</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">Stephen Leppik</a> , <a href="#">Undo</a>
38	ModelPresentationStyles	<a href="#">Dishant Kapadiya</a>
39	MPMediaPickerDelegate	<a href="#">FelixSFD</a> , <a href="#">George Lee</a>
40	MPVolumeView	<a href="#">lostAtSeaJoshua</a>
41	MVP	<a href="#">Oleh Zayats</a>
42	MVVM	<a href="#">JPetric</a>
43	MyLayout	
44	NSArray	<a href="#">Krunal</a> , <a href="#">user5553647</a>
45	NSAttributedString	<a href="#">Bhavin Ramani</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Jinhuan Li</a> , <a href="#">Kirit Modi</a> , <a href="#">Luiz Henrique Guimaraes</a> , <a href="#">Mansi Panchal</a> , <a href="#">Stephen Leppik</a> , <a href="#">Tim</a> , <a href="#">Tim Ebenezer</a> , <a href="#">Undo</a>
46	NSHTTPCookieStorage	<a href="#">balagurubaran</a>
47	NSInvocation	<a href="#">Md. Ibrahim Hassan</a>
48	NSNotificationCenter	<a href="#">Alex Kallam</a> , <a href="#">Alex Koshy</a> , <a href="#">Anand Nimje</a> , <a href="#">Bence Pattogato</a> , <a href="#">Bright Future</a> , <a href="#">Ichthyocentaurs</a> , <a href="#">Jacopo Penzo</a> , <a href="#">James P</a> , <a href="#">Kirit Modi</a> , <a href="#">Tarun Seera</a>
49	NSPredicate	<a href="#">Brendon Roberto</a> , <a href="#">Joshua</a> , <a href="#">Mehul Chuahan</a>
50	NSURL	<a href="#">Adnan Aftab</a> , <a href="#">ApolloSoftware</a> , <a href="#">tharkay</a>
51	NSURLConnection	<a href="#">byJeevan</a>
52	NSURLSession	<a href="#">bluey31</a> , <a href="#">dasdom</a> , <a href="#">dgatwood</a> , <a href="#">Duly Kinsky</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Narendra Pandey</a> , <a href="#">Otávio</a> , <a href="#">R P</a> , <a href="#">sage444</a> , <a href="#">Stephen Leppik</a>
53	NSUserActivity	<a href="#">Samuel Spencer</a>
54	NSUserDefaults	<a href="#">Anand Nimje</a> , <a href="#">Emptyless</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Husein Behboodi Rad</a> , <a href="#">J F</a> , <a href="#">James P</a> , <a href="#">Josh Caswell</a> , <a href="#">Kirit Modi</a> , <a href="#">Mr. Xcoder</a> , <a href="#">Roland Keesom</a> , <a href="#">Seyyed Parsa Neshaei</a> , <a href="#">user3760892</a> , <a href="#">william205</a>
55	Objective-C	<a href="#">halil_g</a>
56	Objective-C	<a href="#">Noam</a>
57	OpenGL	<a href="#">Fonix</a>



58	PDFiOS	<a href="#">Mansi Panchal</a> , <a href="#">Narendra Pandey</a>
59	plist iOS	<a href="#">SNarula</a>
60	QR	<a href="#">Bluewings</a> , <a href="#">Efraim Weiss</a>
61	Safari	<a href="#">Arnon Rodrigues</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Kilian Koeltzsch</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">Stephen Leppik</a>
62	SLComposeViewController	<a href="#">Md. Ibrahim Hassan</a>
63	SqlCipher	<a href="#">Nirav</a>
64	StoreKit	<a href="#">askielboe</a>
65	SwiftAppDelegate rootViewControllermain login / onboarding	<a href="#">cleverbit</a>
66	SwiftFCM	<a href="#">Saeed-rz</a>
67	SwiftObjective-C	<a href="#">Harshal Bhavsar</a> , <a href="#">njuri</a> , <a href="#">Stephen Leppik</a>
68	SWRevealViewController	<a href="#">Reinier Melian</a> , <a href="#">tharkay</a>
69	UIActivityViewController	<a href="#">Amandeep</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Stephen Leppik</a> , <a href="#">Vivek Molkar</a>
70	UIAlertController	<a href="#">Andrii Chernenko</a> , <a href="#">Arefly</a> , <a href="#">Bhavin Ramani</a> , <a href="#">FelixSFD</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Irfan</a> , <a href="#">juliand665</a> , <a href="#">Kirit Modi</a> , <a href="#">Muhammad Zohaib Ehsan</a> , <a href="#">Narendra Pandey</a> , <a href="#">Nikita Kurtin</a> , <a href="#">NSNoob</a> , <a href="#">pableiros</a> , <a href="#">Senseful</a> , <a href="#">Seyyed Parsa Neshaei</a> , <a href="#">shim</a> , <a href="#">Stephen Leppik</a> , <a href="#">Sunil Sharma</a> , <a href="#">Suragch</a> , <a href="#">user3480295</a>
