

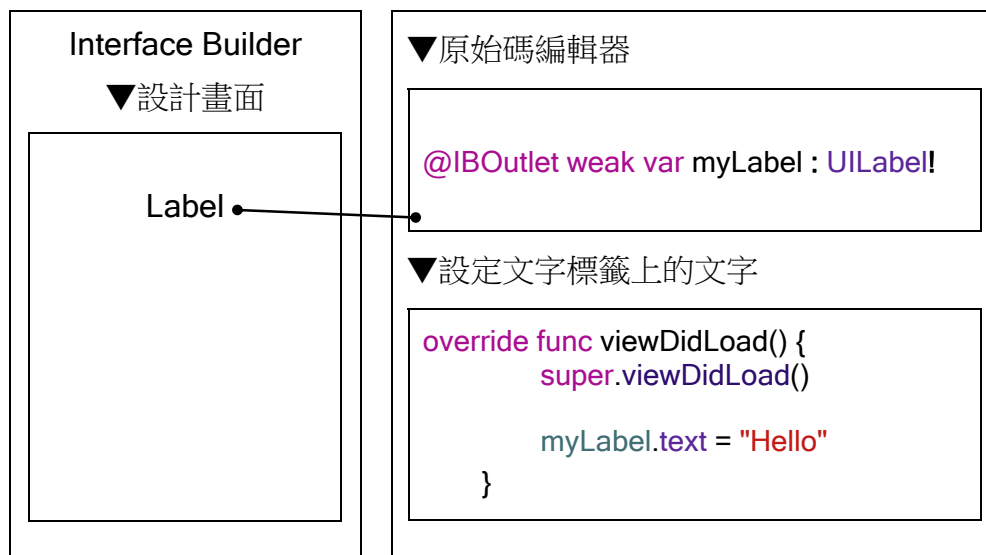
## CHAPTER 5-1

### UILabel：顯示簡短訊息的文字標籤

#### 在文字標籤中顯示文字

學習概念：

1. 首先用 IB 建立【文字標籤】。
2. 接著將【文字標籤】與【程式碼連結】。
3. 最後在實作檔中實作相關程式，在處理畫面載入後所觸發的事件，也就是【ViewDidLoad】方法。
- 4.

