eBook Gratuit

APPRENEZ OSX

eBook gratuit non affilié créé à partir des **contributeurs de Stack Overflow.**

#osx

Table des matières

À propos

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.

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

Chapitre 1: Démarrer avec osx

Remarques

Cette balise est destinée à la documentation sur les API de programmation Mac spécifiques à Apple, telles que AppKit.

Examples

Vue d'ensemble des cadres

Les applications écrites pour macOS sont généralement écrites avec Frameworks d'Apple. Les frameworks que presque chaque application utilisera sont:

- AppKit - pour créer et gérer des éléments d'interface utilisateur
- Foundation - pour les objets et opérations non-IU courants

Il existe d'autres frameworks communs utilisés dans de nombreuses applications, mais pas toutes:

- CoreData - pour le stockage de données
- Dispatch - pour la gestion de plusieurs threads
- CoreGraphics - pour dessiner des tâches liées aux graphiques
- CoreAnimation - pour l'animation d'éléments d'interface utilisateur

Lire Démarrer avec osx en ligne: https://riptutorial.com/fr/osx/topic/2818/demarrer-avec-osx

Chapitre 2: Association de fichier

Examples

Définir mon application comme application par défaut pour un type de fichier

```
- (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;
}
```

la source

Créer une association avec des types de fichiers nouveaux / personnalisés via 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>
```

la source

Lire Association de fichier en ligne: https://riptutorial.com/fr/osx/topic/10926/association-de-fichier

Chapitre 3: Définir des variables d'environnement

Introduction

Sous Mac OS X, vous pouvez définir les variables d'environnement dans l'un des fichiers suivants:

~ / .bashrc

~ / .bash_profile

~ / .profile

Par défaut, Mac OS X n'a pas les fichiers ci-dessus, vous devez le créer manuellement.

Examples

Ajouter un chemin

1. `vim ~/.bash_profile`

Le fichier peut ne pas exister (sinon, vous pouvez simplement le créer).

2. tapez ceci et enregistrez le fichier:

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

Lire Définir des variables d'environnement en ligne:
https://riptutorial.com/fr/osx/topic/10162/definir-des-variables-d-environnement

Chapitre 4: Demander à l'utilisateur un fichier

Introduction

NSOpenPanel fournit une API pour inviter l'utilisateur à ouvrir un fichier. Ce menu est l'interface utilisateur standard présentée par l'élément de menu Ouvrir (O).

Examples

Ouverture de fichiers

Ouvrir un fichier

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

Permettre l'ouverture de plusieurs fichiers

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

Limiter à des types de fichiers spécifiques

```
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
}];
```

Lire Demander à l'utilisateur un fichier en ligne: https://riptutorial.com/fr/osx/topic/9438/demander-

a-l-utilisateur-un-fichier

Chapitre 5: NSFont

Introduction

NSFont est l'objet qui fournit aux applications Mac des informations sur les glyphes et des caractéristiques de police à utiliser principalement pour l'affichage. Vous apprendrez à créer et à utiliser des objets NSFont de diverses manières, communes et peu courantes.

Examples

Créer un objet NSFont

La méthode préférée et la plus courante pour créer un objet NSFont est la suivante:

Objectif c

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

Lire NSFont en ligne: https://riptutorial.com/fr/osx/topic/8881/nsfont

Chapitre 6: NSMenuItem

Remarques

Voir la documentation d'Apple ici:
https://developer.apple.com/library/mac/documentation/Cocoa/Reference/ApplicationKit/Classes/NSMenu

Examples

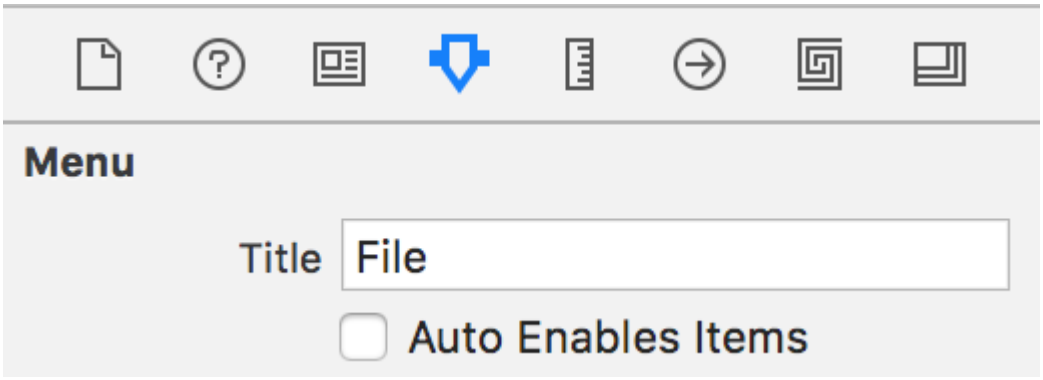
Activation des éléments de menu

Activation manuelle des éléments de menu

Pour contrôler manuellement l'état activé d'un élément de menu, le menu qui le contient doit désactiver l'activation automatique de ses éléments.

Les menus peuvent désactiver l'activation automatique de deux manières:

1. Dans le générateur d'interface



2. Dans du code

