

```

135     }
136
137     override var representedObject: Any? {
138         didSet {
139             // Update the view, if already loaded.
140         }
141     }
142
143     func checkNumber(LabelText: String) -> String {
144         var stringOfNumber: NSNumber
145         stringOfNumber = formatter.number(from: LabelText)!
146         return String(describing: stringOfNumber)
147     }
148 }
    
```

完成後，按下執行，便可以順利地在 OS X 上執行計算器。圖 20-12 是此計算器的初始畫面。



圖 20-12

21

CHAPTER

在 iOS 裝置上製作隨機顯示圖片的 App

本章仿照第 19 章來建立一個新的 iOS 專案，命名為 NextRandomImage。此 App 的概念是，按下按鈕後，畫面上的圖片會隨機換成儲存在 App 中圖片集的其中一張，另外也會提示出，目前顯示的圖片是哪一張圖片。這個簡單的 App 會用到以下幾種 UI 元件：Navigation Bar、Image View、Button、Label。

21.1 UI 設計

圖 21-1 是這個 App 大略的 UI 設計，您不必按照此元件排列方式，大可發揮您的美學佈局。

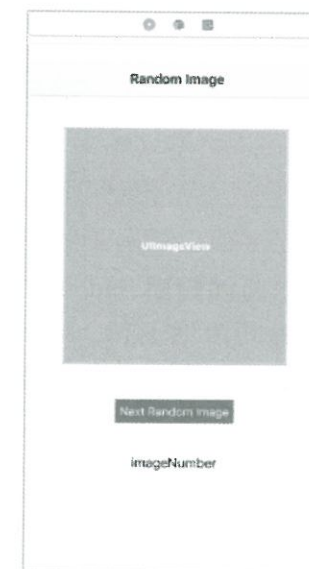


圖 21-1

建立好 NextRandomImage 的專案後，打開 Main.storyboard，並從右下角的 Object Library 選取 Navigation Bar，並將它拖曳到 View Controller 的最上方。接著雙擊「Title」改變其顯示的文字，將它改為「Random Image」。

同樣的方法，再從 Object Library 選取 Image View、Button，以及 Label，並按照圖 21-1 所示，放置於 View Controller 適當的位置上。

點擊 Button 後，在右側的 Attributes inspector 的屬性中，設置成如圖 21-2 與圖 21-3 所示。

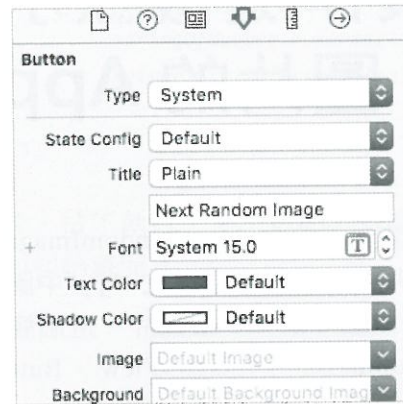


圖 21-2

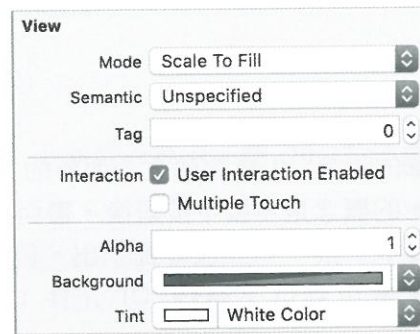


圖 21-3

接著點擊 Label，同樣於右側的 Attributes inspector 的屬性中，設置成如圖 21-4 所示。

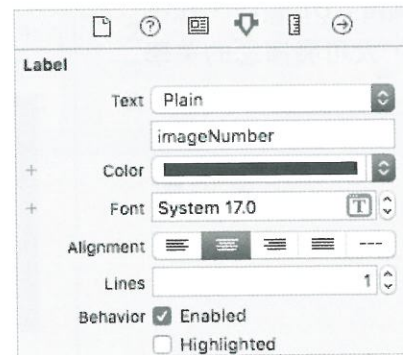


圖 21-4

21.2 撰寫此 App 的程式碼

繼續使用 Main.storyboard，在視窗的右上角點擊圖 21-5 中間的按鈕。



圖 21-5

並在左側的 storyboard 中，點擊 Image View 並且按住 control，拖曳至右側 ViewController.swift 程式碼中，如圖 21-6 所示。

