

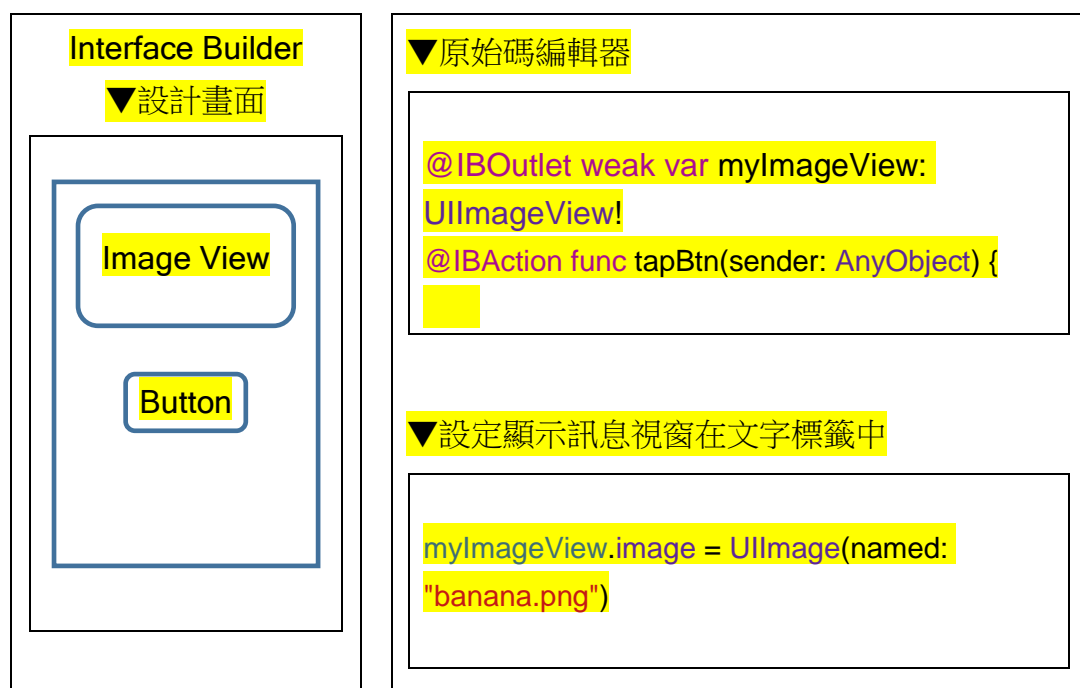
## CHAPTER 10-3

### Table View Cell：表格自訂欄位格式

#### 手動建立 Table View Cell，自訂欄位格式

學習概念：

1. 首先用 IB 建立【圖像方塊】。
2. 將【操作按鈕】及【圖像方塊】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】結合【圖像方塊】，直接顯示在【設計畫面】上的程式。