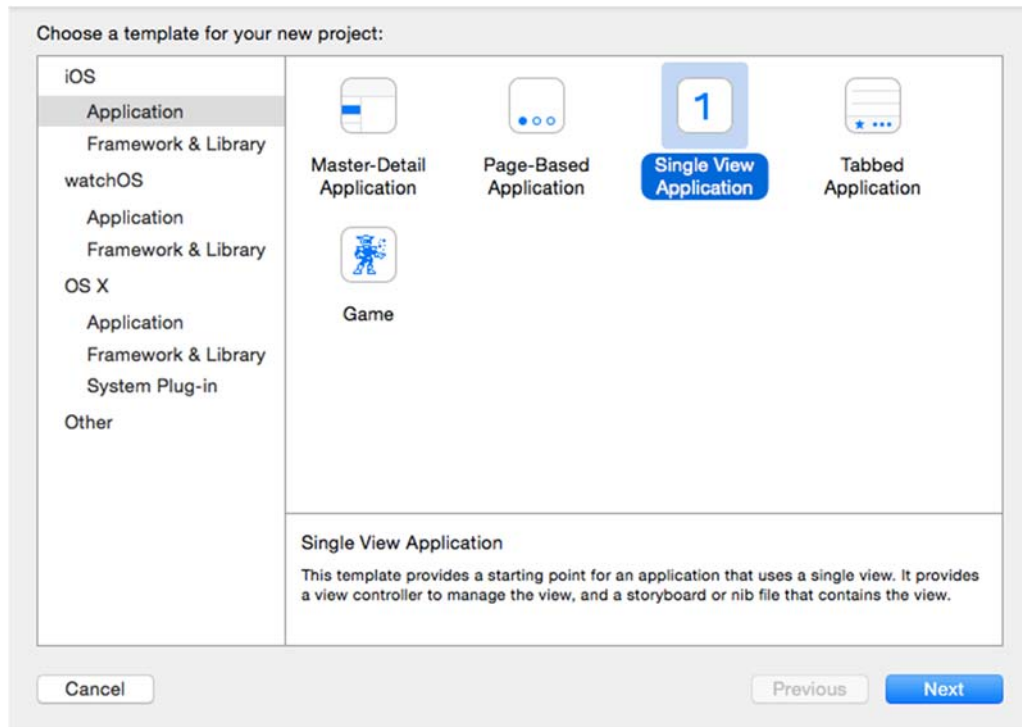


#### 【執行結果】

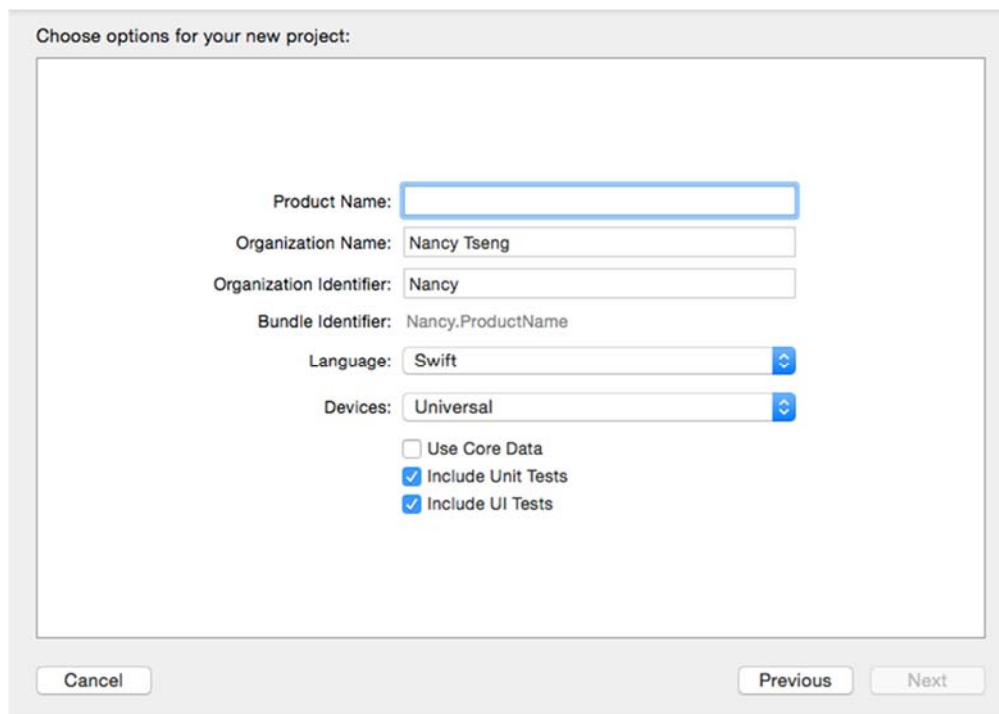
當 App 執行後，設定在【文字標籤】上的文字，將顯示在【myLabel】。

## Step.1

開啟 **xcode** 時會出現的畫面，點選 **iOS** 下的【**Application**】，接著右視窗選擇【**Single View Application**】，點選【**Next**】選項後進入設定的基本視窗。

