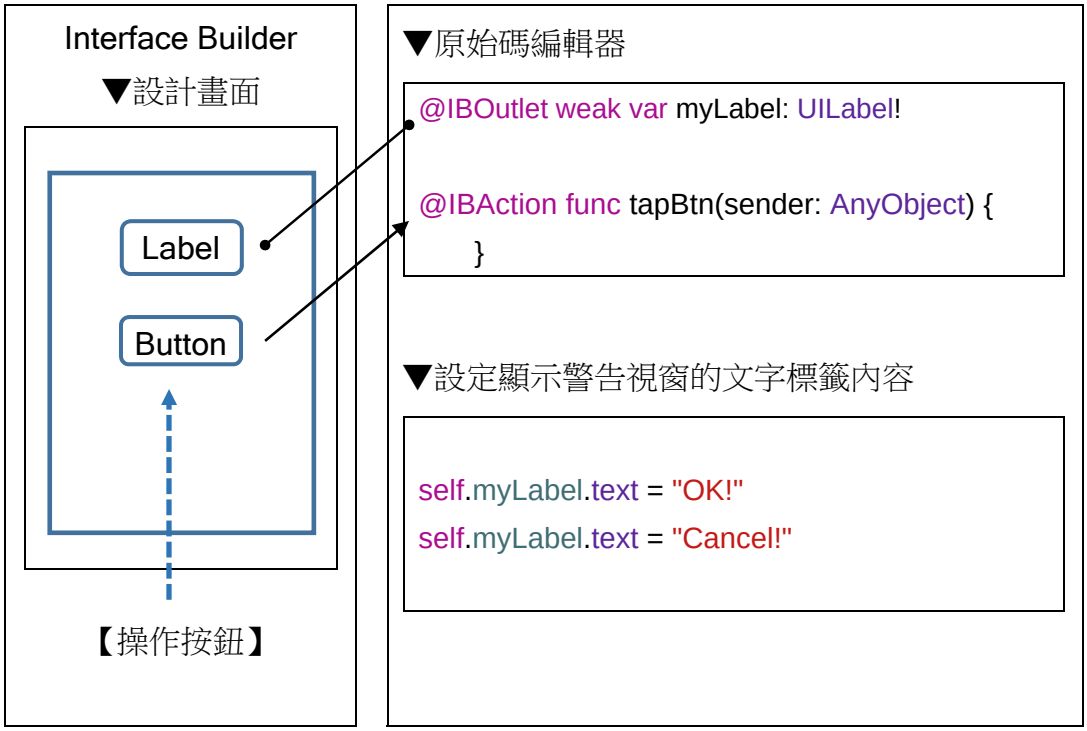
CHAPTER 5-8

UIAlertView：彈出訊息視窗

透過操作按鈕顯示警告訊息

學習概念：

1. 首先用 IB 建立【文字標籤】及【操作按鈕】。
2. 將【文字標籤】及【操作按鈕】與【程式碼】連結。
3. 最後在實作檔中相關程式，於處理載入後所觸發的事件，也就是撰寫利用【操作按鈕】將結果顯示在【文字標籤】上的程式。

