

# 第十周 完善你的程式

## 上架前的準備

你可能已經寫好一個 APP，準備要上架 App Store 了。上架就和學習開發不同了，程式必需是【完整】的，也就是說，不能有草稿 Logo，或權限要求不能只是有字，至少要有正確的要求等，畫面中也不能有空白的圖案，例如 TabBar Icon 必需要設定圖片，不可以空白，特別是在各種畫面中，不能出現 test / beta / 工程中 / 製作中 等字眼。

上架前也有各種考量，上架主要是讓一般的使用者，都能下載使用，不符這個目的，也無法上架

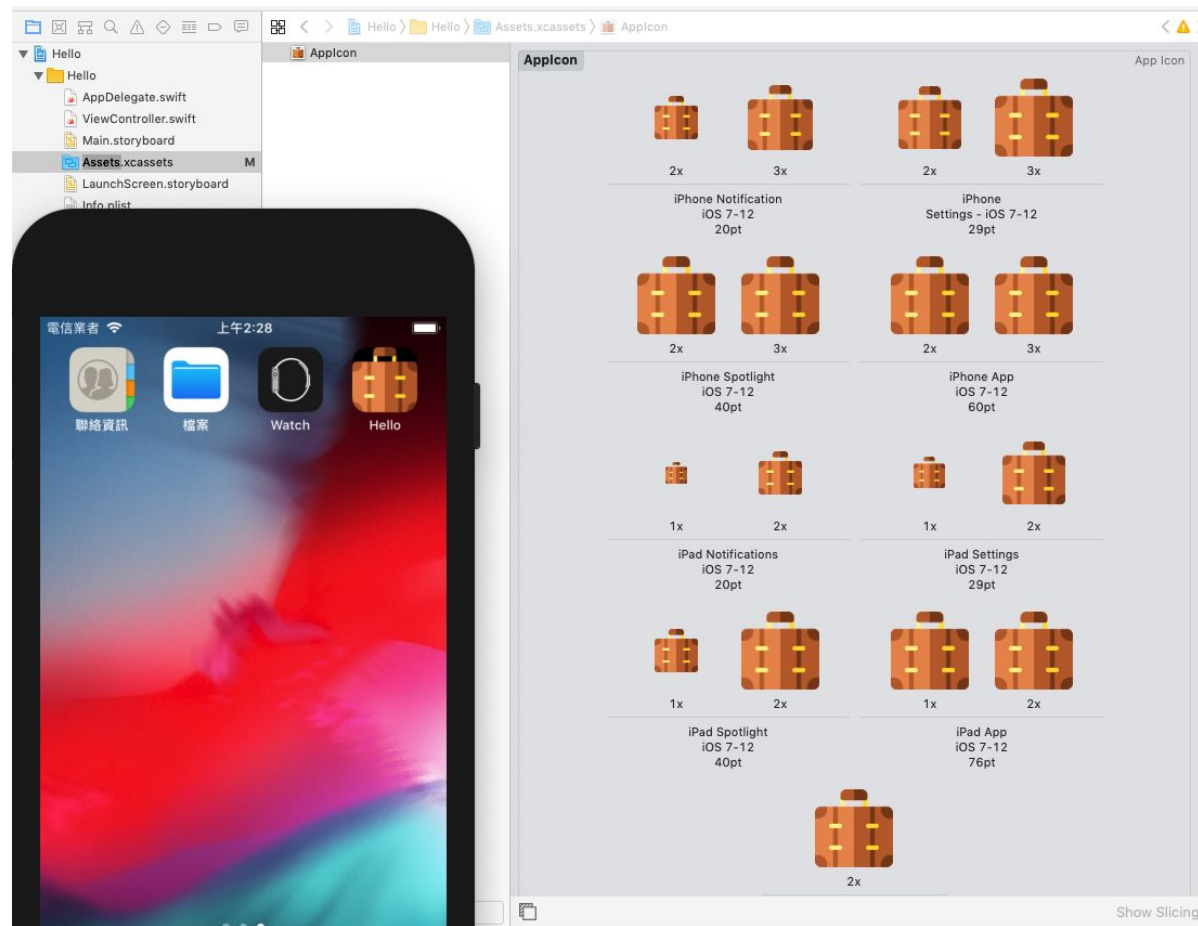
## 啟動畫面

LaunchScreen.storyboard 是 APP 的啟動畫面，你必需完成它才可以上架。基本上它和其他的 storyboard 是相同的，只是由於該畫面會在 Main.storyboard 的 init view controller 完成載入後就消失，所以出現的生命周期很短，通常是拉個載入中的圖片，和版權圖片就可以了。如果你的 APP 並沒有需要等待的載入過程，你至少要把 LaunchScreen.storyboard 調整成透明，或加上一個 Loading.... 的權，才能通過審核