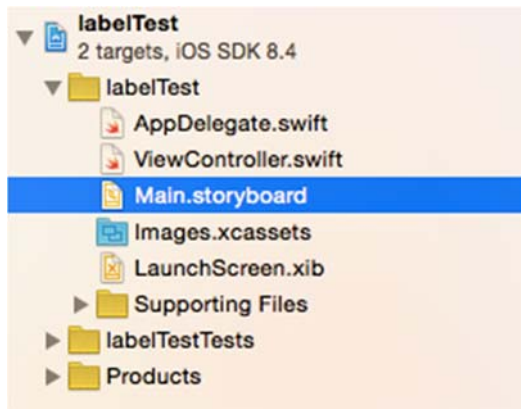


檔名及名稱設定，請將【**Product Name**】設定為 **xxxxxxxx**



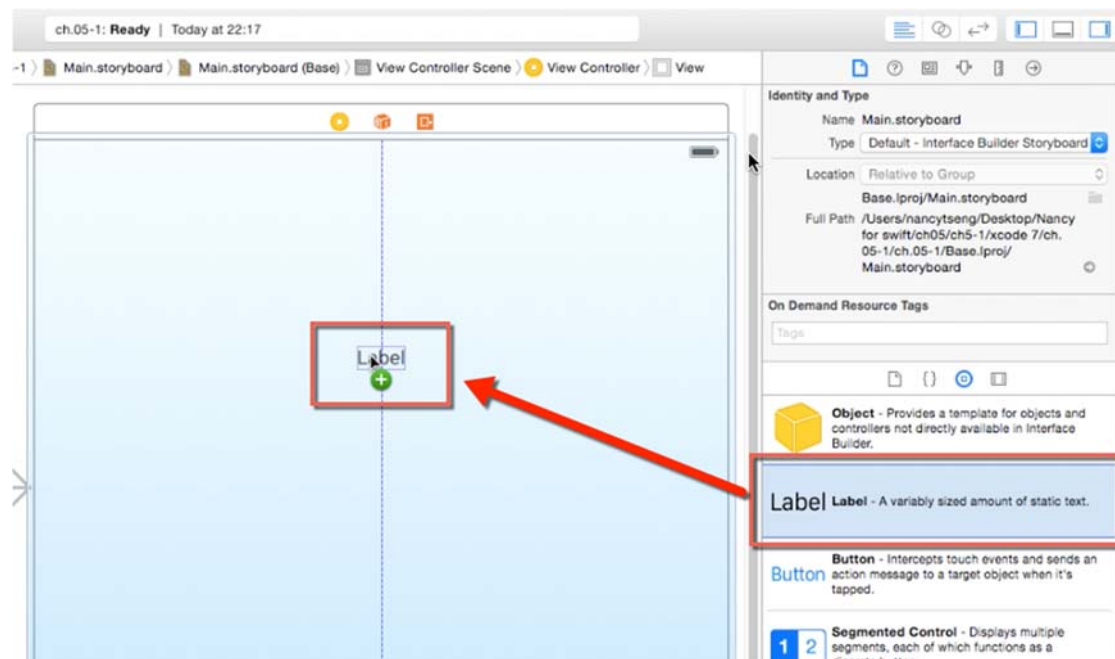
## Step.2

選取【Main.storyboard】



從【物件庫】中拖曳【Label】到畫面中。

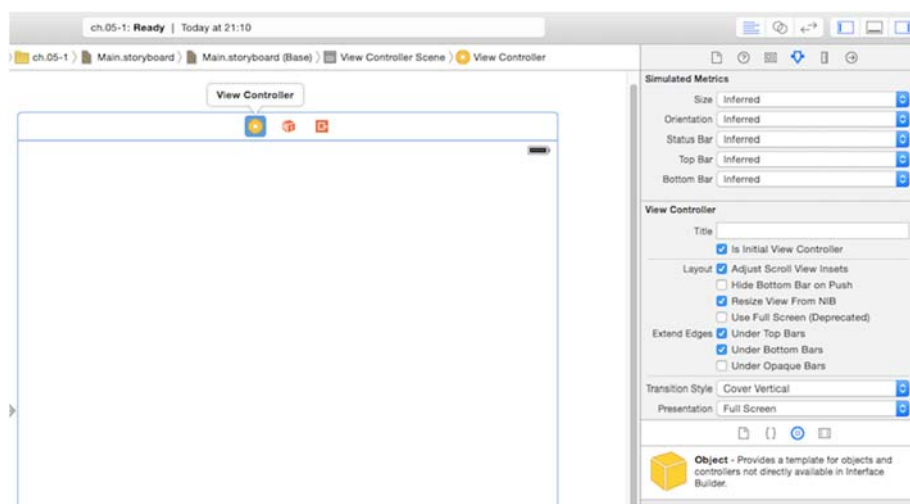
按住【control】鍵，將【Label】拖曳到右邊視窗產生一個【文字標籤】。



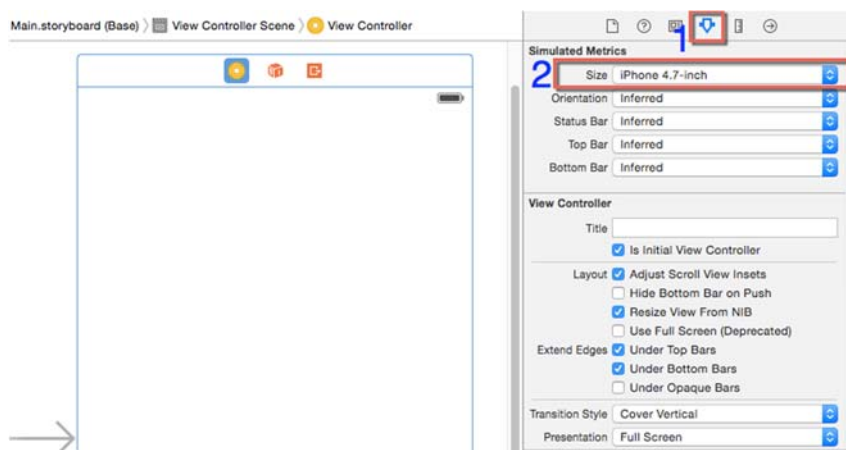


## 屬性設定小技巧

1. 在還沒在進行手機尺寸設定時，建議讀者可以透過選項完成設定。
2. 於【設計畫面】中，點選【View Controller】，就會出現【Simulated Metrics】進行手機尺寸設定。