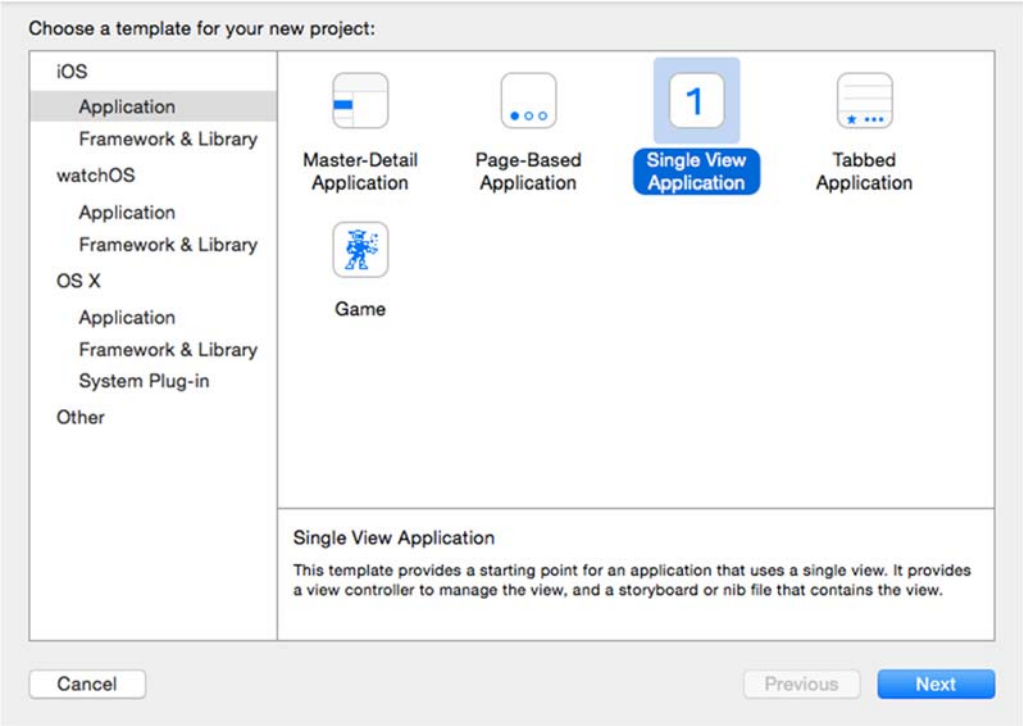


【執行結果】

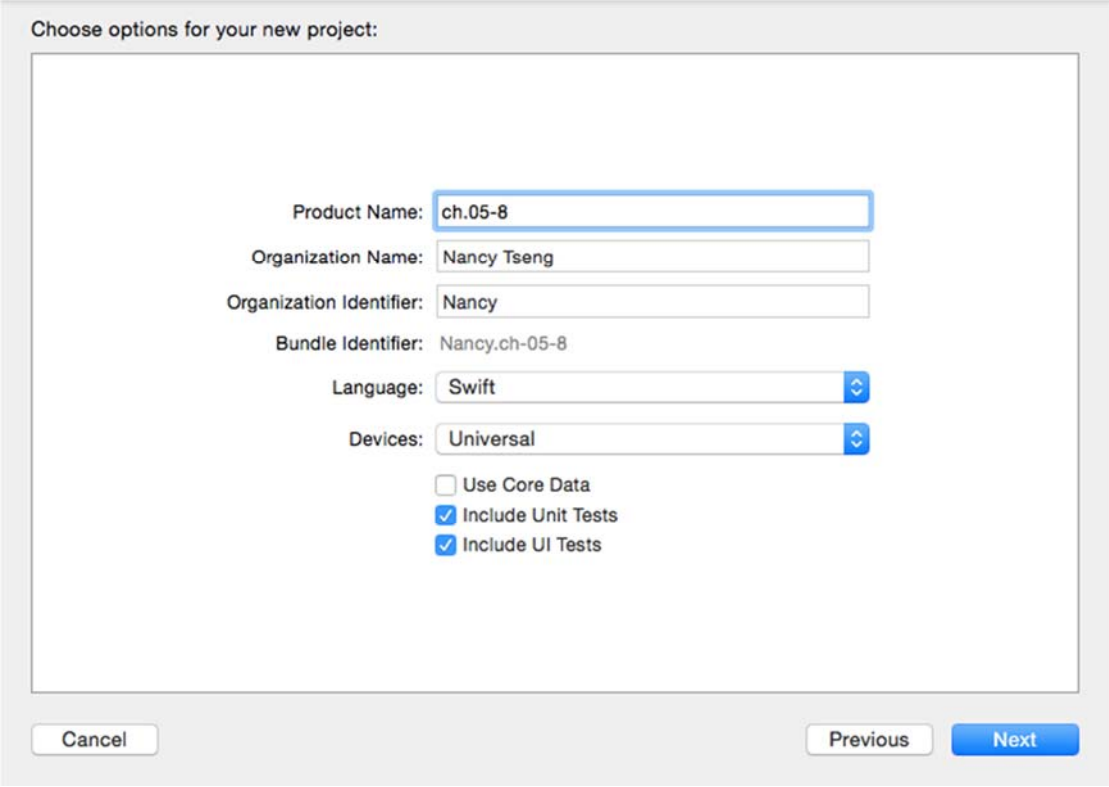
當 App 執行後，操作者在點選【操作按鈕】時會將結果，顯示在【文字標籤】中。

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。

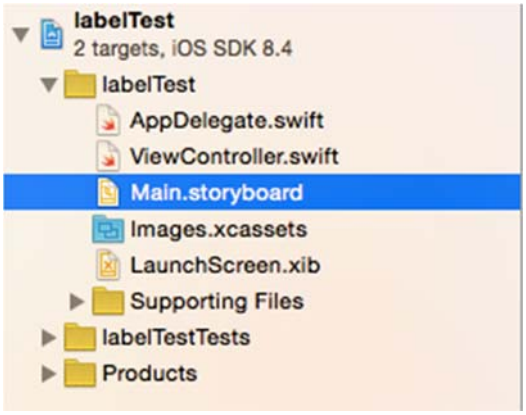


檔名及名稱設定，請將【Product Name】設定為 ch.05-8

