Class 4


TAIWAN
TECH
National Taiwan University of
Science and Technology


HPCLab
High-Performance Computing Laboratory

Table View

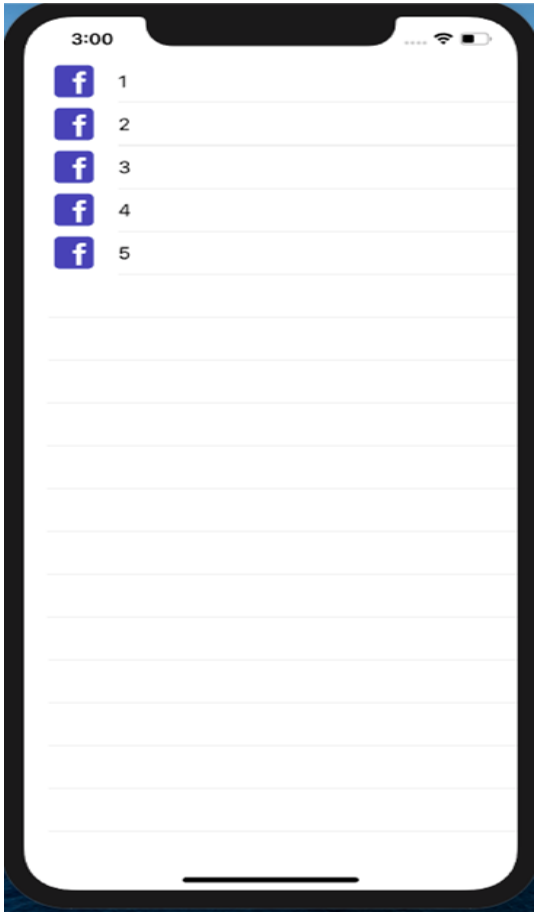


Table View

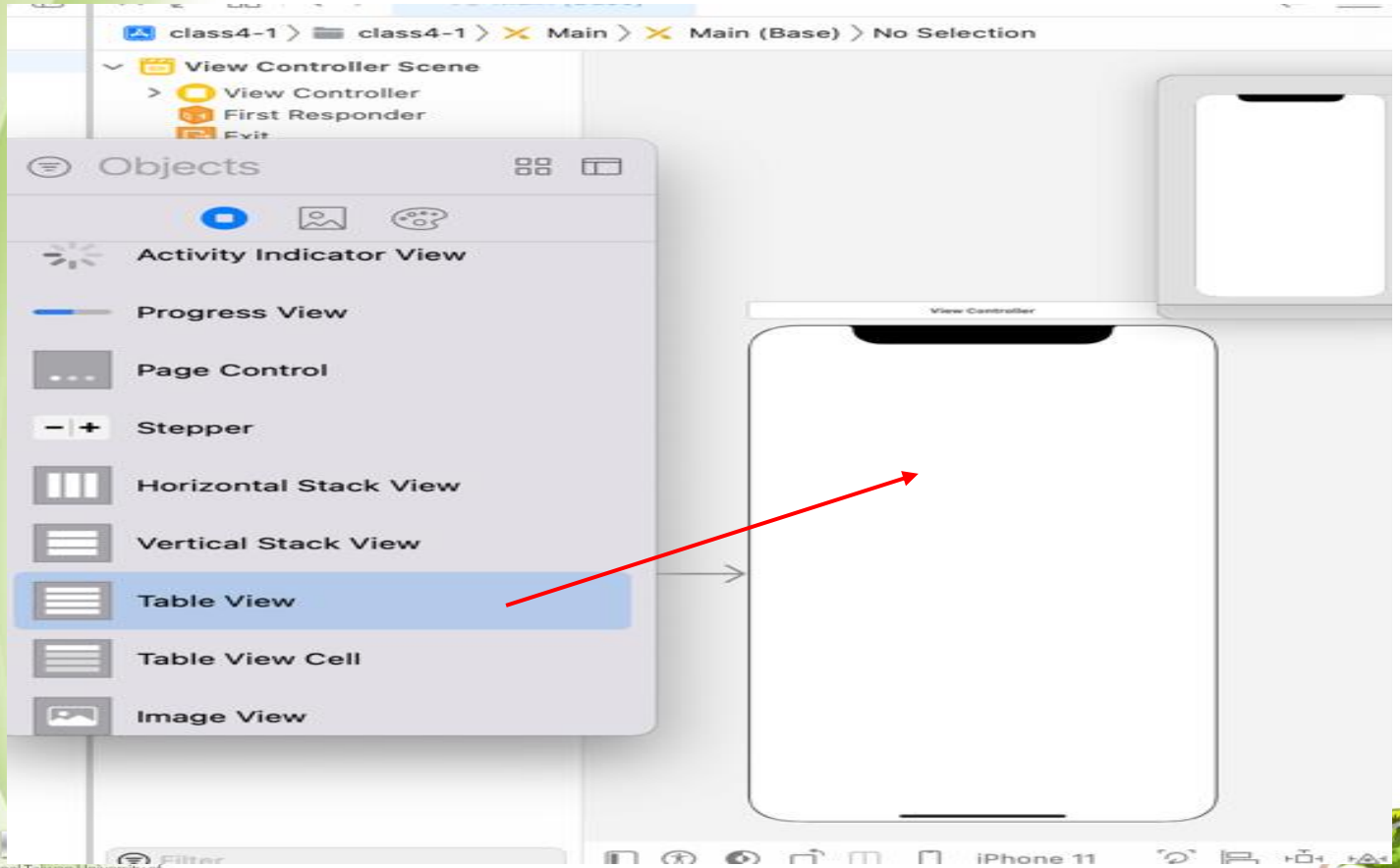

class4-1 > class4-1 > Main > Main (Base) > No Selection
View Controller Scene
View Controller
First Responder
Objects
Activity Indicator View
Progress View
Page Control
Stepper
Horizontal Stack View
Vertical Stack View
Table View
Table View Cell
Image View
iPhone 11

