

# iOS程式開發設計

授課：宋文平 (Perkin)

APP Store：WENPING SUNG

Email：[perkinsung@hotmail.com](mailto:perkinsung@hotmail.com)



# 第一個iPad 電子書App

一個表格

**搞定** 全部の日語動詞變化



App免費下載





# iOS程式開發課程教材

- ◆ Swift程式語言部分：

The Swift Programming Language (Swift 3) (英文版)

The Swift Programming Language (Swift 3) (簡體中文版)

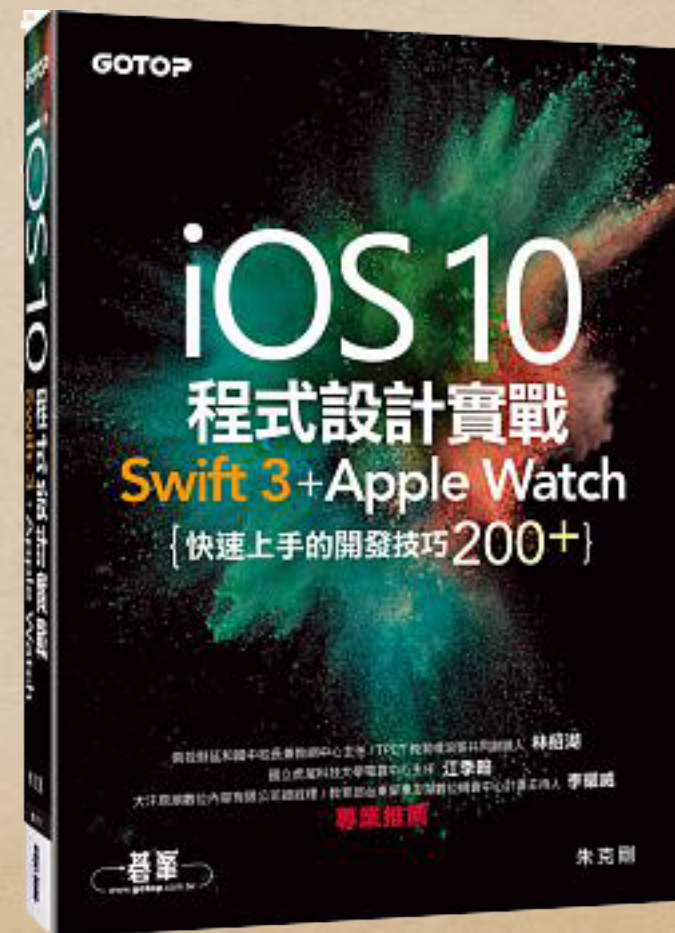
The Swift Programming Language (Swift 1.0) (正體中文版)

- ◆ 實際開發部分：

iOS 10程式設計實戰

Swift 3 + Apple Watch

快速上手的開發技巧200+ (碁峯)





# Mac vs. windows 快速鍵對應

- ◆ command鍵  $\Leftrightarrow$  window鍵

快速鍵作用等同於windows系統中的ctrl鍵

- ◆ option鍵  $\Leftrightarrow$  alt鍵
- ◆ shift鍵  $\Leftrightarrow$  shift鍵
- ◆ control鍵  $\neq$  ctrl鍵



## Mac作業系統常用快速鍵

- ◆ `command + q` : 結束作用中的應用程式
- ◆ `command + n` : 開啟作用中應用程式的新視窗
- ◆ `command + a` : 全選
- ◆ `command + 空白鍵` (放開`command`) : 快速切換上一次的輸入法
- ◆ `command + 空白鍵` (持續按住`command`) : 輸入法選單切換
- ◆ `command + shift + 3` : 螢幕快照(桌面檔案)
- ◆ `control + command + shift + 3` : 螢幕快照(剪貼簿)
- ◆ `command + shift + 4` : 特定範圍的螢幕快照(桌面檔案)
- ◆ `control + command + shift + 4` : 特定範圍的螢幕快照(剪貼簿)



# Mac及windows 系統的磁碟格式

- ◆ FAT、FAT32

Mac、windows通用

- ◆ NTFS

windows可讀寫、Mac唯讀（無法寫入）

- ◆ exFAT（FAT的擴充格式）

Mac、windows通用

- ◆ Mac OS擴充格式

Mac獨有、windows無法讀取



# iPhone解析度

- ◆ iPhone 3G (3.5吋)

Point: 320 X 480

Pixel: 320 X 480

- ◆ iPhone 4S (3.5吋)

Point: 320 X 480

Pixel: 640 X 960 ← @2X

- ◆ iPhone 5 & iPhone 5S (4吋)

Point: 320 X 568

Pixel: 640 X 1136 ← @2X

- ◆ iPhone 6 (4.7吋)

Point: 375 X 667

Pixel: 750 X 1334 ← @2X

- ◆ iPhone 6 Plus (5.5吋)

Point: 414 X 736

Pixel: 1080 X 1920 ← @3X (實際約為2.6x)



# iPad解析度

- ◆ iPad2 (9.7吋)

Point: 768 X 1024

Pixel: 768 X 1024

- ◆ iPad3以後機種、iPad Air (9.7吋)


Point: 768 X 1024

Pixel: 1536 X 2048 ← @2X

- ◆ iPad Pro (12.9吋)

Point: 1024 X 1366

Pixel: 2048 X 2732 ← @2X

 iCon解析度參考文件



# 裝置與App解析度

在單一Storyboard之中，可以呈現各種裝置的版面配置  
(Auto Layout與Trait Variations搭配使用)

\*增加模擬器種類\*

在模擬器啟動的狀態下 → Hardware → Device → Manage Devices



# Auto Layout & Trait Variations<練習一>

佈置一個填滿螢幕的物件

畫面上佈置一個UIImageView，並且選定展示之圖片

增加四個Constraints：

1.Pin～先取消constrain to margins勾選

2.Pin～加入上下左右四個掛勾，並將四邊距離都設為零



# Auto Layout & Trait Variations<練習二>

製作通用版Layout：物體置中

Storyboard內佈置一個Button

增加兩個Constraints：

- 1.Align～水平、垂直位置皆位於Container中央
- 2.Pin～指定寬高



# Auto Layout & Trait Variations <練習三>

製作通用版Layout：將物體固定於左上角

Storyboard內佈置一個Button

增加兩個Constraints：

- 1.Pin～加入上方、左側的掛勾
- 2.Pin～指定寬高



# Auto Layout & Trait Variations <練習四>

製作通用版Layout：兩個物體並列及等寬

Storyboard內佈置兩個Button

- 1.第一個Button加入限制條件定位於左上角（練習三），同時指定寬度100 <共加入三個Constraints>
- 2.第二個Button加入上方掛勾，勾住上一個Button
- 3.同時點選兩個Button（作為基準的物件最後點選）Align～勾選Leading Edges
- 4.同時點選兩個Button，讓兩個Button等寬（Pin～Equal Width）



