



Бесплатная электронная книга

УЧУСЬ

OSX

Free unaffiliated eBook created from
Stack Overflow contributors.

#OSX

.....	1
1: osx	2
.....	2
Examples.....	2
.....	2
2: NSFont	3
.....	3
Examples.....	3
NSFont.....	3
Objective-C.....	3
3: NSMenuItem	4
.....	4
Examples.....	4
.....	4
.....	4
.....	4
.....	5
.....	5
4: NSRunLoop	6
Examples.....	6
.....	6
5: NSStoryboard	7
Examples.....	7
.....	7
6: NSTextView	8
.....	8
Examples.....	8
NSTextView.....	8
.....	8
.....	11

Objective-C.....	12
7:	14
.....	14
Examples.....	14
.....	14
.....	14
.....	14
.....	14
8:	16
Examples.....	16
.....	16
/ Info.plist.....	16
9:	18
.....	18
Examples.....	18
.....	18
.....	19

Около

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

It is an unofficial and free osx 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 osx.

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

глава 1: Начало работы с osx

замечания

Этот тег предназначен для документации по программным API-интерфейсам, разработанным Apple, таким как AppKit.

Examples

Обзор рамок

Приложения, написанные для macOS, обычно пишутся с помощью Apple Framework. Рамки, которые будут использоваться почти каждым приложением:

- AppKit - для создания и управления элементами пользовательского интерфейса
- Foundation - для обычных объектов и операций, не относящихся к пользовательскому интерфейсу

Существуют и другие общие рамки, которые используются во многих, но не во всех приложениях:

- CoreData - для хранения данных
- Диспетчер - для управления несколькими потоками
- CoreGraphics - для рисования графических задач
- CoreAnimation - для анимации элементов пользовательского интерфейса

Прочитайте Начало работы с osx онлайн: <https://riptutorial.com/ru/osx/topic/2818/начало-работы-с-osx>

глава 2: NSFont

Вступление

NSFont - это объект, который предоставляет приложениям Mac информацию о символах и характеристиках шрифта, которые будут использоваться, в основном, для отображения. Вы узнаете, как создавать и использовать объекты NSFont различными способами, как обычными, так и необычными.

Examples

Создание объекта NSFont

Предпочтительным и наиболее распространенным способом создания объекта NSFont является следующее:

Objective-C

```
// Name is PostScript name of font; size is in points.  
NSFont *essayFont = [NSFont fontWithName:@"Times New Roman" size:12.0];
```

Прочитайте NSFont онлайн: <https://riptutorial.com/ru/osx/topic/8881/nsfont>

глава 3: NSMenuItem

замечания

См. Документацию Apple здесь:

<https://developer.apple.com/library/mac/documentation/Cocoa/Reference/ApplicationKit/Classes/NSMenuItem>

Examples

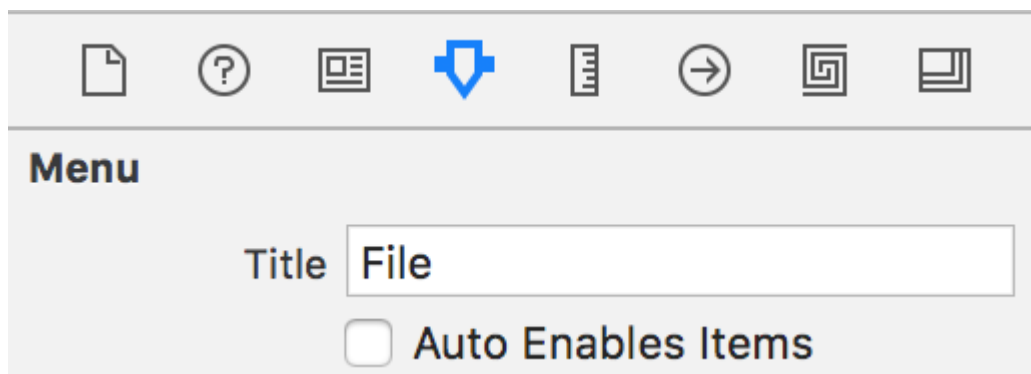
Включение пунктов меню

Ручное включение пунктов меню

Чтобы вручную управлять включенным состоянием элементов меню, содержащее его меню должно отключать автоматическое включение его элементов

Меню могут отключать автоматическое включение одним из двух способов:

1. В интерфейсе Builder



2. В коде

```
menu.autoenablesItems = false
```

Оба механизма установки `autoenablesItems` собственности на `NSMenu` .