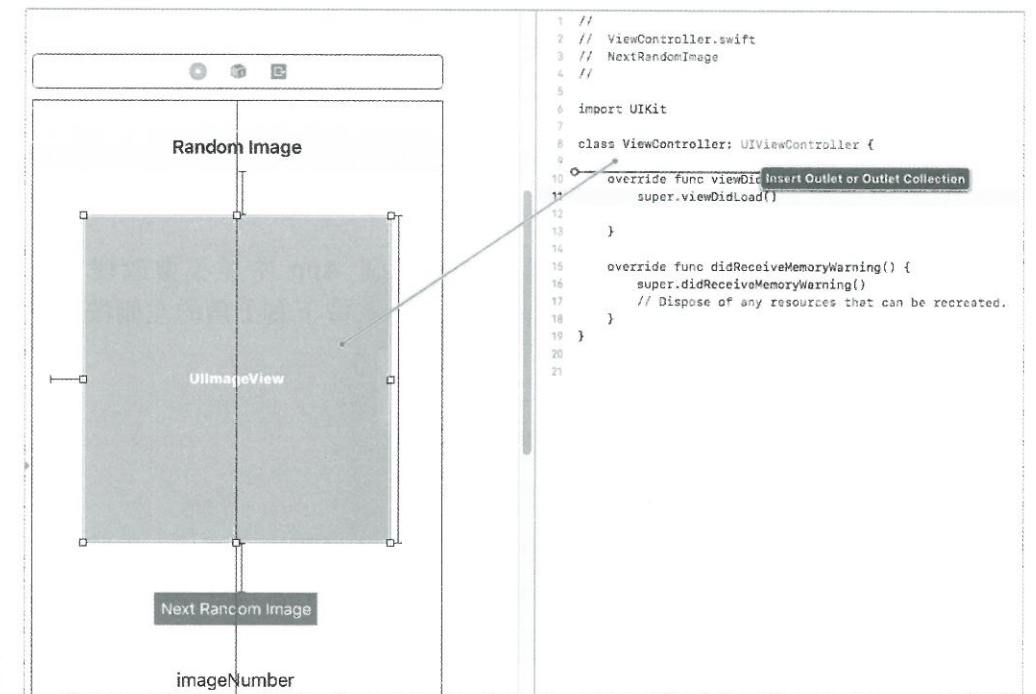


圖 21-6

並依序完成其他的 UI 元件宣告，拖曳 imageNumber 標籤的做法和上述 Image View 的做法相同，但要注意再拖曳 Button 時，所做的設定要將 Connection 的選項改為「Action」，如圖 21-7 所示。

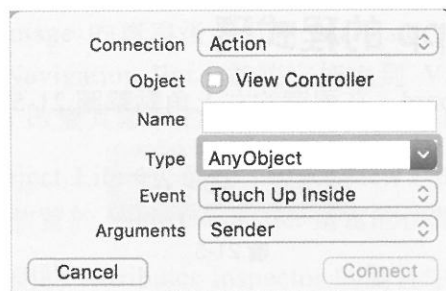


圖 21-7

在宣告時可以自訂您的 UI 元件變數名稱 (Name)，如以下分別取的名稱 imageView、imageNum，以及 randomBtn。完成後如下列的程式碼：

```
@IBOutlet var imageView: UIImageView!
@IBOutlet var imageNum: UILabel!
@IBAction func randomBtn(_ sender: AnyObject) {
}
```

不需宣告 Navigation Bar 的變數，是因為在這個 App 中並不會改變它。到此 App 所有用到的 UI 元件變數就宣告完成了。按下右上角的左側按鈕來恢復顯示單一檔案視窗，如圖 21-8 所示。