Table View

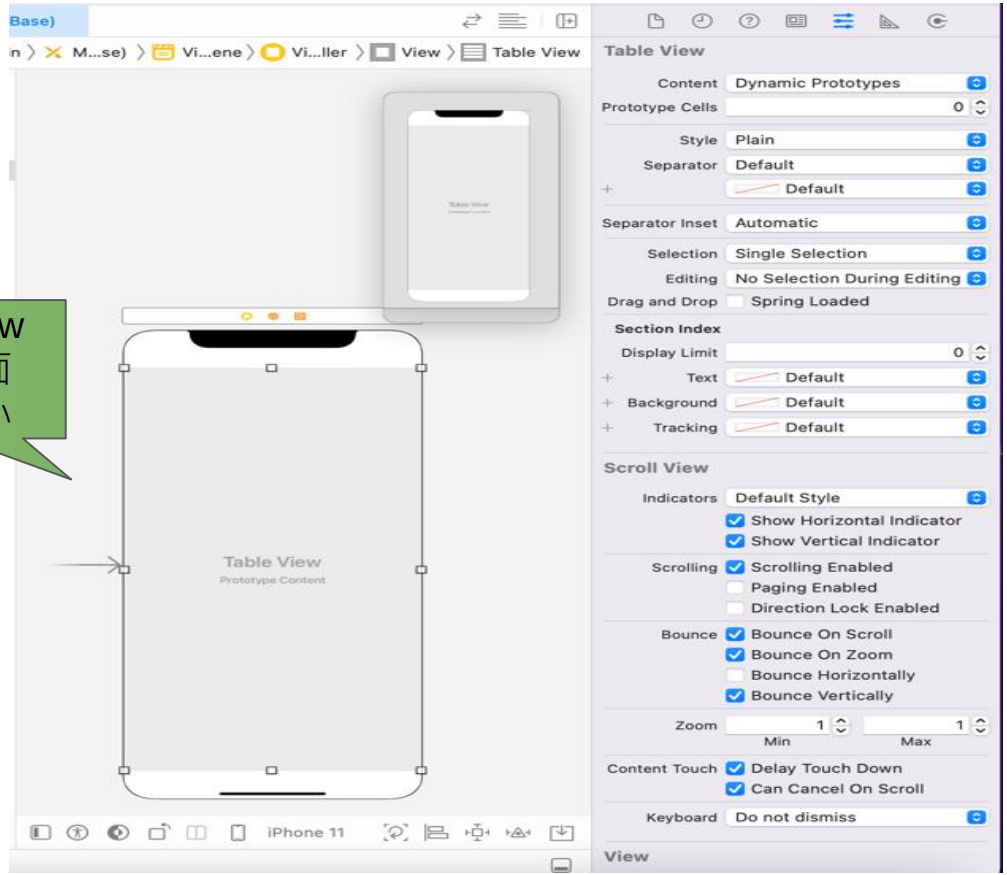


Table View

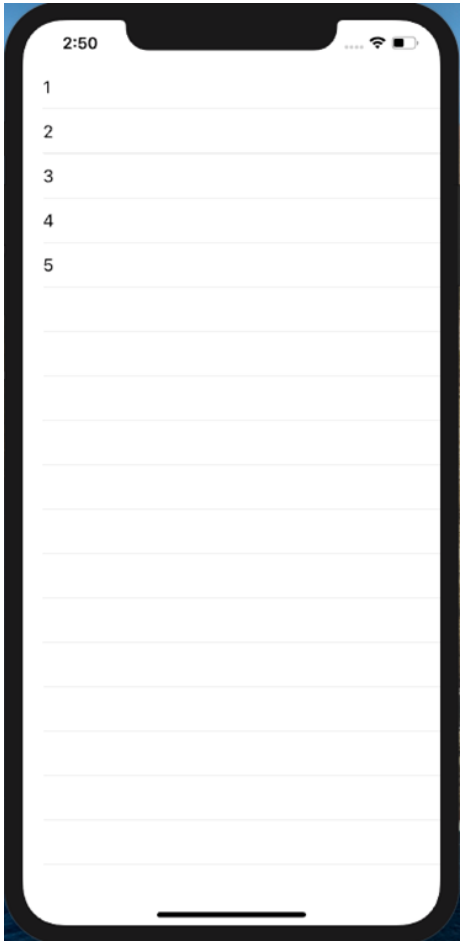


Table View

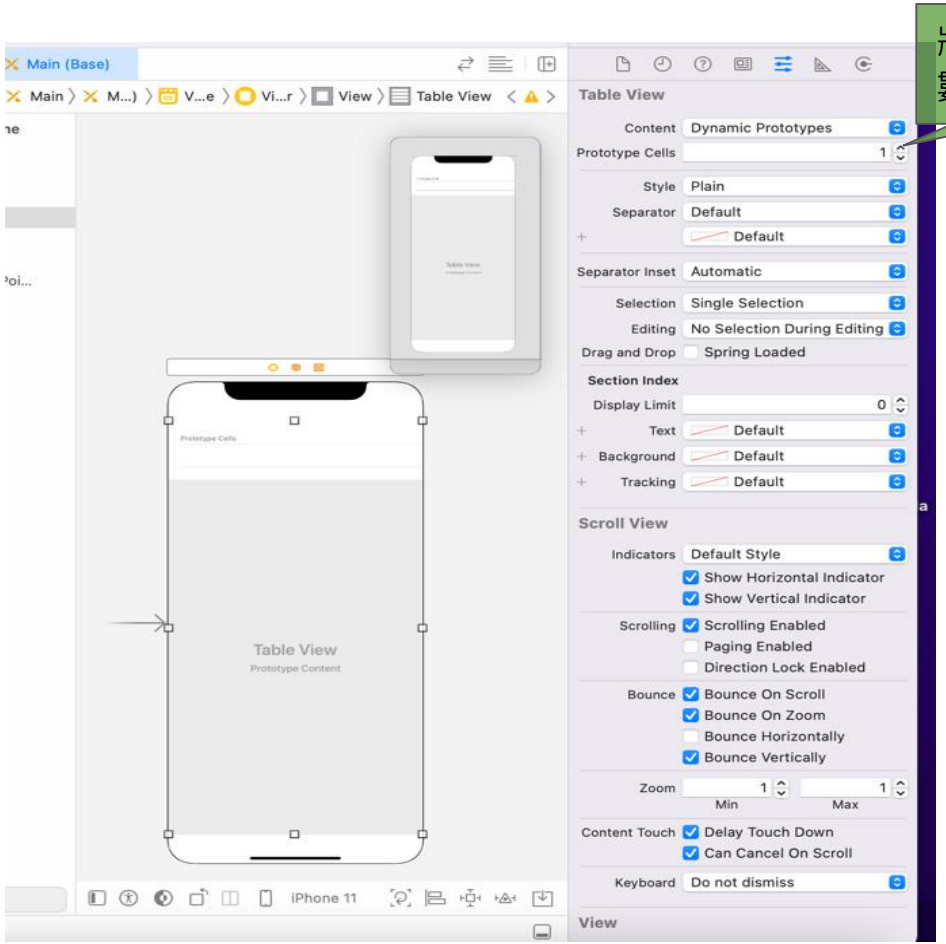

將cell的個數增加
Main (Base)
Main
M...)
V...e
Vi...r
View
Table View
Table View
Content
Dynamic Prototypes
Prototype Cells
1
Style
Plain
Separator
Default
Default
Separator Inset
Automatic
Selection
Single Selection
Editing
No Selection During Editing
Drag and Drop
Spring Loaded
Section Index
Display Limit
0
Text
Default
Background
Default
Tracking
Default
Scroll View
Indicators
Default Style
Show Horizontal Indicator
Show Vertical Indicator
Scrolling
Scrolling Enabled
Paging Enabled
Direction Lock Enabled
Bounce
Bounce On Scroll
Bounce On Zoom
Bounce Horizontally
Bounce Vertically
Zoom
1
1
Min
Max
Content Touch
Delay Touch Down
Can Cancel On Scroll
Keyboard
Do not dismiss
View
Table View
Prototype Content
iPhone 11

Table View

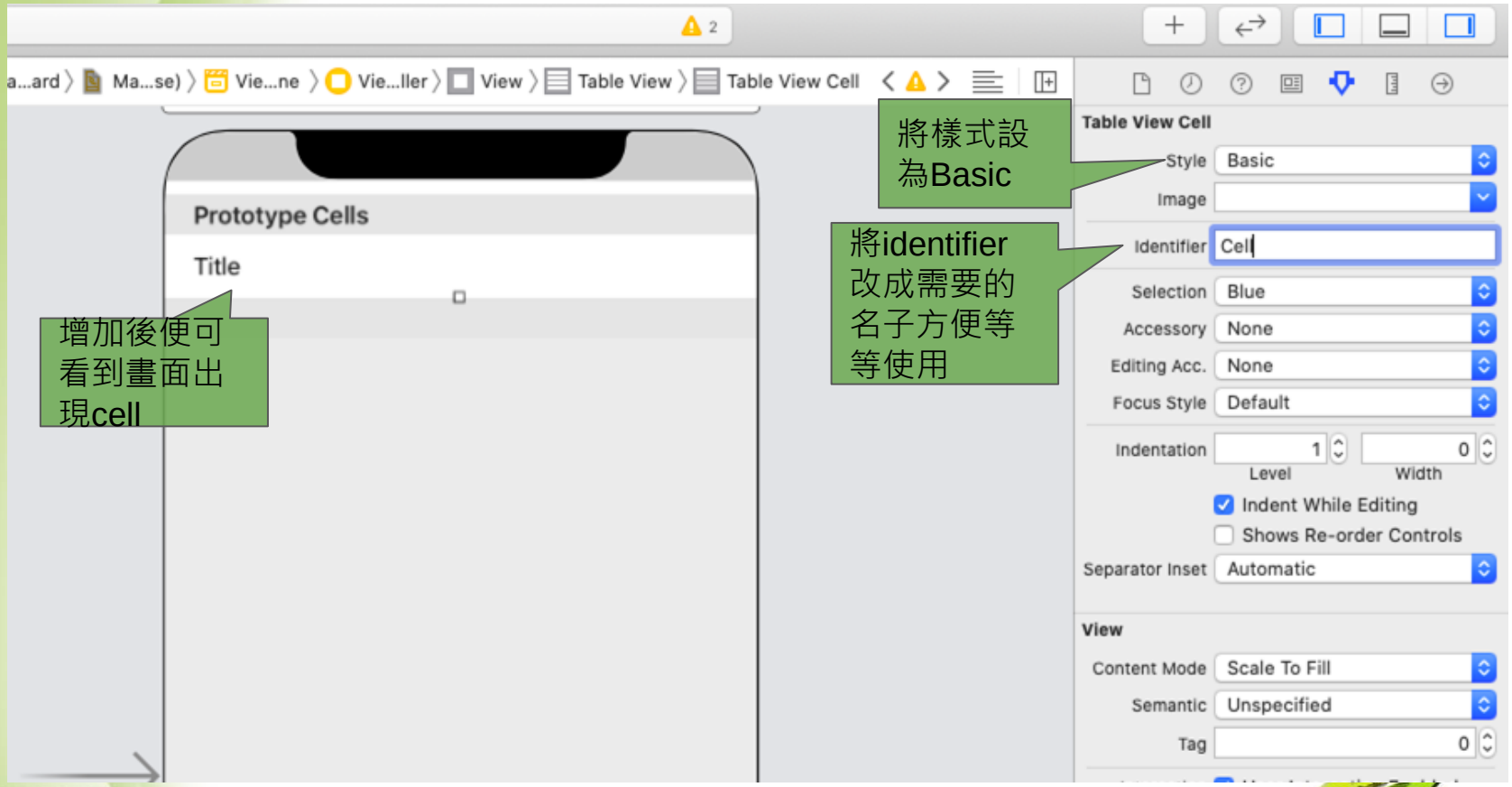

Prototype Cells
Title
增加後便可看到畫面出現cell
將樣式設為Basic
將identifier改成需要的名子方便等等使用
Table View Cell
Style
Basic
Image
Identifier
Cell
Selection
Blue
Accessory
None
Editing Acc.
None
Focus Style
Default
Indentation
1
Level
0
Width
Indent While Editing
Shows Re-order Controls
Separator Inset
Automatic
View
Content Mode
Scale To Fill
Semantic
Unspecified
Tag
0