# Auto Layout & Trait Variations<練習五>

製作特別版Layout：以iPad直式或橫式為例

延續<練習四>

將Storyboard版型更換成wRegular hRegular(iPad直式或橫式)

請將畫面上所有按鈕內的文字，加大其顯示字體：

- 1.點選按鈕後，在屬性視窗中找到Font屬性，點一下左邊的+，在彈出的視窗中，選擇（current）
- 2.在Font下方多出來的x部分，調大字體
- 3.測試iPad和iPhone在模擬器下所呈現的字體大小



# App實機安裝測試

檢查開發憑證：Launchpad → 其他 → 鑰匙圈存取

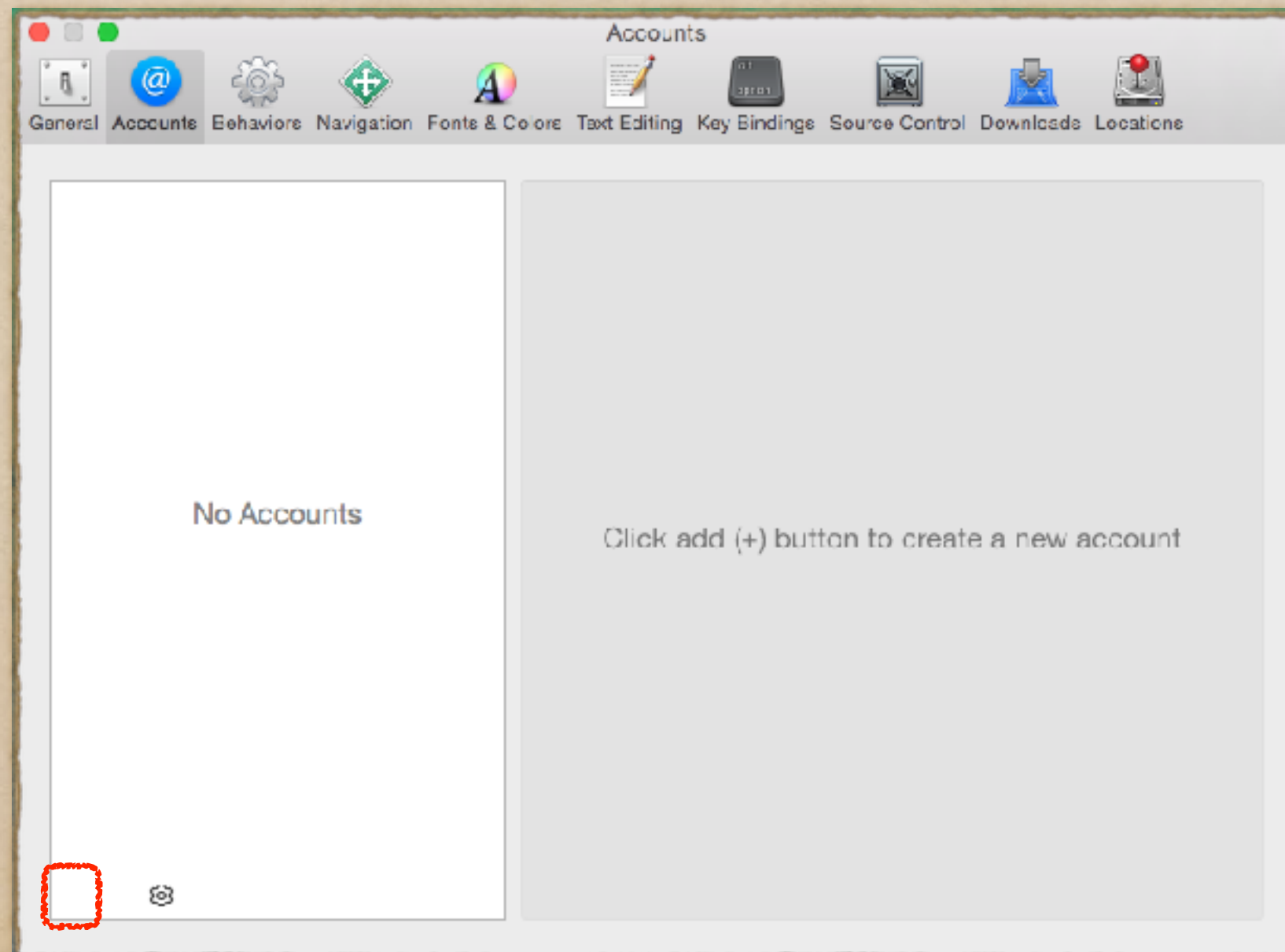