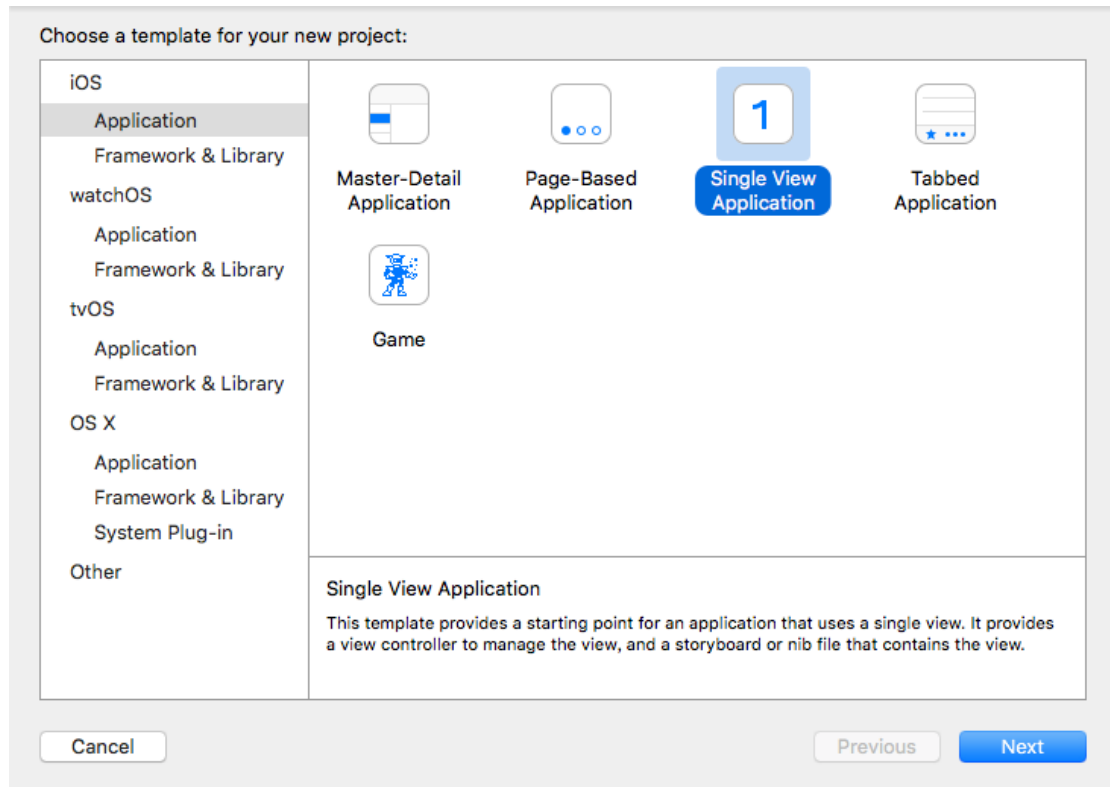


#### 【執行結果】

當 App 執行後，預先顯示 apple.png 圖片，點選【操作按鈕】後，讓【圖像方塊】顯示 banana.png 圖片在〈設計畫面〉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.10-3**

Choose options for your new project:

Product Name:

Organization Name:

Organization Identifier:

Bundle Identifier:

Language:

Devices:

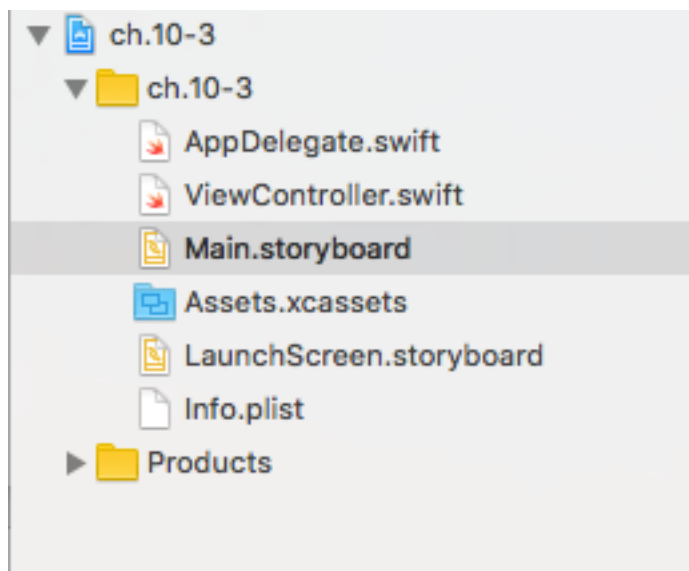
☐ Use Core Data

☐ Include Unit Tests

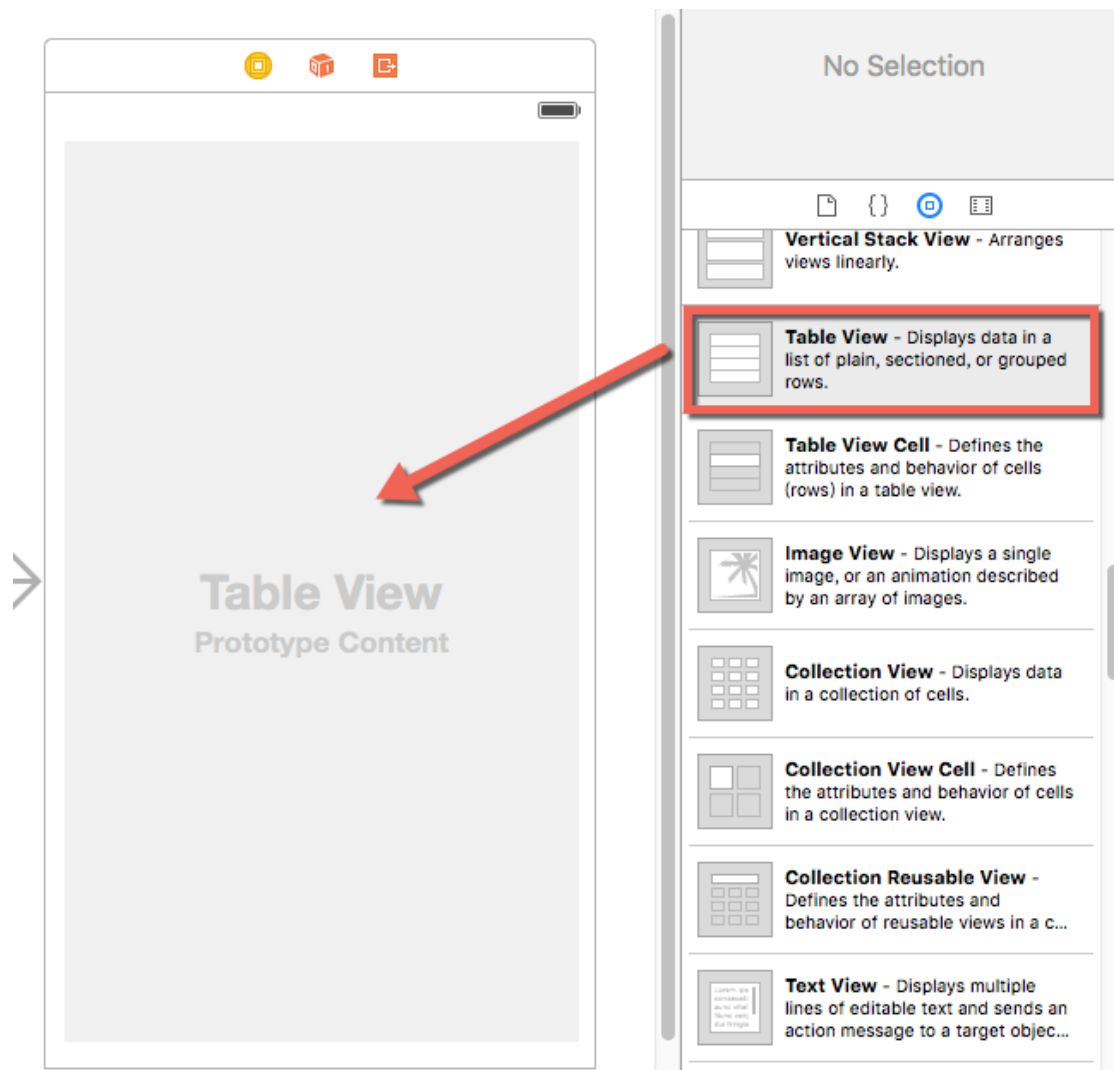
☐ Include UI Tests

## Step.2

選取【Main.storyboard】