# 完成 Icon

要上架的話，必需  
確定每一個 Icon  
都有依照規定的解  
析度製作，並保留  
1024x1024 的圖，  
做為 AppStore 用

忘了的話找一下第  
三周的課程



# 多國語系

---

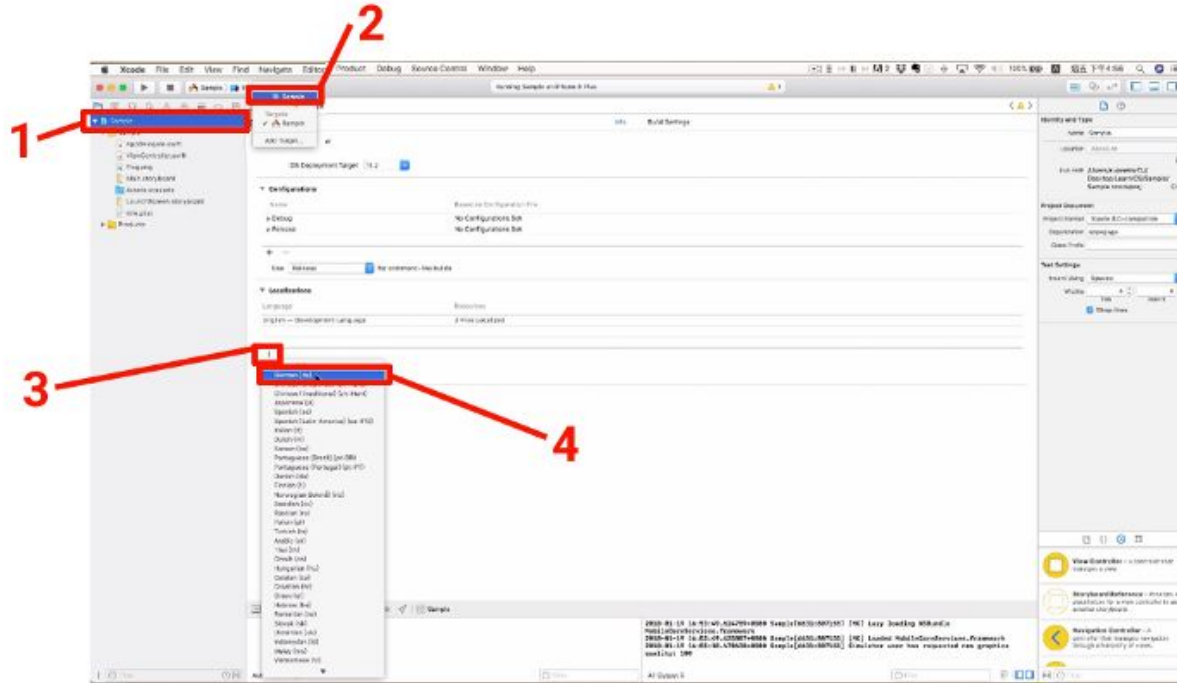
# 如何設定多國語系

1. 設定專案檔語系
2. 新增 Localizable.string
3. 設定字串檔語系
4. 修改程式

# 設定多國語系

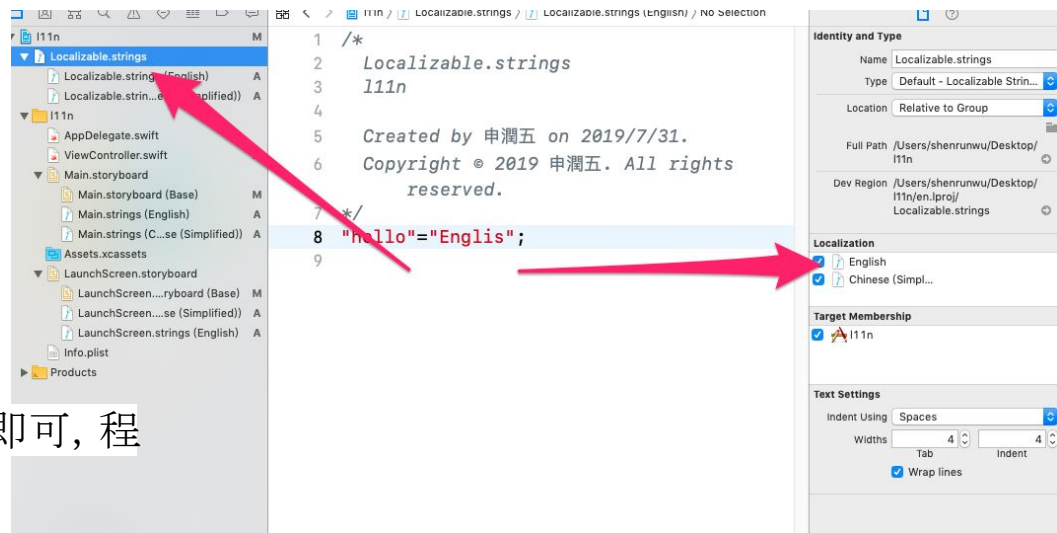
1. 選擇專案
2. 選擇 Target
3. 新增語系
4. 選擇語言

選擇的完成時候，所有的UI檔都可以選多語化，你可以依需求來設定，但因為使用不便，幾乎沒有人這樣做



# 字串資源檔

使用 New->File 後，選 string 檔案類型即可，程式用的必需叫做 Localizable.string



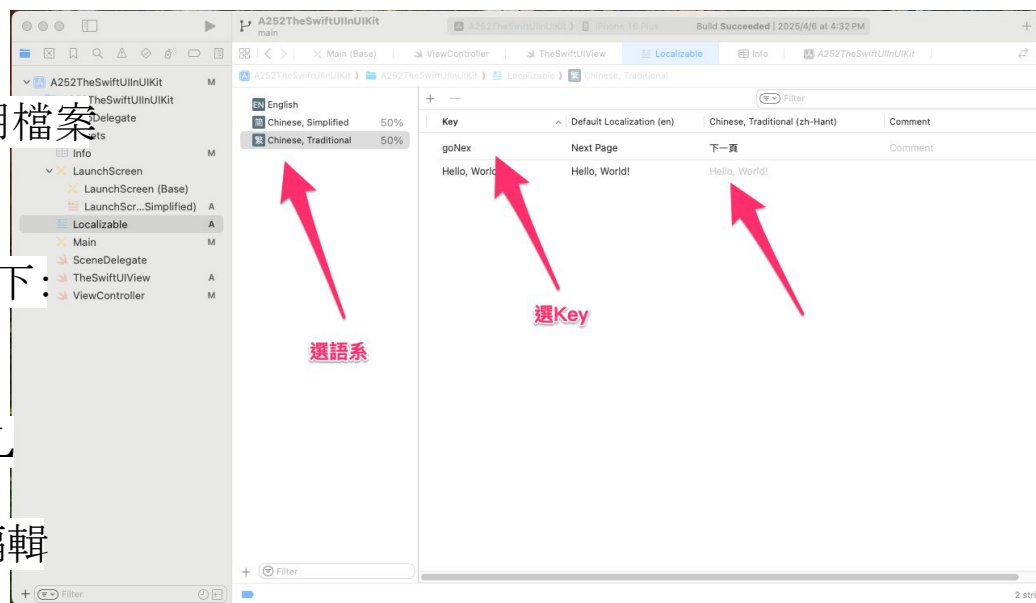
.string 是字串資源檔，你可以用這種檔案來做字串的管理，而不用改寫程式，而其中 Localizable.string 是專為多語系設定的專用檔案名稱。

字串資源檔內容設定也很簡單，格式範例如下：

"字串名稱" = "字串內容";

Xcode 13 之後，預設的多語系，改成了 XML

Xcode 15 之後，預設的UI改成了單一表格編輯





# 多語系的設定方法

NSString(key: String , comment: String)

```
extension String{  
    func localize() -> String{  
        return NSString(self, comment: "")  
    }  
}
```

# APP 顯示名稱的多語系

1. 在 Info.plist 中新增 Application has localized Display Name 設為 YES
2. 建立 InfoPlist.strings, 類似程式用的方法建立各語系檔
3. 在各語系檔新增
  - a. `CFBundleDisplayName = "APP 名稱";`

# APP 內指定語言的作法

iOS 的語言指定, 有一串 SOP, OS 可以設定主要語言, APP 也可以指定語言。在語資源不存在時, 也有一套取代原則, 如系統為英文, APP 指定為正體中文, APP 的資源中, 沒有正體中文, 只有簡體中文時, 是用簡體還是英為作為預設的語言等等的規則。