71	UIAppearance	<a href="#">azimov</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Stephen Leppik</a> , <a href="#">Undo</a>
72	UIBezierPath	<a href="#">Bean</a> , <a href="#">Igor Bidiniuc</a> , <a href="#">Suragch</a> , <a href="#">Teja Nandamuri</a>
73	UICollectionView	<a href="#">Adam Eberbach</a> , <a href="#">AJ9</a> , <a href="#">Alex Koshy</a> , <a href="#">Anand Nimje</a> , <a href="#">Anh Pham</a> , <a href="#">Bhavin Ramani</a> , <a href="#">Bhumit Mehta</a> , <a href="#">Brian</a> , <a href="#">Dalija Prasnikar</a> , <a href="#">ddb</a> , <a href="#">Dima Deplov</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Kevin DiTraglia</a> , <a href="#">Koushik</a> , <a href="#">Mark</a> , <a href="#">Rodrigo de Santiago</a> , <a href="#">Stephen Leppik</a> , <a href="#">Suragch</a> , <a href="#">Undo</a>
74	UIControl -	<a href="#">Brandon</a>
75	UIFeedbackGenerator	<a href="#">beyowulf</a>
76	UIFont	<a href="#">Mr. Xcoder</a>
77	UIGestureRecognizer	<a href="#">Adam Preble</a> , <a href="#">dannyzlo</a> , <a href="#">Dunja Lalic</a> , <a href="#">Harshal Bhavsar</a> ,

		<a href="#">John Leonardo</a> , <a href="#">Josh Caswell</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">Ruby</a> , <a href="#">Stephen Leppik</a> , <a href="#">Sujania</a> , <a href="#">Suragch</a> , <a href="#">Undo</a>
78	UIKit	<a href="#">beyowulf</a> , <a href="#">Mark Stewart</a> , <a href="#">Md. Ibrahim Hassan</a>
79	UILabelattributedString	<a href="#">vp2698</a>
80	UILabel	<a href="#">Md. Ibrahim Hassan</a>
81	UILocalNotification	<a href="#">Bhumit Mehta</a> , <a href="#">Brian</a> , <a href="#">Byte1518</a> , <a href="#">D4ttatraya</a> , <a href="#">David</a> , <a href="#">ElonChan</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">hgwhittle</a> , <a href="#">kamwysoc</a> , <a href="#">KrishnaCA</a> , <a href="#">rajesh sukumaran</a> , <a href="#">Rex</a> , <a href="#">Samuel Spencer</a> , <a href="#">themathsrobot</a> , <a href="#">tksubota</a> , <a href="#">william205</a> , <a href="#">Wolverine</a> , <a href="#">Xenon</a>
82	UINavigationController	<a href="#">dasdom</a> , <a href="#">Oleh Zayats</a> , <a href="#">sage444</a> , <a href="#">Suragch</a> , <a href="#">william205</a> , <a href="#">WMios</a>
83	UIPageViewController	<a href="#">azimov</a> , <a href="#">Bright Future</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Mayuri R Talaviya</a> , <a href="#">Stephen Leppik</a> , <a href="#">stonbrooklyn</a> , <a href="#">Victor M</a>
84	UIPheonix - UI	<a href="#">StackUnderflow</a>
85	UIPickerView	<a href="#">FelixSFD</a> , <a href="#">Hasintha Janka</a> , <a href="#">MCMatan</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">Moritz</a> , <a href="#">NinjaDeveloper</a>
86	UIRefreshControl UITableView	<a href="#">Md. Ibrahim Hassan</a> , <a href="#">Mohammad Rana</a>
87	UIScrollView AutoLayout	<a href="#">Aaron</a> , <a href="#">Brandon</a> , <a href="#">Shrikant K</a>
88	UIScrollView	<a href="#">Bhavin Ramani</a> , <a href="#">LinusGeffarth</a> , <a href="#">maxkonovalov</a> , <a href="#">Rex</a> , <a href="#">sanman</a> , <a href="#">Sujania</a> , <a href="#">Sunil Sharma</a> , <a href="#">Suragch</a> , <a href="#">tharkay</a> , <a href="#">torinpitchers</a>
89	UISearchController	<a href="#">Harshal Bhavsar</a> , <a href="#">Mehul Chuahan</a> , <a href="#">mtso</a> , <a href="#">Stephen Leppik</a> , <a href="#">Tarvo Mäesepp</a>
90	UISegmentedControl	<a href="#">Kamil Harasimowicz</a>
91	UISlider	<a href="#">Andreas</a> , <a href="#">Md. Ibrahim Hassan</a>
92	UISplitViewController	<a href="#">Cerbrus</a> , <a href="#">Koushik</a>