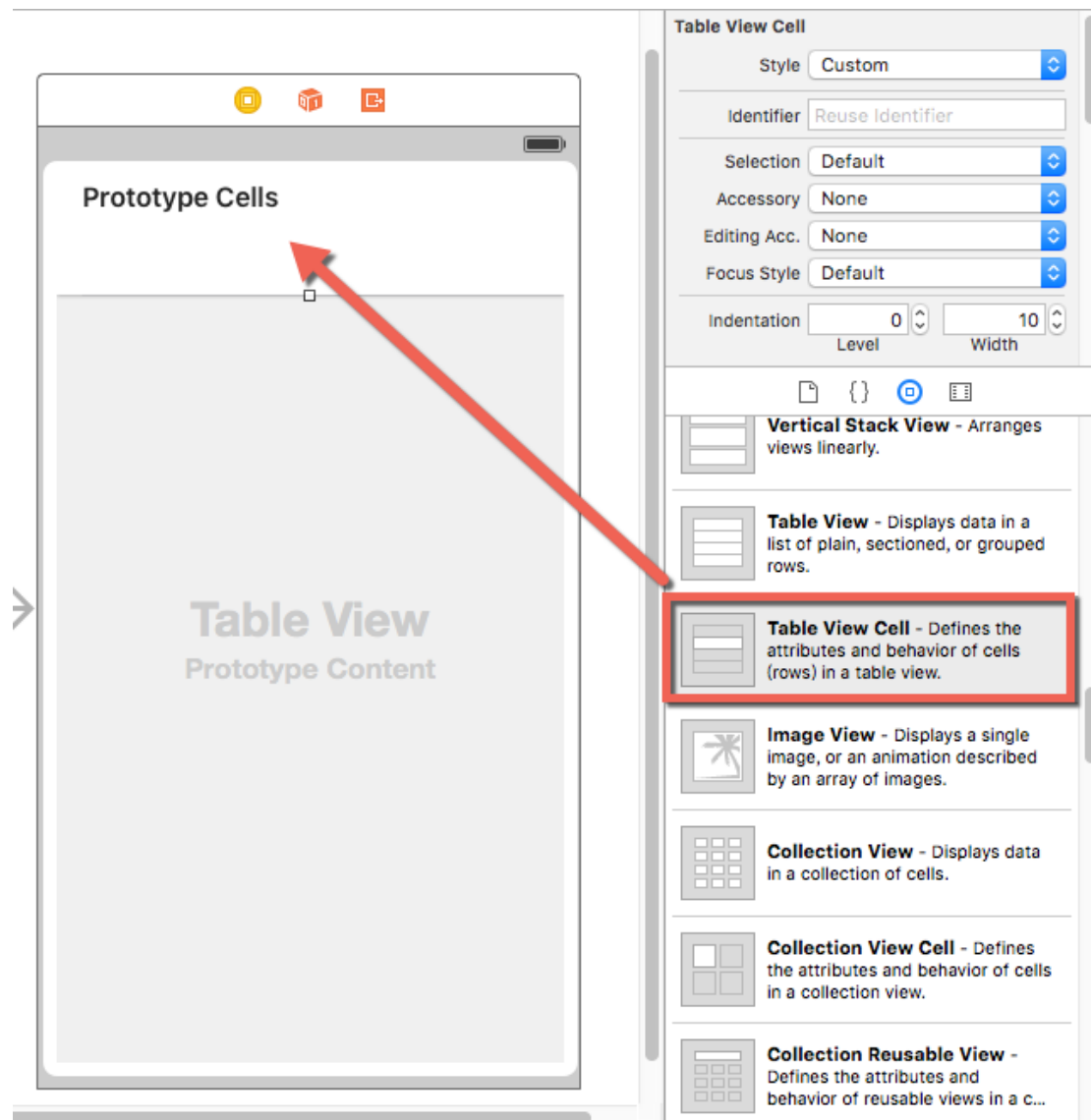


從【物件庫】中拖曳【Table View】到畫面中。

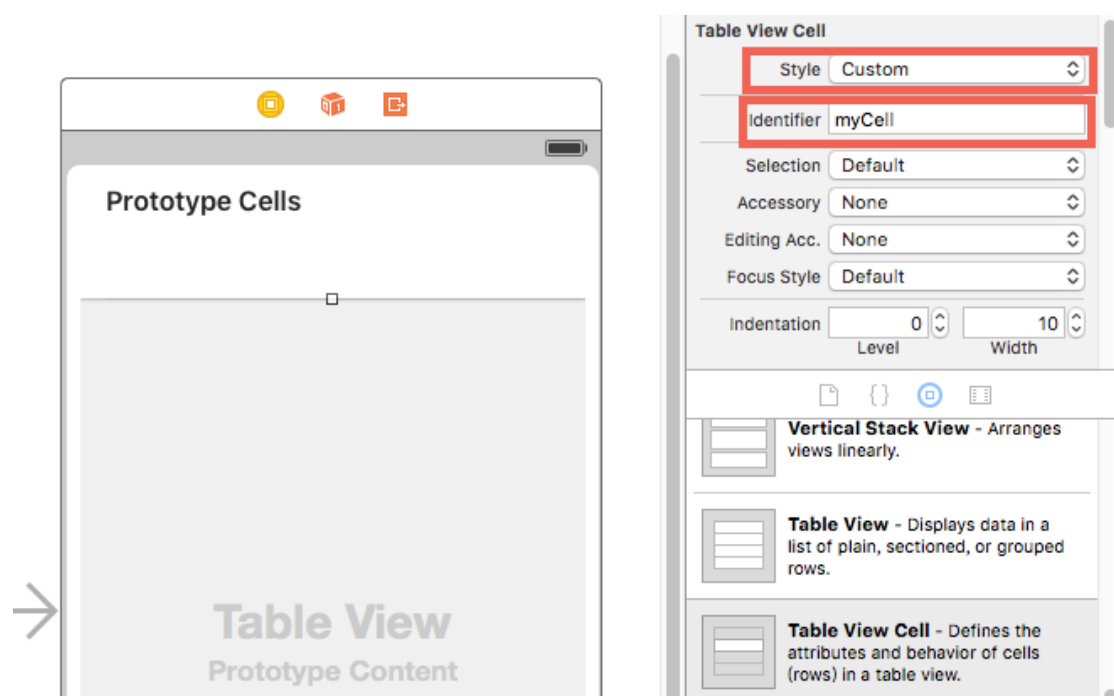


### Step.3.

- (1) 從【物件庫】中拖曳【Table View Cell】到畫面中。

