

iOS程式設計

讓資料井然有序：表格元件

王昱景 Brian Wang

brian.wang.frontline@gmail.com

表格元件 Table View

- Table View 稱為表格元件
- 以一系列方式顯示要呈現的資料
- 每一列資料稱為儲存格 (Cell)
- 每一個儲存格可以包含一個或多個不同型別的元件

Table View 常用屬性

- Table View 繼承 View 類別
- 只要從元件區中拖曳 Table View 元件到 View 中，就可以加入一個 Table View 元件
- Table View 提供的 Content 屬性，可以設定表格中儲存格產生方式

- 預設是 Dynamic Prototypes 表示以動態方式建立儲存格
- 而 Static Cell 則可以靜態方式建立儲存格

Content	設定表格中儲存格的產生方式，預設是 Dynamic
Prototype Cells	設定表格中儲存格的數目，預設為 0
Style	設定表格中儲存格是否以群組方式顯示，預設為 Plain 表示非群組方式，Grouped 則設定為群組方式
Separator	設定表格中儲存格間的分隔線，預設值 Default 表示顯示分隔線，None 則不顯示分隔線，Single Line 會顯示分隔線
Separator Inset	預設為 Default，設定為 Custom 可以定義分隔線的左右間距
Sections	設定表格中包含多少個區段（Section），這個屬性必須是 Content 設定為 Static Cells 才有作用

建立動態方式的儲存格

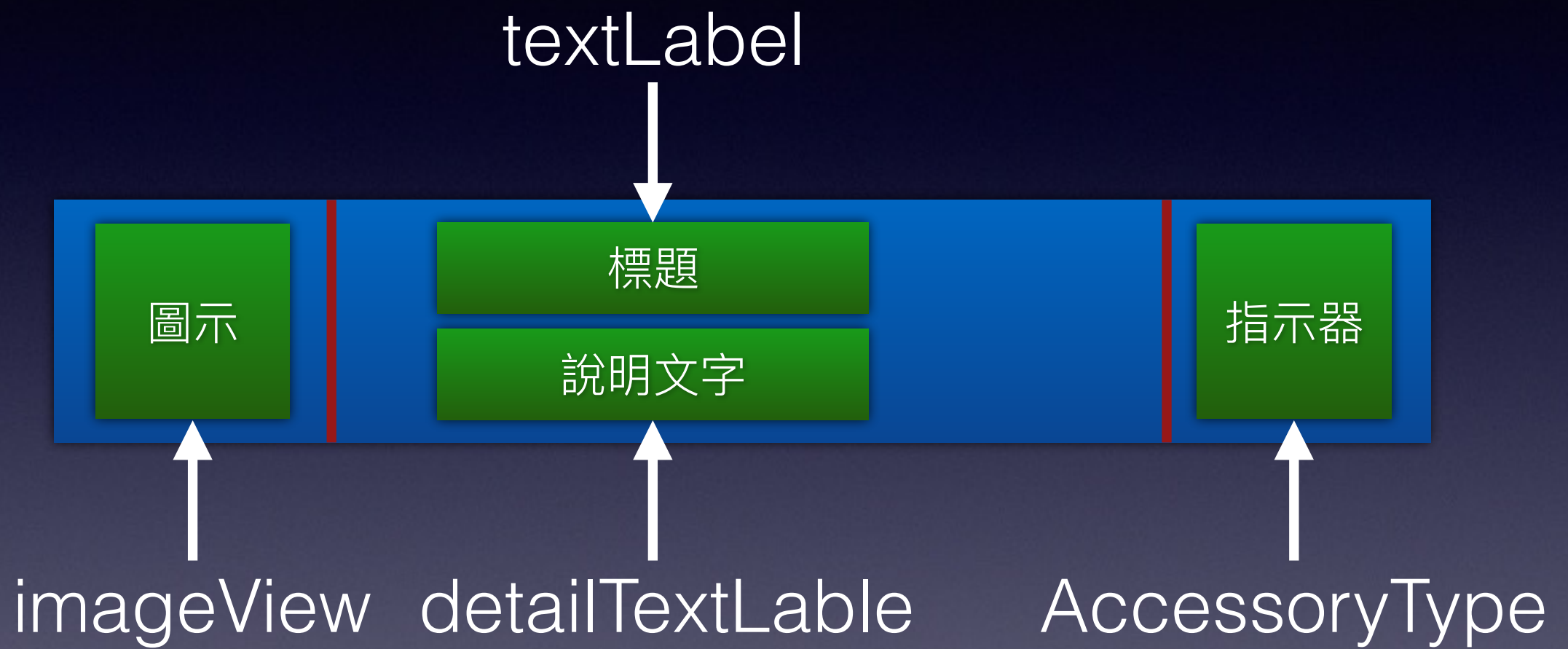
- Table View 表格預設的 Content 屬性為 Dynamic Prototypes
- 以動態方式建立儲存格
- 最常使用的方式，可以顯示動態的資料

- 預設的 Table View 表格並未包含任何儲存格，必須自行加入儲存格
- 加入的儲存格稱為 Table View Cell，其中包含一個 Content View 元件

Style	Swift 提供 Basic、Right Detail、Left Detail、Subtitle、Custom 等不同儲存格的樣板
Image	設定儲存格中的圖示
Identifier	設定儲存格名稱
Accessory	設定表格中儲存格的指示器 None：不顯示 Disclosure Indicator：表示按下後連結到另一個細節的頁面 Detail Disclosure Checkmark：指示選取的狀態 Detail
Separator	設定分隔線 Default Insets：顯示部分分隔線 Custom Insets：顯示完整分隔線

- 指示器主要是用來提示使用者如何操作這個儲存格，以及按下這個儲存格會顯示哪些資料
- 善用指示器可以讓使用者操作時更加自如
- 程式中設定指示器是以
UITableViewCellStyleAccessoryType.Type 設定
- Type 中包含 Checkmark、DetailButton、
DetailDisclosureButton、DisclosureIndicator 和
None 等狀態

- 有兩個方法可以在 Table View 表格中加入儲存格：
- 設定 Table View 的 Prototype Cell 屬性
- 從元件區中拖曳 Table View Cell 到 Table View 中



設定 Table View 和 dataSource、delegate 的關聯

- 在 Main.storyboard 中，選擇 Table View 表格
- 按右鍵在快顯功能表中，分別將 dataSource、delegate 分別拖曳到 View Controller 圖示中
- 即可設定 Table View 表格和 dataSource、delegate 的關聯

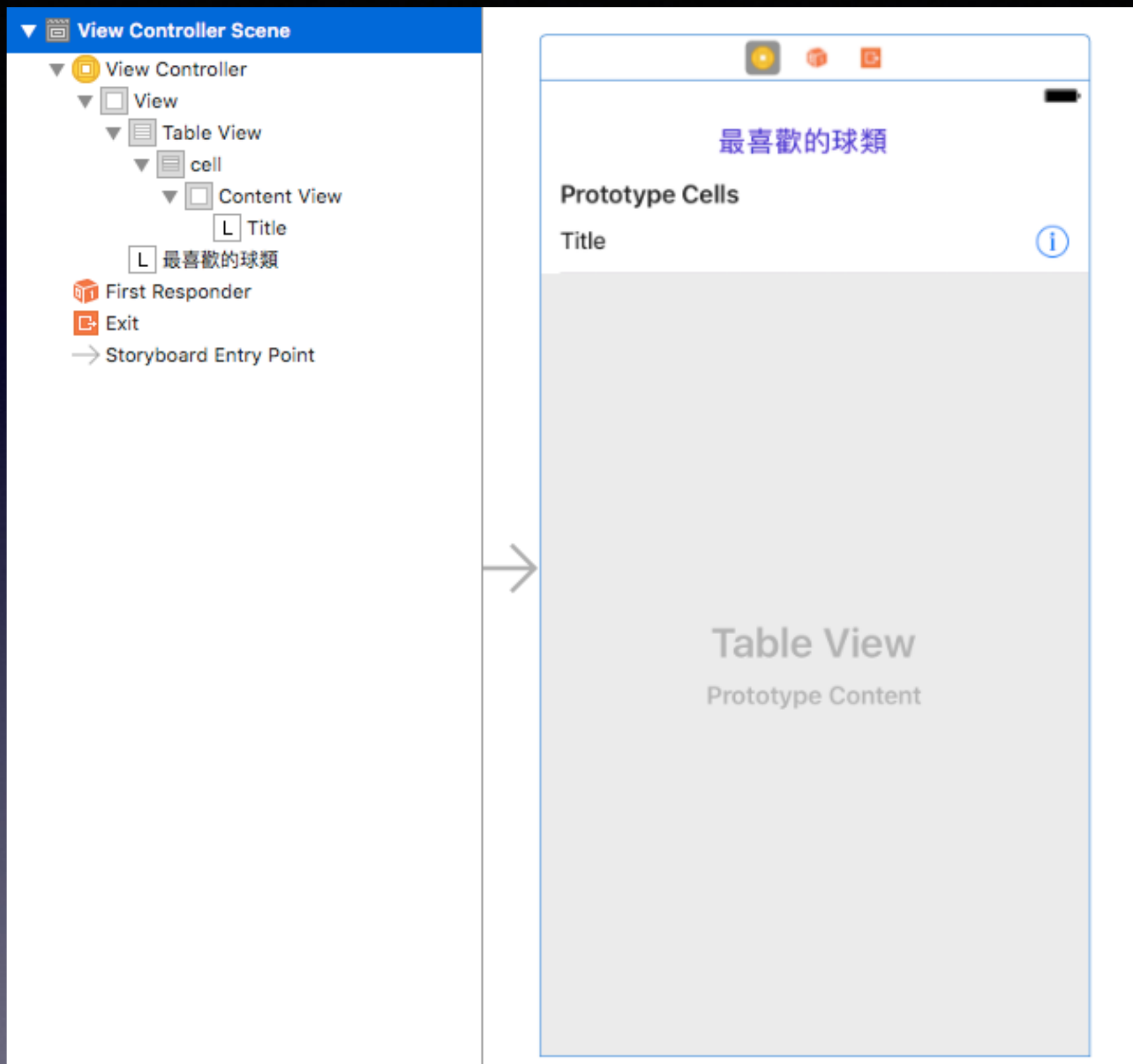
加入表格資料

- 要讓動態建立的儲存格顯示資料，必須實作 UITableViewDataSource、UITableViewDelegate
- UITableViewDataSource 告訴表格共有多少區，每一區有多少列，每一列的資料是什麼
- UITableViewDelegate 設定表格的外觀以及表格觸發的方法

numberOfSectionsInTableView	設定共有多少的區段，省略時預設為 1 個區段
numberOfRowsInSection	設定每一區段有多少列
cellForRowAtIndexPath	設定表格中儲存格的內容

以 Table View 顯示陣列資料

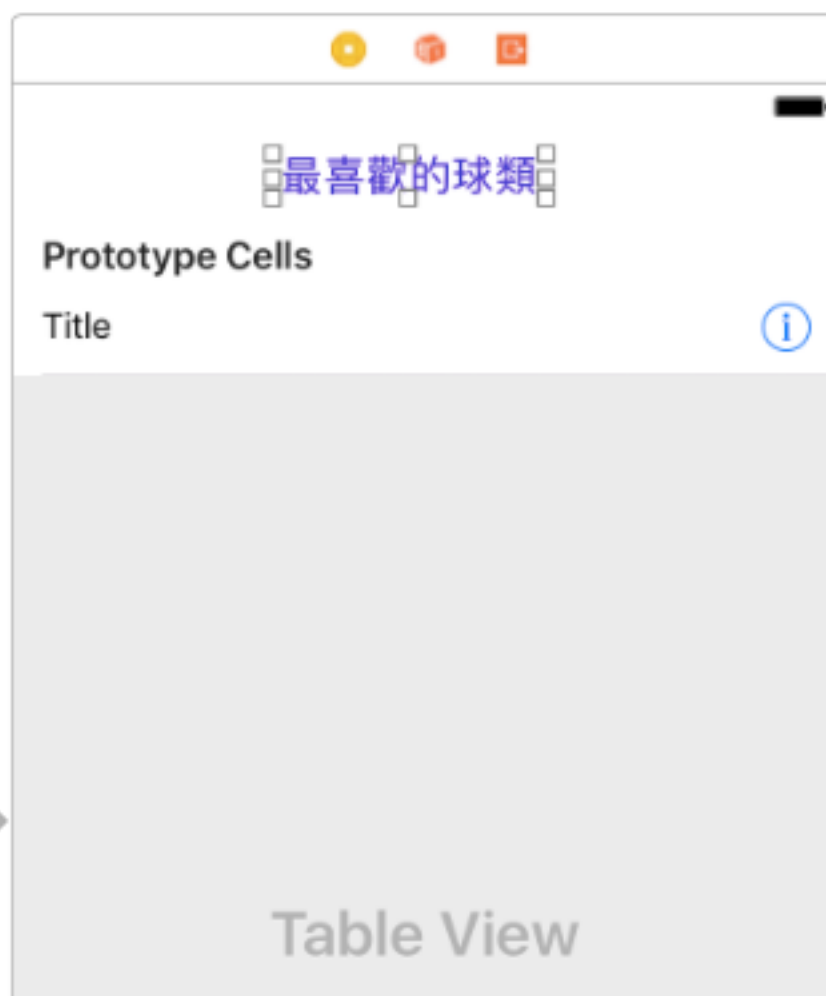
- 建立名稱為 TableView01 的 Single View 專案
- Table View 表格以動態方式顯示
- 儲存格的樣板為 Basic 樣式
- 指示器的圖示為 Detail 樣式



TableView01 > Ta...w01 > Mai...ard > Mai...se) > Vie...ene > Vie...ller > View > L 最喜歡的球類 < ⚠ >