名稱為iOS Developer或iPhone Developer開頭者，即為開發憑證，若過期或發生問題，可以全部刪除後，再重新製作憑證。



# App實機安裝測試

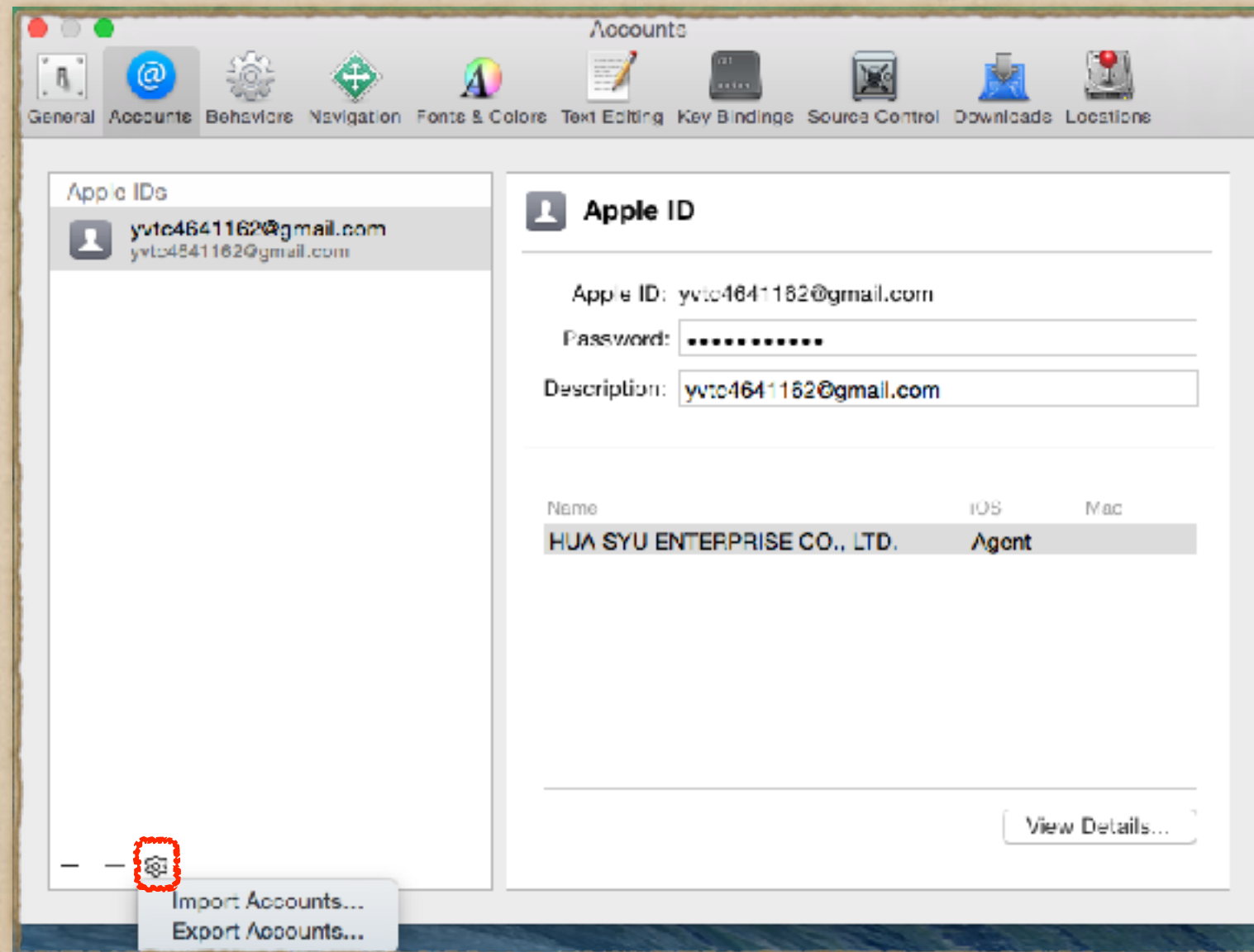
加入付費或免費開發帳號：Xcode → Preferences → +





# App實機安裝測試

開發憑證之匯入、匯出：Xcode → Preferences → 工具圖示



主要的開發憑證同一時間僅能從一台Mac電腦認證。

多台電腦共用同一開發者帳號時，僅能使用匯出、匯入功能來分享憑證。



# MVC架構

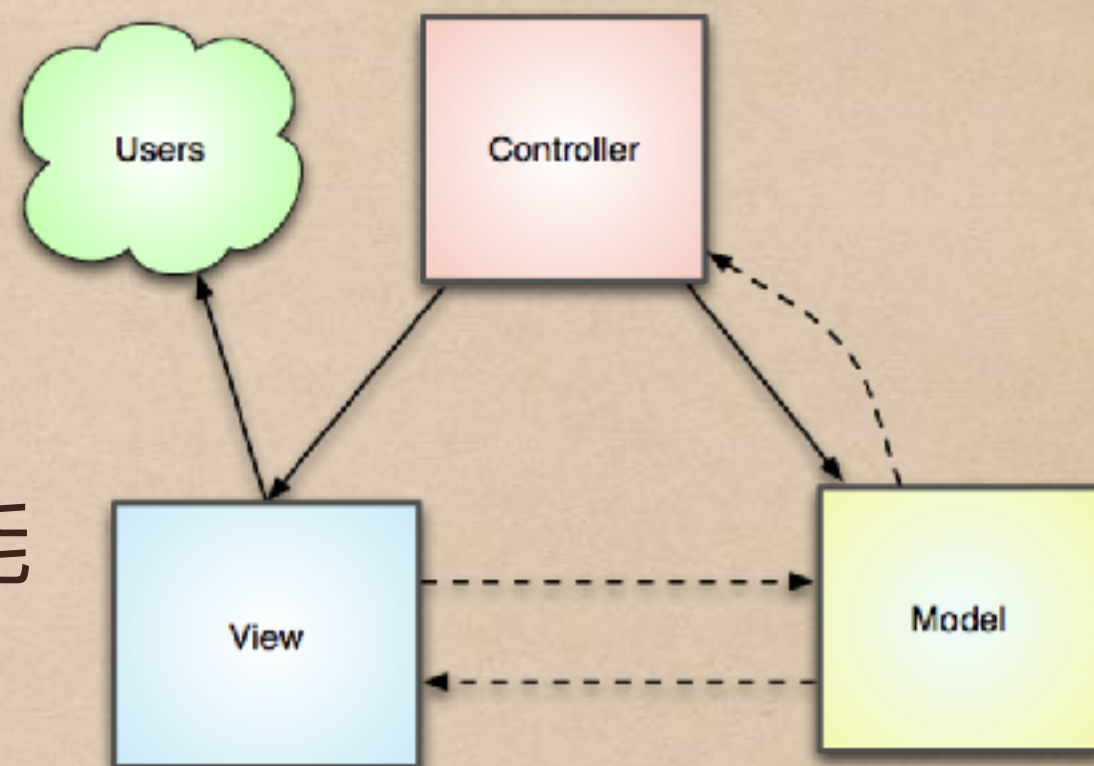
- ◆ Model  
資料存取、通訊等功能

- ◆ View

使用者可見、可操作的視覺元件 (UI)