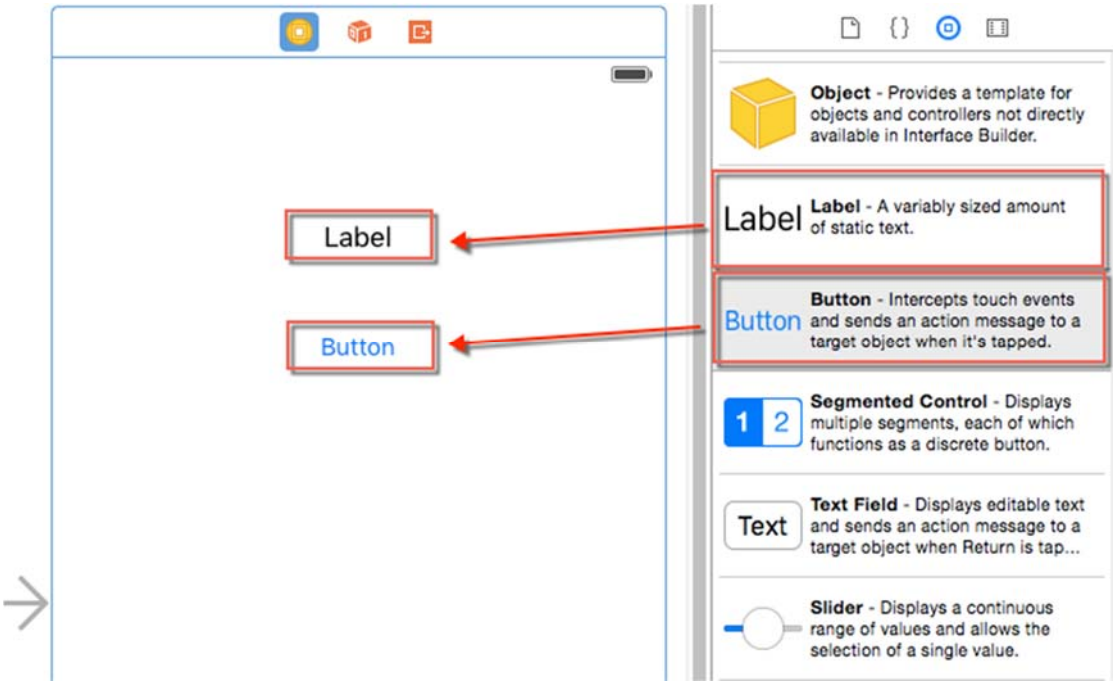


Step.2

選取【Main.storboard】



本章操作已選擇【iPhone 4.7-inch】 操作頁面。(詳見 5-1 屬性設定小技巧)
從【物件區】中拖曳【Label】及【Button】到 IB 畫面中。



Step.3.

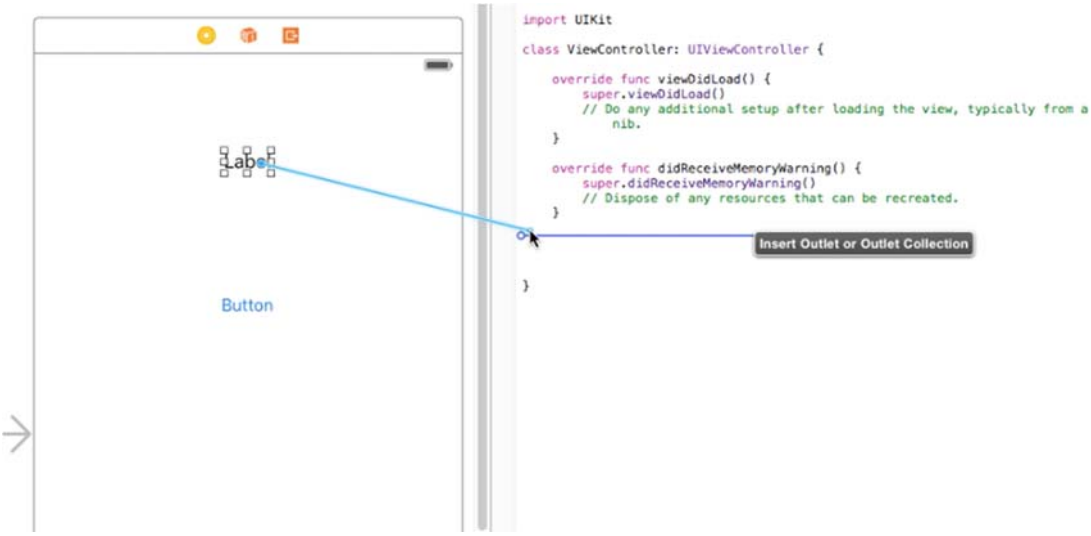
接著點選右上方工具列視窗【輔助編碼器】，就是雙圈符號【2】，進行程式碼編輯。

(詳見 5-1 屬性設定小技巧)