View Controller Scene

- View Controller
 - View
 - Table View
 - cell
 - Content View
 - L 最喜歡的球類
- First Responder
- Exit
- Storyboard Entry Point



Label

Text Plain

最喜歡的球類

Color

Font System 19.0

Alignment

Lines 1

Behavior ☒ Enabled ☐ Highlighted

Baseline Align Baselines

Line Break Truncate Tail

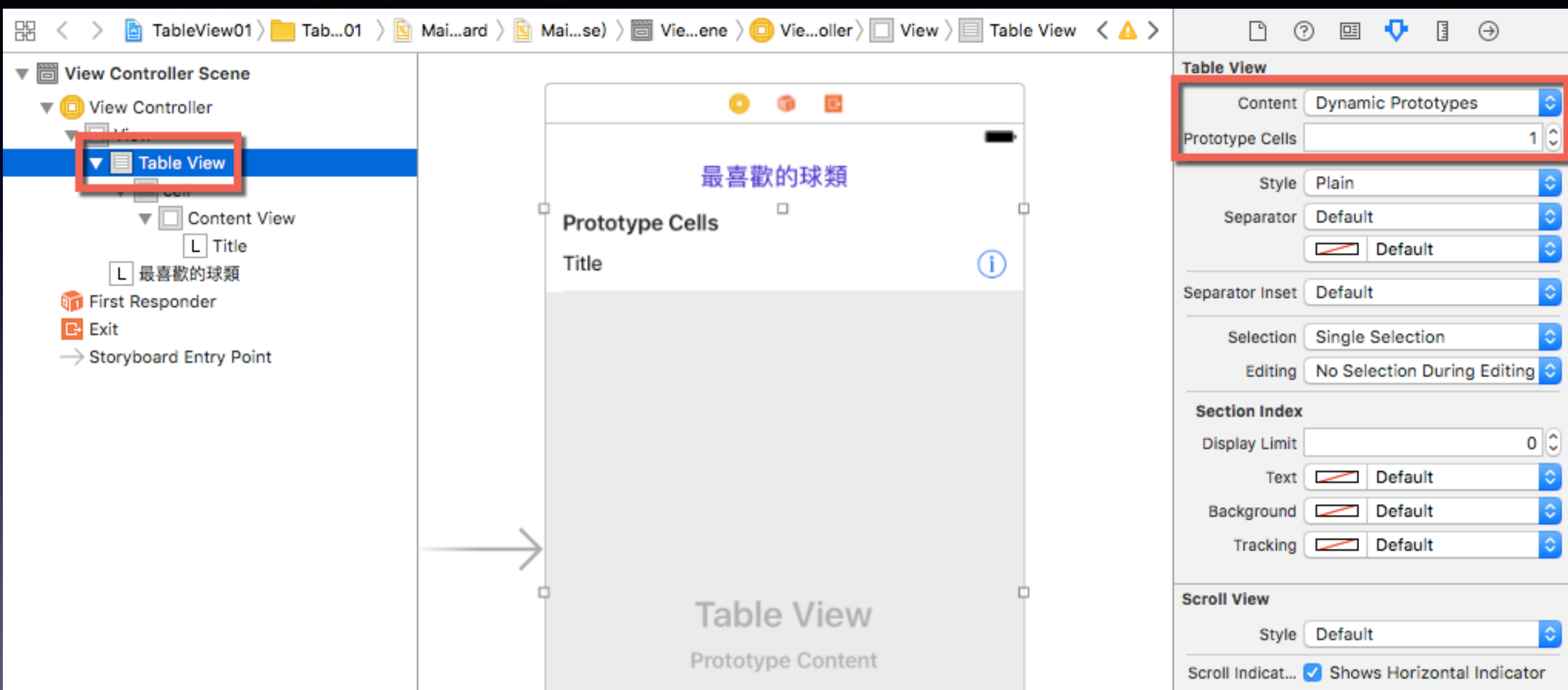
Autoshrink Fixed Font Size

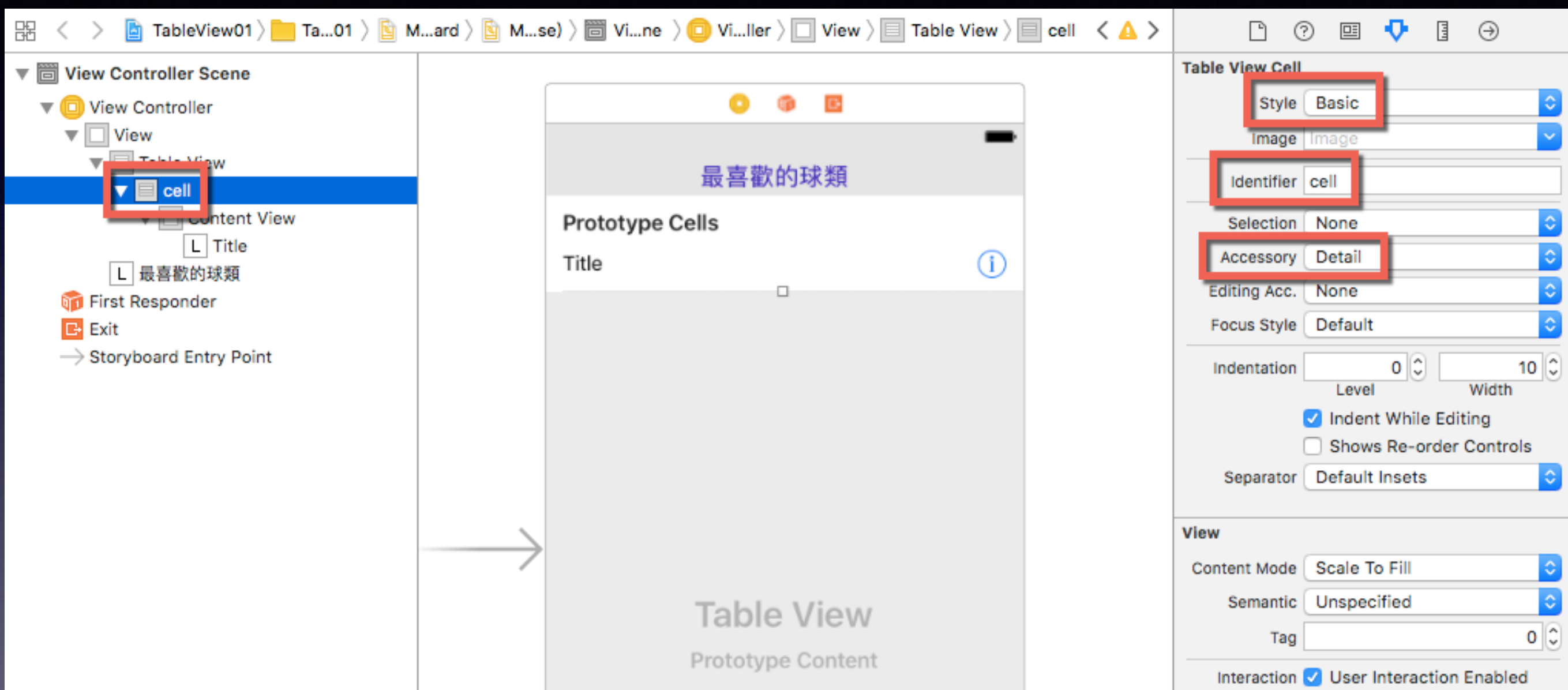
☐ Tighten Letter Spacing

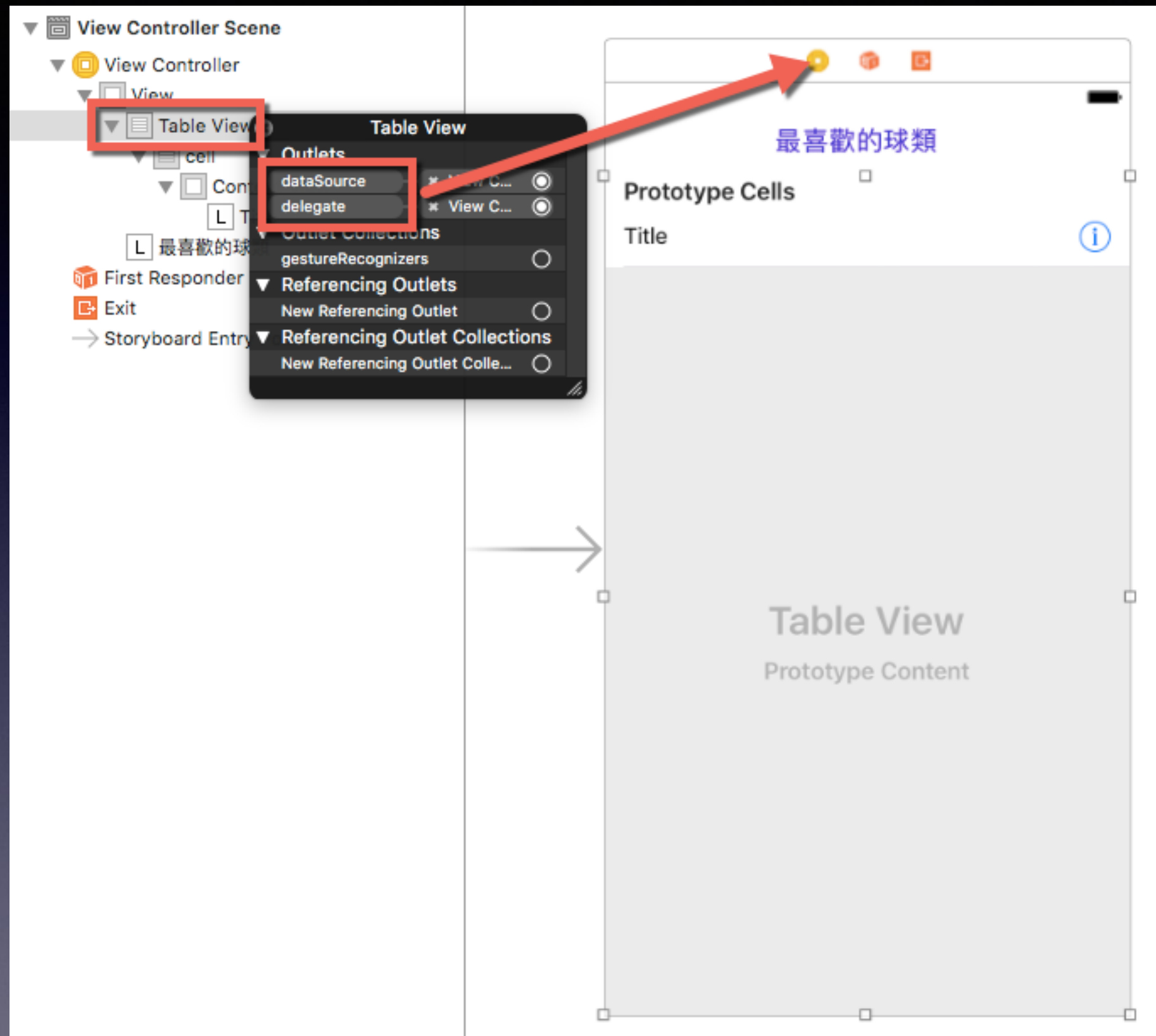
Highlighted Default

Shadow Default

Shadow Offset Width 0 Height -1








```
1 import UIKit
2
3 class ViewController: UIViewController,UITableViewDataSource{
4     var balls:Array<String> = ["籃球","足球","棒球","其他"]
5
6     // 設定表格只有一個區段
7     func numberOfSections(in tableView: UITableView) -> Int {
8         return 1
9     }
10
11     // 設定表格的列數
12     func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int{
13         return balls.count
14     }
15
16     // 表格的儲存格設定
17     func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell{
18         // 設定儲存格的內容
19         let cell:UITableViewCell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath)
20         as UITableViewCell
21         cell.textLabel!.text = balls[indexPath.row]
22         return cell
23     }
24
25     override func viewDidLoad() {
26         super.viewDidLoad()
27         // Do any additional setup after loading the view, typically from a nib.
28     }
29
30     override func didReceiveMemoryWarning() {
31         super.didReceiveMemoryWarning()
32         // Dispose of any resources that can be recreated.
33     }
34 }
35
```

儲存格的選取和指示器使用

- 當按下儲存格可以加入點選後的動作
- 指示器可用來提示使用者如何操作這個儲存格
- 以及按下這個儲存格會顯示哪些資料

儲存格的選取

- 要處理儲存格點選後的動作，必須實作 `UITableViewDelegate` 委派
- 最重要的是實作 `didSelectRowAtIndexPath` 方法

didSelectRowAtIndexPath	點選儲存格發生的事件
willSelectionRowAtIndexPath	儲存格即將被選取時發生的事件
editingStyleForRowAtIndexPath	設定滑動後顯示紅色刪除按鈕

Table View 的選取

- 建立名稱為 TableView02 的 Single View 專案
- 以 Table View 顯示陣列資料
- 按下儲存格後在上面的 Label 中顯示最喜歡的球類

View Controller Scene

- View Controller
 - view
 - L Label Ball**
 - Table View
 - cell
 - Content View
 - L Title
 - First Responder
 - Exit
 - Storyboard Entry Point