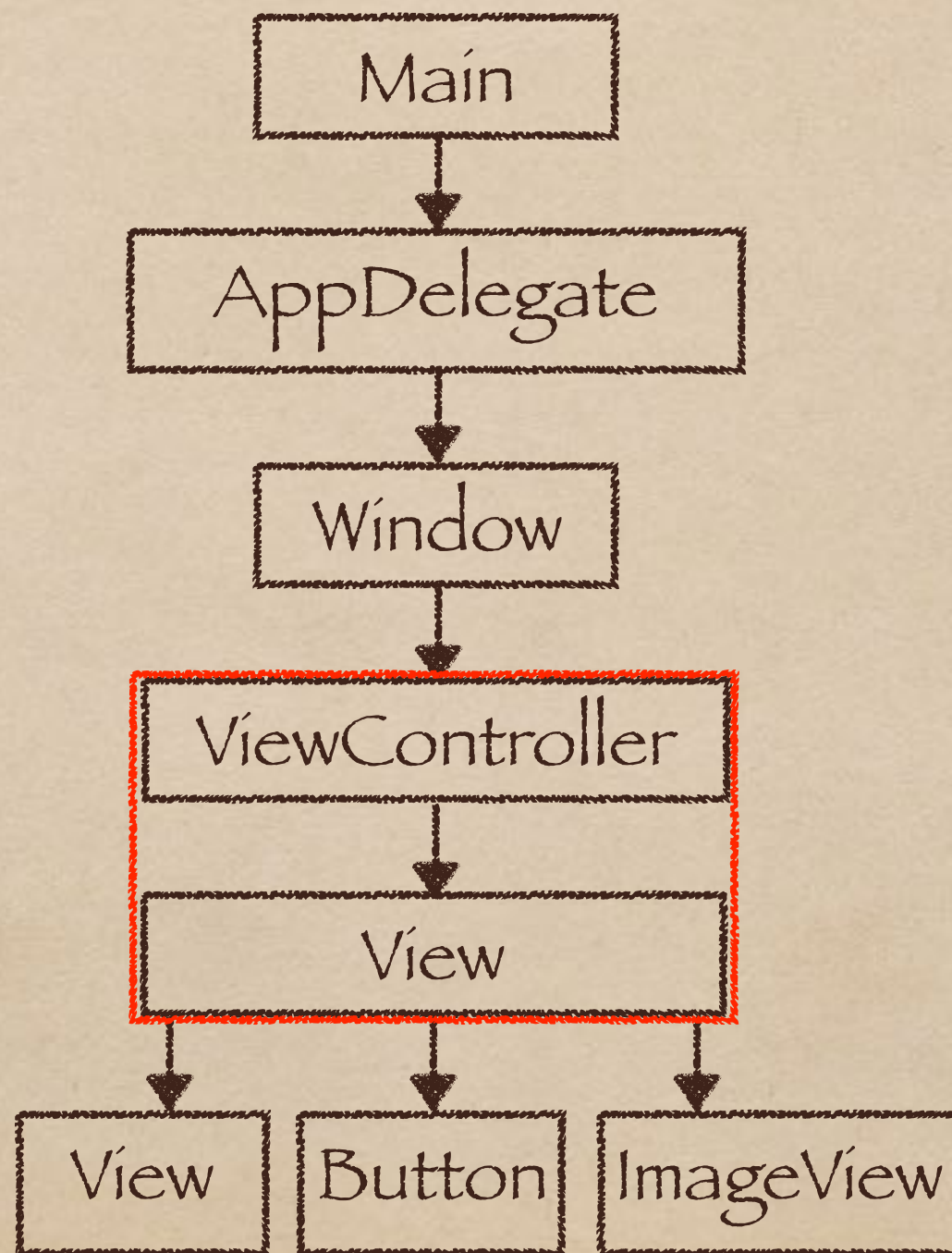
- ◆ Controller

控制UI元件之展現、與Model溝通以存取資料





# iOS軟體架構





# Application Life Cycle

1. `didFinishLaunchingWithOptions` : 載入App
2. `applicationDidBecomeActive` : 啟動App
3. `applicationWillResignActive` : App即將進入背景
4. `applicationDidEnterBackground` : App進入背景
5. `applicationWillEnterForeground` : App即將從背景回到前景
6. `applicationWillTerminate` : App終止