若我們想在 APP 中直接指定語言規則時, 事情就有點麻煩, 但只要我們能控制出現的語言, 規則就是 PM 或客戶的問題了

所以我們可以深入了解 NSLocalizedString 看看是否有自行指定語言的可能性

## NSString function 的完整寫法

NSString(key: String , comment: String)

的完整寫法是

```
func NSString(_ key: String, tableName: String? = nil,  
bundle: Bundle = Bundle.main, value: String = "", comment: String)  
-> String
```

可以指定 tableName (字串檔名稱) 與 bundle (資源目錄) 的

## 自行製作指定目錄 Bundle

**Bundle** 就是資源群組，它也可以指專案一個目錄下的所有檔案，而多語系的作法實作就是把相同名稱的資源，分別放到不同目錄下，所以我們可以實作不同的資源，執行 **NSBundle**

```
let path = Bundle.main.path(forResource: "zh-Hant", ofType: "lproj")! //取得目錄
let bundle = Bundle(path:path)! //把該目錄設下的所有檔案當作資源
let localText = NSLocalizedString(key, tableName: "Localizable", bundle:bundle,
comment: "")
```

## 自行指定語言實作

```
var currentLocal = "zh-Hant"
```

```
public func LocalizedString(_ key: String, comment: String =  
"", languagePath: String? = nil) -> String {
```

```
    let path = Bundle.main.path(forResource: languagePath ?? currentLocal,  
ofType: "lproj") ?? ""
```

```
    let localText = NSLocalizedString(key, tableName: "Localizable", bundle:  
Bundle(path: path) ?? Bundle.main, comment: "")
```

```
    return localText  
}
```

# iOS - App 上架

在上架 App 之前必須先準備好開發者帳號，但申請開發者帳號需之前已經申請好了，故就跳過此步驟，直接從產生憑證到上傳 App 開始講起。首先要將自己辛苦寫好的 App 送審的話，則要依序做完下列幾件事情即可。

天下沒有白吃的午餐，想上架 App，請記得付給 Apple 年費 99 美金，才有資格上架，沒付錢的只能在旁邊玩沙。

這兒假設已設好憑證檔，已可在實體機測試的狀況下，做上架的準備

# 上架前的檢查

必需符合規定才能上架

<https://developer.apple.com/app-store/review/guidelines/>

常見問題：

1. 不良內容(情色, 博弈, 或其他法律問題等)
2. 程式撰寫不良(易閃退等bug, 未製作 Logo 等, 啟動畫面)
3. 商業模式不良(很多, 如使用共用Web套件開發即馬甲APP, 限公司內部使用無公開需求等)
4. 不符最低功能需求(沒有功能或太簡單, 如手電筒, 單網頁)
5. IDFA 與各種權限聲明(位置, 相簿等)



# 上架流程

1. 在 App Store Connect 建立 App
2. 設定各種 APP 上架需要的資料
  - a. 定價, 地區, 行銷文字等
  - b. APP 各種解析度的預覽
  - c. APP 說明
  - d. APP Store 用的縮圖與 Banner
  - e. APP Store 的搜尋 Keyword
  - f. APP 支援網站
  - g. 其他, 如測試帳號等
3. 使用 Xcode Archive 專案
4. 使用 Xcode-Organizer 發佈
5. 等待審核

# 審核問題處理

## 1. 依問題修改程式

- a. 寫程式其實經常出現在我的電腦是好的等狀況，例如不同時區下就會閃退等問題
- b. 測試機種有時也會產生 UI 錯誤，如在 iPhone 上正常，iPad / macOS 就會出問題等，可以設定限制只在 iPhone 上運作解決問題

## 2. 提出各種相關說明

- a. 部份引用他人網站/商標/形象時，要取得授權書

## 3. 如無法由審核者實測的功能，需拍影片證實

- a. 如需整合硬體測試者，可拍整合過程的影片去實作

雖然回信都是英文，但若你怕回信的說明被誤解，使用中文回信也是會被接受處理的

# 補充教材

# 程式總是學不完的

iOS 有成千上萬個 SDK, 是不可能教完和學完的, 在最後的時間, 會選擇學員工作上最需要的, 或最想學的來實作。

語音說話程式

Siri 整合

UIAnimation 動畫

Facebook 發文

字串的那些事

文字轉拼音

陣列進階應用

有顏色的字

影片, 音樂播放

APNS + Firebase 推播

各種跨平台方案簡介

如 Flutter, Xamarin

加速器, 電子羅盤, 計步器

Firebase Realtime DB 進階

Firebase Storage

手勢放大縮小旋轉

CoreData

檔案 APP

iCloud

檔案存取

照片與相機

macOS 程式

Apple Map

其他