Step.4

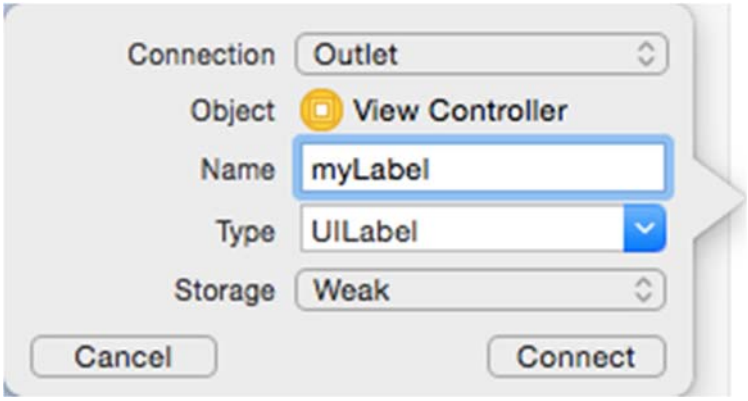
(1) 透過連結即產生的程式碼，控制 IB 建立的元件和【程式碼】連結。

(2) 將【文字標籤】「Label」與「變數名稱」連結。



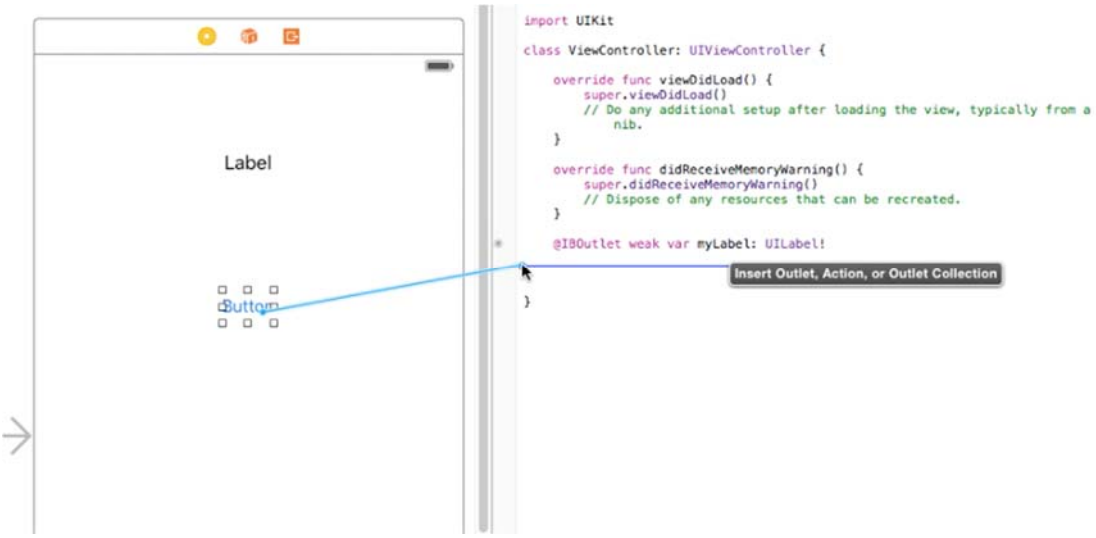
拖曳 Lable 元件

在「Connection」欄位點選【Outlet】、「Name」欄位設定名稱【myLabel】以及在「Type」欄位點選【UILabel】後，按【Connect】按鈕。



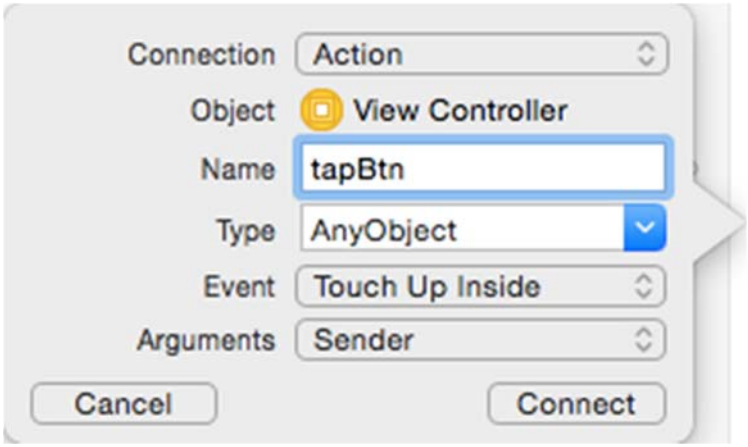
在拖曳後自動彈跳的視窗

(3) 將【操作按鈕】「Button」與「變數名稱」連結。



拖曳 Button 元件

在「Connection」欄位點選【Action】、「Name」欄位設定名稱【tapBtn】以及在「Type」欄位點選【AnyObject】後，按【Connect】按鈕。