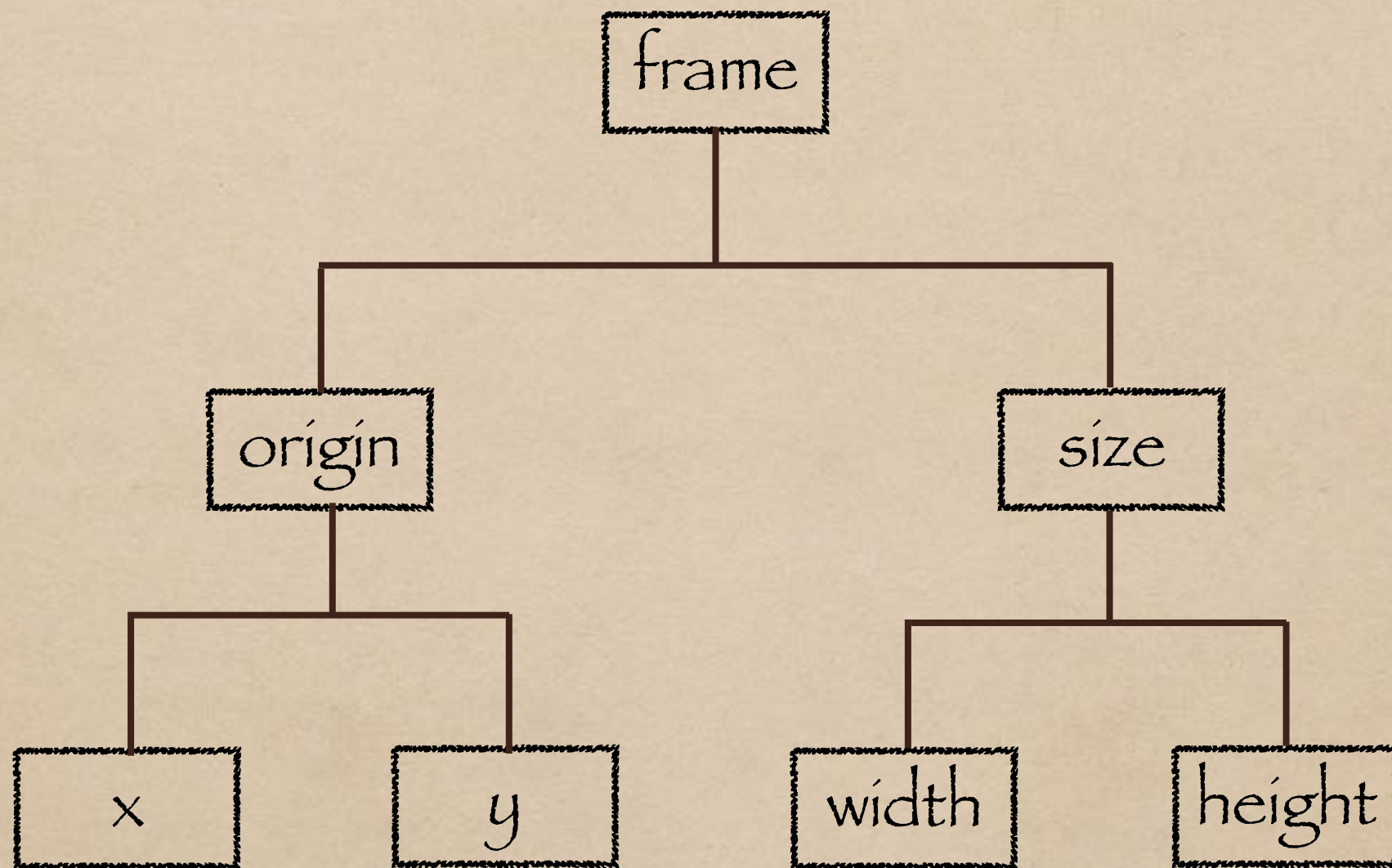


# ViewController Life Cycle

1. **viewDidLoad** : controller的view載入完成
2. **viewWillAppear** : controller的view即將被加入 view hierarchy (view的階層架構)
3. **viewDidAppear** : controller的view已被加入 view的階層架構中
4. **viewDidLayoutSubviews** : controller的view已經完成介面佈置
5. **viewWillDisappear** : controller的view即將從 view的階層架構中移除
6. **viewDidDisappear** : controller的view已經從 view的階層架構中移除



# UIView的frame





# 代理機制

## 制定代理機制

- 1.宣告protocol的代理方法（此時還沒有實作代理方法）
- 2.定義一個類別，其中包含遵循於protocol的delegate或datasource
- 3.在類別中提供特定的方法，讓delegate或datasource可以去啟動protocol的代理方法



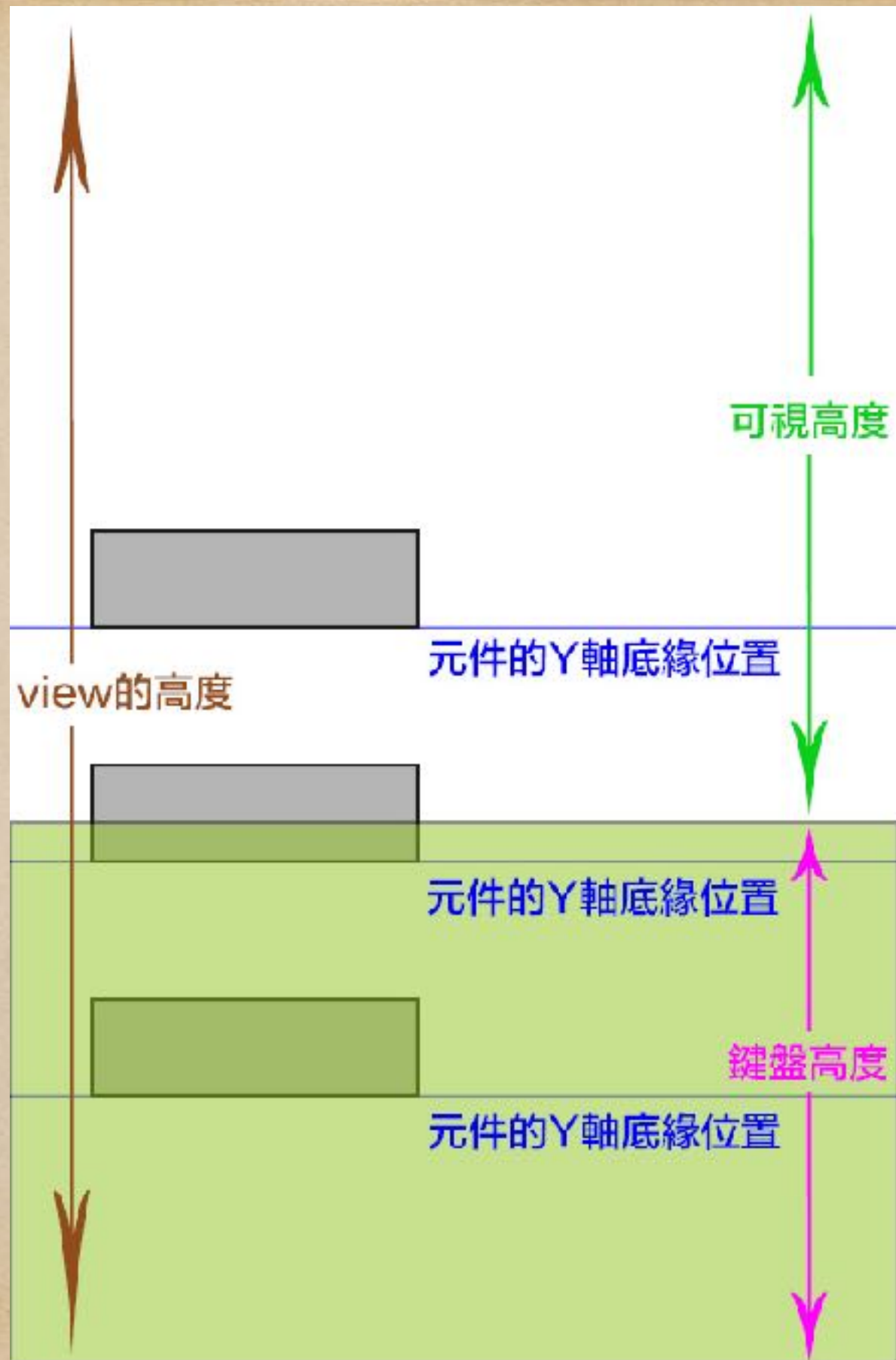
# 代理機制

## 使用代理機制

- 1.採納特定的protocol
- 2.實作代理方法
- 3.生成一個包含代理人（delegate或datasource）的『類別實體』
- 4.指定此類別實體的代理人所在位置