Prototype Cells

Title

最喜歡的球類

Table View

Label

Text Plain

最喜歡的球類

Color

Font System 19.0

Alignment

Lines 1

Behavior ☒ Enabled ☐ Highlighted

Baseline Align Baselines

Line Break Truncate Tail

Autoshrink Fixed Font Size ☐ Tighten Letter Spacing

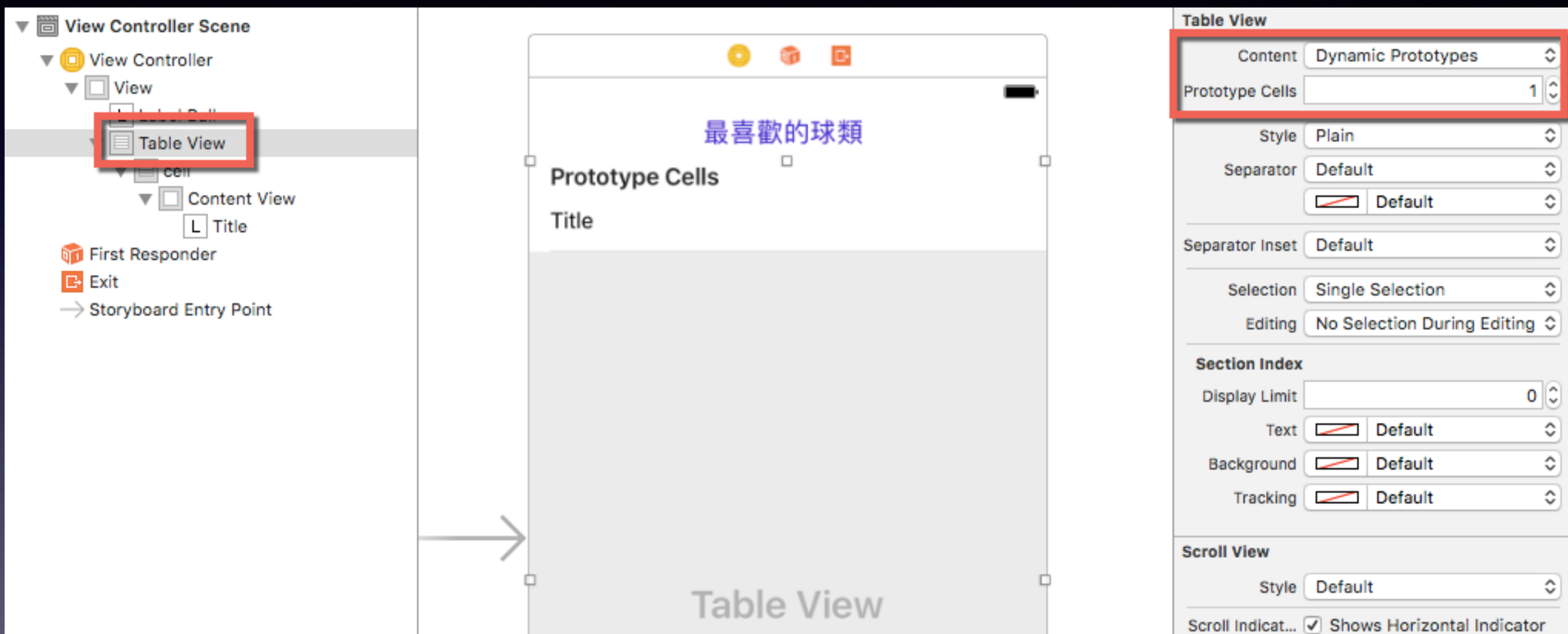
Highlighted ☐ Default

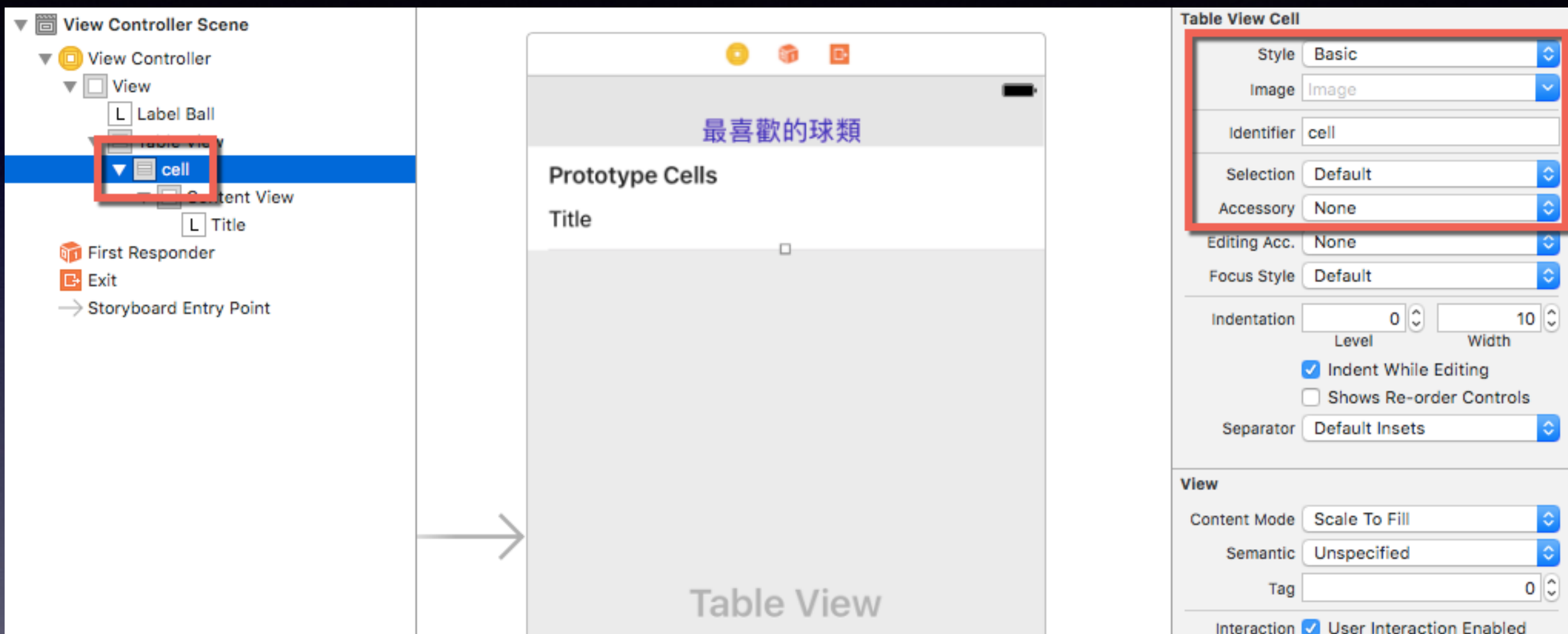
Shadow ☐ Default

Shadow Offset 0 -1

Width Height

View





```

1 import UIKit
2
3 class ViewController: UIViewController,UITableViewDataSource,UITableViewDelegate{
4     @IBOutlet weak var labelBall: UILabel!
5     var balls:Array<String> = ["籃球","足球","棒球","其他"]
6
7     // 點選儲存格的處理
8     func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath){
9         let choice:String = balls[indexPath.row]
10        labelBall.text = "最喜歡的球類:\(choice)"
11    }
12
13    // 設定表格只有一個區段
14    func numberOfSections(in tableView: UITableView) -> Int {
15        return 1
16    }
17
18    // 設定表格的列數
19    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int{
20        return balls.count
21    }
22
23    // 表格的儲存格設定
24    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell{
25        // 設定儲存格的內容
26        let cell:UITableViewCell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath)
27        as UITableViewCell
28        cell.textLabel!.text = balls[indexPath.row]
29        return cell
30    }
31
32    override func viewDidLoad() {
33        super.viewDidLoad()
34        // Do any additional setup after loading the view, typically from a nib.
35    }
36
37    override func didReceiveMemoryWarning() {
38        super.didReceiveMemoryWarning()
39        // Dispose of any resources that can be recreated.
40    }
41 }
42

```


儲存格指示器

- 指示器可用來提示使用者如何操作這個儲存格
- 以及按下這個儲存格會顯示哪些資料
- 例如使用 Checkmark 指示器可以記錄目前選取的狀態，尤其是在表格選取多項儲存格時

顯示 Table View 的選取狀態

- 建立名稱為 TableView03 的 Single View 專案
- 以 Table View 顯示資料，包含圖示、名稱、說明文字
- 按下儲存格後以 Checkmark 圖示顯示該儲存格已選取
- 如果該儲存格已選取、再次按下該儲存格則會取消選取

View Controller Scene

- View Controller
 - View
 - L 最喜歡的球類**
 - Table View
 - cell
 - Content View
 - Title
 - Subtitle
 - First Responder
 - Exit
 - Storyboard Entry Point

Prototype Cells

Title

Subtitle

Table View

Prototype Content

結果：

Label

Text Plain

最喜歡的球類

Color

Font System 19.0

Alignment

Lines 1

Behavior ☒ Enabled ☐ Highlighted

Baseline Align Baselines

Line Break Truncate Tail

Autoshrink Fixed Font Size ☐ Tighten Letter Spacing

Highlighted

Shadow

Shadow Offset 0 -1

Width Height

View

View Controller Scene

- View Controller
 - View
 - 最喜歡的球類
 - Table View**
 - Cell
 - Content View
 - Title
 - Subtitle
 - Label Result
- First Responder
- Exit
- Storyboard Entry Point

最喜歡的球類

Prototype Cells

Title

Subtitle

Table View

Prototype Content

結果：

Table View

Content: Dynamic Prototypes

Prototype Cells: 1

Style: Plain

Separator: Single Line

Separator Inset: Custom

Selection: Single Selection

Editing: No Selection During Editing

Section Index

Display Limit: 0

Text: Default

Background: Default

Tracking: Default

Scroll View

Style: Default

View Controller Scene

- View Controller
 - View
 - 最喜歡的球類
 - Table View
 - cell**
 - Content View
 - Title
 - Subtitle
 - Label Result
 - First Responder
 - Exit
 - Storyboard Entry Point

Prototype Cells

Title

Subtitle

Table View

Prototype Content

結果：

Table View Cell

Style Subtitle

Image Image

Identifier cell

Selection Default

Accessory None

Editing Acc. None

Focus Style Default

Indentation 0 10

Level Width

☒ Indent While Editing

☐ Shows Re-order Controls

Separator Default Insets

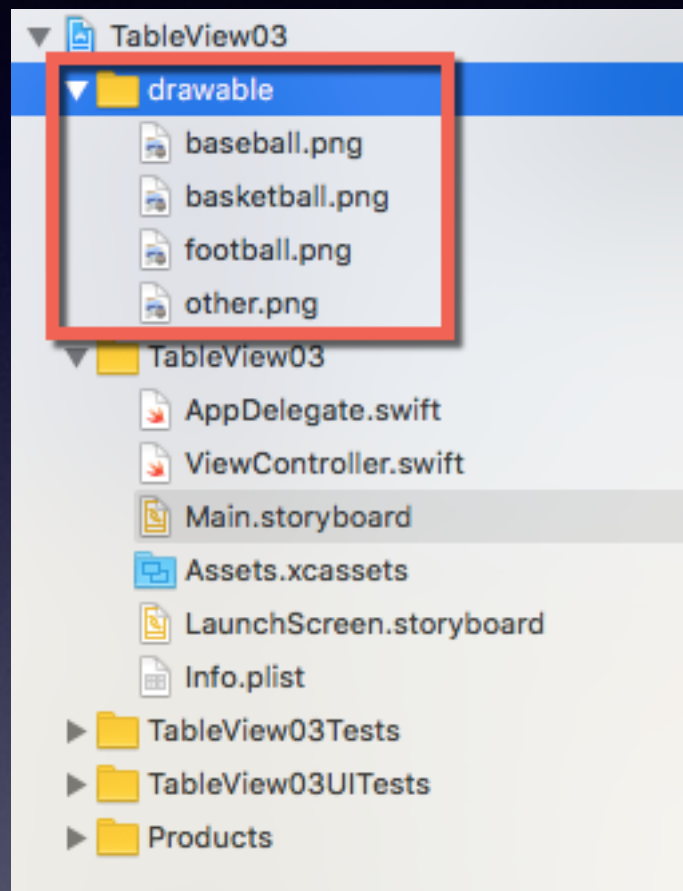
View

Content Mode Scale To Fill

Semantic Unspecified

Tag 0

Interaction ☒ User Interaction Enabled



```

1 import UIKit
2
3 class ViewController: UIViewController,UITableViewDataSource,UITableViewDelegate{
4
5     @IBOutlet weak var labelResult: UILabel!
6
7     // balls 字典記錄 名稱、價格、圖示名稱
8     var balls:Array<Dictionary<String,String>> =
9     [{"name":"籃球","value":"600","imageName":"basketball"},
10      [{"name":"足球","value":"500","imageName":"football"},
11      [{"name":"棒球","value":"250","imageName":"baseball"},
12      [{"name":"其他","value":"350","imageName":"other"}]]
13
14     // aryCheck 記錄選取狀態
15     var aryCheck:Array<Bool> = [false,false,false,false]
16
17     // 點選儲存格的處理
18     func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath){
19         // 取得目前選取的儲存格
20         let cell:UITableViewCell? = tableView.cellForRow(at: indexPath) as UITableViewCell?
21
22         aryCheck[indexPath.row] = !aryCheck[indexPath.row]
23
24         // 依 aryCheck 狀態設定指示器
25         if aryCheck[indexPath.row]{
26             cell!.accessoryType = UITableViewCellAccessoryType.checkmark
27         }else{
28             cell!.accessoryType = UITableViewCellAccessoryType.none
29         }
30
31         // 顯示選取項目
32         showResult()
33     }
34
35     // 顯示選取項目
36     func showResult(){
37         labelResult.text = "選取項目:"
38
39         for i in 0 ..< balls.count {
40             if aryCheck[i]{
41                 var ball:Dictionary<String,String> = balls[i]
42                 labelResult.text = labelResult.text! + ball["name"]! + " "
43             }
44         }
45     }
46