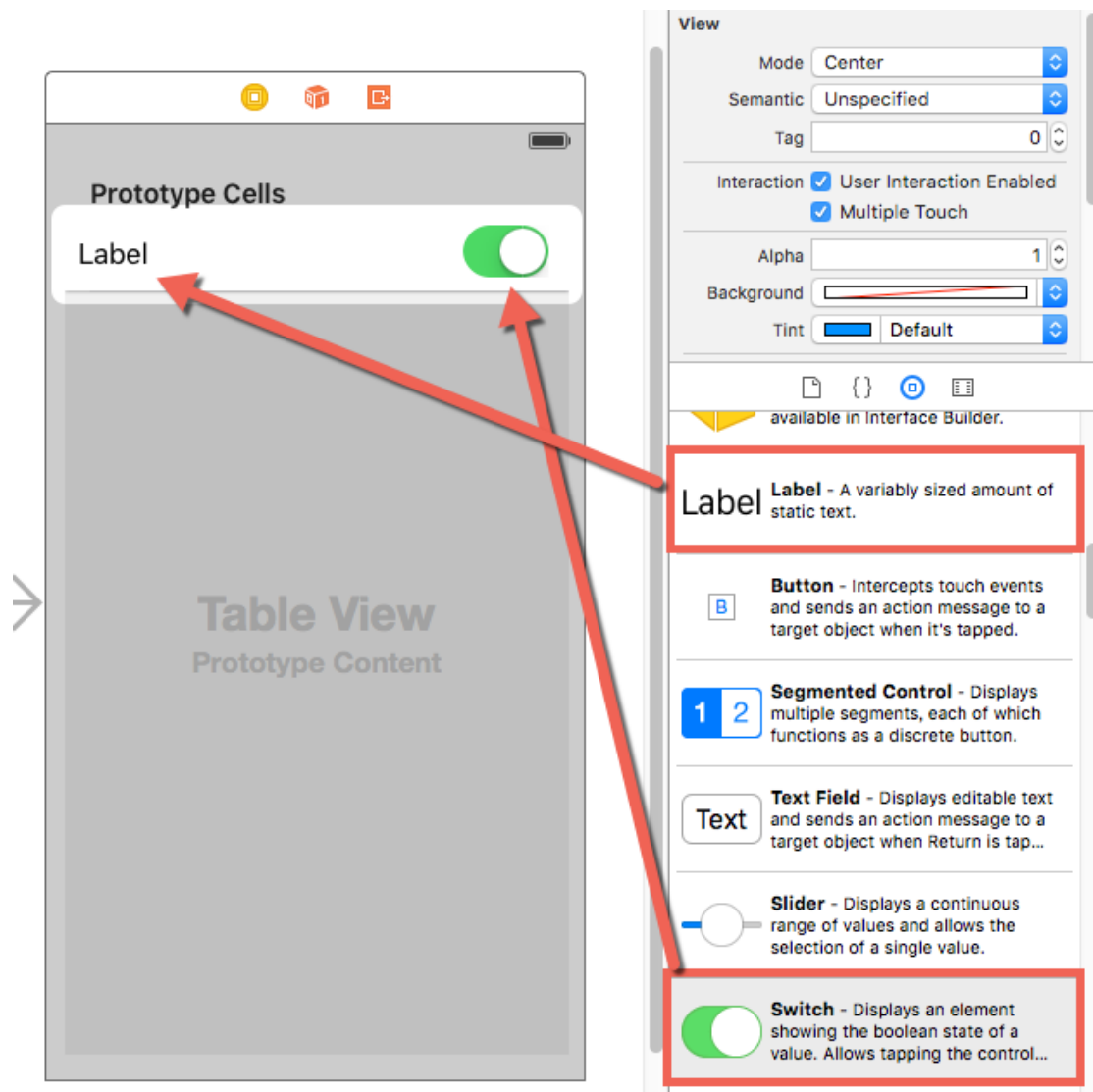


- (2) 設定 Table View Cell 的屬性  
點選新建立的 Table View Cell  
設定 Style 為 Custom  
設定 Identifier 為 myCell