Table View


Main (Base)
ViewController
class4-1 > class4-1 > ViewController > ViewController
1 //
2 // ViewController.swift
3 // class4-1
4 //
5 // Created by nick on 2022/3/4.
6 //
7
8 import UIKit
9
10 class ViewController:
UIViewController,UITableViewDataSource,UITableViewDelegate
{
11
12 override func
13 super.view
14 // Do any
15 }
16
17
18 }
19
20
Type 'ViewController' does not conform to protocol 'UITableViewDataSource'
Do you want to add protocol stubs?
Fix
如同PickView一樣需要透過Delegate與DataSource來使用
按下Fix便能知道要實作哪些必要函式

Table View

```
import UIKit

class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
    var names = ["1","2","3","4","5"]

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return names.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
        UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)
        cell.textLabel?.text = names[indexPath.row]
        return cell
    }
```

宣告一個字串陣列等等將他輸出到tableView

每個section要有幾個row但這裡只有一個

使用變數去取得在tableView上的cell

將每個欄位的Label設置成陣列的字串

tableView有一種回收機制，當你在滾動畫面使得欄位超出畫面時並不會創建新的欄位給你裝新的資料而是回收舊的欄位給你使用

Table View

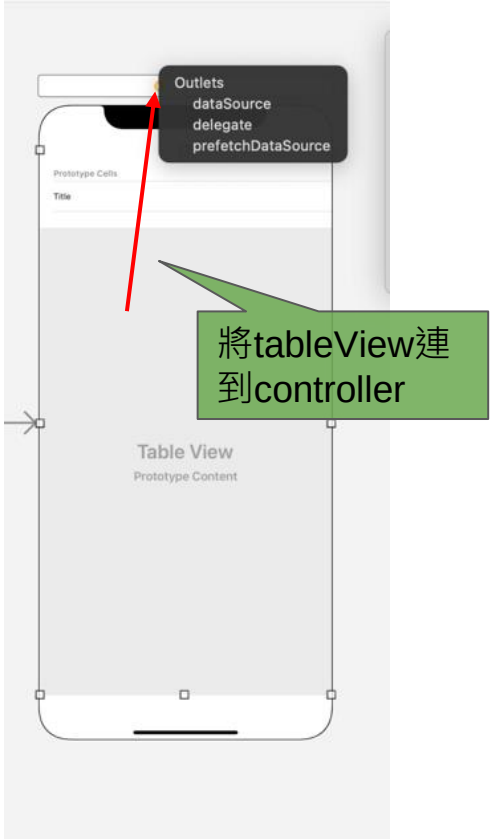

Outlets
dataSource
delegate
prefetchDataSource
Prototype Cells
Title
將tableView連到controller
Table View
Prototype Content

Table View

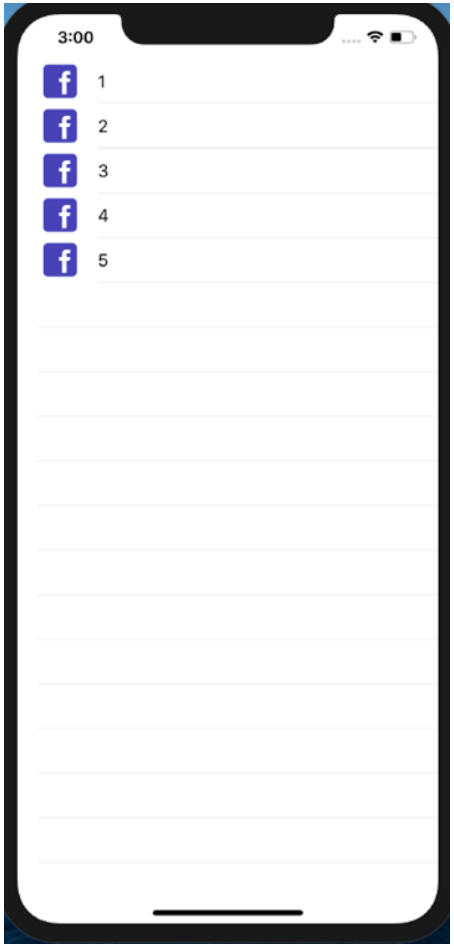


Table View

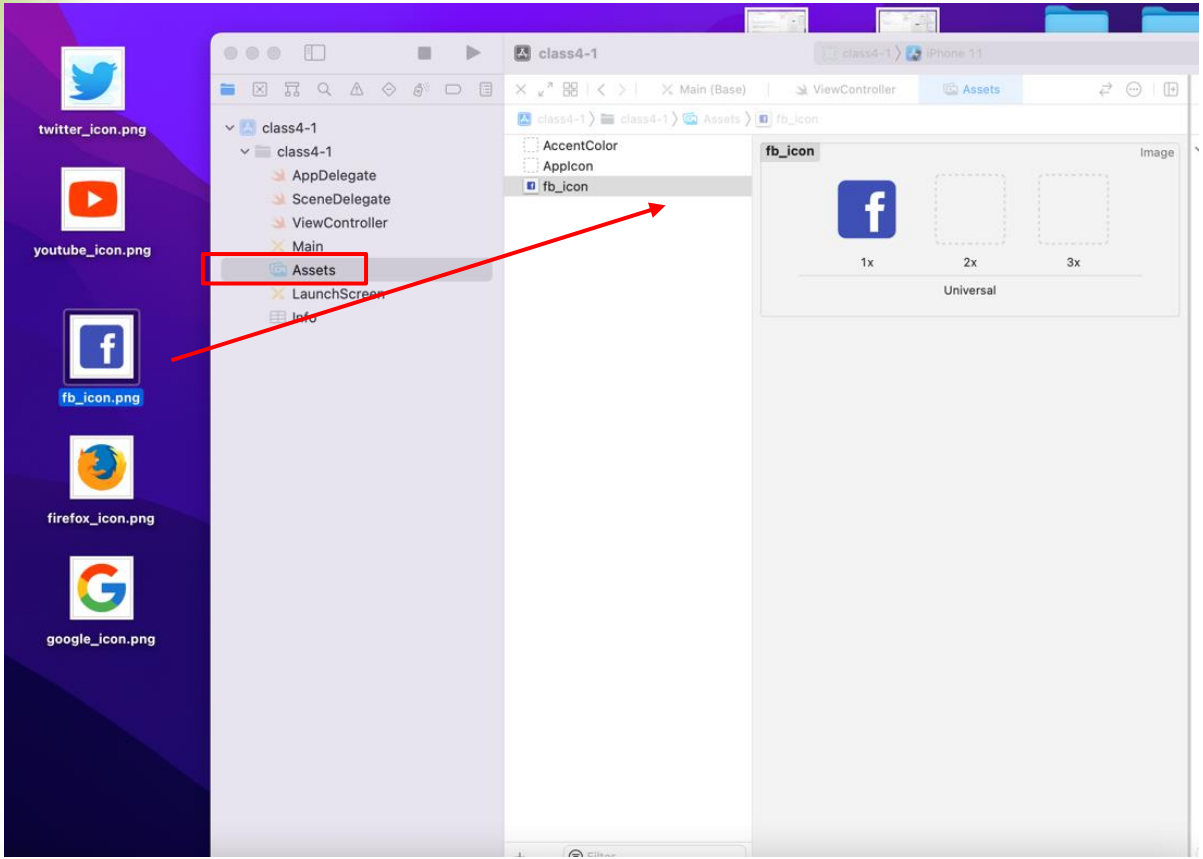

twitter_icon.png
youtube_icon.png
fb_icon.png
firefox_icon.png
google_icon.png
class4-1
class4-1
AppDelegate
SceneDelegate
ViewController
Main
Assets
LaunchScreen
Info
AccentColor
AppIcon
fb_icon
fb_icon
Image
1x
2x
3x
Universal