93	UIStackView	<a href="#">Anuj Joshi</a> , <a href="#">danshevluk</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Kof</a> , <a href="#">Lior Pollak</a> , <a href="#">Sally</a> , <a href="#">sasquatch</a> , <a href="#">Stephen Leppik</a> , <a href="#">william205</a>
94	UIStoryboard	<a href="#">Adriana Carelli</a> , <a href="#">Mr. Xcoder</a> , <a href="#">Vignan</a>
95	UISwitch	<a href="#">Bhavin Ramani</a> , <a href="#">FelixSFD</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">Mr. Xcoder</a> , <a href="#">RamenChef</a> , <a href="#">Sujay</a>

96	UITableView	<a href="#">AJ9</a> , <a href="#">Alex Koshy</a> , <a href="#">Andres Kievsky</a> , <a href="#">Anh Pham</a> , <a href="#">animuson</a> , <a href="#">Bean</a> , <a href="#">Brendon Roberto</a> , <a href="#">Brian</a> , <a href="#">dasdom</a> , <a href="#">DeyaEldeen</a> , <a href="#">Dima Deplov</a> , <a href="#">Dunja Lalic</a> , <a href="#">Erik Godard</a> , <a href="#">Glorfindel</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Jojodmo</a> , <a href="#">Kof</a> , <a href="#">Luca D'Alberti</a> , <a href="#">Luis</a> , <a href="#">Meng Zhang</a> , <a href="#">Nathan</a> , <a href="#">Nirav Bhatt</a> , <a href="#">Nirav D</a> , <a href="#">RamenChef</a> , <a href="#">Rex</a> , <a href="#">RodolfoAntonici</a> , <a href="#">Ruby</a> , <a href="#">Samuel Spencer</a> , <a href="#">Seslyn</a> , <a href="#">simple_code</a> , <a href="#">Srinija</a> , <a href="#">Steve Moser</a> , <a href="#">Sujania</a> , <a href="#">Sujay</a> , <a href="#">Suragch</a> , <a href="#">Tamarous</a> , <a href="#">user3480295</a>
97	UITextField	<a href="#">Andreas</a> , <a href="#">animuson</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">midori</a> , <a href="#">Ruby</a>
98	UIView	<a href="#">Bright Future</a> , <a href="#">Darshit Shah</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">SpaceDog</a>
99	UIWebView	<a href="#">Allan Burleson</a> , <a href="#">dchar4life80X</a> , <a href="#">iOS BadBoy</a> , <a href="#">J F</a> , <a href="#">Julian135</a> , <a href="#">KANGKANG</a> , <a href="#">Kevin DiTraglia</a> , <a href="#">maxkonovalov</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">Ortwin Gentz</a> , <a href="#">Ramkumar chintala</a> , <a href="#">Sunil Sharma</a>
100	UI	<a href="#">P. Pawluś</a>
101	UUID	<a href="#">Anand Nimje</a> , <a href="#">FelixSFD</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">James P</a> , <a href="#">Mehul Chuahan</a> , <a href="#">Rahul Vyas</a> , <a href="#">Seyyed Parsa Neshaei</a> , <a href="#">shim</a> , <a href="#">Stephen Leppik</a> , <a href="#">sushant jagtap</a>
102	WCSessionDelegate	<a href="#">pkc456</a>
103	WKWebView	<a href="#">Brandon</a> , <a href="#">byJeevan</a> , <a href="#">Mahmoud Adam</a> , <a href="#">Yevhen Dubinin</a>
104	Xcode	<a href="#">Kyle Decot</a> , <a href="#">Shardul</a>
105	XCTest -	<a href="#">D4ttatraya</a> , <a href="#">dasdom</a> , <a href="#">Jan ATAC</a> , <a href="#">Josh Brown</a> , <a href="#">msohng</a> , <a href="#">Raphael Silva</a> , <a href="#">Seyyed Parsa Neshaei</a> , <a href="#">Tarun Seera</a>
106	NSBundle	<a href="#">wdywayne</a>
107		<a href="#">Doc</a> , <a href="#">Fonix</a> , <a href="#">Juan Campa</a> , <a href="#">Kevin DiTraglia</a> , <a href="#">Tien</a>
108		<a href="#">HaemEternal</a>
109	CoreBluetooth	<a href="#">Beto Caldas</a>
110	CoreImage / OpenCV	<a href="#">Md. Ibrahim Hassan</a>
111	GPXiOS	<a href="#">Uma</a>
112	UICollectionViewUIKit Dynamics	<a href="#">beyowulf</a>
113	Aseets	<a href="#">D4ttatraya</a>

114		Tien
115	UIView	Md. Ibrahim Hassan
116	XIBUIViews	<a href="#">backslash-f</a> , <a href="#">Code.Warrior</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">idocode</a> , <a href="#">Nirav Bhatt</a> , <a href="#">Sharpkits Innovations</a> , <a href="#">Stephen Leppik</a>
117		<a href="#">Faran Ghani</a> , <a href="#">simple_code</a>