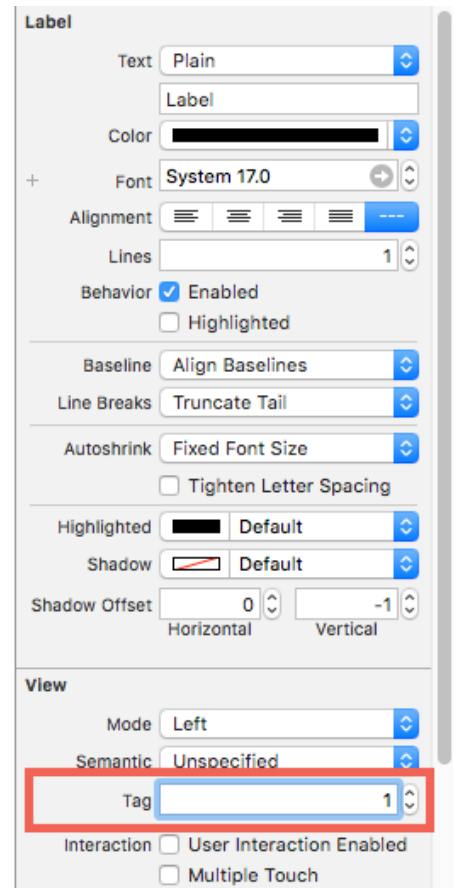
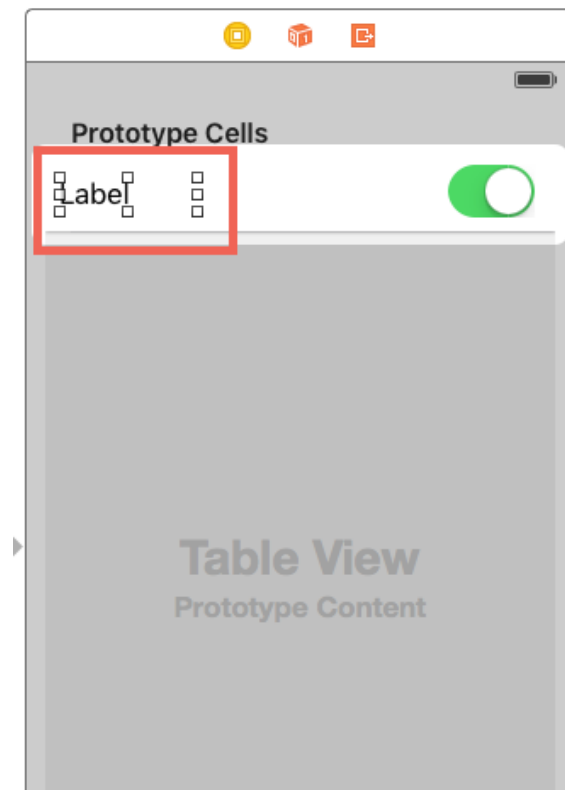