```
menu.autoenablesItems = false
```

Les deux mécanismes mis la autoenablesItems propriété sur NSMenu .

Une fois que le menu est un menu qui ne permet plus d'activer ou de désactiver les éléments de menu, les éléments de menu peuvent être définis par programmation de deux manières différentes.

1. Dans Interface Builder
2. Dans du code

```
menuItem.enabled = true
```

Activation automatique des éléments de menu

Les éléments de menu peuvent être automatiquement activés en connectant les actions des éléments de menu au premier répondant et en implémentant l'action fournie sur un objet de la chaîne de répondeur, comme décrit dans la documentation d'Apple.

Prise en charge des actions de menu par défaut

Les menus agissent comme tous les éléments de contrôle standard. Ils ont une action qui est la fonction à appeler et une cible qui est l'objet à envoyer à la fonction. Si la cible est définie sur un objet, lorsqu'un utilisateur sélectionne un élément de menu, la méthode d'action est envoyée à l'objet cible. Si l'élément de menu a une action, mais pas une cible, la cible sera sélectionnée dynamiquement à partir du premier objet parmi les suivants qui répondent à l'action:

1. Le premier répondant
2. La hiérarchie des vues
3. Fenêtre
4. Contrôleur de fenêtre
5. `NSApplication`
6. `NSApplication.delegate`
7. `NSApplication.nextResponder`

Implémenter l'élément de menu Open ((O) par défaut peut être accompli en implémentant la méthode `openDocument` sur n'importe quel objet de la liste ci-dessus.

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

}
```

Ajout et suppression d'éléments dans un menu

```
// add an item to a menu
menu.addItem(item)

// remove and item from a menu
menu.removeItem(item)
```

Lire NSMenuItem en ligne: https://riptutorial.com/fr/osx/topic/6038/nsmenuitem

Chapitre 7: NSRunLoop

Examples

application démon simple

Un processus démon exécute un programme en arrière-plan, généralement sans intervention de l'utilisateur. L'exemple ci-dessous montre comment créer un démon et enregistrer un écouteur, qui surveille toutes les applications ouvertes. La partie principale est l'appel de fonction `NSRunLoop.mainRunLoop().run()` , qui démarre le démon.

```
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()
```

Vous pouvez également utiliser ce code comme base pour un processus serveur.

Lire NSRunLoop en ligne: https://riptutorial.com/fr/osx/topic/6667/nsrunloop

Chapitre 8: NSStoryBoard

Examples

Ouvrir un nouveau contrôleur de fenêtre

Pour ouvrir une nouvelle fenêtre, ajoutez le code suivant quelque part où vous pouvez conserver une référence à la nouvelle fenêtre (IE, le délégué d'application).