После того, как меню перестает включать и отключать пункты меню, пункты меню могут быть программно заданы одним из двух способов

1. В интерфейсе Builder
2. В коде

```
menuItem.enabled = true
```

Автоматическое включение пунктов меню

Элементы меню могут автоматически включаться путем подключения действий пунктов меню к первому ответчику и реализации поставленного действия на объект в цепочке ответчиков, как описано в [документах](#) Apple.

Поддержка действий меню по умолчанию

Меню действуют как все стандартные элементы управления. У них есть **действие**, которое является вызываемой функцией, и **целью**, для которой объект отправляется. Если цель установлена на объект, тогда, когда пользователь выбирает пункт меню, метод действия будет отправлен целевому объекту. Если элемент меню имеет действие, но не цель, то цель будет динамически выбрана из первого объекта из следующего, который отвечает на действие:

1. Первый ответчик
2. Иерархия представлений
3. Окно
4. Контроллер окна
5. `NSApplication`
6. `NSApplication.delegate`
7. `NSApplication.nextResponder`

Реализация пункта меню Open (O) по умолчанию может быть реализована путем реализации метода `openDocument` для любого объекта в приведенном выше списке.

```
- (IBAction)openDocument:(id) sender {  
  
}
```

Добавление и удаление элементов в меню

```
// add an item to a menu  
menu.addItem(item)  
  
// remove and item from a menu  
menu.removeItem(item)
```

Прочитайте `NSMenuItem` онлайн: <https://riptutorial.com/ru/osx/topic/6038/nsmenuitem>

глава 4: NSRunLoop

Examples

простое приложение-демон

Процесс daemon выполняет программу в фоновом режиме, как правило, без взаимодействия с пользователем. В приведенном ниже примере показано, как создать демона и зарегистрировать прослушиватель, который контролирует все открытые приложения. Основная часть - вызов функции `NSRunLoop.mainRunLoop().run()`, который запускает демона.

```
class MyObserver: NSObject
{
    override init() {
        super.init()

        // app listeners
        NSWorkspace.sharedWorkspace().notificationCenter.addObserver(self, selector:
"SwitchedApp:", name: NSWorkspaceDidActivateApplicationNotification, object: nil)
    }

    func SwitchedApp(notification: NSNotification!)
    {
        print(notification)
    }
}

let observer = MyObserver()

// simply to keep the command line tool alive - as a daemon process
NSRunLoop.mainRunLoop().run()
```

Вы также можете использовать этот код в качестве основы для процесса сервера.

Прочитайте NSRunLoop онлайн: <https://riptutorial.com/ru/osx/topic/6667/nsrunloop>

глава 5: NSStoryboard

Examples

Откройте новый контроллер окна

Чтобы открыть новое окно, добавьте следующий код где-нибудь, где вы можете сохранить ссылку на новое окно (IE, делегат приложения).

стриж

```
let storyboard:NSStoryboard = NSStoryboard(name: "Main", bundle: nil)
guard let controller:NSWindowController =
    storyboard.instantiateControllerWithIdentifier("myWindowController") as? NSWindowController
else { return /*or handle error*/ }
controller.showWindow(self)
```

Objective-C

```
NSStoryboard *storyBoard = [NSStoryboard storyboardWithName:@"Main" bundle:nil]; // get a
reference to the storyboard
myController = [storyBoard instantiateControllerWithIdentifier:@"secondWindowController"]; //
instantiate your window controller
[myController showWindow:self];
```

После создания `controller` убедитесь, что вы храните ссылку на него где-то вне вызова функции. Это можно сделать, создав переменную `NSWindowController` в `NSWindowController` приложения и назначив новый контроллер переменной.