Table View

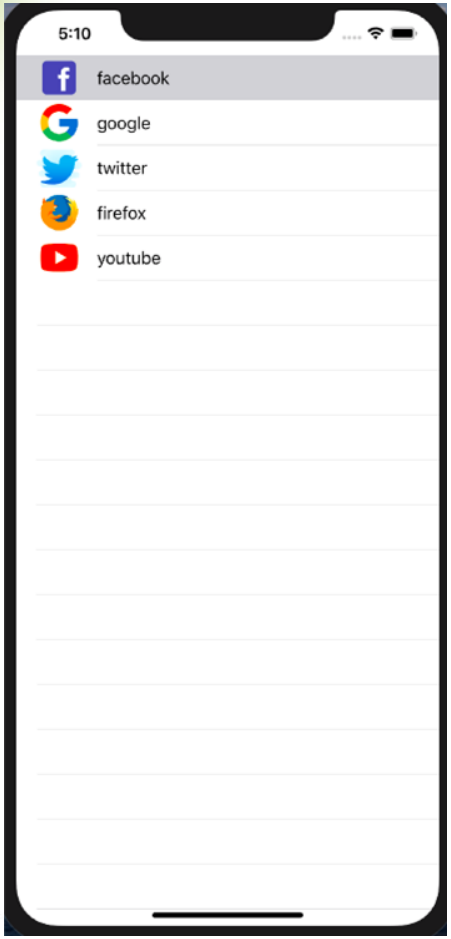
Table View 〉 Table View 〉 ViewController.swift 〉 tableView(_:cellForRowAt:)

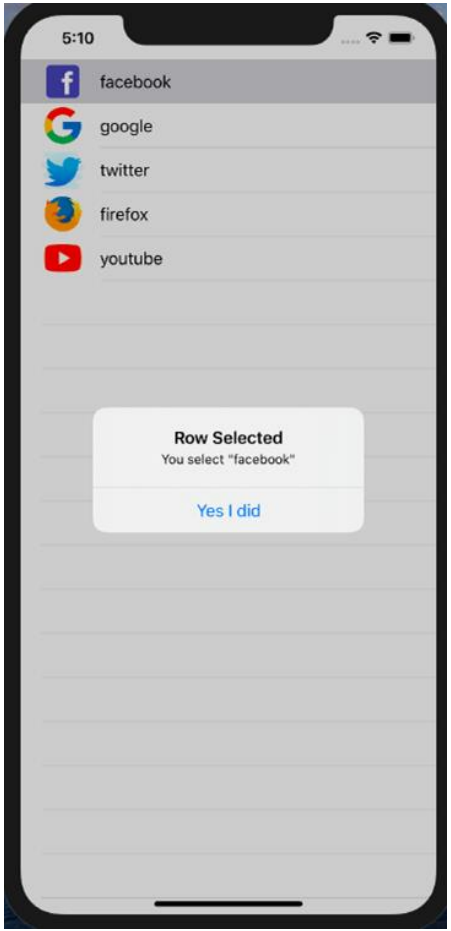
有可用的更新項目
您要立即安裝更新項目嗎？

```
7  //
8
9  import UIKit
10
11 class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
12     var names = ["1","2","3","4","5"]
13
14     func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
15         return names.count
16     }
17
18     func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
           UITableViewCell {
19         let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)
20         cell.textLabel?.text = names[indexPath.row]
21         cell.imageView?.image = UIImage(named: "facebook")
22         return cell
23     }
24
```