## Step.4.

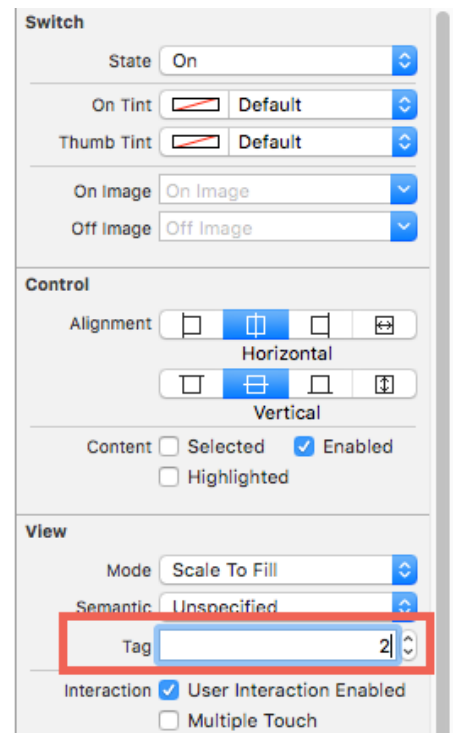
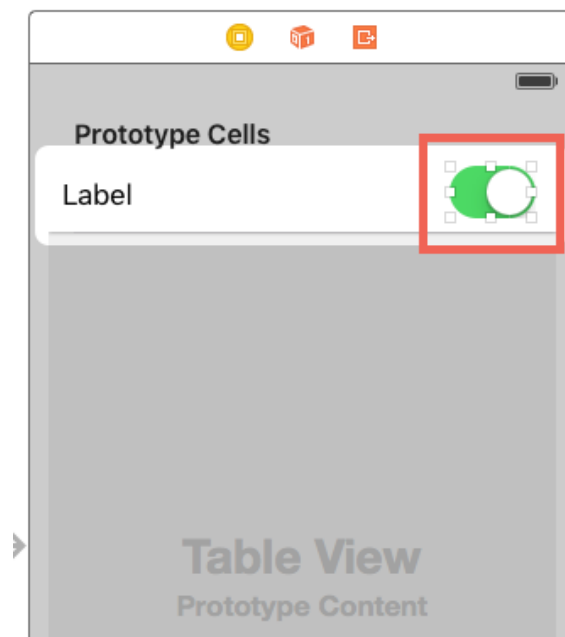
(1) 於 Cell 中 加入一個 Label 與一個 Switch



(2) 設定 新加入的 Label 的識別編號 Tag 為 1；



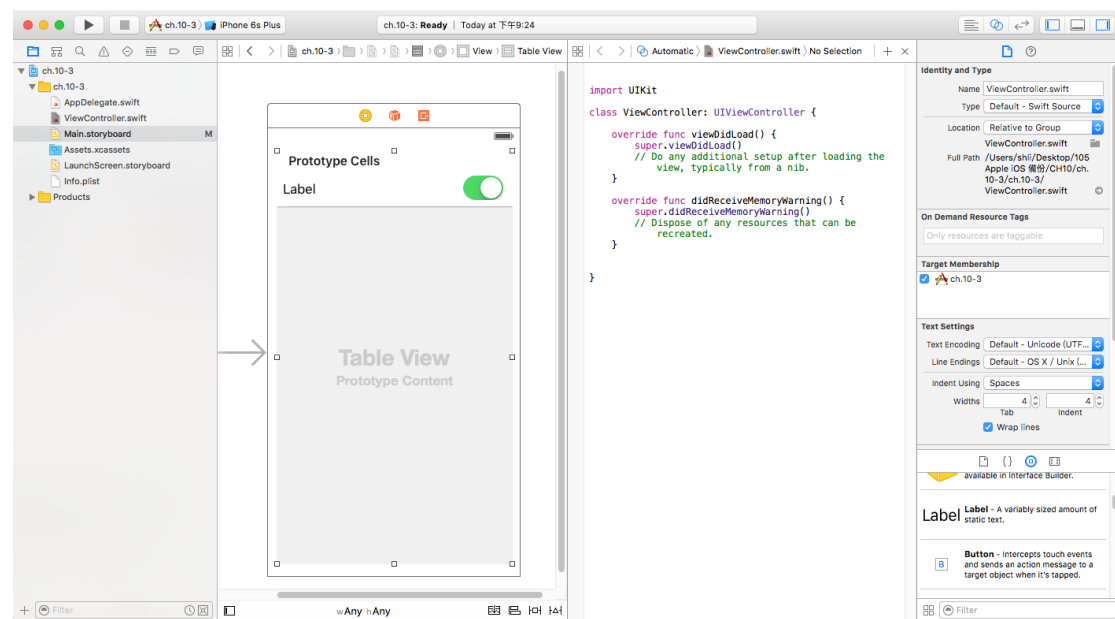
(3) 設定新加入的 Switch 的識別編號 Tag 為 2



Step.5.