```

```

46
47 // 設定表格的列數
48 func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int{
49     return balls.count
50 }
51
52 // 表格的儲存格設定
53 func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell{
54     // 設定儲存格的內容
55     let cell:UITableViewCell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath) as UITableViewCell
56     var ball:Dictionary<String,String> = balls[indexPath.row]
57
58     cell.textLabel!.text = ball["name"] //名稱
59     cell.detailTextLabel?.text = ball["value"] //價格
60     cell.imageView!.image = UIImage(named:ball["imageName"]!) //圖示
61
62     return cell
63 }
64
65 override func viewDidLoad() {
66     super.viewDidLoad()
67     // Do any additional setup after loading the view, typically from a nib.
68 }
69
70 override func didReceiveMemoryWarning() {
71     super.didReceiveMemoryWarning()
72     // Dispose of any resources that can be recreated.
73 }
74
75 }
76

```

編輯儲存格

- 有時候我們必須對表格做編輯，例如：將輸入資料加入儲存格，刪除指定的儲存格，或是改變儲存格的順序
- 要改變儲存格的順序，必須使用 NSMutableArray 動態的陣列

- NSMutableArray 動態陣列使用 addObject() 方法可以新增一筆資料
- removeObjectAtIndex(index) 則可以刪除指定索引的資料
- 要處理儲存格編輯的動作必須實作 UITableViewDataSource 和 UITableViewDelegate 方法

editingStyleForRowAtIndexPath	設定滑動後顯示紅色刪除按鈕
commitEditingStyle	刪除該儲存格的資料
canMoveRowAtIndexPath	設定儲存格是否可拖曳
moveRowAtIndexPath	將動態陣列中的資料移動到指定的位置

新增一筆資料

- 在表格中新增資料只要將資料加入陣列或資料庫中，然後再以 reloadData() 方法更新表格即可

- 有兩種方法可將資料新增至表格中：
- 第一種方法較簡單，先為表格取個 IBOutlet 名稱，然後執行 reloadData() 方法
- 第二種方法較麻煩，從 ViewController 中篩選所有的 UIView 元件，如果找到的 UIView 元件是 UITableView 元件，即以此 UITableView 元件呼叫 reloadData() 方法

刪除一筆資料

- iOS 刪除資料的動作是在儲存格中由右往左滑動
- 此時儲存格的右方會出現紅色的刪除按鈕
- 然後按下 刪除 按鈕即可將該筆資料刪除

- 要讓表格中出現紅色刪除按鈕需要兩個步驟
 - 必須實作 `editingStyleForRowAtIndexPath` 方法，並傳回 `UITableViewCellEditingStyle.Delete`
 - 實作 `commitEditingStyle` 方法，如果沒有實做這個方法，則紅色的刪除按鈕不會出現
 - 除了刪除該儲存格的資料，同時也要將實體的陣列或是資料庫的資料刪除

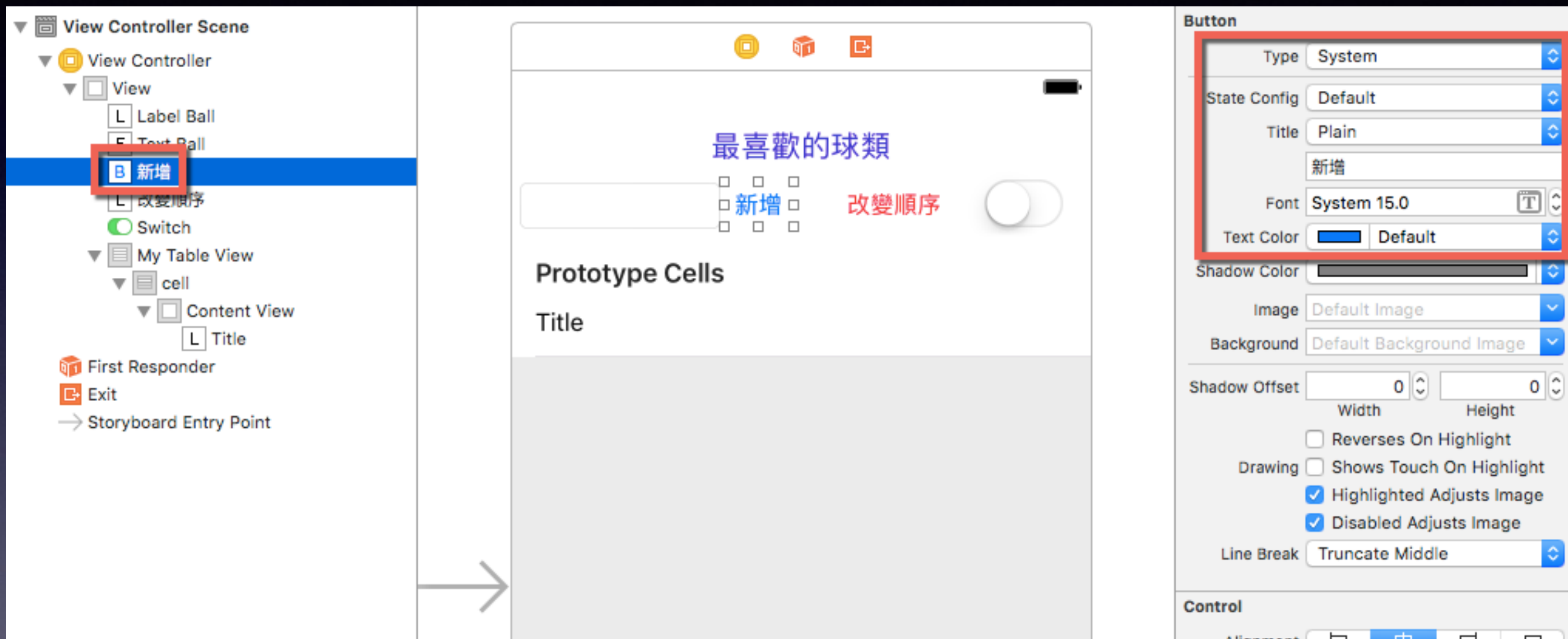
儲存格的拖曳

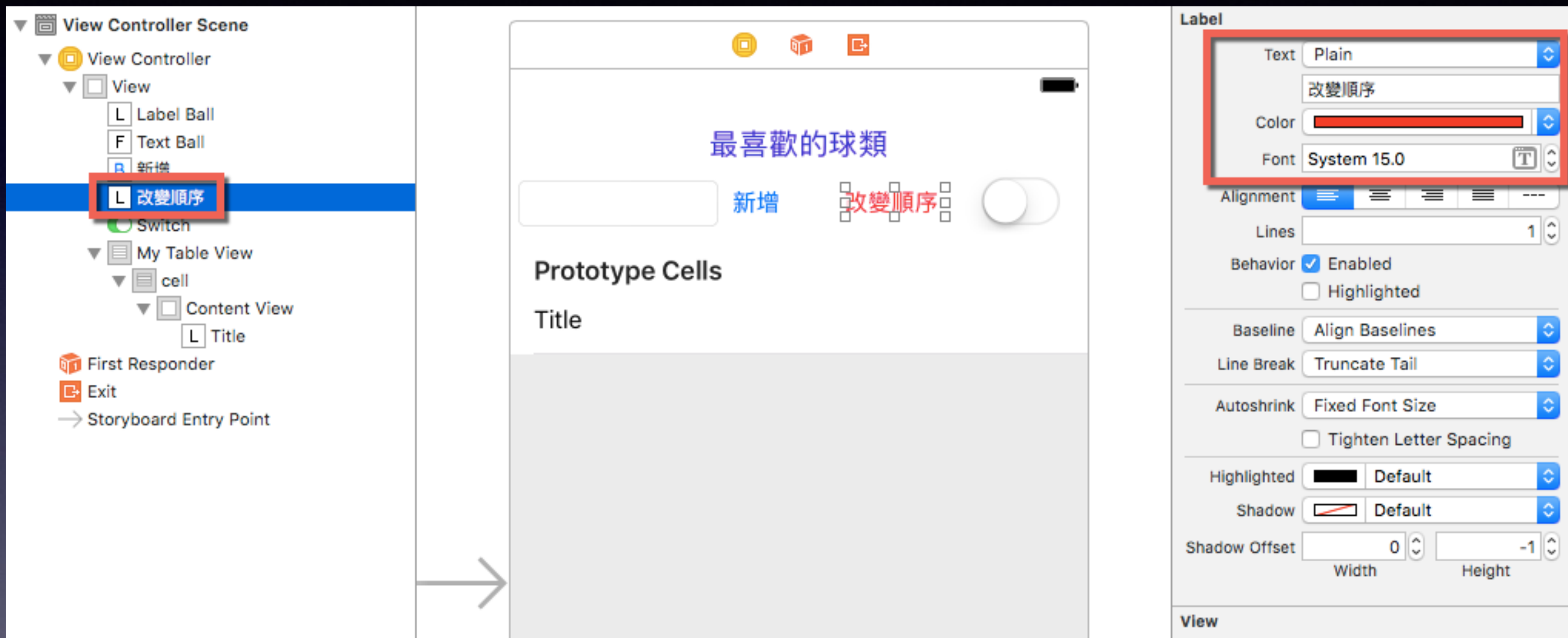
- 預設儲存格式不可拖曳
- 要設定儲存格為可拖曳需要三個步驟：
 - 設定 editing 屬性為 true
 - 必須實作 canMoveRowAtIndexPath 方法，並傳回 true
 - 實作 moveRowAtIndexPath 方法，其中要以 exchangeObjectAtIndex 方法將動態陣列中的資料移動到指定的位置

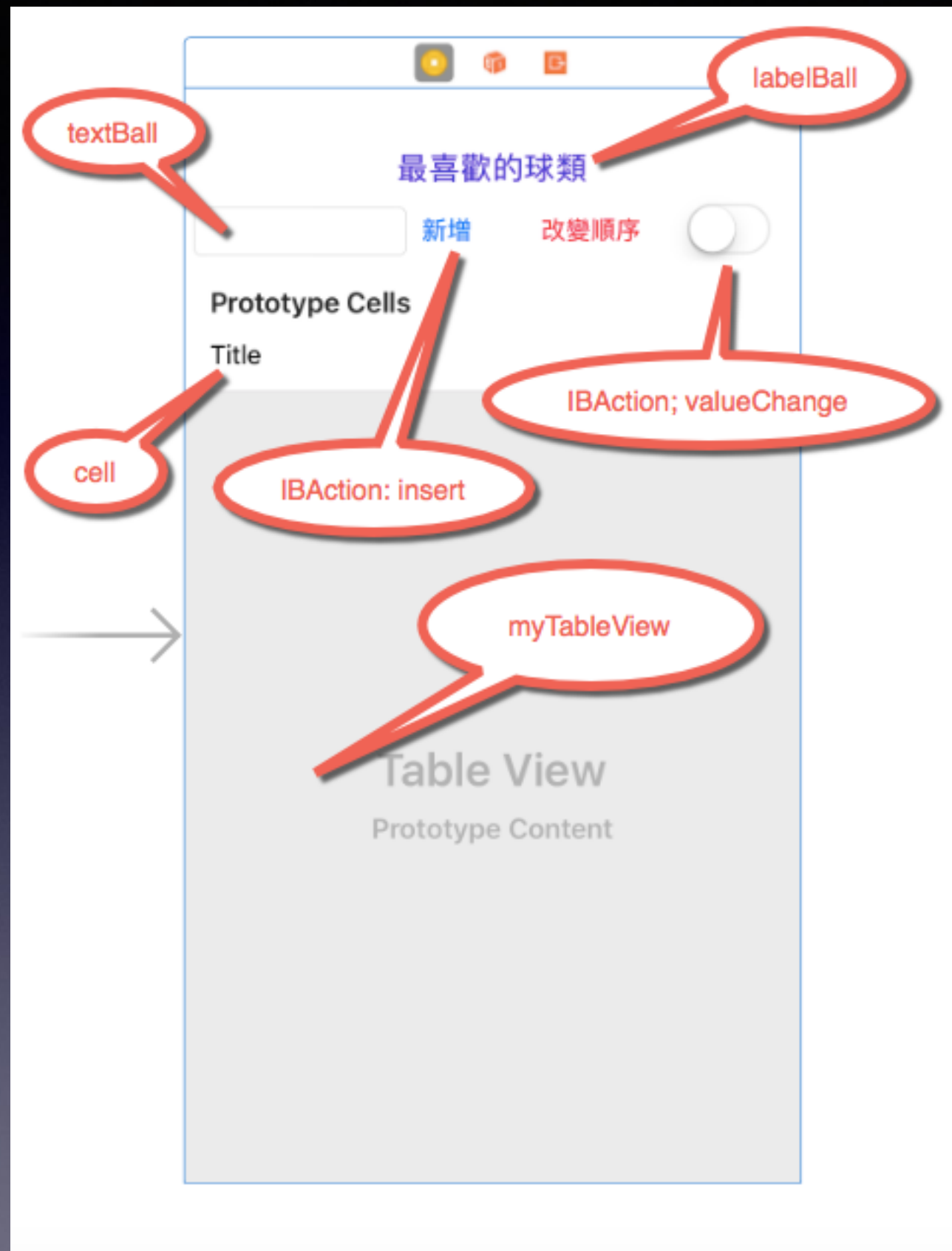
編輯儲存格

- 建立名稱為 TableView_Add 的 Single View 專案
- 以 Table View 顯示資料，在文字方塊中輸入資料後按 新增 鈕，可以將該筆資料加入表格中
- 在儲存格上由右往左拖曳會出現紅色 Delete 鈕，按下 Delete 鈕會刪除該筆資料
- 切換 Switch 即可拖曳表格的資料，改變表格資料的順序









```

1 import UIKit
2
3 class ViewController: UIViewController,UITableViewDataSource,UITableViewDelegate {
4
5     @IBOutlet weak var labelBall: UILabel!
6     @IBOutlet weak var textBall: UITextField!
7     @IBOutlet weak var myTableView: UITableView!
8
9     // 建立 NSMutableArray 型別陣列
10    var balls:NSMutableArray = ["籃球","足球","棒球","其他"]
11
12    // 新增資料
13    @IBAction func insert(_ sender: UIButton) {
14        balls.add(textBall.text!)
15        myTableView.reloadData()
16    }
17
18    // 設定滑動後顯示紅色刪除按鈕
19    func tableView(_ tableView: UITableView, editingStyleForRowAt indexPath: IndexPath) -> UITableViewCellEditingStyle{
20        return UITableViewCellEditingStyle.delete
21    }
22
23    // 按下刪除按鈕，刪除該儲存格資料
24    func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCellEditingStyle, forRowAt indexPath: IndexPath){
25        // 刪除陣列資料
26        balls.removeObject(at: indexPath.row)
27        // 刪除該儲存格資料
28        tableView.deleteRows(at: [indexPath], with: .fade)
29    }
30
31    // UISwitch 切換
32    @IBAction func valueChanged(_ sender: UISwitch) {
33        if sender.isOn {
34            self.myTableView.isEditing = true
35        }else{
36            self.myTableView.isEditing = false
37        }
38    }
39
40    // 設定表格的列數
41    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int{
42        return balls.count
43    }
44