# 虛擬鍵盤 & 輸入元件





# SQLite資料庫

SQLite Studio 下載：<http://sqlitestudio.pl>

SQLite 官方網站：<https://www.sqlite.org/index.html>

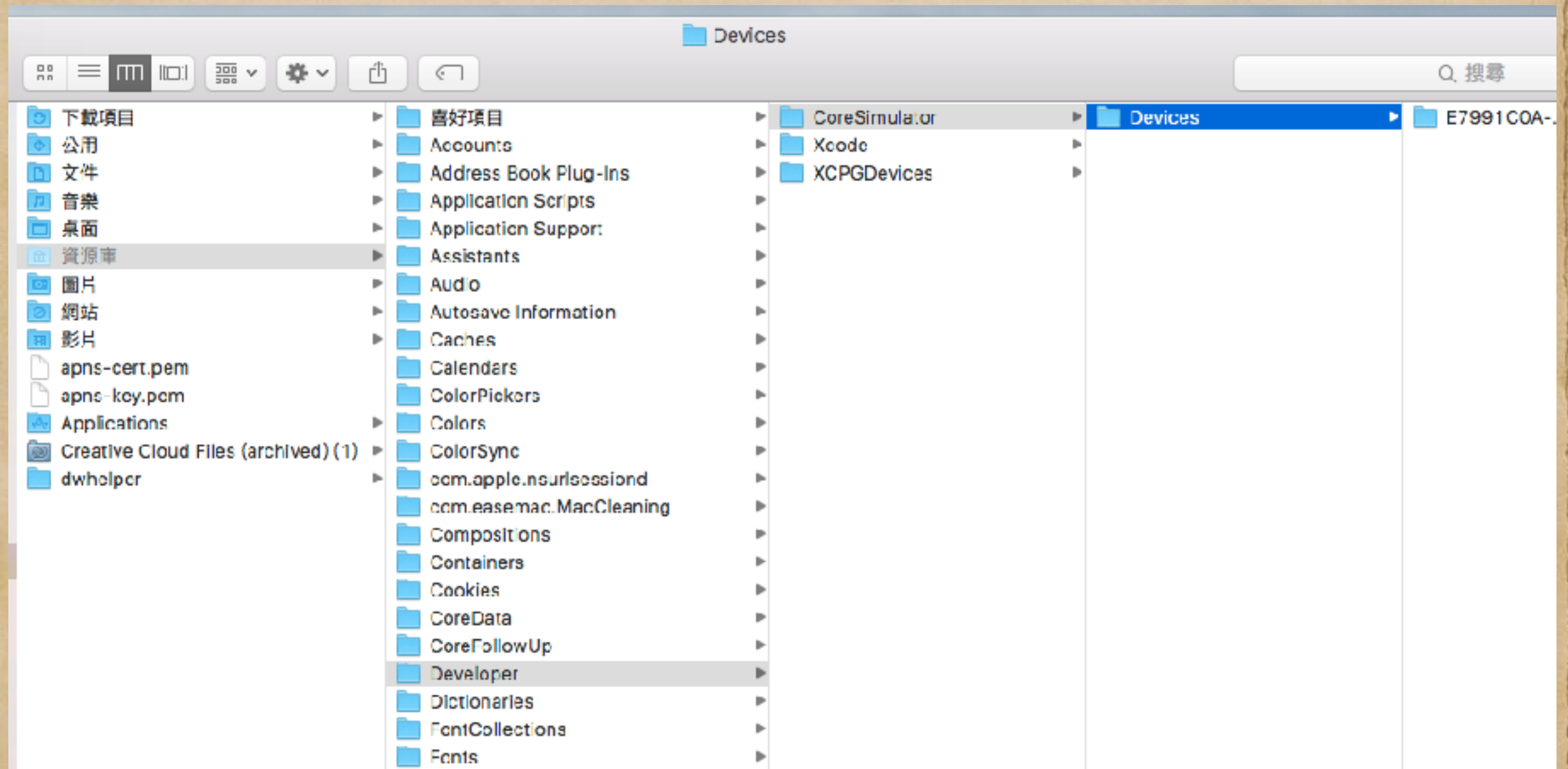
SQLite 資料型態：<https://www.sqlite.org/datatype3.html>

SQLite 日期函式：[https://www.sqlite.org/lang\\_datefunc.html](https://www.sqlite.org/lang_datefunc.html)