圖 21-8

接著在左側的導覽列中打開 Assets.xcassets，準備五張圖片，並一一地將圖片拖曳到 Assets.xcassets 右側的 appicon 的空白處，再把圖片依序命名為「image01」、「image02」、「image03」、「image04」，以及「image05」，完成後的畫面如圖 21-9 所示。

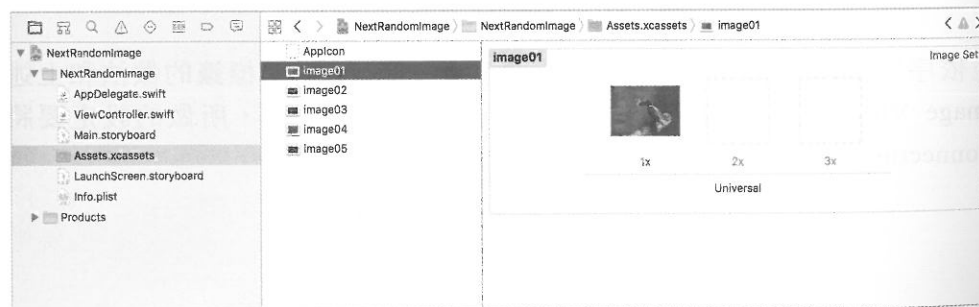


圖 21-9

最後在左側的導覽列中打開 ViewController.swift，開始撰寫其餘的程式碼。首先，先宣告要使用的圖片名稱，輸入下列的程式碼：

```
let imageArray = ["image01", "image02", "image03", "image04", "image05"]
```

之後隨機產生圖片，表示在字串陣列中隨機挑選一個字串，再將 Image View 顯示的圖片改變。

接著在 viewDidLoad() 中輸入下列的程式碼：

```
imageView.image = UIImage(named: imageArray[0])
imageNum.text = imageArray[0]
```

第一行程式碼表示將第一張圖片指定給 Image View，而第二行程式碼是將第一張圖片的名稱指定給 Label 顯示。

現在可以按下左上角的執行按鈕，看看目前的程式碼的執行結果，結果如圖 21-10 所示。

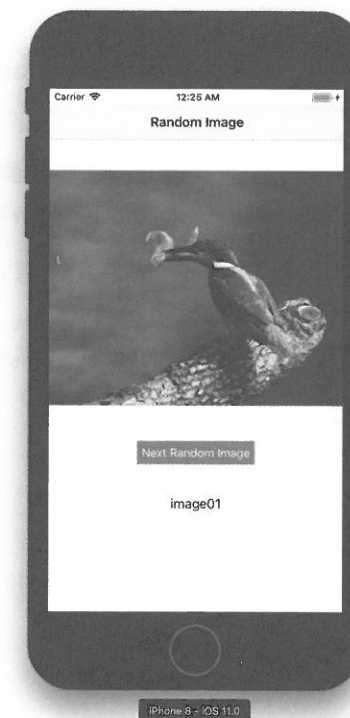


圖 21-10

圖片已經可以正常顯示了，但只有一張，而且 Next Random Image 的按鈕仍沒有任何作用。

現在回到 ViewController.swift，在 randomBtn(_ sender: AnyObject) 中加入以下程式碼：

```
let itemInArray = imageArray.count
let random = Int(arc4random_uniform(UInt32(itemInArray)))
imageView.image = UIImage(named: imageArray[random])
imageNum.text = imageArray[random]
```

第 1 行程式碼是計算 imageArray 中的元件數量，並將它指定給常數 itemInArray，第 2 行程式碼是以第 1 行程式碼的結果，使用 arc4random_uniform 在 0 到 imageArray-1 的數字間，取一隨機亂數並指定給常數 random。由於 arc4random_uniform 接收的參數為 UInt32 型態，所以需要先經過型態轉換，而且回傳值也為是 UInt32 型態，為了之後變數的使用，在此也將它的型態轉為 Int。