```

```

44
45 // 表格的儲存格設定
46 func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell{
47     // 設定儲存格的內容
48     let cell:UITableViewCell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath) as UITableViewCell
49     cell.textLabel!.text = balls.object(at: indexPath.row) as! NSString as String
50     return cell
51 }
52
53 // 點選儲存格的處理
54 func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath){
55     let choice:AnyObject = balls[indexPath.row] as AnyObject
56     labelBall.text = "最喜歡的球類：\(choice)"
57 }
58
59 // 允許拖曳
60 func tableView(_ tableView: UITableView, canMoveRowAt indexPath: IndexPath) -> Bool{
61     return true
62 }
63
64 // 移動資料
65 func tableView(_ tableView: UITableView, moveRowAt sourceIndexPath: IndexPath, to destinationIndexPath: IndexPath){
66     balls.exchangeObject(at: sourceIndexPath.row, withObjectAt: destinationIndexPath.row)
67 }
68
69 override func viewDidLoad() {
70     super.viewDidLoad()
71     // Do any additional setup after loading the view, typically from a nib.
72 }
73
74 override func didReceiveMemoryWarning() {
75     super.didReceiveMemoryWarning()
76     // Dispose of any resources that can be recreated.
77 }
78
79 }
80

```

Custom 儲存格

- Swift 預設的儲存格只有固定的格式
- 想要自訂具有個人風格的表格，就必須使用 Custom 自訂儲存格的方式
- 如果要使用 Custom 自訂儲存格，則儲存格的 Style 必須設定為 Custom，Identifier 則可自行命名，同時 Class 欄位內容可以繼承自建的類別

自訂儲存格

- 建立名稱為 TableView_Custom 的 Single View 專案
- 以 Table View 顯示資料，並以自訂的 CustomCell 類別建立自訂的儲存格

- 儲存格命名為 cell，其中包含有 imageName、labelName 和 labelValue 等 3 欄位分別顯示圖示、項目名稱和價格
- 也將奇、偶數儲存格以不同的背景色呈現

View Controller Scene

- View Controller
 - View
 - Table View
 - cell
 - Content View
 - Image Name
 - L Label Ball**
 - Label Name
 - Label Value
- First Responder
- Exit
- Storyboard Entry Point

Prototype Cells

Label

Label

Table View

Prototype Content

Label

Text Plain

最喜歡的球類

Color

Font System 19.0

Alignment

Lines

Behavior ☒ Enabled ☐ Highlighted

Baseline Align Baselines

Line Break Truncate Tail

Autoshrink Fixed Font Size

☐ Tighten Letter Spacing

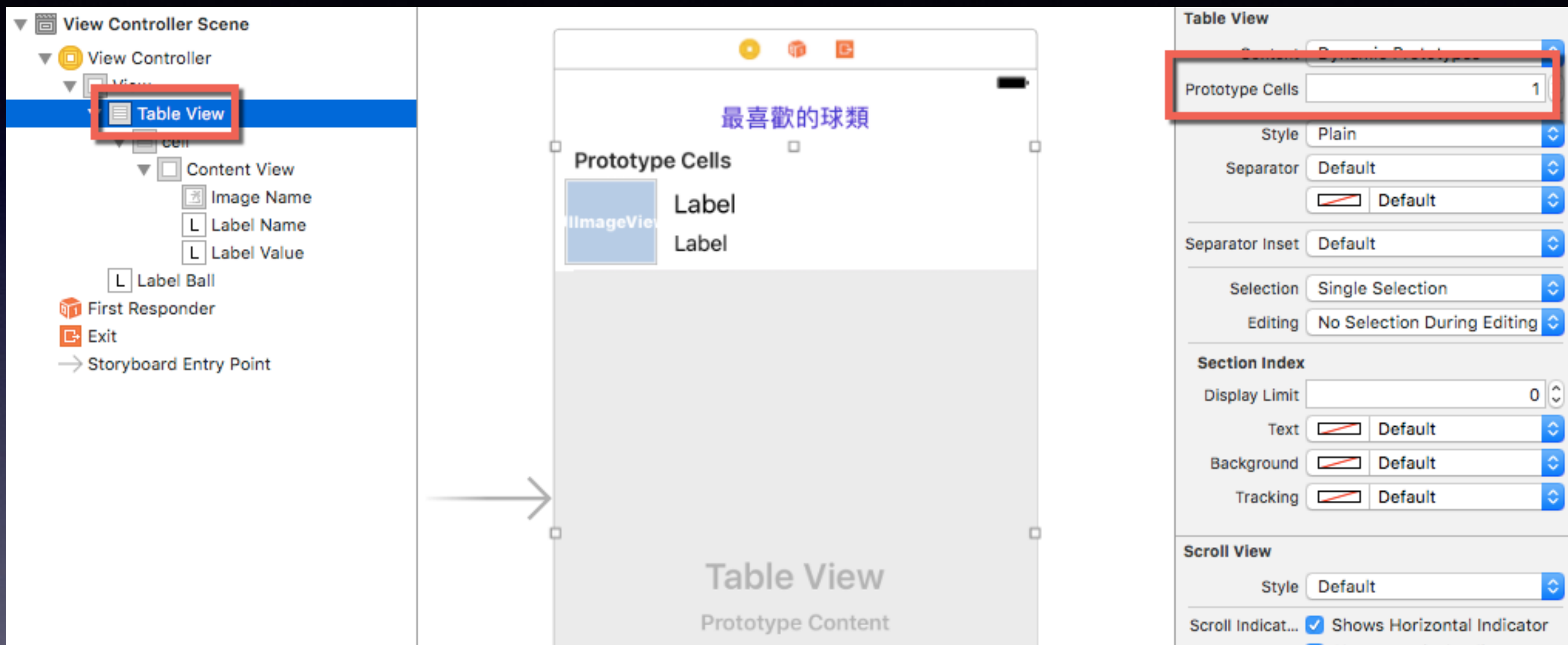
Highlighted ☐ Default

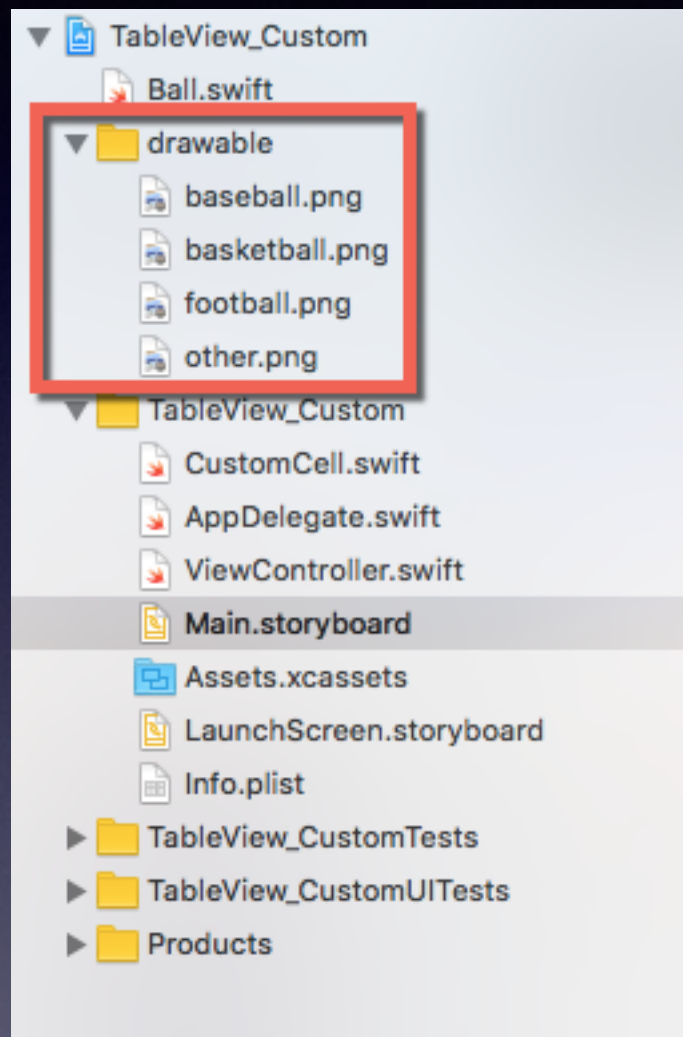
Shadow ☐ Default

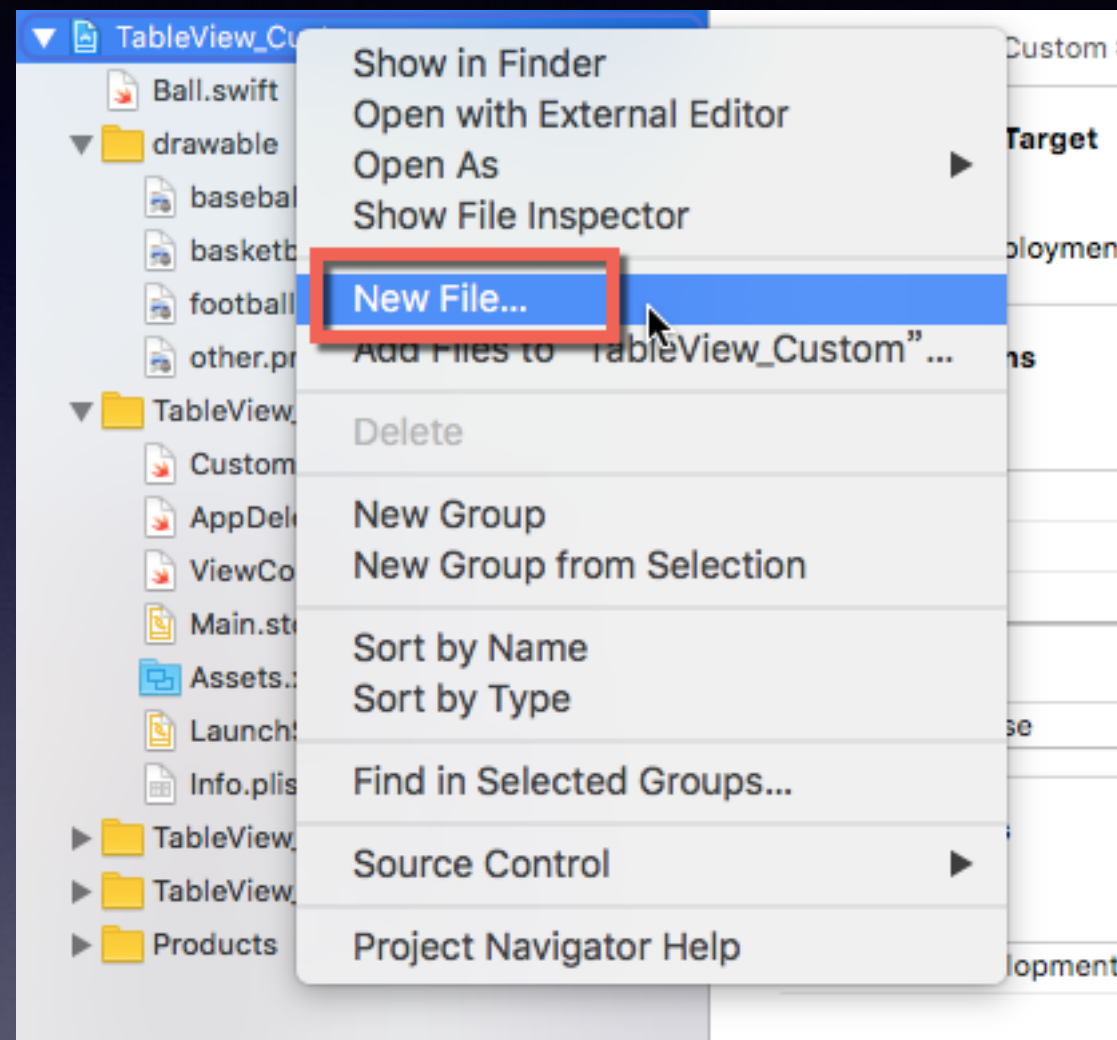
Shadow Offset 0 -1

Width Height

View







Choose a template for your new file:

iOS


watchOS

tvOS


macOS

Filter


Source




Cocoa Touch Class




UI Test Case Class




Unit Test Case Class




Playground




Swift File




Objective-C File




Header File



C File




C++ File




Metal File


User Interface




Storyboard



View



Empty



Launch Screen

Cancel

Previous

Next

Choose options for your new file:

Class:

Subclass... ▼

☐ Also create XIB file

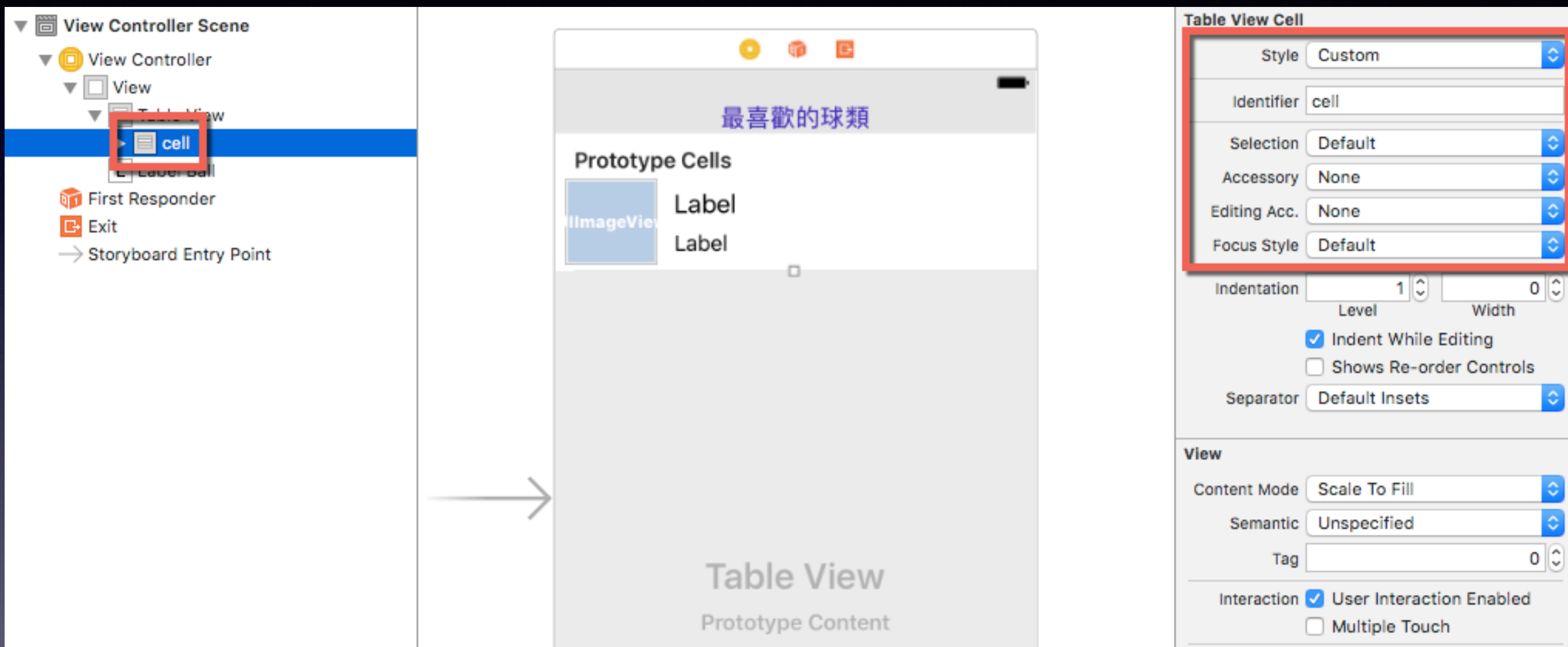
Language: ⌵

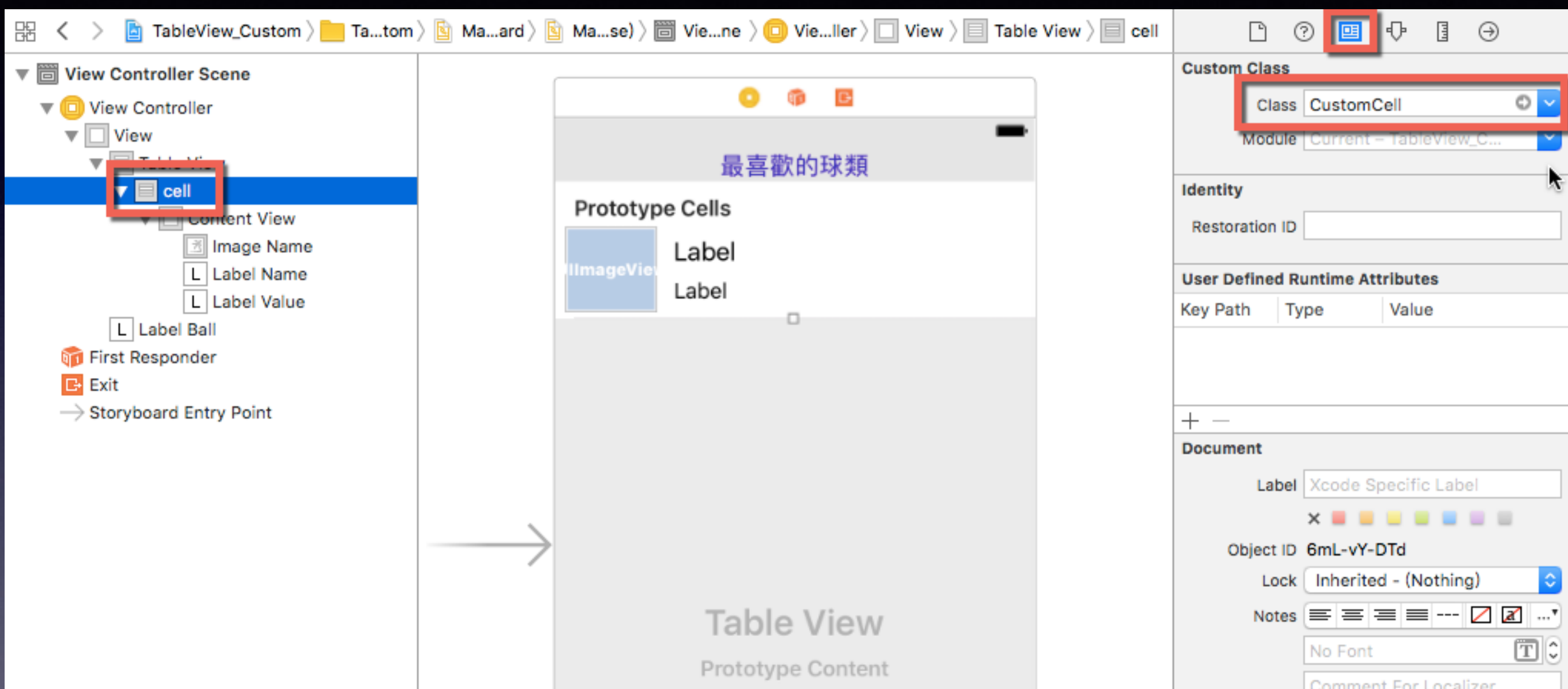
Cancel

Previous

Next

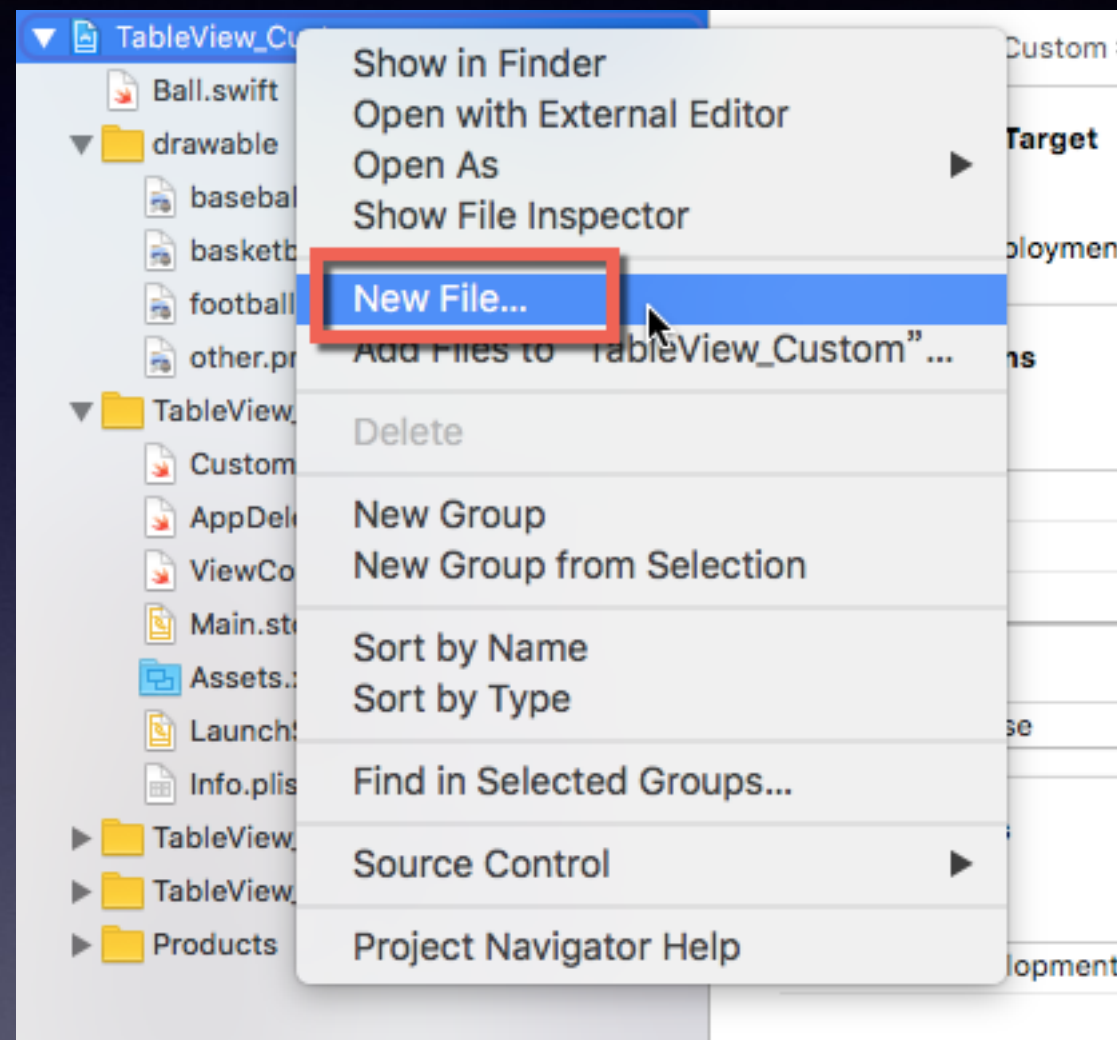
```
1 import UIKit
2
3 class CustomCell: UITableViewCell {
4     // 定義 IBOutlet
5     @IBOutlet var imageName: UIImageView!
6     @IBOutlet var labelName: UILabel!
7     @IBOutlet var labelValue: UILabel!
8
9     // 設定儲存格內容
10    func setCell(_ labelName:String, labelValue:Int, imageName:String) {
11        self.labelName.text = labelName
12        self.labelValue.text = String(labelValue)
13        self.imageName.image = UIImage(named: imageName)
14    }
15
16    override func awakeFromNib() {
17        super.awakeFromNib()
18        // Initialization code
19    }
20
21    override func setSelected(_ selected: Bool, animated: Bool) {
22        super.setSelected(selected, animated: animated)
23
24        // Configure the view for the selected state
25    }
26
27 }
28
```







```
1 import UIKit
2
3 class CustomCell: UITableViewCell {
4     // 定義 IBOutlet
5     @IBOutlet var imageName: UIImageView!
6     @IBOutlet var labelName: UILabel!
7     @IBOutlet var labelValue: UILabel!
8
9     // 設定儲存格內容
10    func setCell(_ labelName:String, labelValue:Int, imageName:
11        String) {
12        self.labelName.text = labelName
13        self.labelValue.text = String(labelValue)
14        self.imageName.image = UIImage(named: imageName)
15    }
16
17    override func awakeFromNib() {
18        super.awakeFromNib()
19        // Initialization code
20    }
21
22    override func setSelected(_ selected: Bool, animated: Bool) {
23        super.setSelected(selected, animated: animated)
24
25        // Configure the view for the selected state
26    }
27 }
28
```



Choose a template for your new file:

iOS


watchOS


tvOS


macOS


Filter


Source



Cocoa Touch Class



UI Test Case Class



Unit Test Case Class



Playground



Swift File


Objective-C File



Header File



C File



C++ File



Metal File

User Interface


Storyboard


View


Empty


Launch Screen

Cancel

Previous

Next

Save As **Ball.swift**

Tags:



TableView_Custom

Search

Favorites

- Recents
- iCloud Drive
- 應用程式
- 桌面
- 文件
- 下載項目
- 影片
- 音樂
- 圖片
- brianwang

- Ball.swift
- CustomCell.swift
- drawable
- TableView_Custom
- TableView_C...om.xcodeproj
- TableView_CustomTests
- TableView_CustomUITests