在拖曳後自動彈跳的視窗

(4) 將會自動插入程式碼，作為與 IB 的連結。

```
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!

    @IBAction func tapBtn(sender: AnyObject) {

    }
}
```

連結後出現的程式碼。

Step.5

(1) 最後，我們在【紅框】中加入設定 swift 的程式碼，將透過串接執行程式後，將結果顯示在【文字標籤】「Label」裡。

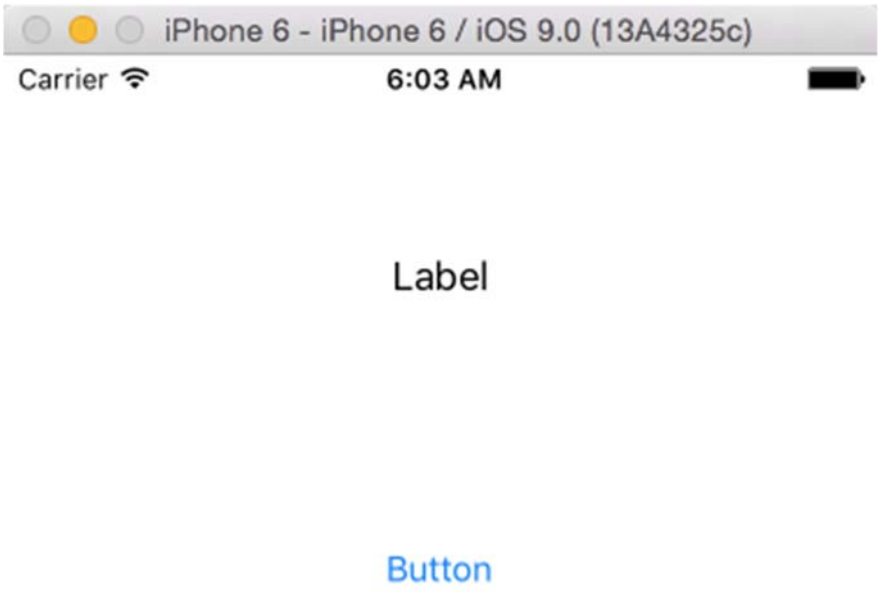
```
import UIKit

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!


    @IBAction func tapBtn(sender: AnyObject) {
        let alertController = UIAlertController (title: "Alert Test", message: "OK
            Cancel Test", preferredStyle: .Alert)

        alertController.addAction(UIAlertAction(title: "OK", style: .Default,
            handler: {action in self.myLabel.text = "OK!" }
            ))

        alertController.addAction(UIAlertAction(title: "Cancel", style: .Default,
            handler: {action in self.myLabel.text = "Cancel!" }
            ))

        presentViewController(alertController, animated: true, completion: nil)
    }
}
```