# 模擬器路徑

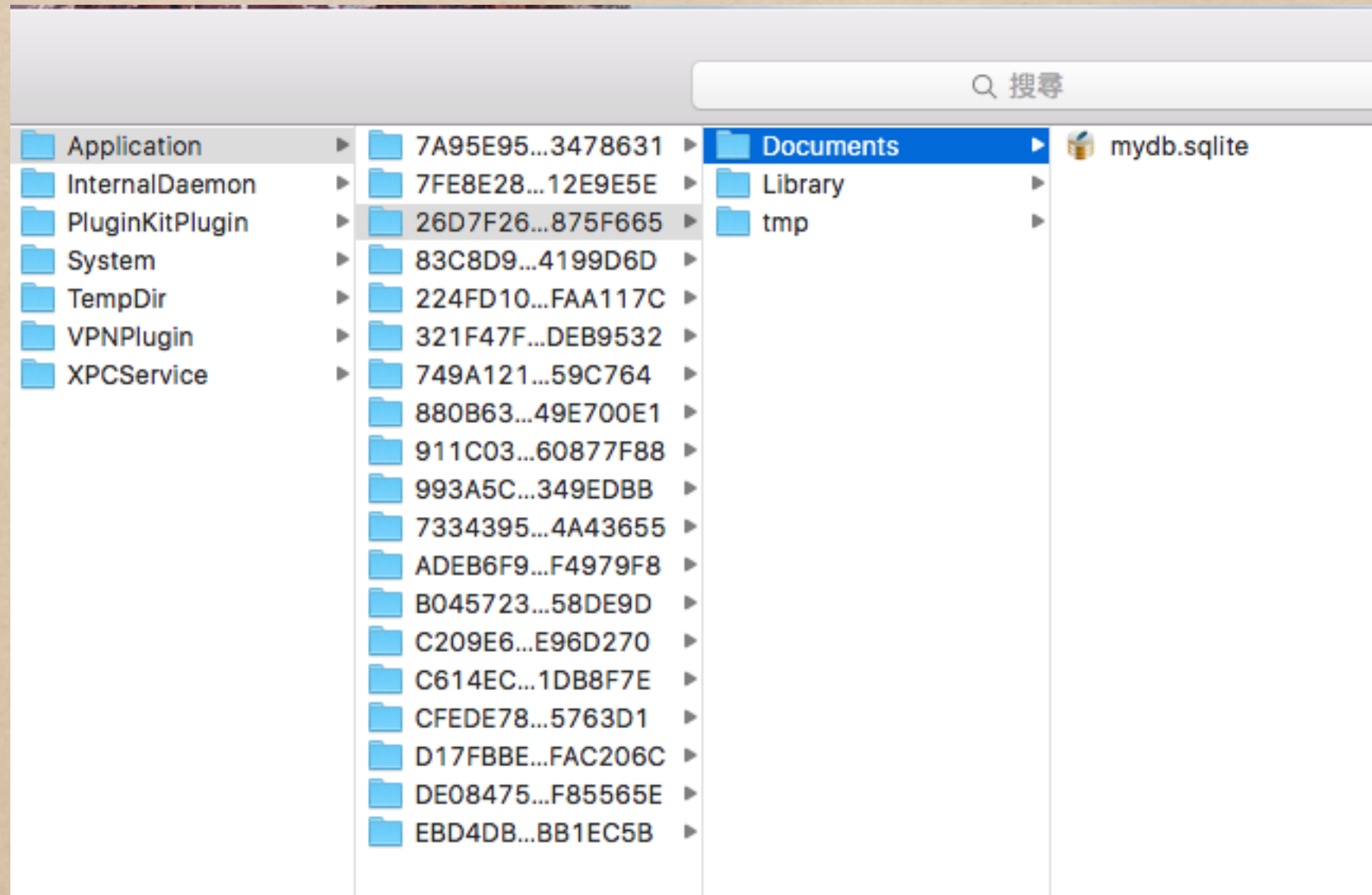
Finder → 前往 → （加按option鍵）資源庫 →  
Developer → Devices





# 模擬器路徑

(承上頁) 找到特定App的Documents路徑



PS.上述畫面的App編碼，可以從AppDelegate中藉由檔案管理員印出！



## 字串格式化 (適用於String & NSString)

型態修飾詞	資料 型態	所佔 byte 數	表示範圍	使用 NSLog 函 式的輸出格式
	char	1	-128 ~ 127	%c
	int	4	-2,147,483,648 ~ 2,147,483,647	%i
	float	4	3.4E-38 ~ 3.4E+38	%f
	double	8	1.7E-308 ~ 1.7E+308	%f
short	int	2	-32,768 ~ 32767	%hi
long	int	8		%li
long	double	16		%Lf
★ unsigned	int	4	0 ~ 4,294,967,295	%u
unsigned short	int	2	0 ~ 65,535	%hu
unsigned long	int	8		%lu
	char[] or char *			%s

object

%@



# UIScrollView元件



contentSize：可捲動區域大小

contentOffset：UIScrollView的頂點跟目前可視範圍的偏移量  
(捲動畫面時，此偏移量隨之改變)

contentInset：畫面捲動到最左上角時，放開之後畫面的回彈位置



# Xcode常用快速鍵

- ◆ `command + z` : 復原修改資料
- ◆ `command + ]` : 縮排
- ◆ `command + [` : 復原縮排
- ◆ `command + /` : 整段註解
- ◆ `option(alt) + 單次點擊` : 快速查閱相關功能、定義
- ◆ `option(alt) + 快點兩下` : 查閱開發文件
- ◆ 選定畫面上已存在物件 + `option` + 滑鼠拖曳 : 複製物件



# Xcode常用快速鍵

- ◆ 在storyboard與程式碼雙畫面呈現時，點選畫面物件 + control + 滑鼠拖曳到程式碼視窗：建立Action或Outlet
- ◆ 點選畫面上特定按鈕 + control + 拖曳到下一個畫面：製作segue(畫面轉換)