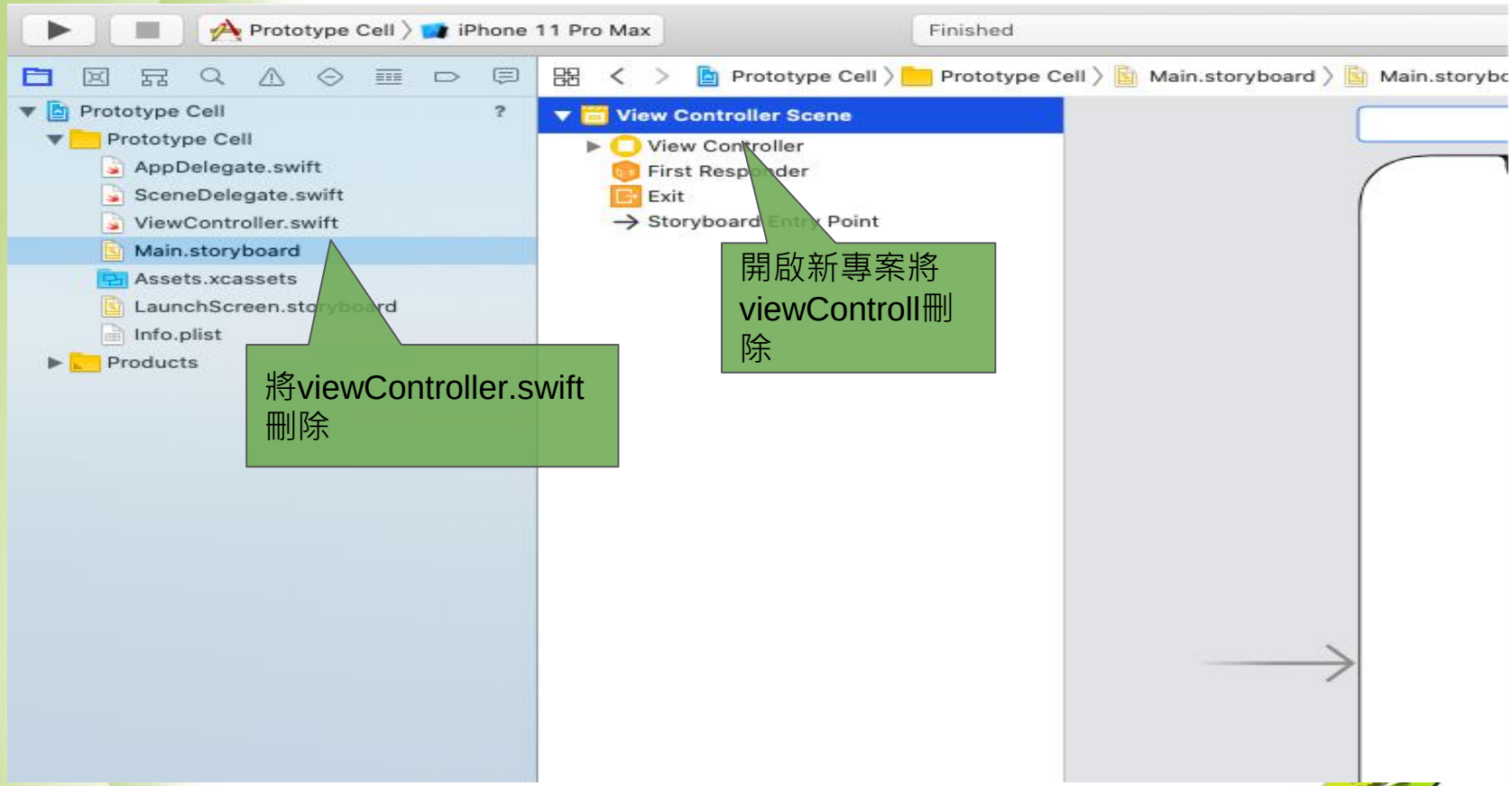
將圖案新增到欄位上

TableViewController

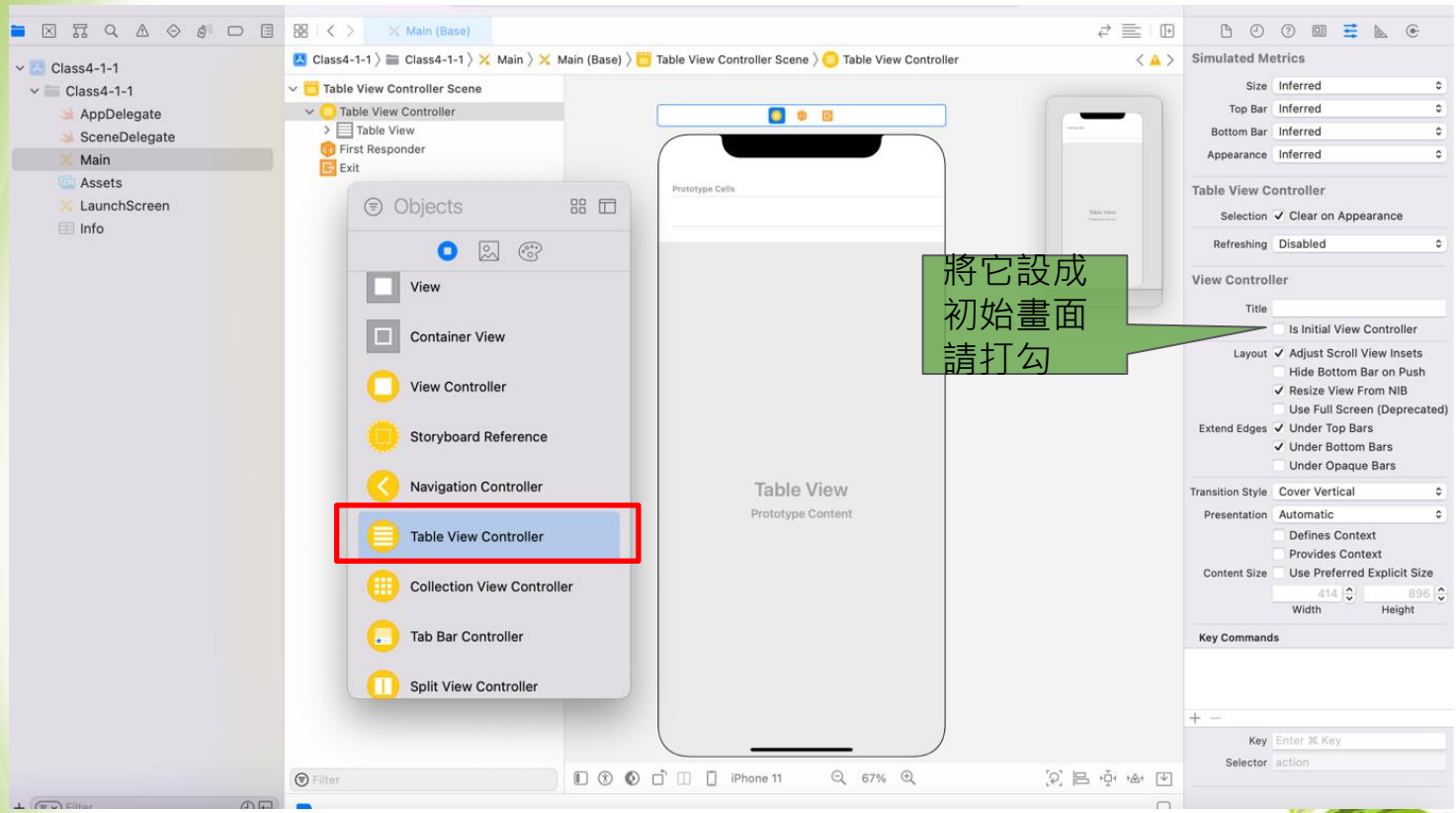




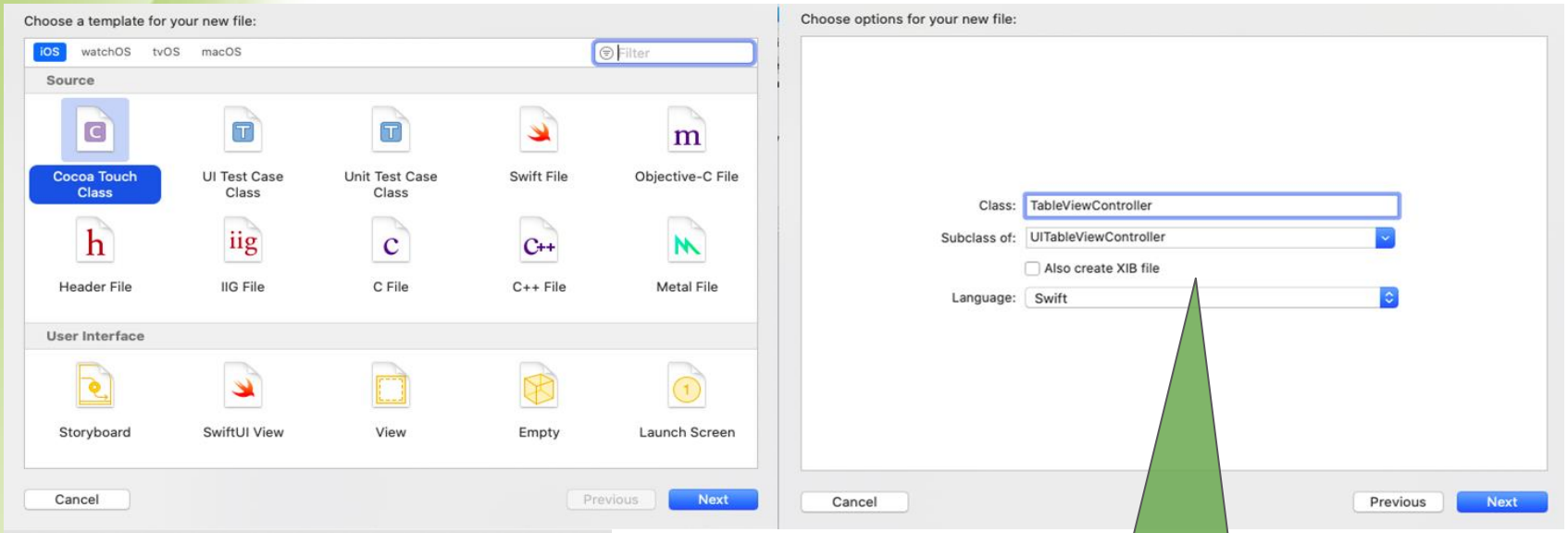
TableViewController

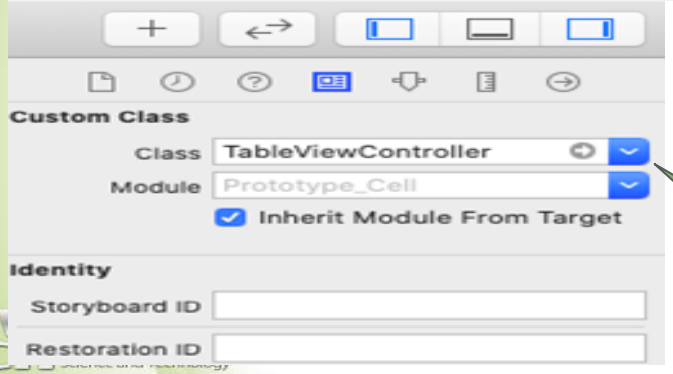


TableViewController



TableViewController

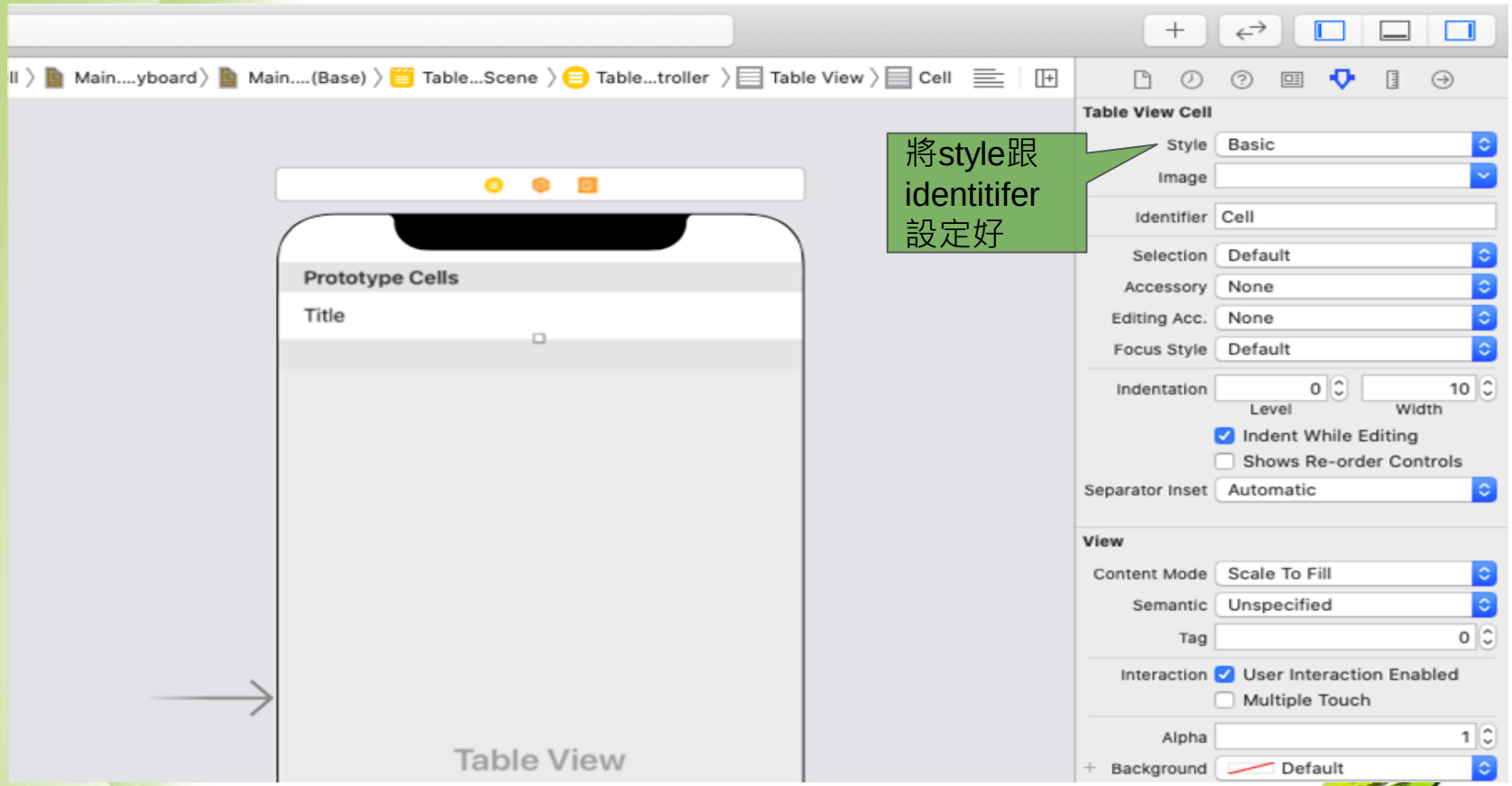




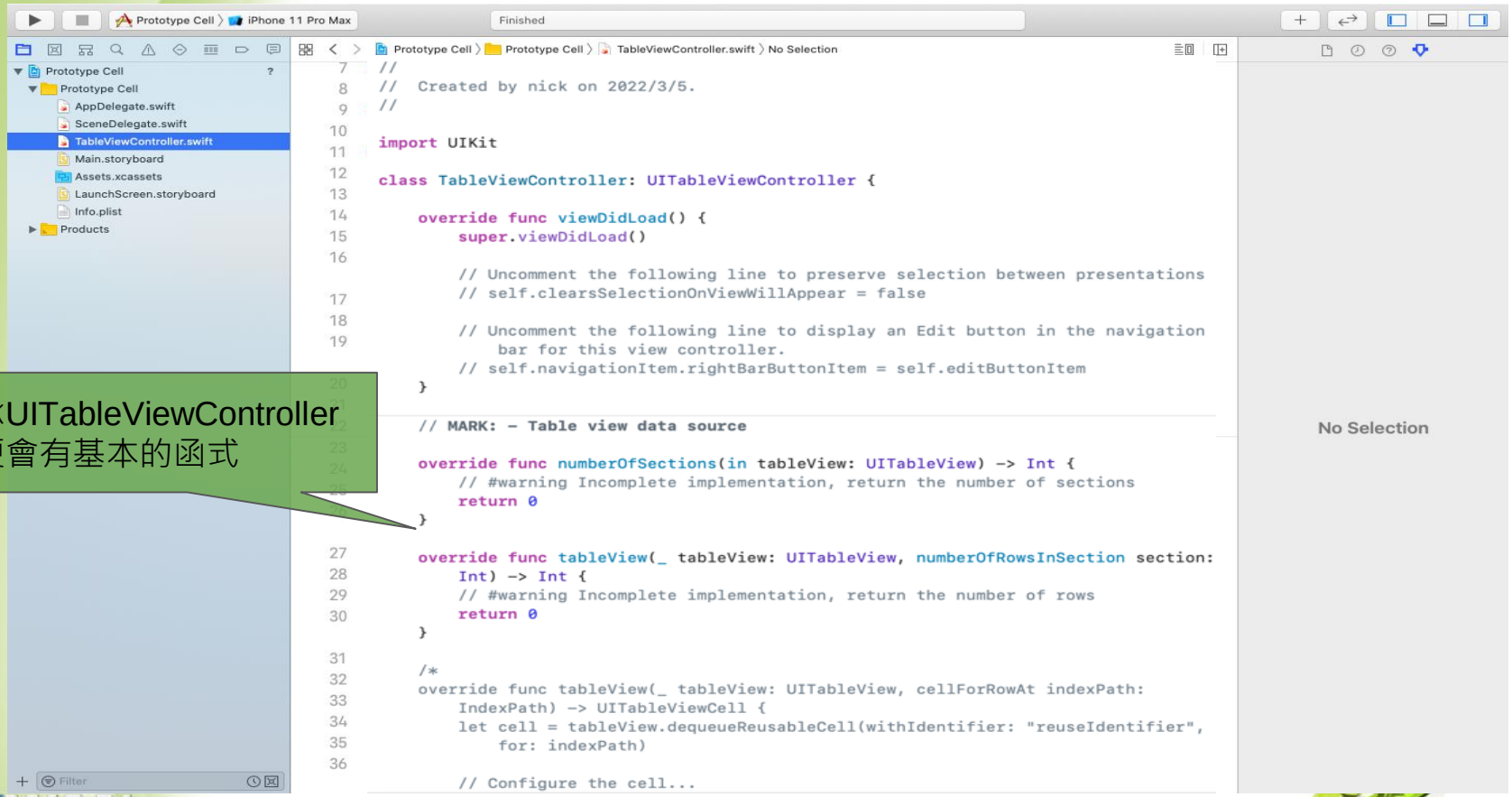
創新檔案將它繼承
UITableViewController

將class
設置好

TableViewController



TableViewController



繼承UITableViewController後便會有基本的函式

TableViewController

```swift
var names = ["facebook","google","twitter","firefox","youtube"]
var imgs = ["fb_icon","google_icon","twitter_icon","firefox_icon","youtube_icon"]
```

宣告名字與圖片名稱陣列

```swift
override func numberOfSections(in tableView: UITableView) -> Int {
    // #warning Incomplete implementation, return the number of sections
    return 1
}
```

回傳有幾個section

```swift
override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    // #warning Incomplete implementation, return the number of rows
    return names.count
}
```

每個section有幾個row

```swift
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
    UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)
    cell.textLabel?.text = names[indexPath.row]
    cell.imageView?.image = UIImage(named: imgs[indexPath.row])

    return cell
}
```

基本都與之前相同差別在圖片是用陣列來決定

TableViewController

```