Devices

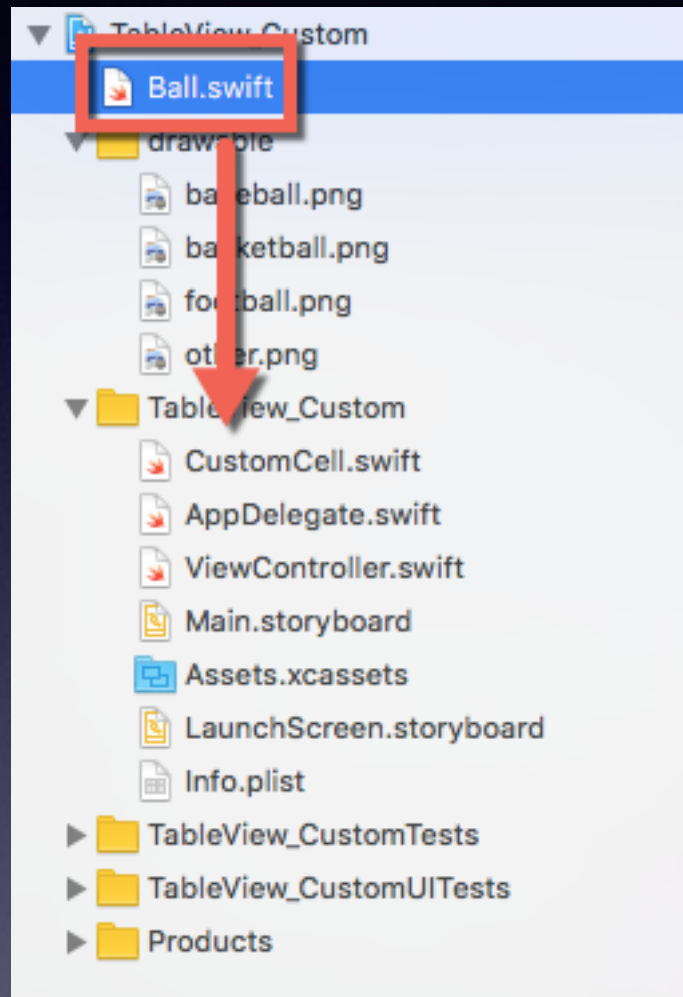
Group TableView_Custom

- Targets
- ☒ TableView_Custom
 - ☐ TableView_CustomTests
 - ☐ TableView_CustomUITests

New Folder

Cancel

Create




```
1 import Foundation
2
3 class Ball {
4
5     var name = "name"
6     var value = 0
7     var imageName = "go"
8
9     init(name:String, value:Int, imageName:String) {
10         self.name = name
11         self.value = value
12         self.imageName = imageName
13     }
14
15 }
16 |
```

```
1 import UIKit
2
3 class ViewController: UIViewController, UITableViewDataSource, UITableViewDelegate {
4
5     @IBOutlet weak var labelBall: UILabel!
6     var arrBalls:Array<Ball> = []
7
8     // 設定表格的列數
9     func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
10         return arrBalls.count
11     }
12
13     // 表格的儲存格設定
14     func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
15         // 以 CustomCell 做為 cell 類別
16         let cell:CustomCell = tableView.dequeueReusableCell(withIdentifier: "cell") as! CustomCell
17
18         if indexPath.row % 2 == 0 {
19             cell.backgroundColor = UIColor.yellow
20         } else {
21             cell.backgroundColor = UIColor.orange
22         }
23
24         // 加入項目內容
25         let ball = arrBalls[indexPath.row]
26         cell.setCell(ball.name, labelValue:ball.value, imageName:ball.imageName)
27
28         return cell
29     }
30 }
```

```
30
31 func setupBalls() {
32     let ball1 = Ball(name: "籃球", value: 600, imageName: "basketball")
33     let ball2 = Ball(name: "足球", value: 500, imageName: "football")
34     let ball3 = Ball(name: "棒球", value: 250, imageName: "baseball")
35     let ball4 = Ball(name: "其他", value: 350, imageName: "other")
36
37     arrBalls.append(ball1)
38     arrBalls.append(ball2)
39     arrBalls.append(ball3)
40     arrBalls.append(ball4)
41 }
42
43 // 點選儲存格的處理
44 func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
45     let choice:String = arrBalls[indexPath.row].name
46     labelBall.text = "最喜歡的球類：\(choice)"
47 }
48
49 override func viewDidLoad() {
50     super.viewDidLoad()
51     self.setupBalls()
52 }
53
54 override func didReceiveMemoryWarning() {
55     super.didReceiveMemoryWarning()
56     // Dispose of any resources that can be recreated.
57 }
58
59 }
60
```

建立靜態方式的儲存格

- 要以 Static Cells 靜態方式建立儲存格，必須配合 Table View Controller 才能建立
- 由於預設建立的 Single View 專案中，Main.storyboard 中預設繼承的是 View Controller
- 若在 View Controller 中建立 Static Cells 靜態方式儲存格將會產生錯誤

靜態儲存格

- 建立名稱為 TableView_Static 的 Single View 專案
- 以 Single View Application 方式建立專案後，刪除預設建立的 View Controller
- 然後從工具區中拖曳 Table View Controller，並核選 Is Initial View Controller 做為啟動執行的頁面

Table View Controller Scene

Table View Controller

- Table View
 - Table View Section
 - Table View Cell
 - Content View
 - Button
 - Table View Cell
 - Content View
 - Label
 - Table View Cell
 - Content View
 - Switch
- First Responder
- Exit
- Storyboard Entry Point

Button

Label

Switch

Table View
Static Content

Simulated Metrics

SizeFreeform

Status BarInferred

Top BarInferred

Bottom BarInferred

Table View Controller

Selection☒ Clear on Appearance

RefreshingDisabled

View Controller

Title

☒ Is Initial View Controller

Layout☒ Adjust Scroll View Insets

☐ Hide Bottom Bar on Push

☒ Resize View From NIB

☐ Use Full Screen (Deprecated)

Extend Edges☒ Under Top Bars☒ Under Bottom Bars☐ Under Opaque Bars

Navigation Controller - A controller that manages navigation through a hierarchy of views.

Table View Controller - A controller that manages a table view.

Collection View Controller - A controller that manages a collection view.

Simulated Metrics

Size	Freeform	↕
Status Bar	Inferred	↕
Top Bar	Inferred	↕
Bottom Bar	Inferred	↕

Table View Controller

Selection	<input checked="" type="checkbox"/> Clear on Appearance	
Refreshing	Disabled	↕

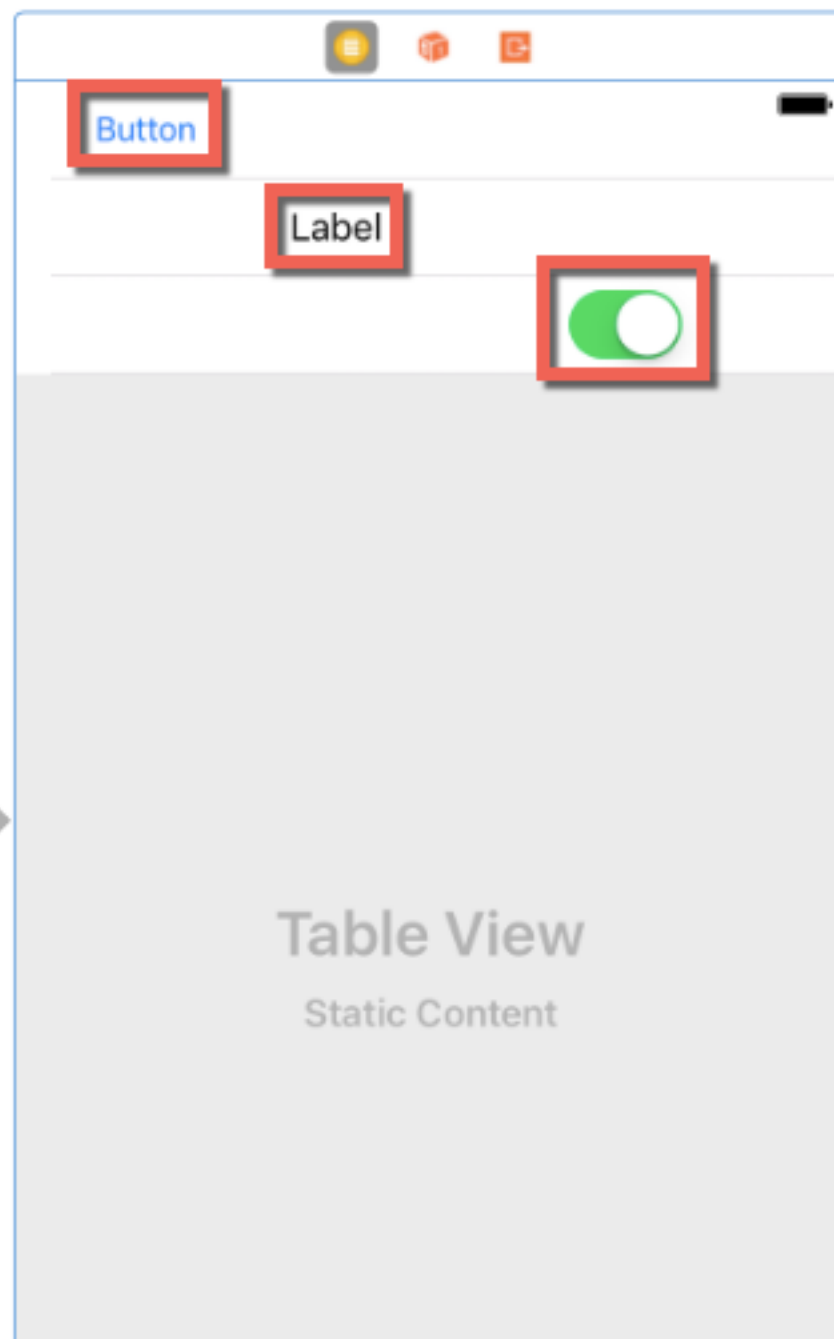
View Controller

Hide

Title		
	<input checked="" type="checkbox"/> Is Initial View Controller	
Layout	<input checked="" type="checkbox"/> Adjust Scroll View Insets	
	<input type="checkbox"/> Hide Bottom Bar on Push	
	<input checked="" type="checkbox"/> Resize View From NIB	
	<input type="checkbox"/> Use Full Screen (Deprecated)	
Extend Edges	<input checked="" type="checkbox"/> Under Top Bars	
	<input checked="" type="checkbox"/> Under Bottom Bars	
	<input type="checkbox"/> Under Opaque Bars	
Transition Style	Cover Vertical	↕

- Table View Controller 預設會建立一個 Table View，Table View 中建立一個 Table View Cell 儲存格中則包含一個 Content View
- 設定 Table View 提供的 Content 屬性為 Static Cells 靜態方式建立儲存格

- Table View Controller Scene
 - Table View Controller
 - Table View
 - Table View Section
 - Table View Cell
 - Content View
 - Button
 - Table View Cell
 - Content View
 - Label
 - Table View Cell
 - Content View
 - Switch
 - First Responder
 - Exit
 - Storyboard Entry Point



Collection View 網格元件

- Collection View 元件和 Table View 元件類似，但每列可以顯示多個儲存格
- 要以 Collection View 元件顯示資料，必須配合 Collection View Controller 才能建立

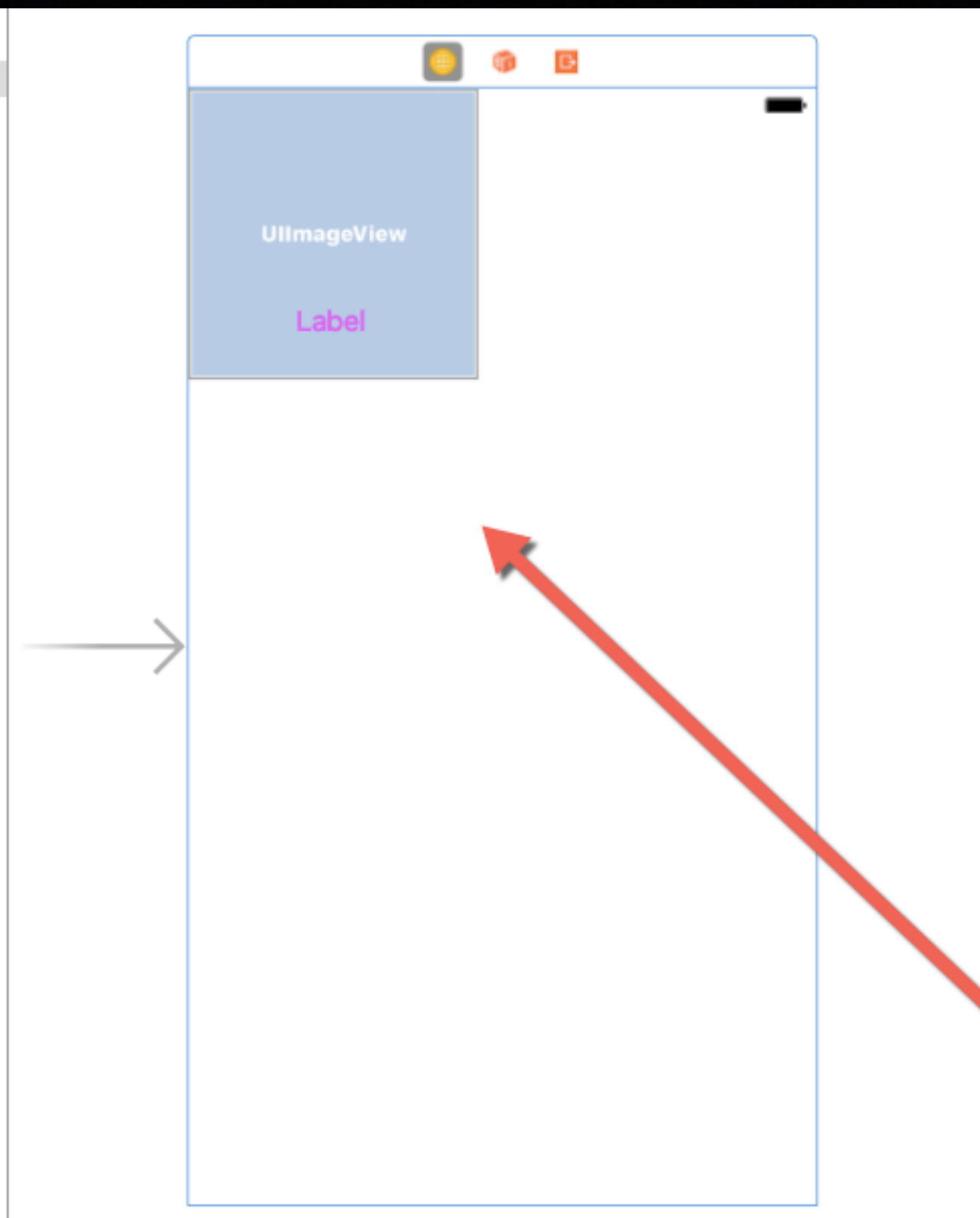
網格元件

- 建立名稱為 CollectionView 的 Single View 專案
- 以 Single View Application 方式建立專案後，刪除預設建立的 View Controller
- 然後從工具區中拖曳 Collection View Controller，並核選 Is Initial View Controller 做為啟動執行的頁面