Rapide

```
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)
```

Objectif 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];
```

Une fois que vous avez créé votre `controller` assurez-vous de le conserver en dehors de l'appel de fonction. Cela peut être fait en créant une variable `NSWindowController` dans votre délégué d'application et en assignant votre nouveau contrôleur à la variable.

Lire NSStoryBoard en ligne: https://riptutorial.com/fr/osx/topic/4287/nsstoryboard

Chapitre 9: NSTextView

Introduction

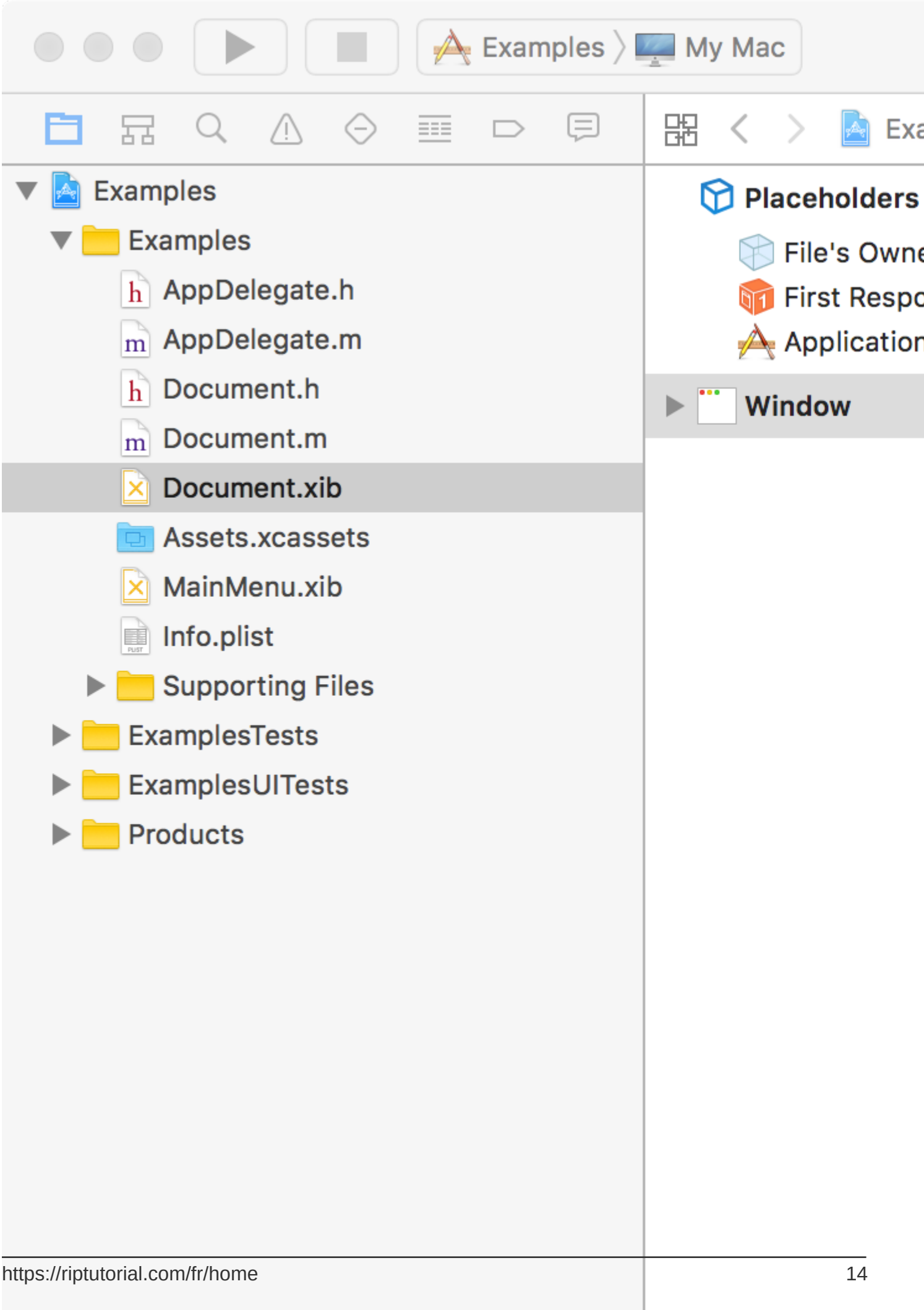
NSTextView est le gestionnaire principal du système de texte AppKit d'Apple. Il contient tout ce dont vous avez besoin pour créer un espace d'affichage / éditeur de texte dans les applications OS X (renommées MacOS).

Examples

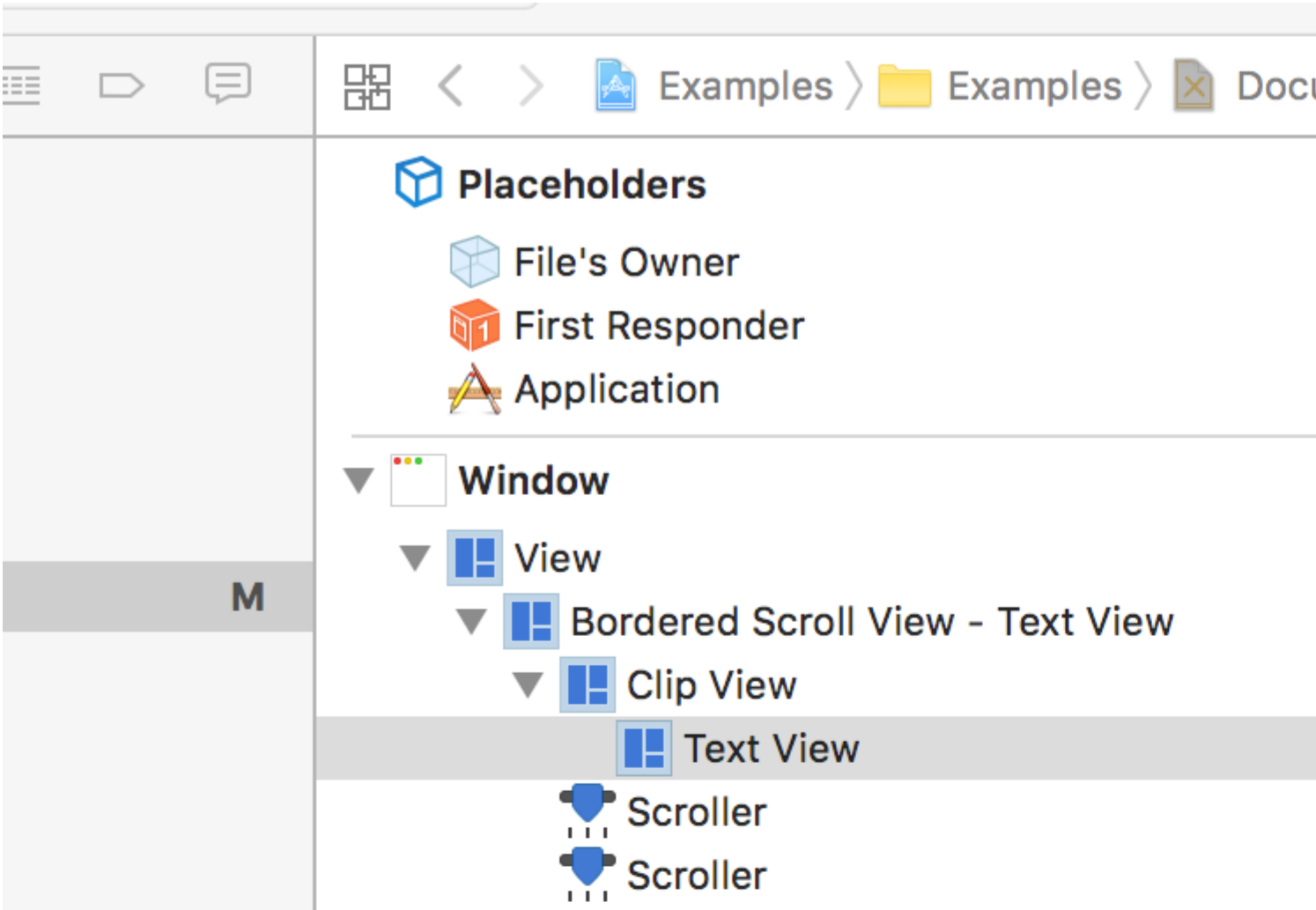
Créer un NSTextView

Graphiquement

Dans XCode, un simple NSTextView peut être créé en en faisant glisser un depuis la bibliothèque d'objets.


Examples
My Mac
Examples
Examples
AppDelegate.h
AppDelegate.m
Document.h
Document.m
Document.xib
Assets.xcassets