override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    let alert = UIAlertController(title: "Row Selected", message: "You select \"\(names[indexPath.row])\"",
        preferredStyle: .alert)
    let action = UIAlertAction(title: "Yes I did", style: .default, handler: nil)
    alert.addAction(action)

    present(alert, animated: true, completion: nil)
}
```

此函式是指當選則row時會觸發並且能從"indexPath.row"來判斷是哪個被選

選到後便跳出訊息

TableViewController

```
override func viewDidLoad() {
    super.viewDidLoad()
    tableView.isEditing = true

    // Uncomment the following line to preserve selection between pres
    // self.clearsSelectionOnViewWillAppear = false

    // Uncomment the following line to display an Edit button in the n
        view controller.
    // self.navigationItem.rightBarButtonItem = self.editButtonItem
}
```

開啟編輯狀態
(通常是用按鈕
來觸發開關)

TableViewController

```
// Override to support editing the table view.
override func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt
    indexPath: IndexPath) {
    if editingStyle == .delete {
        names.remove(at: indexPath.row)
        imgs.remove(at: indexPath.row)
        tableView.deleteRows(at: [indexPath], with: .fade)
        tableView.reloadData()
    } else if editingStyle == .insert {
        // Create a new instance of the appropriate class, insert it into the array, and add a new row to the
            view
    }
}
```

