



KKBOX iOS/Mac OS X 基本開發教材

一份讓新成員可以進入 KKBOX 開發維護工作的學習指南 —
Weizhong Yang

[Read Online](#)[Download ▾](#)

[📖 ABOUT THIS BOOK](#)[☰ TABLE OF CONTENTS](#)[ⓘ DETAILS](#)

★ Star

165

[🔔 Subscribe](#)

這份教材是為了 KKBOX iOS/Mac OS X 開發部門的新人訓練所設計，目的是培養 出可以開發、維護 KKBOX 的 iOS 與 Mac OS X 版本，以及我們其他軟體產品的 工程師。

寫給誰的？

這不太算是一本入門的教材。編寫這份教材的時候，我們假設的讀者是已經寫過 半年左右的 iOS 程式，甚至有一兩個 App 在架上，因為畢竟是為了新人訓練所 設計，要我們雇用完全沒有經驗的 iOS 工程師，我想也很困難（其實也有這樣 的案例，但真的為數不多）。所以我們假設在使用這份教材之前：

1. 你已經會操作 Mac 電腦，也知道怎麼安裝 Xcode
2. 你已經知道一些 Objective-C 語法
3. 你已經知道一些 Foundation 物件怎麼使用
4. 你已經知道怎樣使用一些 UIKit 物件
5. 你會使用 Quartz 2D 畫圖
6. 你知道怎麼處理在 iOS 裝置上實機執行的 certificate 與 provision profile

你可以期待在這份教材看到什麼？

這份教材的主要方向是把一些 iOS 工程師天天都在使用，但往往模模糊糊懵懵 懂懂的觀念說清楚，首先從 Objective-C 這門語言是怎麼運作講起，再進一步 講解在 iOS 與 Mac OS X 裝置上的 GUI App 如何運作，中間也會帶過一些重要 的 Design Pattern，在閱讀的過程中再搭配實際練習，透過 coding 實際體會 這些重要概念，最後具備有 coding，以及能夠清楚解決各種問題的能力。

我們所重視的不是如何快速上手，不是如何用 CocoaPods 或 Carthage 的套件 拼出一個 App，而是偏重由下而上

（Bottom-up）的學習：先了解整個開發框架 的底層，以及整個框架的基本觀念，然後才去一個個去看在這個框架中有哪些 API、以及有哪些第三方 library 可以使用。

這和我們的工作型態有關，在一些以專案為主的公司裡頭，可能注重的是如何快速完成專案、如何以最快的速度完成新 App，交付 App 之後就不再維護。但 KKBOX 算是一個有一些年紀的產品，我們在 2008 年九月推出第一版 Mac OS X 版本，在 2009 年一月推出第一版 iOS 版本，一路從 Mac OS X 10.4、iPhoneOS 2 的時代寫到現在，我們會花上許多時間解決、並且避免軟體中出現 的問題。而要對系統到底怎麼運作要有一定的認識，才能夠知道怎樣閱讀 crash report，知道怎麼辨識問題並修正。

iOS 的技術不斷變動，我們也多從不太會變動的基礎概念著手。不過，我們也會 按照 KKBOX 自己的需要以及技術的改變，隨時擴充或改變這份教材的內容。

同時，不同於坊間大多數的 iOS 開發書籍，因為 KKBOX 同時有 iOS 與 Mac OS X 的版本，在這份教材中，我們會同時講到 iOS 與 Mac OS X，但如果同一個重要觀念同時出現在 iOS 與 Mac OS X 中，會以 iOS 為主。

在程式語言上，我們先講解 Objective-C，因為我們的主要產品大部分的 code 還是用 Objective-C 寫成的，只有使用少量的 Swift。而就我們的經驗，學過 Objective-C 之後，（只要搞定了 Optional 這個語言特性）轉換到 Swift 上 也不會花太多的時間。而且，就算先學了 Swift 語法，還是得搞清楚 selector、protocol、delegate 等，而這些概念 Swift 與 Objective-C 還是互通的。

在各種 API 與 library 的介紹上，也以 KKBOX 產品中會用到的為主，像我們 在這份教材中，會打算講 Audio 相關的部份，絕大多數的 iOS 開發者可能並不需要知道如何在 iOS 上處理 Audio，但 KKBOX 是一家做音樂服務的公司。至於 像遊戲開發等，雖然我們之前也用過像 Cocos2D 這些功能做過一些小專案，但 不會在這邊佔上篇幅，畢竟這份教材完全是為了 KKBOX 的需要而打造。

其實在這份教材中絕大多數材料，都可以在蘋果文件與 WWDC 影片中找到，其實 蘋果官方的文件比絕大多數的書籍都還要好，但這些資料非常多，當你想直接從 蘋果文件學習開發的時候，這份教材也提供一份怎樣在文件之海中探索的方向。

雖然在敘述上，我們只能夠按照章節順序排列，但是在整個 Cocoa Framework 中，許多觀念其實互為因果或是互相糾纏，所以在某個章節中，可能會事先講到跟後面章節有關的事情，但是先不要介意，如果遇到這樣的狀況，我們會在後面繼續說清楚。

↔ About

💬 [Discussions](#)

[ABOUT](#) [BLOG](#) [CONTACT](#) [PRICING](#) [EDITOR](#) [MORE](#)