3. 接著點選【方框 1】後，再選擇【方框 2】設定手機尺寸大小。
4. 這裡建議初學讀者，選擇【iPhone 4.7-inch】較為容易操作。



### Step.3.

接著點選右上方工具列視窗【輔助編碼器】，就是雙圈符號【2】，進行程式碼編輯。



### 屬性設定小技巧

各選項名稱與進階說明：

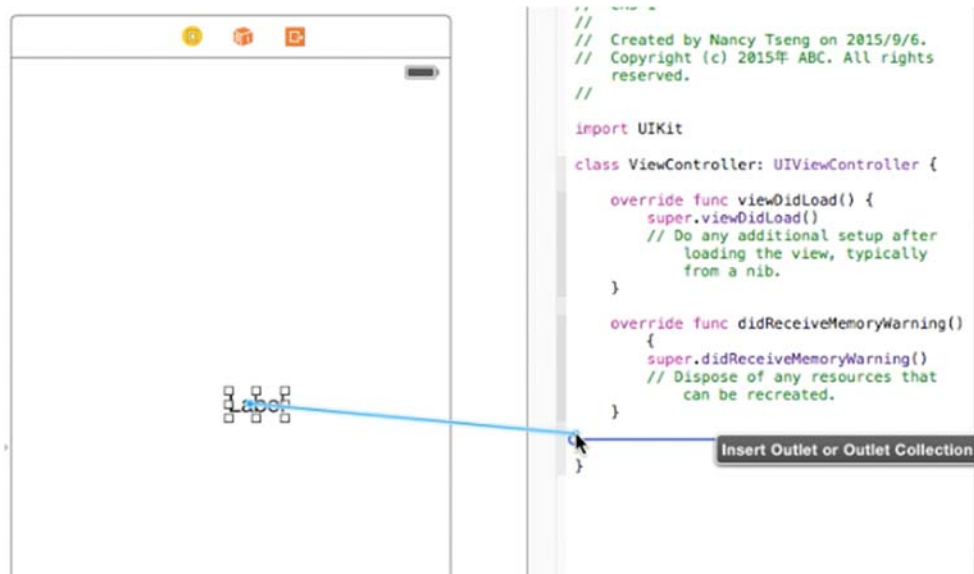
1. Show the Standard editor：顯示標準視窗
2. Show the Assistant editor：輔助編碼器
3. Show the Version editor：歷史編輯記錄比較
4. Hide or show the Navigator：隱藏或顯示導覽區
5. Hide or show the Debug area：隱藏或顯示除錯區
6. Hide or show the Utilities：隱藏或顯示工具區