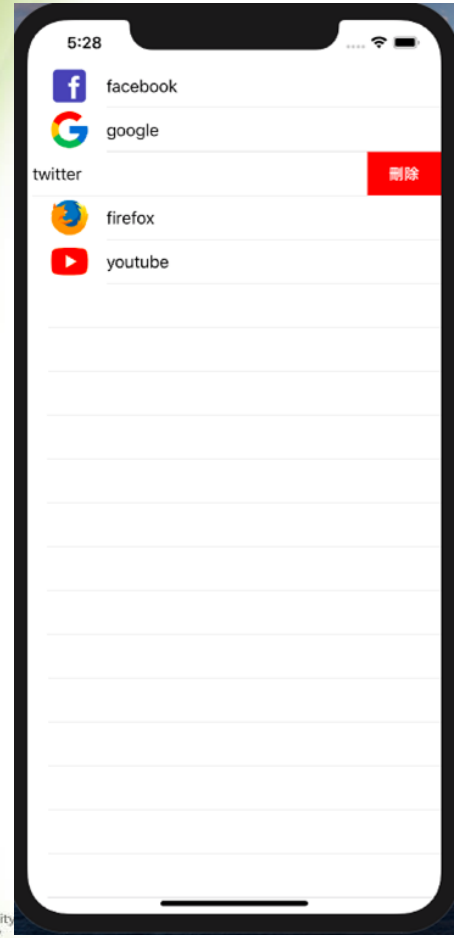
將元素從陣列中移除不然資料對不上會出錯

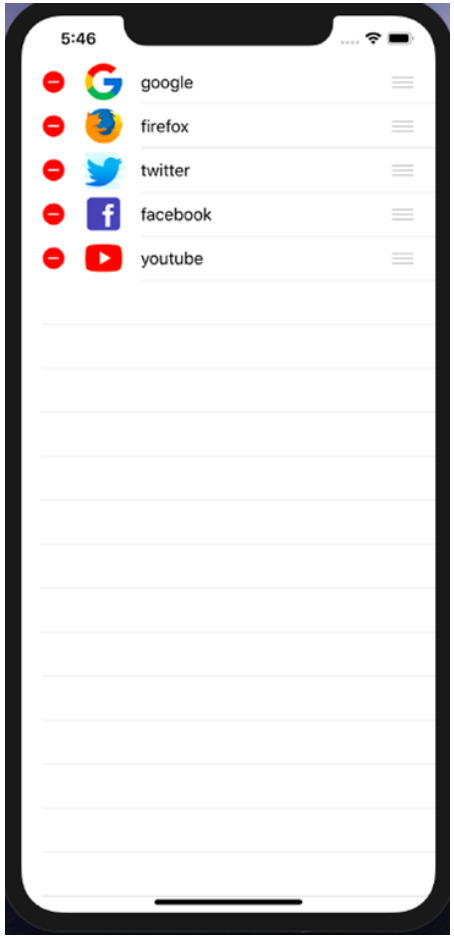
此函式在下面是註解的可以直接解開註解使用，裡面是編輯時要做的事

此行是將某一欄刪掉

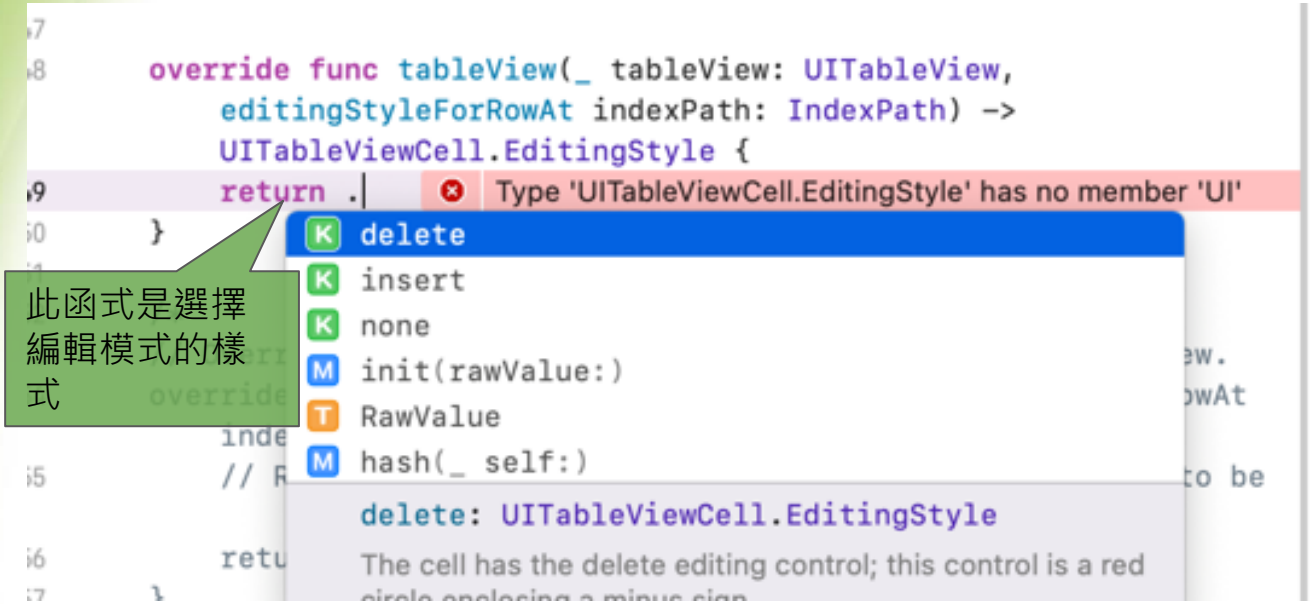
當更新資料時通常會將tableView刷新，但這裡其實已經幫你刷新過了因此可加可不加