118		Jano
119	.ipaApplicationloader appstore	<a href="#">Anuj Joshi</a>
120	ID	<a href="#">yogesh wadhwa</a>
121		<a href="#">J.Paravicini</a>
122	UIStackView	<a href="#">Harshal Bhavsar</a> , <a href="#">Rahul</a> , <a href="#">Stephen Leppik</a>
123		<a href="#">Alvin Abia</a> , <a href="#">H. M. Madrone</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">James P</a> , <a href="#">Stephen Leppik</a>
124		<a href="#">Ashish Kakkad</a>
125	Coreplot	<a href="#">MarmiK</a> , <a href="#">Md. Ibrahim Hassan</a>
126	iOS	<a href="#">Saeed-rz</a>
127		<a href="#">Cyril Ivar Garcia</a> , <a href="#">Martin</a> , <a href="#">rigdonmr</a> , <a href="#">WMios</a>
128		<a href="#">Arulkumar</a> , <a href="#">Ashish Kakkad</a> , <a href="#">BorisE</a> , <a href="#">Bright Future</a> , <a href="#">Dima Deplov</a> , <a href="#">dispute</a> , <a href="#">FelixSFD</a> , <a href="#">Honey</a> , <a href="#">ignotusverum</a> , <a href="#">Irfan</a> , <a href="#">Jake Runzer</a> , <a href="#">juanjo</a> , <a href="#">Kasun Randika</a> , <a href="#">Kendall Lister</a> , <a href="#">Kyle KIM</a> , <a href="#">Luca D'Alberti</a> , <a href="#">muazhud</a> , <a href="#">OhadM</a> , <a href="#">RamenChef</a> , <a href="#">rustproofFish</a> , <a href="#">salabaha</a> , <a href="#">StackUnderflow</a> , <a href="#">Steve Moser</a> , <a href="#">Suragch</a> , <a href="#">Tamarous</a> , <a href="#">timbroder</a> , <a href="#">Undo</a> , <a href="#">WMios</a> , <a href="#">Yagnesh Dobariya</a>
129	MessageBox-Concept	<a href="#">StackUnderflow</a>
130	I / O.	<a href="#">Idan</a>
131		<a href="#">4444</a> , <a href="#">animuson</a> , <a href="#">Joshua</a> , <a href="#">Mehul Chuahan</a> , <a href="#">Ruby</a> , <a href="#">Tamarous</a> , <a href="#">user459460</a>
132		<a href="#">Daniel Ormeño</a>
133		<a href="#">Rahul</a>
134		<a href="#">Tim</a>

135		<a href="#">D4ttatraya</a>
136	<a href="#">HTMLNSAttributed</a>	<a href="#">Md. Ibrahim Hassan</a>
137	<a href="#">NSAttributedString UIImage</a>	<a href="#">Md. Ibrahim Hassan</a>
138	<a href="#">UIImage</a>	<a href="#">Md. Ibrahim Hassan</a>
139		<a href="#">Md. Ibrahim Hassan</a> , <a href="#">Mehul Chuahan</a>
140	<a href="#">StackViewUIScrollView</a>	<a href="#">mourodrigo</a>
141		<a href="#">Tiko</a>
142		<a href="#">J F</a> , <a href="#">KrauseFx</a> , <a href="#">SM18</a> , <a href="#">tharkay</a>
143		<a href="#">midori</a>
144		<a href="#">Nermin Sehic</a>
145	<a href="#">ATS</a>	<a href="#">breakingobstacles</a> , <a href="#">D4ttatraya</a> , <a href="#">esthepiking</a> , <a href="#">FelixSFD</a> , <a href="#">Mehul Chuahan</a> , <a href="#">nathan</a>
146		<a href="#">Amanpreet</a> , <a href="#">Anh Pham</a> , <a href="#">Ashish Kakkad</a> , <a href="#">Bhadresh Kathiriy</a> , <a href="#">BloodWoork</a> , <a href="#">Bonnie</a> , <a href="#">Honey</a> , <a href="#">Hossam Ghareeb</a> , <a href="#">iOS BadBoy</a> , <a href="#">J F</a> , <a href="#">Patrick Beard</a> , <a href="#">Pavel Gurov</a> , <a href="#">sanman</a> , <a href="#">Seyyed Parsa Neshaei</a> , <a href="#">tilo</a>
147	<a href="#">- iOS 10</a>	<a href="#">Oleh Zayats</a>
148		<a href="#">Harshal Bhavsar</a> , <a href="#">Kirit Vaghela</a> , <a href="#">Stephen Leppik</a> , <a href="#">Tommie C.</a>
149		<a href="#">Nikhlesh Bagdiya</a>
150		<a href="#">Alex Rouse</a> , <a href="#">danshevluk</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Mr. Xcoder</a> , <a href="#">shim</a> , <a href="#">Stephen Leppik</a> , <a href="#">Steve Moser</a> , <a href="#">william205</a> , <a href="#">WMios</a>
151		<a href="#">4444</a> , <a href="#">animuson</a> , <a href="#">Joshua</a> , <a href="#">Ruby</a> , <a href="#">WMios</a>