Прочитайте NSStoryboard онлайн: <https://riptutorial.com/ru/osx/topic/4287/nsstoryboard>

глава 6: NSTextView

Вступление

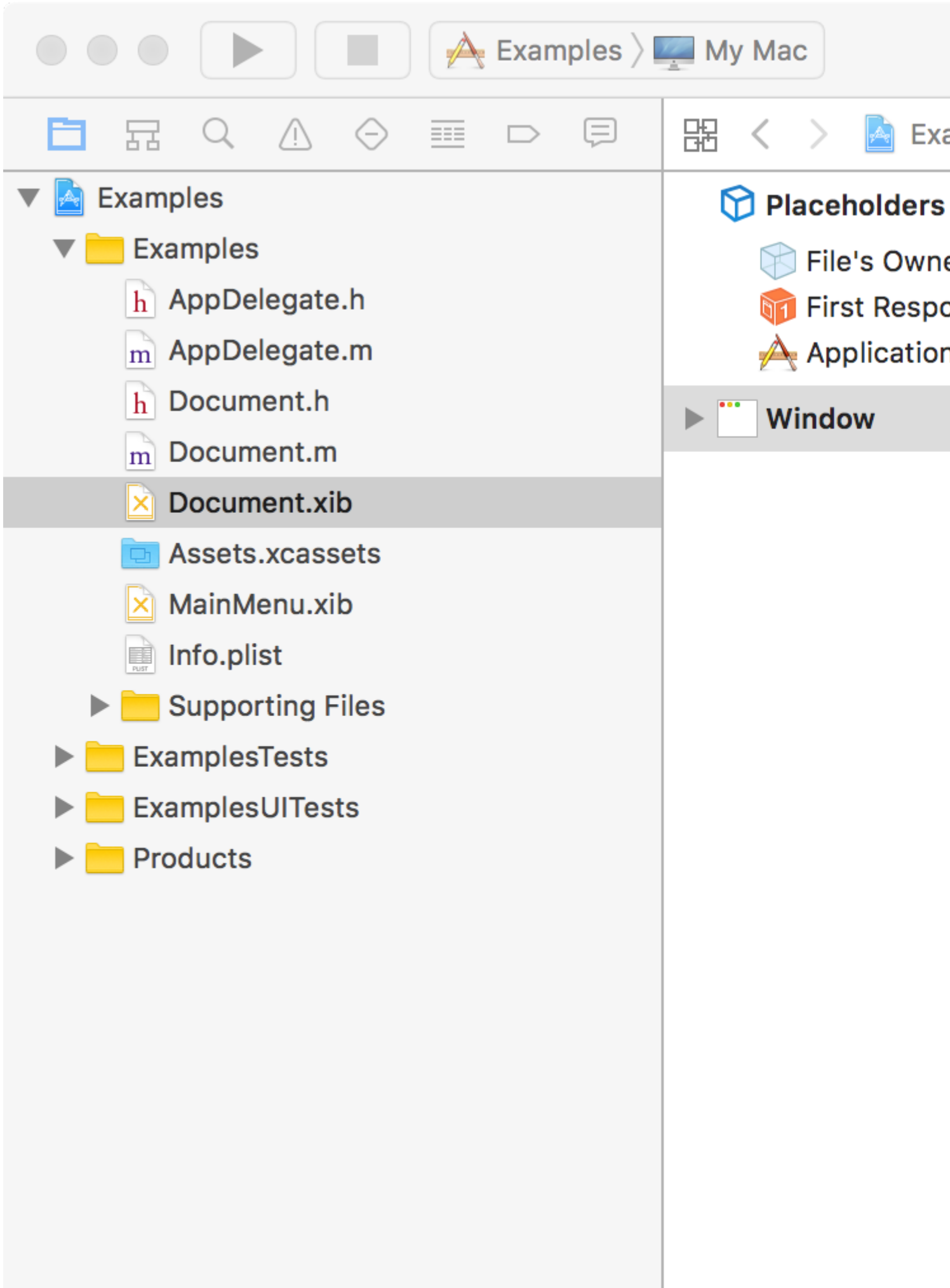
NSTextView является основным обработчиком Apple в текстовой системе AppKit. Он содержит все необходимое для создания пространства для просмотра текстовых файлов / редактора в приложениях OS X (переименованных макросов).