## 『建立』 自己的Framework (API)

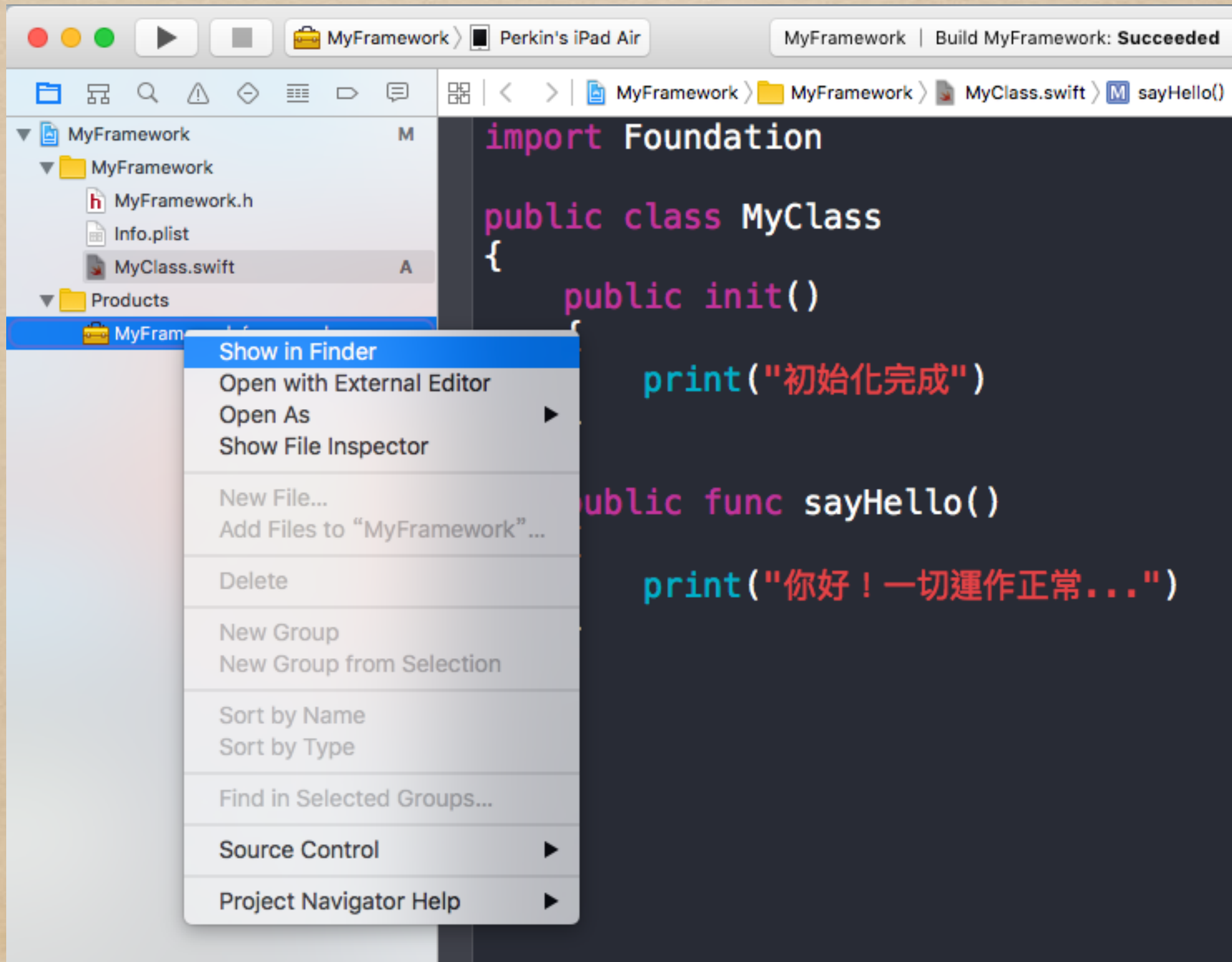
請依序執行以下步驟：

- ◆ Xcode ► File ► New ► Project ► iOS ► Framework & Library ► Cocoa Touch Framework
- ◆ 增加<自訂類別>，並將類別、特定的屬性和方法設定為『Public存取權限』（測試程式可參考下頁截圖）
- ◆ 產生Framework ~ Xcode ► Product ► Build



# 『建立』自己的Framework (API)

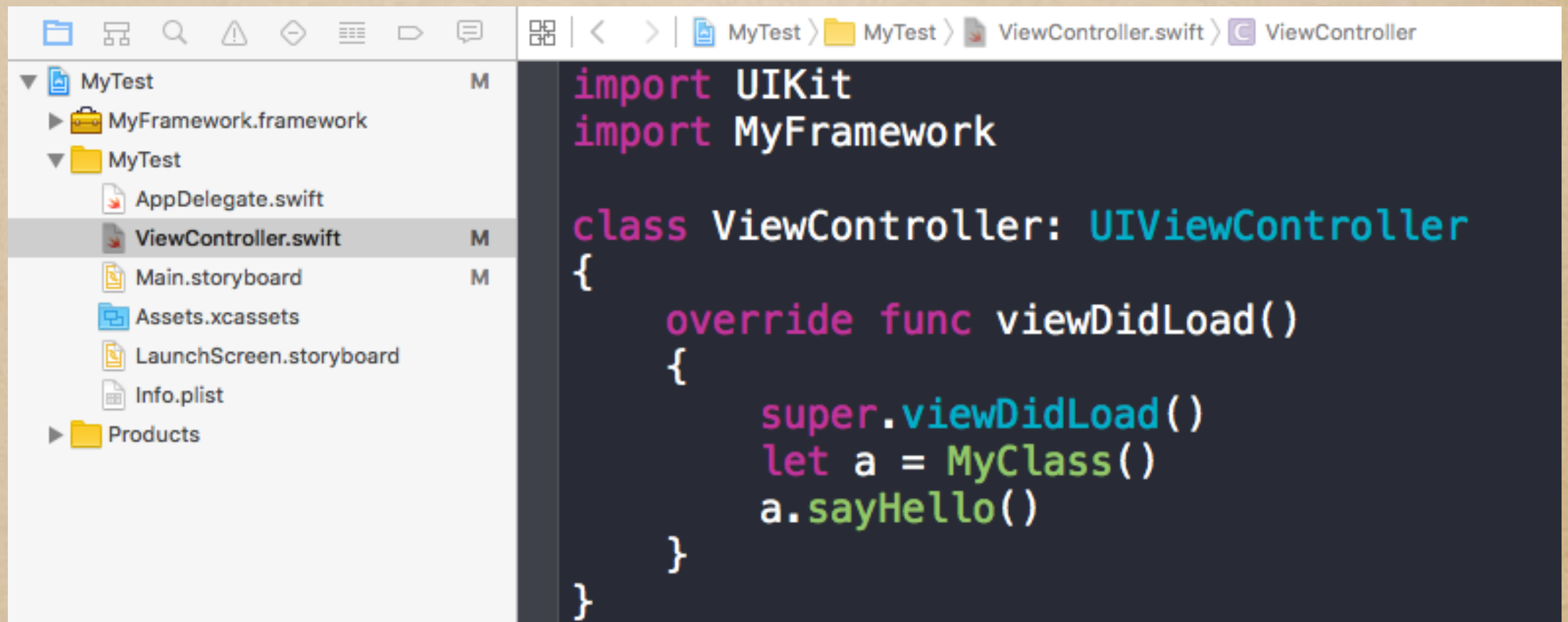
- ◆ 參照下圖，找到新Framework的檔案所在位置





# 使用自己的App

- ◆ 將製作好的Framework檔案，拖曳到需要使用的專案中，import之後呼叫其中的類別方法，如下圖：



- ◆ 執行之前，請檢查下一頁截圖中的設定，在Target的General頁籤中，『Embedded Binaries』與『Linked Frameworks and Libraries』必須同時出現自己的Framework，才能正常運行。



# 使用自己的API

MyTest

General

Capabilities

Resource Tags

Info

Build Settings

Build Phases

Build Rules

▼ Deployment Info

Deployment Target

9.3

Devices

Universal

Main Interface

Main

Device Orientation

☒ Portrait

☐ Upside Down

☒ Landscape Left

☒ Landscape Right

Status Bar Style

Default

☐ Hide status bar

☐ Requires full screen

▼ App Icons and Launch Images

App Icons Source

AppIcon

Launch Images Source

Use Asset Catalog

Launch Screen File

LaunchScreen

▼ Embedded Binaries

MyFramework.framework

+

—

▼ Linked Frameworks and Libraries

Name	Status
<div><div>MyFramework.framework</div></div>	Required
<div><div>MyFramework.framework</div></div>	Required
<div><div>+</div><div>—</div></div>	