152		<a href="#">Harshal Bhavsar</a> , <a href="#">Mayuri R Talaviya</a> , <a href="#">Mehul Chuahan</a> , <a href="#">mtso</a> , <a href="#">quant24</a> , <a href="#">Stephen Leppik</a> , <a href="#">sushant jagtap</a> , <a href="#">william205</a>
153		<a href="#">Md. Ibrahim Hassan</a> , <a href="#">RamenChef</a>
154		<a href="#">Dunja Lalic</a> , <a href="#">Josh Caswell</a> , <a href="#">Seyyed Parsa Neshaei</a> , <a href="#">Sunil Sharma</a> , <a href="#">Unheilig</a>
155		<a href="#">Ankit chauhan</a> , <a href="#">Md. Ibrahim Hassan</a>

156		<a href="#">Seyyed Parsa Neshaei</a>
157		<a href="#">Durai Amuthan.H</a>
158	iOS	<a href="#">Bhavin Ramani</a> , <a href="#">byJeevan</a> , <a href="#">James P</a> , <a href="#">Joshua</a> , <a href="#">njuri</a> , <a href="#">Samuel Teferra</a> , <a href="#">Sandy</a>
159		<a href="#">ajmccall</a> , <a href="#">breakingobstacles</a> , <a href="#">Mick MacCallum</a> , <a href="#">pableiros</a> , <a href="#">sushant jagtap</a>
160	SWIFT BRIDGING HEADER	<a href="#">yogesh wadhwa</a>
161		<a href="#">Harshal Bhavsar</a> , <a href="#">Justin</a> , <a href="#">Ruby</a> , <a href="#">Stephen Leppik</a> , <a href="#">Zev Eisenberg</a>
162		<a href="#">Vinod Kumar</a>
163	/	<a href="#">Abhijit</a>
164	MKMapView	<a href="#">Arnon Rodrigues</a> , <a href="#">Brian</a> , <a href="#">FelixSFD</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Kosuke Ogawa</a> , <a href="#">Mahesh</a> , <a href="#">Mehul Thakkar</a> , <a href="#">Ortwin Gentz</a> , <a href="#">Reinier Melian</a> , <a href="#">Stephen Leppik</a>
165	NSData	<a href="#">Felipe Cypriano</a> , <a href="#">maxkonovalov</a> , <a href="#">Seyyed Parsa Neshaei</a>
166	NSDate	<a href="#">Bonnie</a> , <a href="#">Charles</a> , <a href="#">dasdom</a> , <a href="#">Dunja Lalic</a> , <a href="#">ERbittuu</a> , <a href="#">FelixSFD</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Jon Snow</a> , <a href="#">Josh Caswell</a> , <a href="#">lostAtSeaJoshua</a> , <a href="#">maxkonovalov</a> , <a href="#">Mehul Thakkar</a> , <a href="#">NSNoob</a> , <a href="#">Nykholas</a> , <a href="#">OhadM</a> , <a href="#">Sally</a> , <a href="#">Samuel Teferra</a> , <a href="#">Sandy</a> , <a href="#">Seyyed Parsa Neshaei</a> , <a href="#">Stephen Leppik</a> , <a href="#">tharkay</a> , <a href="#">tobeiosdeveloper</a>
167	NSTimer	<a href="#">AJ9</a> , <a href="#">James P</a> , <a href="#">Maddyツヅ</a> , <a href="#">Samuel Teferra</a> , <a href="#">tfrank377</a> , <a href="#">That lazy iOS Guy</a> , <a href="#">Undo</a> , <a href="#">william205</a>
168	UIBarButtonItem	<a href="#">Ahmed Khalaf</a> , <a href="#">Dunja Lalic</a> , <a href="#">hgwhittle</a> , <a href="#">Suragch</a> , <a href="#">william205</a>
169	UIButton	<a href="#">Aleksei Minaev</a> , <a href="#">Arefly</a> , <a href="#">dasdom</a> , <a href="#">ddb</a> , <a href="#">Fabio Berger</a> , <a href="#">FelixSFD</a> , <a href="#">fredpi</a> , <a href="#">James</a> , <a href="#">James P</a> , <a href="#">Jojodmo</a> , <a href="#">Joshua</a> , <a href="#">mattblessed</a> , <a href="#">Mr. Xcoder</a> , <a href="#">mtso</a> , <a href="#">Nate Lee</a> , <a href="#">NSNoob</a> , <a href="#">P. Pawluś</a> , <a href="#">Quantm</a> , <a href="#">RamenChef</a> , <a href="#">Roland Keesom</a> , <a href="#">Sachin S P</a> , <a href="#">tharkay</a> , <a href="#">Viktor Simkó</a> , <a href="#">william205</a> , <a href="#">WMios</a>
170	UIColor	<a href="#">Amanpreet</a> , <a href="#">Anh Pham</a> , <a href="#">Avineet Gupta</a> , <a href="#">Brett Ponder</a> , <a href="#">Cin316</a> , <a href="#">Community</a> , <a href="#">dasdom</a> , <a href="#">DeyaEldeen</a> , <a href="#">Douglas Hill</a> , <a href="#">Elias Datler</a> , <a href="#">Fabio Berger</a> , <a href="#">FelixSFD</a> , <a href="#">Gary Riches</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Honey</a> , <a href="#">ing0</a> , <a href="#">iphonic</a> , <a href="#">Irfan</a> , <a href="#">JAL</a> , <a href="#">Jaleel Nazir</a> , <a href="#">Jojodmo</a> , <a href="#">Luca D'Alberti</a> , <a href="#">maxkonovalov</a> , <a href="#">mtso</a> ,