Examples

Создание NSTextView

Графически

В XCode простой NSTextView может быть создан путем перетаскивания одного из библиотеки объектов.



который автоматически устанавливается вертикально с текстовым видом. Убедитесь, что при настройке () -dragging вы делаете подключения к текстовому виду, а не просмотру прокрутки.

Placeholders

- File's Owner
- First Responder
- Application

Window

View

Bordered Scroll View - Text View

Clip View

Text View

- Scroller
- Scroller

M

программно позволяет повысить контроль и настройку. Это немного сложнее и требует знания текстовой системы, чтобы использовать ее в полном объеме. Более подробную информацию о текстовой системе и т. Д. Можно найти [здесь](#) . Основы, необходимые для программного просмотра текста, выглядят следующим образом:

- Для полноценного функционирования NSTextView требуются три других объекта:
 1. NSLayoutManager - выполняет глиф / макет символов.
 2. NSTextContainer - управляет графическим пространством, в котором могут находиться глифы / символы.
 3. NSTextStorage - содержит фактические строковые данные, отображаемые NSTextView.
- NSTextStorage может иметь много NSLayoutManager, но NSLayoutManager может иметь только один NSTextStorage. Это полезно, если вы хотите показывать одни и те же данные по-разному.
- NSLayoutManager может иметь много NSTextContainer. Полезно для разбитого на страницы текста.
- В NSTextView может быть только один NSTextContainer за раз.
- Некоторые вещи, встроенные в NSTextView, на момент написания статьи не ограничиваются. Например, встроенные функции Find-and-Replace не могут быть настроены, но могут быть переопределены пользовательскими функциями.

Более подробную информацию о способах использования текстовой системы можно найти [здесь](#) .

Теперь для кода. Этот код создаст простой NSTextView, даже не прокручивая. Такие вещи, как прокрутка и разбивка на страницы, будут в другом примере.

Objective-C

```
// This code resides in an NSDocument object's windowControllerDidLoadNib:(NSWindowController
*)windowController method.
// This is done simply because it is easy and automatically gets called upon.

// This method is also where the following NSRect variable gets size information. We need this
information for this example.
NSRect windowFrame = windowController.window.contentView.frame;
NSTextStorage *textStorage = [[NSTextStorage alloc] initWithString:@"Example text!"];
NSLayoutManager *manager = [[NSLayoutManager alloc] init];
NSTextContainer *container = [[NSTextContainer alloc]
initWithContainerSize:NSMakeSize(windowFrame.size.width, windowFrame.size.height)];
NSTextView *textView = [[NSTextView alloc] initWithFrame:windowFrame textContainer:container];

[textStorage addLayoutManager:manager];
[manager addTextContainer:container];
```

```
[viewController.window setContentView:textView];
```

Поздравляем! Вы внесли NSTextView программно!



Untitled ▾

Example text!

Прочитайте NSTextView онлайн: <https://riptutorial.com/ru/osx/topic/8880/nstextview>

глава 7: Вызов пользователя для файла

Вступление

NSOpenPanel предоставляет API для запроса пользователю открыть файл. Это меню является стандартным пользовательским интерфейсом, представленным пунктом меню Open (O).