在【選項 2.】會將編輯切割成 2 個編輯器，在撰寫程式上較為便利。

### Step.4

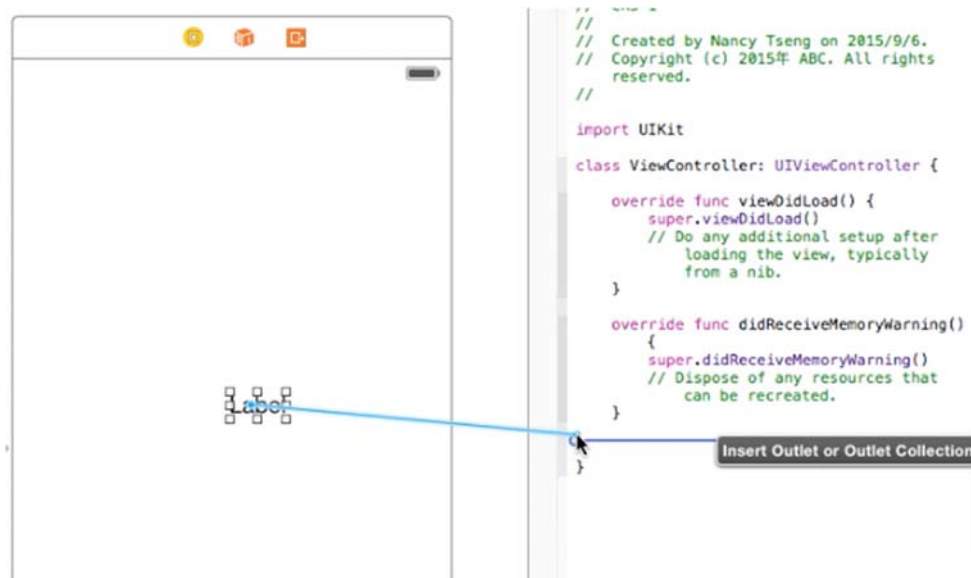
- (1) 將【文字標籤】和【名稱連結】。
- (2) 在 IB 視窗上點選【文字標籤】按住滑鼠，拖曳到右邊視窗與【程式碼】連結。



## Step.5

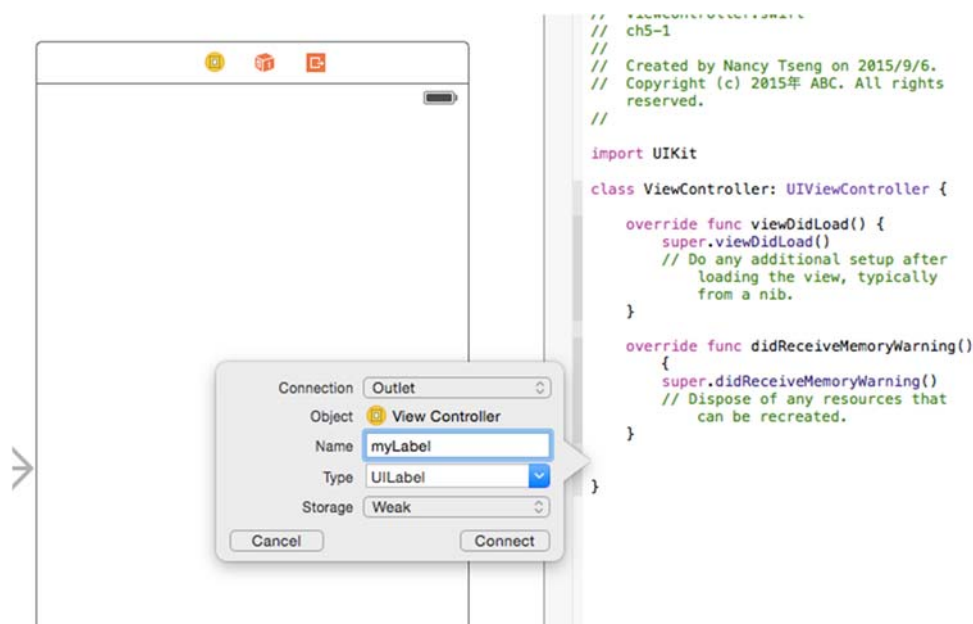
將【文字標籤】和【名稱連結】。

在 IB 視窗上點選【文字標籤】按住滑鼠，拖曳到右邊視窗與【程式碼】連結。



## Step.6.

拖曳後【文字標籤】會彈出視窗，將【Name】命名為【myLabel】按下【Connect】鍵。



透過連結即產生紅框處的程式碼，控制 IB 建立的元件。

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.