將編輯畫面分割為四部分，以方便連結元件與程式



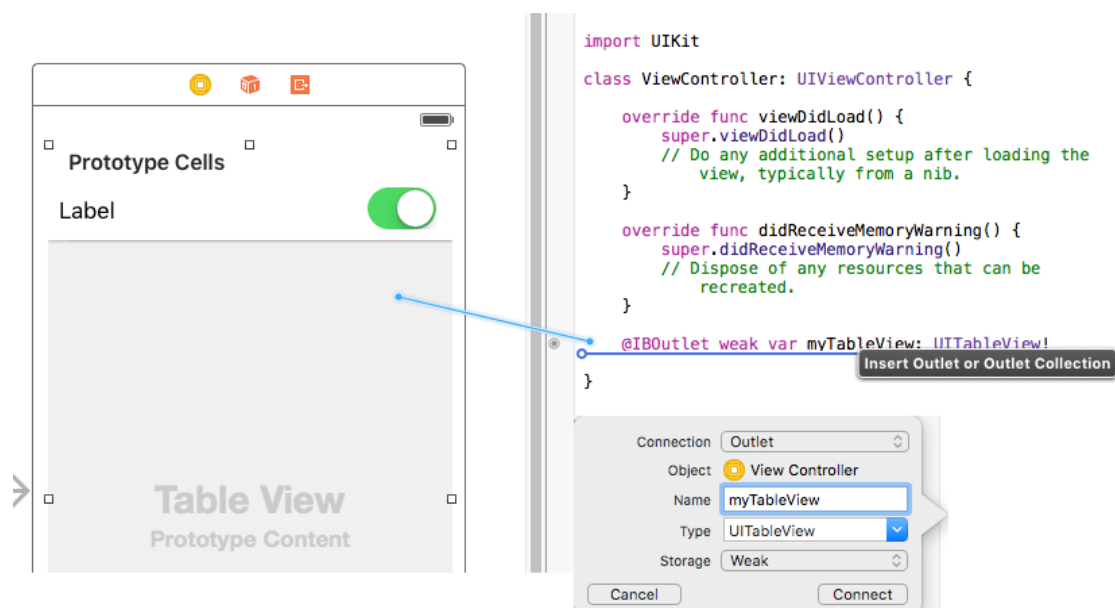
## Step.6

- (1) 加入 TableView 所要使用的類別 UITableViewDataSource, UITableViewDelegate 於 class 處

```
class ViewController: UIViewController, UITableViewDataSource, UITableViewDelegate {
```

注意 此時會出現錯誤提示，先不用理會它，待後續完成程式碼撰寫後即會消失。

- (2) 將【table View】與程式連結，命名 myTableView



## Step.7

- (1) 設定 myTableView 的資料來源為自己。

```
myTableView.dataSource = self
```

- (2) 設置表格中各區段要列出的筆數，本例中要 2 列。

```
func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
    return 2  
}
```

- (3) 設置表格中各列的內容，我們設定 Tabel View Cell 的 identifier 為 myCell，因此本例下方，需要指定

```
dequeueReusableCellWithIdentifier("myCell")。
```

- (4) 所設定的兩個 Label 分別給予了 Tag 編號，因此，本步驟中將使用 viewWithTag 的函數將這兩個 Label 直接給予指派變數

```
let myLabel : UILabel = cell.viewWithTag(1) as! UILabel  
let mySwitch : UISwitch = cell.viewWithTag(2) as! UISwitch
```

```
func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) ->  
UITableViewCell {  
    let cell:UITableViewCell = (tableView.dequeueReusableCellWithIdentifier("myCell") as  
UITableViewCell?)!  
    let myLabel : UILabel = cell.viewWithTag(1) as! UILabel  
    let mySwitch : UISwitch = cell.viewWithTag(2) as! UISwitch  
    if indexPath.row == 0 {  
        myLabel.text = "ON"  
        mySwitch.on = true  
    } else {  
        myLabel.text = "OFF"  
        mySwitch.on = false  
    }  
    return cell  
}
```

```

import UIKit

class ViewController: UIViewController, UITableViewDataSource, UITableViewDelegate {

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

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

    @IBOutlet weak var myTableView: UITableView!

    func tableView(tableView: UITableView, numberOfRowsInSectionSection section: Int) -> Int {
        return 2
    }

    func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath)
    -> UITableViewCell {

        let cell:UITableViewCell = (tableView.dequeueReusableCellWithIdentifier("myCell")
        as UITableViewCell?)!

        let myLabel : UILabel = cell.viewWithTag(1) as! UILabel
        let mySwitch : UISwitch = cell.viewWithTag(2) as! UISwitch

        if indexPath.row == 0 {
            myLabel.text = "ON"
            mySwitch.on = true
        } else {
            myLabel.text = "OFF"
            mySwitch.on = false
        }

        return cell
    }
}

```

## Step.8

執行程式。

可以看到畫面中的表格中包含了使用者自訂格式的兩列。

ON



OFF



---

---

---

---

---

---

---

---

---

---