Step.6

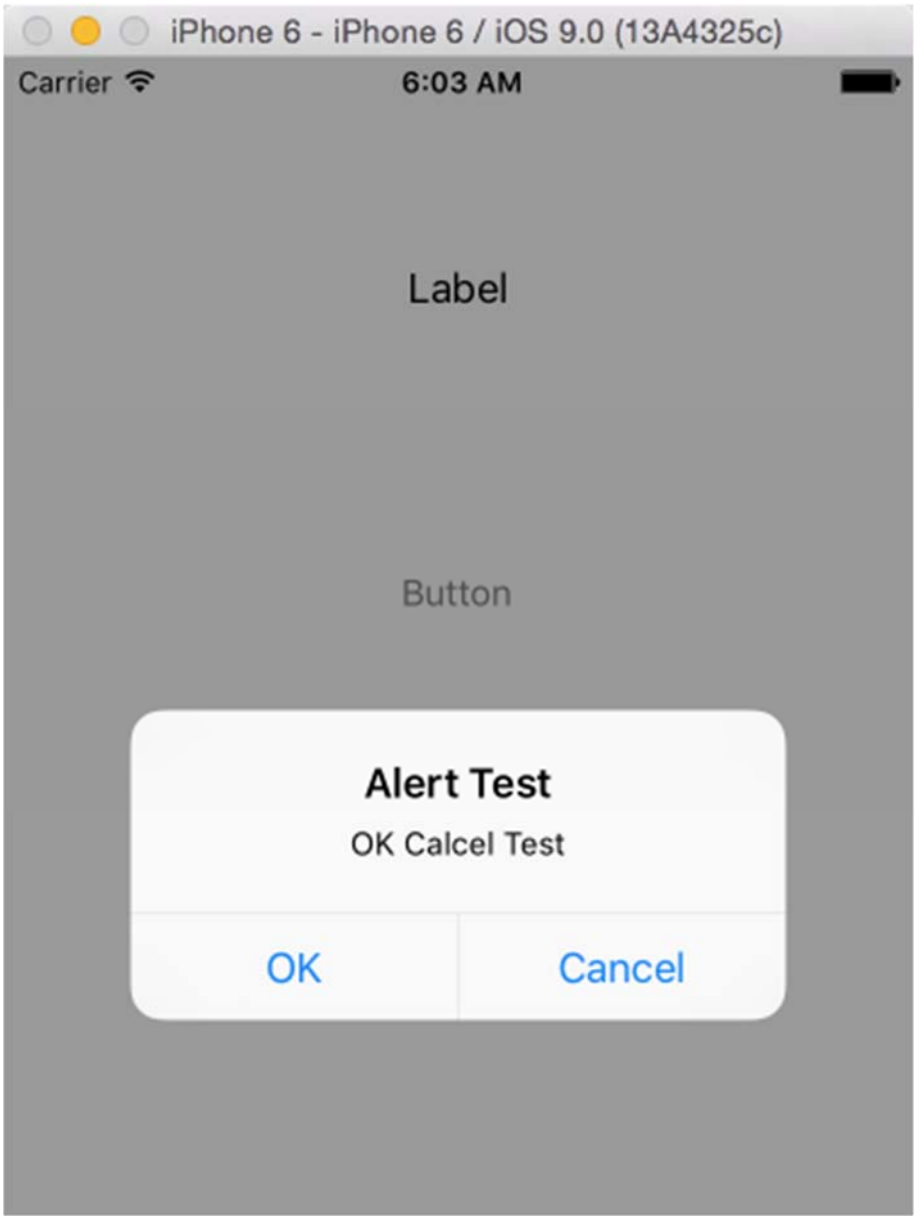
在上方工具列按下【執行鍵▶】(Build and then run the current scheme)，啟動模擬器執行程式。

Step.7

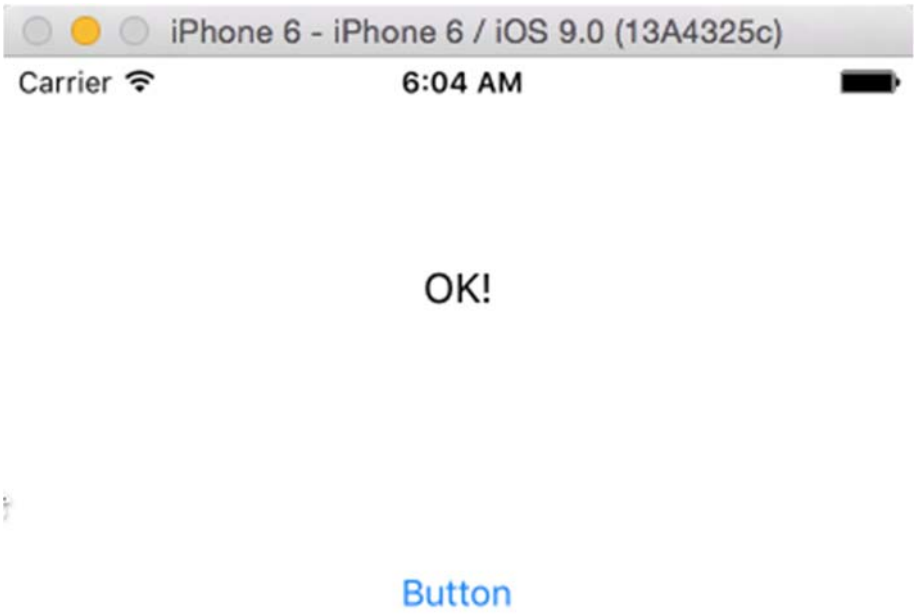
當 App 啟動後在顯示畫面時，在點選【操作按鈕】「Button」，將會自動帶出所設定的內容，顯示在【文字標籤】也就是「Label」中的數值。