		<a href="#">nielsbot</a> , <a href="#">NSNoob</a> , <a href="#">pableiros</a> , <a href="#">Reinier Melian</a> , <a href="#">Rex</a> , <a href="#">Sally</a> , <a href="#">Samer Murad</a> , <a href="#">Sandy</a> , <a href="#">shim</a> , <a href="#">The_Curry_Man</a> , <a href="#">Tommie C.</a> , <a href="#">Viktor Simkó</a> , <a href="#">WMios</a> , <a href="#">Yagnesh Dobariya</a>
171	UIDatePicker	<a href="#">Pavel Gatilov</a>
172	UIDevice	<a href="#">Bhavin Ramani</a> , <a href="#">FelixSFD</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">Mehul Chuahan</a> , <a href="#">Nef10</a> , <a href="#">pableiros</a> , <a href="#">Ramkumar chintala</a>
173	UIImage	<a href="#">Adrian Schönig</a> , <a href="#">Alexander Tkachenko</a> , <a href="#">Bean</a> , <a href="#">Bhavin Ramani</a> , <a href="#">Dipen Panchasara</a> , <a href="#">Dunja Lalic</a> , <a href="#">Emptyless</a> , <a href="#">FelixSFD</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Heberti Almeida</a> , <a href="#">Jimmy James</a> , <a href="#">Mahmoud Adam</a> , <a href="#">maxkonovalov</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">Muhammad Zeeshan</a> , <a href="#">RamenChef</a> , <a href="#">Reinier Melian</a> , <a href="#">Rex</a> , <a href="#">rob180</a> , <a href="#">Ronak Chaniyara</a> , <a href="#">sage444</a> , <a href="#">Sandy</a> , <a href="#">Seyyed Parsa Neshaei</a> , <a href="#">Sujania</a> , <a href="#">Sunil Sharma</a> , <a href="#">The_Curry_Man</a> , <a href="#">user3480295</a> , <a href="#">Vineet Choudhary</a>
174	UIImagePickerController	<a href="#">Brian</a> , <a href="#">stonybrooklyn</a> , <a href="#">william205</a>
175	UIImageView	<a href="#">Adam Eberbach</a> , <a href="#">Anh Pham</a> , <a href="#">Bean</a> , <a href="#">Caleb Kleveter</a> , <a href="#">DeyaEldeen</a> , <a href="#">Dunja Lalic</a> , <a href="#">FelixSFD</a> , <a href="#">il Malvagio Dottor Prosciutto</a> , <a href="#">Irfan</a> , <a href="#">Joshua</a> , <a href="#">mattblessed</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">njuri</a> , <a href="#">Quantm</a> , <a href="#">Reinier Melian</a> , <a href="#">Rex</a> , <a href="#">Rob</a> , <a href="#">Samuel Spencer</a> , <a href="#">Sunil Sharma</a> , <a href="#">Suragch</a> , <a href="#">william205</a>
176	UILabel	<a href="#">4oby</a> , <a href="#">Akilan Arasu</a> , <a href="#">Alex Koshy</a> , <a href="#">alvarolopez</a> , <a href="#">Andres Canella</a> , <a href="#">Andrii Chernenko</a> , <a href="#">Anh Pham</a> , <a href="#">Ashwin Ramaswami</a> , <a href="#">AstroCB</a> , <a href="#">Barlow Tucker</a> , <a href="#">bentford</a> , <a href="#">Bhumit Mehta</a> , <a href="#">Brian</a> , <a href="#">byJeevan</a> , <a href="#">Caleb Kleveter</a> , <a href="#">Chathuranga Silva</a> , <a href="#">Chris Brandsma</a> , <a href="#">Cin316</a> , <a href="#">Code.Warrior</a> , <a href="#">Community</a> , <a href="#">Daniel Bocksteger</a> , <a href="#">Daniel Stradowski</a> , <a href="#">danshevluk</a> , <a href="#">dasdom</a> , <a href="#">ddb</a> , <a href="#">DeyaEldeen</a> , <a href="#">Dunja Lalic</a> , <a href="#">Eric</a> , <a href="#">Erwin</a> , <a href="#">esthepiking</a> , <a href="#">Fabio Berger</a> , <a href="#">Fahim Parkar</a> , <a href="#">Felix</a> , <a href="#">FelixSFD</a> , <a href="#">Franck Dernoncourt</a> , <a href="#">gadu</a> , <a href="#">ggrana</a> , <a href="#">GingerHead</a> , <a href="#">gvuksic</a> , <a href="#">HaemEternal</a> , <a href="#">hankide</a> , <a href="#">Hans Sjunnesson</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Hossam Ghareeb</a> , <a href="#">idobn</a> , <a