    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBOutlet weak var myLabel: UILabel!
}
```

## Step.7.

讓【文字標籤】顯示” Hello” 。

於 viewDidLoad() 函式區段中，加入 myLabel.text = “Hello”

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
        myLabel.text = "Hello";
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBOutlet weak var myLabel: UILabel!
}
```



## 屬性設定小技巧

Swift 語言教學：

### 【設定文字內容】

透過文字標籤 `text` 屬性，顯示內容設定：

```
文字標籤.text = “文字” ;
```

範例：`myLabel.text = “Hello” ;`

## Step.8.

在上方工具列按下【執行鍵▶】（Build and then run the current scheme），啟動模擬器執行程式。

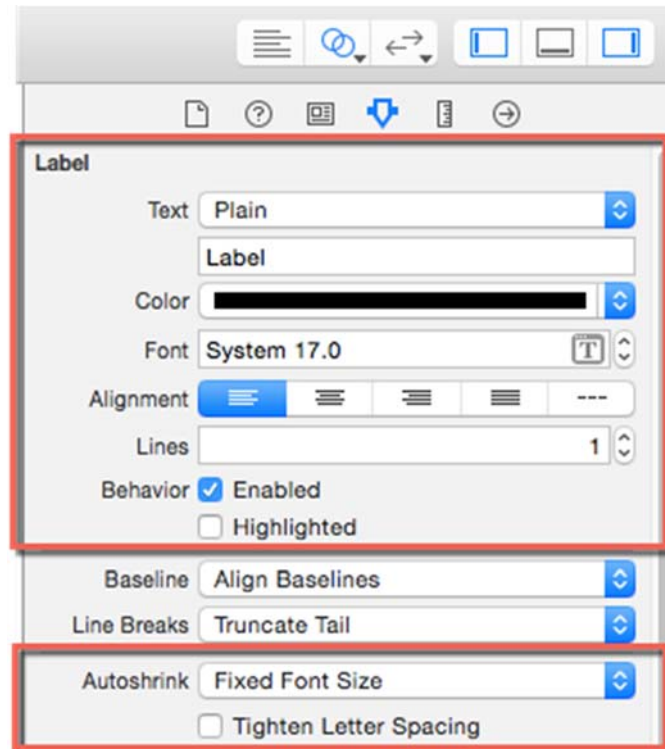






## 屬性設定小技巧

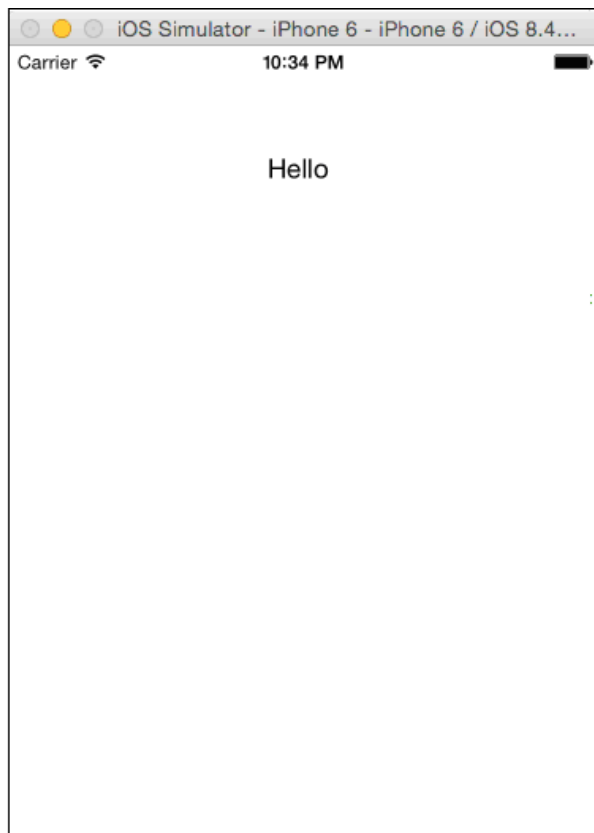
在經過了【文字標籤】連結設定後，透過直接選項讓【文字標籤】除了基本的設定之外更加活潑生動，增進讀者們的功力：



選 項		功 能 說 明
1.	Text	文字內容
2.	Label	要顯示的文字
3.	Color	文字顏色
4.	Font	字型及字體大小
5.	Alignment	文字對齊方式
6.	Lines	最大行數
7.	Autoshrink	文字超出顯示範圍時的處理方式

## Step.9

當 App 啟動後在顯示畫面時，將【myLabel】的顯示文字設定為「Hello」。



（小編在這裡設定為 iPhone6 畫面，供各位讀者們參考）

# 自我練習

## 實作：

於畫面中增加四個標籤，  
第一個使用預設文字方式  
第二個調整文字的顏色  
第三個調整字體為粗體字  
第四個使用動態方式設定文字內容  
（設定畫面如右圖）



## 執行後結果