MainMenu.xib
Info.plist
Supporting Files
ExamplesTests
ExamplesUITests
Products
Placeholders
File's Owne
First Respo
Application
Window

qui est automatiquement configuré pour se développer verticalement avec la vue de texte. Assurez-vous que lorsque vous choisissez l'option (), vous créez des connexions à la vue du texte et non à la vue de défilement.



Drag out to expose

1. Un NSLayoutManager - effectue la disposition de glyphes / caractères.
2. Un NSTextContainer - contrôle l'espace graphique que peuvent occuper les glyphes / caractères.
3. Un NSTextStorage - contient les données de chaîne réelles affichées par NSTextView.

- Un NSTextStorage peut avoir plusieurs NSLayoutManager, mais un NSLayoutManager ne peut en avoir qu'un seul. Ceci est utile si vous souhaitez afficher les mêmes données de différentes manières en même temps.
- NSLayoutManager peut avoir beaucoup de NSTextContainer. Utile pour le texte paginé.
- NSTextView ne peut avoir qu'un seul NSTextContainer à la fois.
- Certaines choses intégrées à NSTextView sont hors limites au moment de la rédaction. Par exemple, les fonctions intégrées de recherche et de remplacement ne peuvent pas être personnalisées, mais elles peuvent être remplacées par des fonctions personnalisées.

Vous trouverez plus d'informations sur les manières d'utiliser le système de texte ici .

Maintenant pour le code. Ce code créera un NSTextView simple, sans même défiler. Des choses comme le défilement et la pagination seront dans un autre exemple.

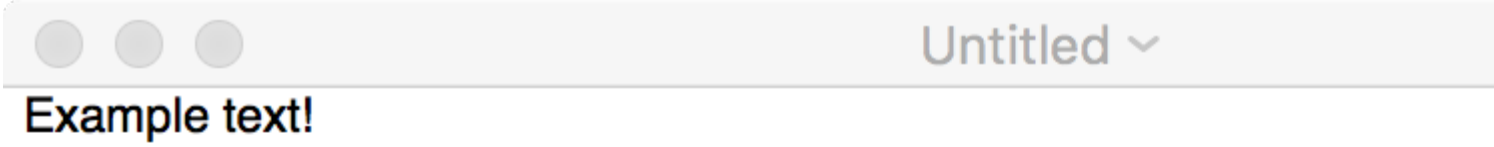
Objectif c

```
// 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];
[windowController.window setContentView:textView];
```

Toutes nos félicitations! Vous avez créé un NSTextView par programmation!



Lire NSTextView en ligne: https://riptutorial.com/fr/osx/topic/8880/nstextview

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec osx	Andrew Hoos, Community, l'L'l, tbodt
2	Association de fichier	bikram990
3	Définir des variables d'environnement	Kuhan
4	Demander à l'utilisateur un fichier	Andrew Hoos
5	NSFont	malicedShade
6	NSMenuItem	Andrew Hoos, Barlow Tucker
7	NSRunLoop	Marco Pashkov
8	NSStoryBoard	Barlow Tucker
9	NSTextView	malicedShade