href="#">Imanou Petit</a> , <a href="#">iOS BadBoy</a> , <a href="#">iphonic</a> , <a href="#">Irfan</a> , <a href="#">J F</a> , <a href="#">Jacky</a> , <a href="#">Jacobanks</a> , <a href="#">johnpenning</a> , <a href="#">Jojodmo</a> , <a href="#">Josh Brown</a> , <a href="#">Joshua</a> , <a href="#">Joshua J. McKinnon</a> , <a href="#">jtbandes</a> , <a href="#">juanjo</a> , <a href="#">kabiroberai</a> , <a href="#">Kai Engelhardt</a> , <a href="#">KANGKANG</a> , <a href="#">Khanh Nguyen</a> , <a href="#">Kireyin</a> , <a href="#">leni</a> , <a href="#">Luca D'Alberti</a> , <a href="#">lufritz</a> , <a href="#">Lukas</a> , <a href="#">Luke Patterson</a> , <a href="#">Lumialxk</a> , <a href="#">Mad Burea</a> , <a href="#">Mahmoud Adam</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">Moshe</a> , <a href="#">Nadzeya</a> , <a href="#">Narendra Pandey</a> , <a href="#">Nathan Levitt</a> , <a href="#">Nirav D</a> , <a href="#">njuri</a> , <a href="#">noelicus</a> , <a href="#">NSNoob</a> , <a href="#">Ollie</a> , <a href="#">Quantm</a> , <a href="#">Radagast the Brown</a> , <a href="#">Rahul Vyas</a> , <a href="#">RamenChef</a> , <a href="#">ramsserio</a> , <a href="#">rfarry</a> , <a href="#">sage444</a> , <a href="#">Scotow</a> , <a href="#">Seyyed Parsa Neshaei</a> , <a href="#">Shahabuddin Vansiwala</a> , <a href="#">solidcell</a> , <a href="#">Sravan</a> , <a href="#">stackptr</a> , <a href="#">Sunil Sharma</a> ,

		<a href="#">Suragch</a> , <a href="#">sushant jagtap</a> , <a href="#">TDM</a> , <a href="#">tharkay</a> , <a href="#">The_Curry_Man</a> , <a href="#">Tibor Molnár</a> , <a href="#">Tyler</a> , <a href="#">Undo</a> , <a href="#">user3480295</a> , <a href="#">vasili111</a> , <a href="#">Vignan</a> , <a href="#">Viktor Simkó</a> , <a href="#">william205</a> , <a href="#">WMios</a> , <a href="#">Yagnesh Dobariya</a>
177	UITabBarController	<a href="#">Alexi</a> , <a href="#">Anand Nimje</a> , <a href="#">Cristina</a> , <a href="#">Mehul Chuahan</a> , <a href="#">Quantm</a> , <a href="#">Srinija</a>
178	UITableViewCell	<a href="#">Rahul</a>
179	UITableViewController	<a href="#">Aju</a>
180	UITextField	<a href="#">Alex Koshy</a> , <a href="#">Ali Elsokary</a> , <a href="#">Ashvinkumar</a> , <a href="#">Duly Kinsky</a> , <a href="#">Fabio Berger</a> , <a href="#">FelixSFD</a> , <a href="#">J F</a> , <a href="#">Joshua</a> , <a href="#">Kof</a> , <a href="#">Luiz Henrique Guimaraes</a> , <a href="#">Maddyツヅ</a> , <a href="#">P. Pawluś</a> , <a href="#">RamenChef</a> , <a href="#">Reinier Melian</a> , <a href="#">Ruby</a> , <a href="#">samwize</a> , <a href="#">sasquatch</a> , <a href="#">shim</a> , <a href="#">SourabhV</a> , <a href="#">Suragch</a> , <a href="#">sushant jagtap</a> , <a href="#">tharkay</a> , <a href="#">william205</a> , <a href="#">WMios</a>
181	UITextView	<a href="#">Anh Pham</a> , <a href="#">animuson</a> , <a href="#">Bole Tzar</a> , <a href="#">Bright Future</a> , <a href="#">Cris</a> , <a href="#">Dunja Lalic</a> , <a href="#">Eonil</a> , <a href="#">gadu</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Hejazi</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">njuri</a> , <a href="#">Roland Keesom</a> , <a href="#">Ruby</a> , <a href="#">Suragch</a> , <a href="#">sushant jagtap</a> , <a href="#">william205</a> , <a href="#">WMios</a>