Examples

Открытие файлов

Открытие любого файла

```
NSOpenPanel *openPanel = [NSOpenPanel openPanel];
[openPanel beginWithCompletionHandler:^(NSInteger result) {
    NSURL *url = openPanel.URL;
    if (result == NSFileHandlingPanelCancelButton || !url) {
        return;
    }
    // do something with a URL
}];
```

Разрешение открытия нескольких файлов

```
NSOpenPanel *openPanel = [NSOpenPanel openPanel];
openPanel.allowsMultipleSelection = YES;
[openPanel beginWithCompletionHandler:^(NSInteger result) {
    NSArray <NSURL *>*urls = openPanel.URLs;
    // do things
}];
```

Ограничение на конкретные типы файлов

```
NSOpenPanel *openPanel = [NSOpenPanel openPanel];
openPanel.allowedFileTypes = @[@"png", @"jpg"];
[openPanel beginWithCompletionHandler:^(NSInteger result) {
    NSURL *url = openPanel.URL;
    if (result == NSFileHandlingPanelCancelButton || !url) {
        return;
    }
    // do something with a picture
}];
```

Прочитайте Вызов пользователя для файла онлайн: <https://riptutorial.com/ru/osx/topic/9438/вызов-пользователя-для-файла>

глава 8: Связывание файлов

Examples

Установите приложение в качестве приложения по умолчанию для типа файла

```
- (NSString *) UTIforFileExtension:(NSString *) extension {
    NSString * UTIString = (NSString
*)UTTypeCreatePreferredIdentifierForTag(kUTTagClassFilenameExtension,
                                        (CFStringRef)extension,
                                        NULL);

    return [UTIString autorelease];
}

- (BOOL) setMyselfAsDefaultApplicationForFileExtension:(NSString *) fileExtension {
    OSStatus returnStatus = LSSetDefaultRoleHandlerForContentType (
                                        (CFStringRef) [self
UTIforFileExtension:fileExtension],
                                        kLSRolesAll,
                                        (CFStringRef) [[NSBundle
mainBundle] bundleIdentifier]
                                        );

    if (returnStatus != 0) {
        NSLog(@"Got an error when setting default application - %d", returnStatus);
        // Please see the documentation or LSInfo.h
        return NO;
    }

    return YES;
}
```

ИСТОЧНИК

Создавать связь с новыми / настраиваемыми типами файлов через Info.plist

```
<key>CFBundleDocumentTypes</key>
<array>
  <dict>
    <key>CFBundleTypeIconFile</key>
    <string>Icon file for associated file</string>
    <key>CFBundleTypeName</key>
    <string>My file format</string>
    <key>CFBundleTypeRole</key>
    <string>Viewer</string> <!-- The value can be Editor, Viewer, Shell, or None. This key
is required. -->
    <key>LSItemContentTypes</key>
    <array>
      <string>UTI of the file</string> <!-- Existing UTI or create a UTI for your new
```

```
file type -->
  </array>
  <key>LSHandlerRank</key>
  <string>Owner</string>
</dict>
</array>
```

ИСТОЧНИК

Прочитайте Связывание файлов онлайн: <https://riptutorial.com/ru/osx/topic/10926/связывание-файлов>

глава 9: Установите переменные среды

Вступление

В Mac OS X вы можете установить переменные среды в один из следующих файлов:

~/ .Bashrc

~/ .Bash_profile

~/ .Profile

По умолчанию Mac OS X не имеет файлов выше, вам необходимо создать их вручную.

Examples

Добавить путь

1. `vim ~/.bash_profile`

Файл может не существовать (если нет, вы можете просто создать его).

2. type в этом и сохраните файл:

```
export PATH=$PATH:YOUR_PATH_HERE
```

Прочитайте Установите переменные среды онлайн: <https://riptutorial.com/ru/osx/topic/10162/установите-переменные-среды>

кредиты

S. No	Главы	Contributors
1	Начало работы с osx	Andrew Hoos , Community , l'l' , tbodt
2	NSFont	malicedShade
3	NSMenuItem	Andrew Hoos , Barlow Tucker
4	NSRunLoop	Marco Pashkov
5	NSStoryboard	Barlow Tucker
6	NSTextView	malicedShade
7	Вызов пользователя для файла	Andrew Hoos
8	Связывание файлов	bikram990
9	Установите переменные среды	Kuhan