想了解更多 arc4random_uniform 的使用方法，可參考第 4 章 4.3 節，或透過 Xcode 查詢文件的功能，按住 control 並點擊 arc4random_uniform 來獲得更進一步的資訊。

而第 3、4 行的程式碼與之前的程式碼相似，是將得到的亂數帶進陣列中，並將其元件指定給 imageView 的 image 以及 imageNum 的 text。

現在可以再執行一次 App，可發現按鈕已經可以發揮它的作用了。

完整的 ViewController.swift 程式碼如下：

```
01 //
02 // ViewController.swift
03 // NextRandomImage
04 //
05
06 import UIKit
07
08 class ViewController: UIViewController {
09     // 宣告UI 元件
10     @IBOutlet var imageView: UIImageView!
11     @IBOutlet var imageNum: UILabel!
12     @IBAction func randomBtn(_ sender: AnyObject) {
```

```
13     // 設定按下按鈕後隨機改變圖片
14     // 取得陣列裡的字串數量
15     let itemInArray = imageArray.count
16
17     // 在取得的數量之間產生亂數
18     let random = Int(arc4random_uniform(UInt32(itemInArray)))
19
20     // 改變圖片與下方的圖片名稱
21     imageView.image = UIImage(named: imageArray[random])
22     imageNum.text = imageArray[random]
23 }
24
25 // 宣告使用到的圖片名稱
26 let imageArray = ["image01", "image02", "image03", "image04", "image05"]
27
28 override func viewDidLoad() {
29     super.viewDidLoad()
30
31     // 載入程式後顯示預設的圖片
32     imageView.image = UIImage(named: imageArray[0])
33
34     // 載入預設圖片的名稱
35     imageNum.text = imageArray[0]
36 }
37
38 override func didReceiveMemoryWarning() {
39     super.didReceiveMemoryWarning()
40     // Dispose of any resources that can be recreated.
41 }
42 }
```

行文至此，這不是終點，而是另一階段的起點，盼望各位讀者有了初步的體驗後，能進一步的實作其他實用的 App，為自己創造更美好的明天。祝福您。

學會 Swift 4 程式設計的 21 堂課

作者：蔡明志
企劃編輯：蔡彤孟
文字編輯：詹祐甯
設計裝幀：張寶莉
發行人：廖文良

發行所：基峰資訊股份有限公司
地址：台北市南港區三重路 66 號 7 樓之 6
電話：(02)2788-2408
傳真：(02)8192-4433
網站：www.gotop.com.tw
書號：ACL052600
版次：2018 年 01 月初版
建議售價：NT\$450

商標聲明：本書所引用之國內外公司各商標、商品名稱、網站畫面，其權利分屬合法註冊公司所有，絕無侵權之意，特此聲明。

版權聲明：本著作物內容僅授權合法持有本書之讀者學習所用，非經本書作者或基峰資訊股份有限公司正式授權，不得以任何形式複製、抄襲、轉載或透過網路散佈其內容。
版權所有 • 翻印必究

國家圖書館出版品預行編目資料

學會 Swift 4 程式設計的 21 堂課 / 蔡明志著. -- 初版. -- 臺北市：
基峰資訊, 2018.01
面；公分
ISBN 978-986-476-723-6(平裝)
1.電腦程式語言 2.物件導向程式
312.2 107000498

讀者服務

- 感謝您購買基峰圖書，如果您對本書的內容或表達上有不清楚的地方或其他建議，請至基峰網站：「聯絡我們」\「圖書問題」留下您所購買之書籍及問題。(請註明購買書籍之書號及書名，以及問題頁數，以便能儘快為您處理)
<http://www.gotop.com.tw>
- 售後服務僅限書籍本身內容，若是軟、硬體問題，請您直接與軟體廠商聯絡。
- 若於購買書籍後發現有破損、缺頁、裝訂錯誤之問題，請直接將書寄回更換，並註明您的姓名、連絡電話及地址，將有專人與您連絡補寄商品。
- 歡迎至基峰購物網
<http://shopping.gotop.com.tw>
選購所需產品。

ISBN 978-986-476-723-6



ACL052600 NT\$450