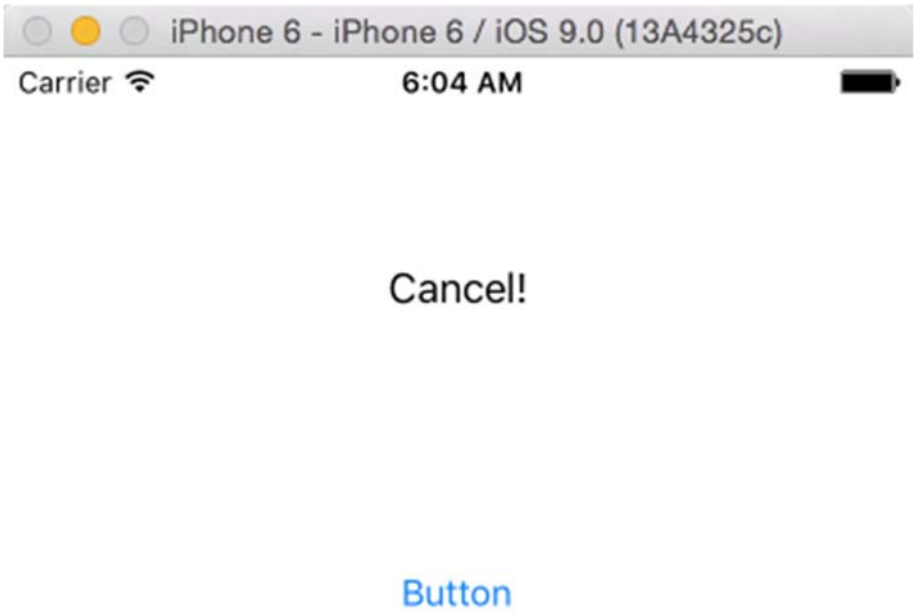
試試看按一下【操作按鈕】「Button」會有什麼事情發生～



當出現訊息視窗時，請選取其中選項「OK」或「Cancel」



當藉由【操作按鈕】「**Button**」選擇「**OK**」時，「**Label**」會跳出顯示結果為「**OK**」



又當選取【操作按鈕】「Button」選擇「Cancel」時，「Label」會跳出顯示結果為「Cancel」

（小編在這裡設定為 iPhone6 畫面，供各位讀者們參考）

活用【AlertView】進階

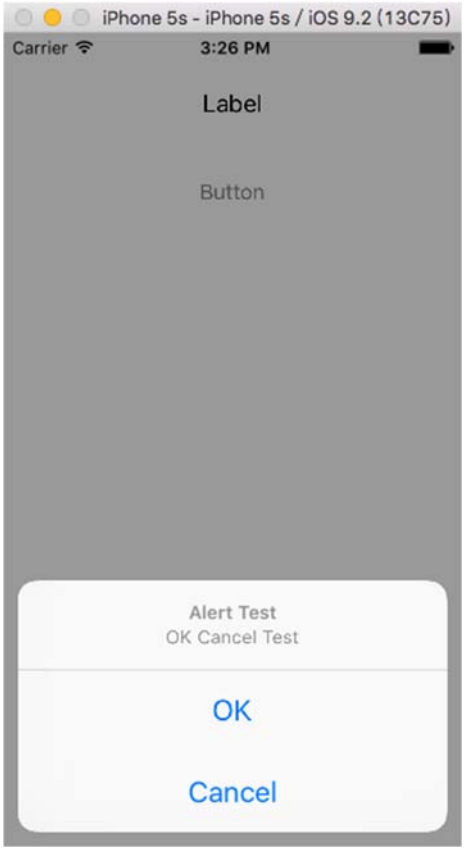
在上例中，我們使用傳統的警告視窗 （透過設定 preferredStyle: .Alert），另一種警告視窗是由下方浮現（ActionSheet），在上例中，僅需設定 preferredStyle: .ActionSheet

原先

```
let alertController = UIAlertController (title: "Alert Test", message: "OK Cancel Test", preferredStyle: .Alert)
```

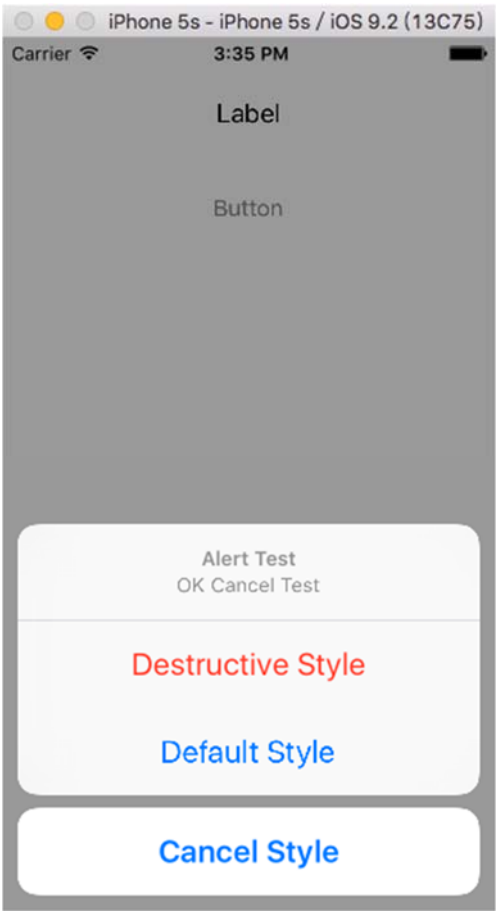
修改為

```
let alertController = UIAlertController (title: "Alert Test", message: "OK Cancel Test", preferredStyle: .ActionSheet)
```