182	UIView	<a href="#">Adam Preble</a> , <a href="#">alaphao</a> , <a href="#">Anh Pham</a> , <a href="#">Caleb Kleveter</a> , <a href="#">Community</a> , <a href="#">Cory Wilhite</a> , <a href="#">D4ttatraya</a> , <a href="#">ddb</a> , <a href="#">DeyaEldeen</a> , <a href="#">Douglas Starnes</a> , <a href="#">hgwhittle</a> , <a href="#">iphonic</a> , <a href="#">Irfan</a> , <a href="#">James</a> , <a href="#">Jojodmo</a> , <a href="#">Jota</a> , <a href="#">Kotha Sai Ram</a> , <a href="#">Luca D'Alberti</a> , <a href="#">maxkonovalov</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">muazhud</a> , <a href="#">Narendra Pandey</a> , <a href="#">Nikhil Manapure</a> , <a href="#">NSNoob</a> , <a href="#">pableiros</a> , <a href="#">pckill</a> , <a href="#">Peter DeWeese</a> , <a href="#">Rahul Vyas</a> , <a href="#">sasquatch</a> , <a href="#">shallowThought</a> , <a href="#">Sunil Sharma</a> , <a href="#">That lazy iOS Guy</a> , <a href="#">The_Curry_Man</a> , <a href="#">Viktor Simkó</a> , <a href="#">william205</a>
183	UIViewController	<a href="#">dasdom</a> , <a href="#">Dunja Lalic</a> , <a href="#">shim</a> , <a href="#">Suragch</a> , <a href="#">tassinari</a> , <a href="#">william205</a>
184		<a href="#">FelixSFD</a> , <a href="#">Md. Ibrahim Hassan</a>
185		<a href="#">Alexander Tkachenko</a> , <a href="#">Greg</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Mr. Xcoder</a> , <a href="#">Richard Ash</a> , <a href="#">Shog9</a> , <a href="#">sxl</a> , <a href="#">Stephen Leppik</a> , <a href="#">Steve Moser</a> , <a href="#">Suragch</a> , <a href="#">V1P3R</a> , <a href="#">william205</a> , <a href="#">WMios</a>
186		<a href="#">Md. Ibrahim Hassan</a>
187		<a href="#">Md. Ibrahim Hassan</a> , <a href="#">Seyyed Parsa Neshaei</a>
188		<a href="#">Seyyed Parsa Neshaei</a>
189		<a href="#">Ashish Kakkad</a>
190		<a href="#">alaphao</a> , <a href="#">amar</a> , <a href="#">Anuj Joshi</a> , <a href="#">Bean</a> , <a href="#">Bhumit Mehta</a> ,



		<a href="#">BlackDeveraux</a> , <a href="#">dasdom</a> , <a href="#">Dennis</a> , <a href="#">Dima Deplov</a> , <a href="#">Dinesh Raja</a> , <a href="#">Đông An</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Hasintha Janka</a> , <a href="#">Irfan</a> , <a href="#">Jano</a> , <a href="#">juanjo</a> , <a href="#">keithbhunter</a> , <a href="#">Mahesh</a> , <a href="#">Mert Buran</a> , <a href="#">Mr. Xcoder</a> , <a href="#">NSNoob</a> , <a href="#">ozgur</a> , <a href="#">Pärserk</a> , <a href="#">Rajesh</a> , <a href="#">Sally</a> , <a href="#">Sandy</a> , <a href="#">Stephen Leppik</a> , <a href="#">Suragch</a> , <a href="#">Undo</a> , <a href="#">user3480295</a> , <a href="#">Vignan</a>
191	UITableViewCell	<a href="#">Kamil Harasimowicz</a>
192	UITextField	<a href="#">D4ttatraya</a>
193		<a href="#">Alexi</a> , <a href="#">Dima Deplov</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Maddy`ツ`ツ</a> , <a href="#">njuri</a> , <a href="#">Stephen Leppik</a> , <a href="#">Tommie C.</a>
194		<a href="#">Md. Ibrahim Hassan</a>
195	URL	<a href="#">azimov</a> , <a href="#">Brian</a> , <a href="#">Dunja Lalic</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">James P</a> , <a href="#">Stephen Leppik</a>
196		<a href="#">Adriana Carelli</a> , <a href="#">Andreas</a> , <a href="#">Bhadresh Kathiriya</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Md. Ibrahim Hassan</a> , <a href="#">user459460</a>
197		<a href="#">Seyyed Parsa Neshaei</a>
198	UIImage	<a href="#">Rahul</a>
199		<a href="#">NobodyNada</a>
200		<a href="#">Koushik</a>
201	iOS	<a href="#">Md. Ibrahim Hassan</a>
202		<a href="#">Harshal Bhavsar</a> , <a href="#">Irfan</a> , <a href="#">satheeshwaran</a> , <a href="#">Stephen Leppik</a> , <a href="#">Vineet Choudhary</a>
203	iOS	<a href="#">Phani Sai</a>
204		<a href="#">abjurato</a> , <a href="#">avojak</a> , <a href="#">Matthew Seaman</a> , <a href="#">Mehul Chuahan</a>
205	-	<a href="#">D4ttatraya</a> , <a href="#">Harshal Bhavsar</a> , <a href="#">Mehul Chuahan</a> , <a href="#">Mihriban Minaz</a> , <a href="#">Mithrandir</a> , <a href="#">Muhammad Zohaib Ehsan</a> , <a href="#">Pärserk</a> , <a href="#">sanman</a>
206	MKBlockQueue	<a href="#">StackUnderflow</a>
207		<a href="#">subv3rsion</a>