▼ My Collection View Controller Scene

▼ My Collection View Controller

- ▼ Collection View
 - ▼ cell
 - Image Name
 - Label Name
 - Collection View Flow Layout
- First Responder
- Exit
- Storyboard Entry Point



Simulated Metrics

Size

Status Bar

Top Bar

Bottom Bar

Collection View Controller

Selection ☒ Clear on Appearance

View Controller

Title

☒ Is Initial View Controller

Layout ☒ Adjust Scroll View Insets

☐ Hide Bottom Bar on Push

☒ Resize View From NIB

☐ Use Full Screen (Deprecated)

Extend Edges ☒ Under Top Bars

☒ Under Bottom Bars

☐ Under Opaque Bars

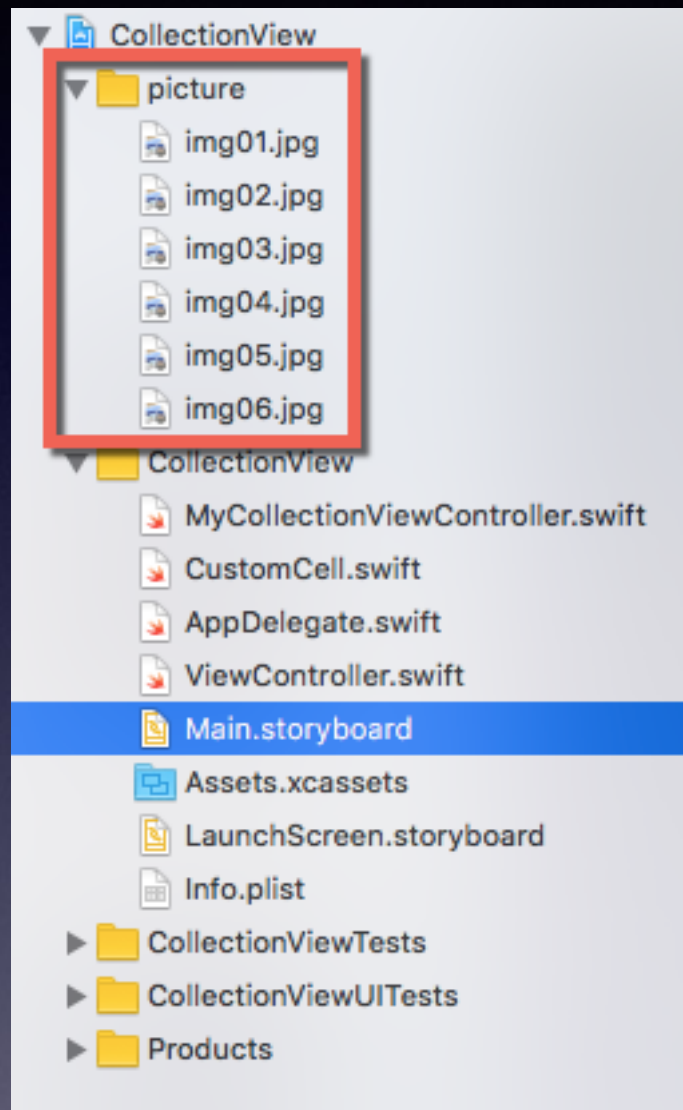
Transition Style

Presentation

Collection View Controller - A controller that manages a collection view.

Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar items.

- Collection View Controller 預設會建立一個 Collection View，Collection View 中建立一個 Collection View Cell 儲存格中則包含一個 Content View



Choose a template for your new file:

iOS


watchOS

tvOS


macOS

Filter


Source




Cocoa Touch Class




UI Test Case Class




Unit Test Case Class




Playground




Swift File




Objective-C File




Header File



C File




C++ File




Metal File


User Interface




Storyboard



View



Empty



Launch Screen

Cancel

Previous

Next

Choose options for your new file:

Class: CustomCell

Subclass... UICollectionViewCell

☐ Also create XIB file

Language: Swift

Cancel

Previous

Next

```
1 import UIKit
2
3 class CustomCell: UICollectionViewCell {
4
5     // 定義 IBOutlet
6     @IBOutlet weak var imageName: UIImageView!
7     @IBOutlet weak var labelName: UILabel!
8
9 }
10
```

My Collection View Controller Scene

- My Collection View Controller
 - Collection View
 - cell
 - Image Name**
 - Label Name**
 - Collection View Flow Layout
 - First Responder
 - Exit
 - Storyboard Entry Point



Label

Text Plain

Label

Color

Font System 17.0

Alignment

Lines 1

Behavior ☒ Enabled ☐ Highlighted

Baseline Align Baselines

Line Break Truncate Tail

Autoshrink Fixed Font Size

☐ Tighten Letter Spacing

Highlighted Default

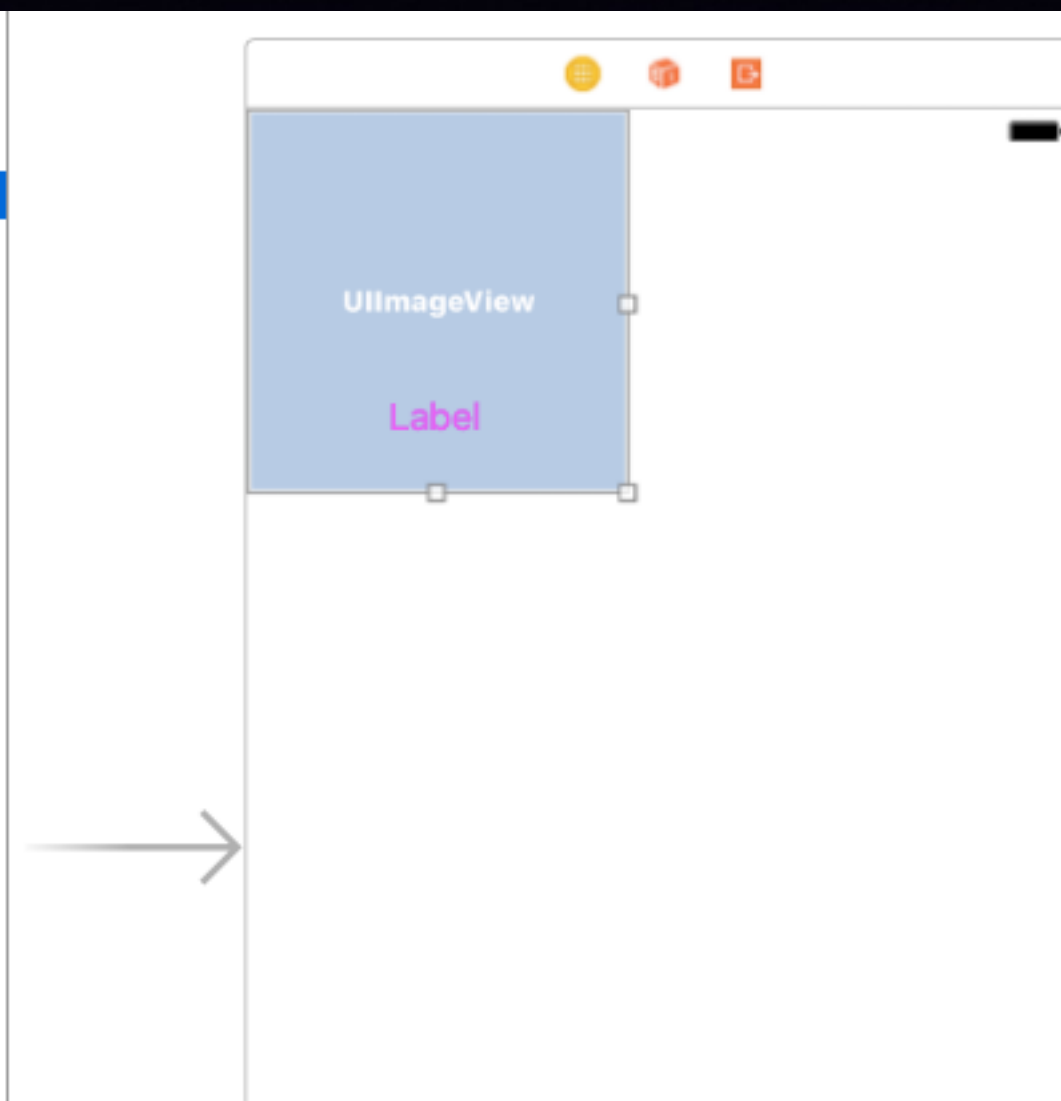
Shadow Default

Shadow Offset 0 -1
Width Height

View

My Collection View Controller Scene

- My Collection View Controller
 - Collection View
 - cell**
 - Image Name
 - Label Name
 - Collection View Flow Layout
 - First Responder
 - Exit
 - Storyboard Entry Point



Collection View Cell

Identifier: cell

View

Content Mode: Center

Semantic: Unspecified

Tag: 0

Interaction: ☒ User Interaction Enabled, ☒ Multiple Touch

Alpha: 1

Background: [gradient bar]

Tint: [blue bar] Default

Drawing: ☐ Opaque, ☐ Hidden, ☒ Clears Graphics Context, ☒ Clip To Bounds, ☒ Autoresize Subviews

Stretching: X (0 to 1), Y (0 to 1)

Custom Class

Class

CustomCell

Module

Current - CollectionVi...

Identity Hide

Restoration ID

User Defined Runtime Attributes

Key Path	Type	Value
----------	------	-------

+ -

Document

Label

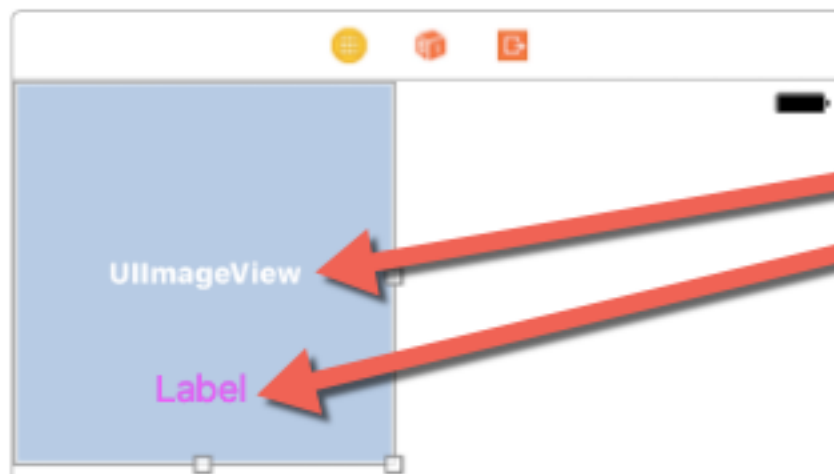
×

Object ID 1LT-Zw-OsW

Lock

Notes

Comment For Localizer



```
1 import UIKit
2
3 class CustomCell: UICollectionViewCell {
4
5     // 定義 IBOutlet
6     @IBOutlet weak var imageName: UIImageView!
7     @IBOutlet weak var labelName: UILabel!
8
9 }
10 |
```

Choose a template for your new file:

iOS


watchOS

tvOS


macOS

Filter


Source




Cocoa Touch Class




UI Test Case Class




Unit Test Case Class




Playground




Swift File




Objective-C File




Header File



C File




C++ File




Metal File


User Interface




Storyboard



View



Empty



Launch Screen

Cancel

Previous

Next

▼ My Collection View Controller Scene

▼ My Collection View Controller

Collection View

cell

Image Name

Label Name

Collection View Flow Layout

First Responder

Exit

→ Storyboard Entry Point

UIImageView

Label

Custom Class

Class MyCollectionViewCont... ▼

Module Current - CollectionVi... ▼

Identity

Storyboard ID

Restoration ID

☐ Use Storyboard ID

User Defined Runtime Attributes

Key Path	Type	Value
----------	------	-------

+ -

Document

Label Xcode Specific Label

Object ID PCs-6w-HBQ

Lock Inherited - (Nothing) ▾

Notes

```

1 import UIKit
2
3 class MyCollectionViewController: UICollectionViewController {
4
5     @IBOutlet var collectionView: UICollectionView!
6
7     var imgAry = ["img01.jpg", "img02.jpg", "img03.jpg", "img04.jpg", "img05.jpg", "img06.jpg"]
8
9     // 設定表格的數目
10    override func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
11        return imgAry.count
12    }
13
14    // 儲存格設定
15    override func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
16        // 取得 CustomCell 自訂類別的 cell 儲存格
17        let cell = collectionView.dequeueReusableCell(withReuseIdentifier: "cell", for: indexPath) as! CustomCell
18
19        cell.labelName.text = "編號: \(indexPath.row)"
20        cell.imageName.image = UIImage(named: imgAry[indexPath.row])
21
22        return cell
23    }
24
25    // 點選儲存格的處理
26    override func collectionView(_ collectionView: UICollectionView, didSelectItemAt indexPath: IndexPath) {
27        print("選取 \(imgAry[indexPath.row]) 圖片")
28    }
29
30
31    override func viewDidLoad() {
32        super.viewDidLoad()
33    }
34
35    override func didReceiveMemoryWarning() {
36        super.didReceiveMemoryWarning()
37        // Dispose of any resources that can be recreated.
38    }
39
40 }
41

```

Choose options for your new file:

Class: MyCollectionViewController

Subclass... UICollectionViewController

☐ Also create XIB file

Language: Swift

Cancel

Previous

Next