另外，對於每個新加入的 Action 可以設定形式 style，
共有三種形式 .Default, .Cancel, 以及 .Destructive

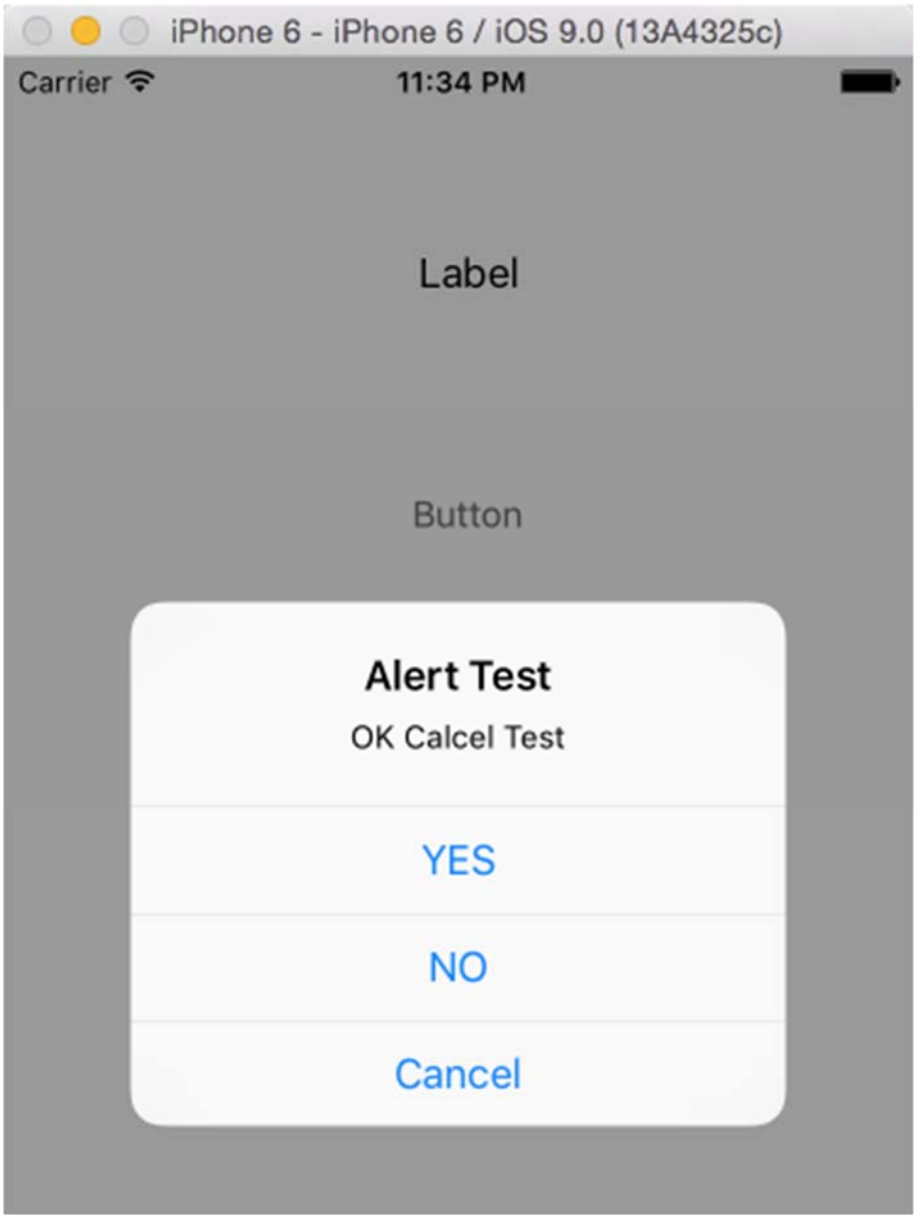
```
        alertController.addAction(UIAlertAction(title: "Destructive Style",
style: .Destructive, handler: {action in self.myLabel.text = "Destructive!" }
            ))

        alertController.addAction(UIAlertAction(title: "Default Style", style: .Default,
handler: {action in self.myLabel.text = "Default!" }
            ))

        alertController.addAction(UIAlertAction(title: "Cancel Style", style: .Cancel,
handler: {action in self.myLabel.text = "Cancel!" }
            ))
```



自 我 練 習

實作執行後結果：

利用【Label】及【Button】程式結合執行，透過【Button】指令選項讓【Label】依據我們所撰寫的內容顯，當執行後會出現選項視窗；分別顯示「YES」、「NO」以及「Cancel」這三者在與【程式碼】連結並串接時，當執行【RUN】後所顯示文字內容。



《《小編提醒您，試試看吧！》》