TableViewController



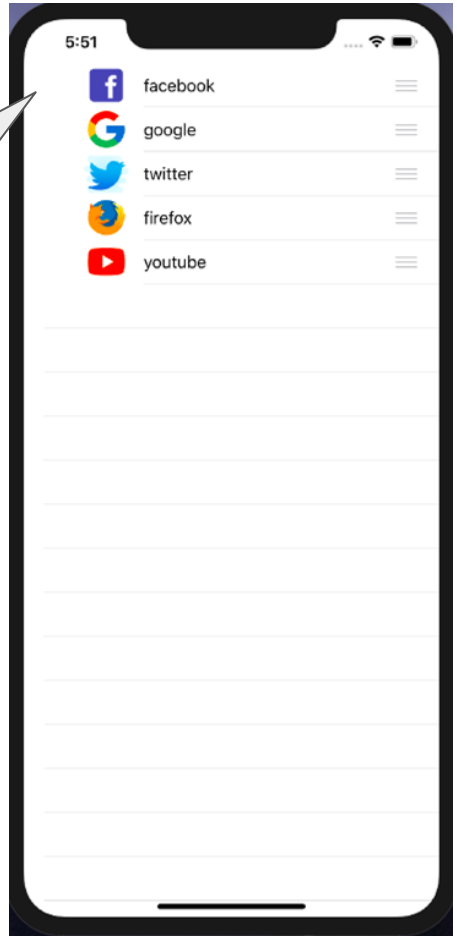


TableViewController

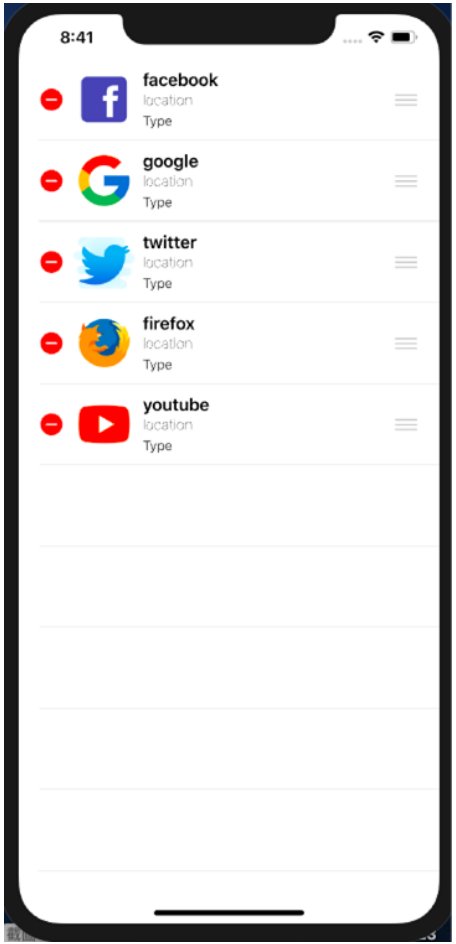


TableViewController

當把編輯樣式改成.none時左邊便會消失



TableViewController



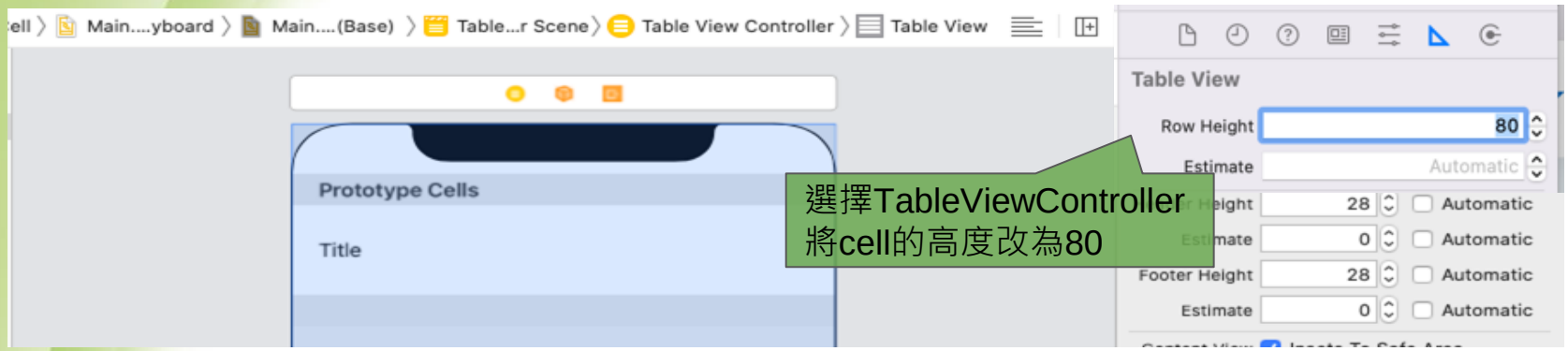
TableViewController

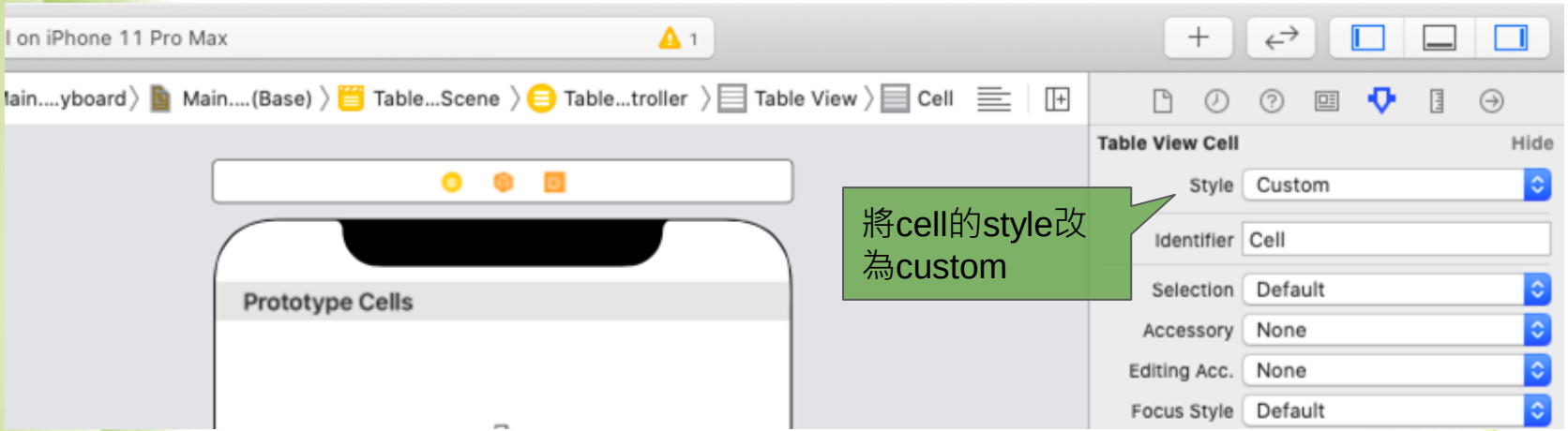
```
// Override to support rearranging the table view.
override func tableView(_ tableView: UITableView, moveRowAt fromIndexPath: IndexPath, to:
    IndexPath) {
    let tmp = names[fromIndexPath.row]
    names.remove(at: fromIndexPath.row)
    names.insert(tmp, at: to.row)
}

// Override to support conditional rearranging of the table view.
override func tableView(_ tableView: UITableView, canMoveRowAt indexPath: IndexPath) -> Bool {
    // Return false if you do not want the item to be re-orderable.
    return true
}
```

此函式是移動欄位時做的事情這裡是將陣列的元素位置排好

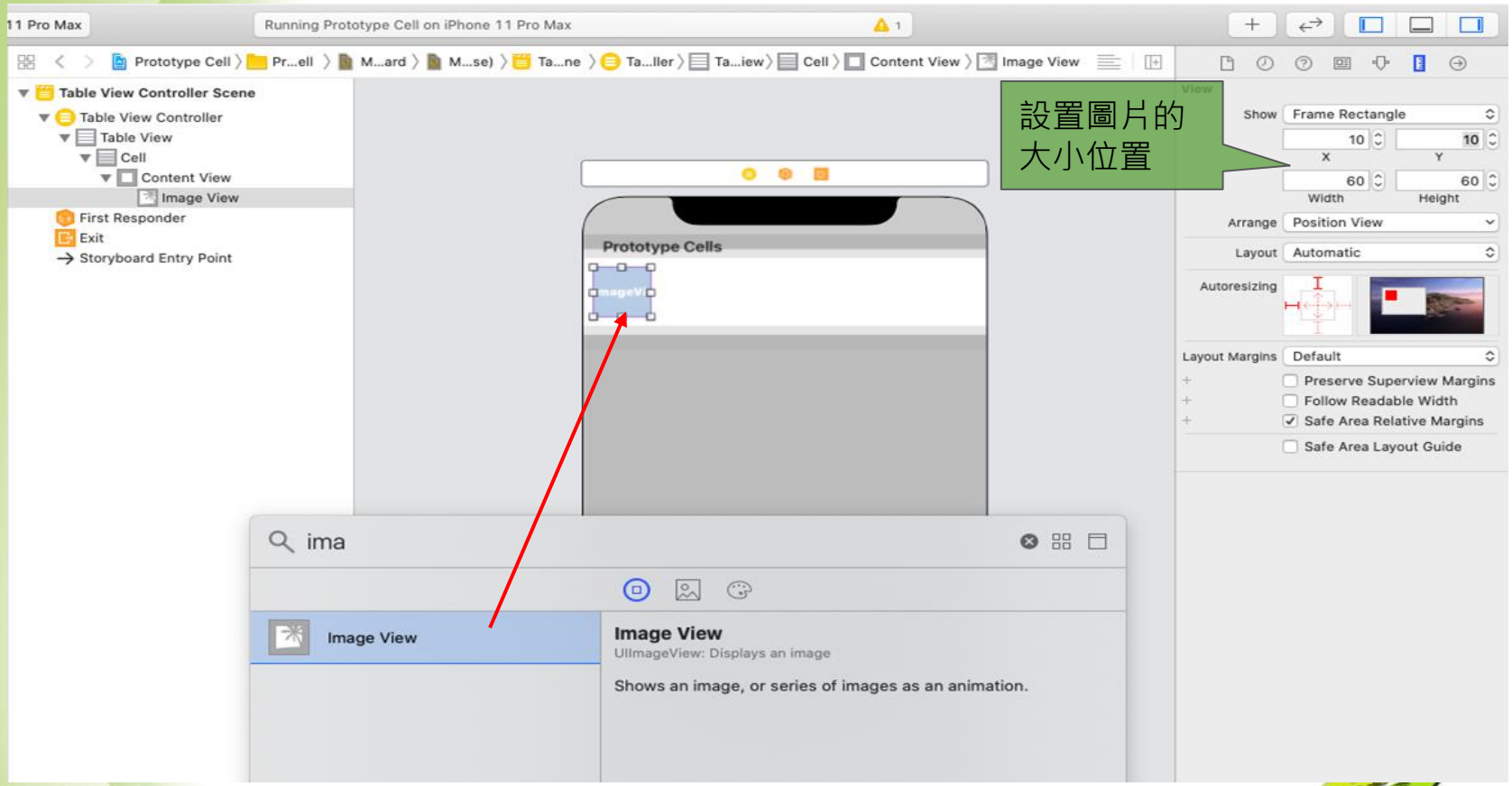
是否可以移動欄位

TableViewController

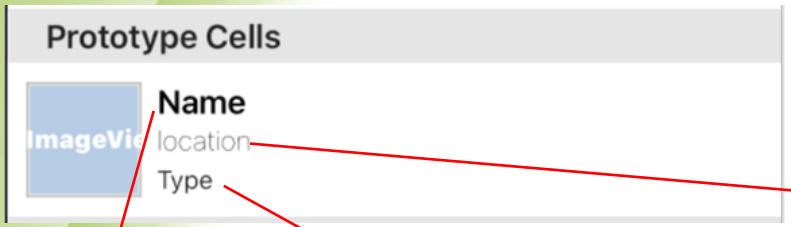


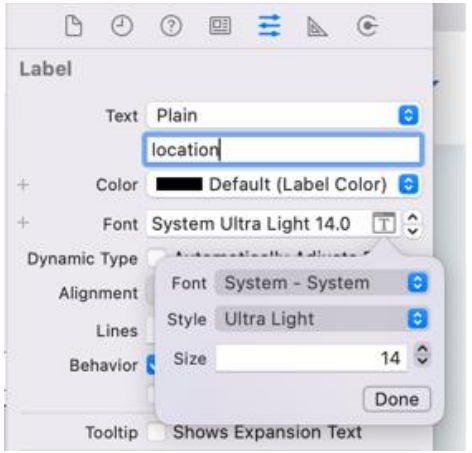


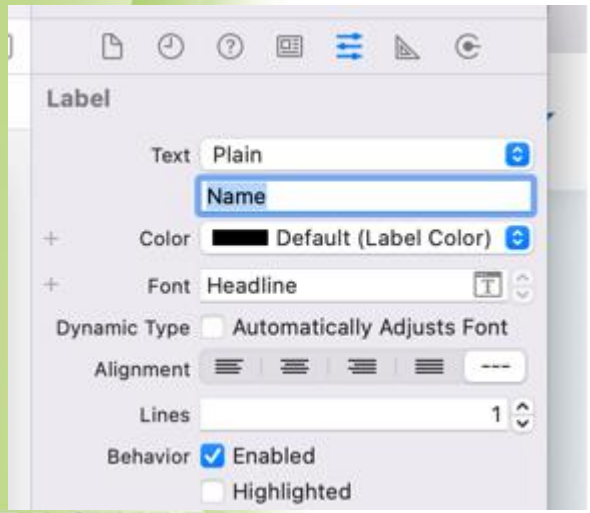
TableViewController

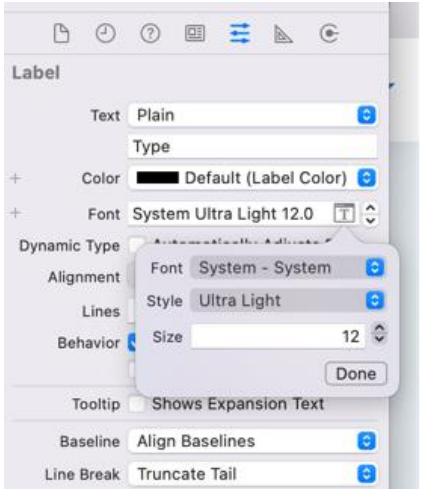


TableViewController

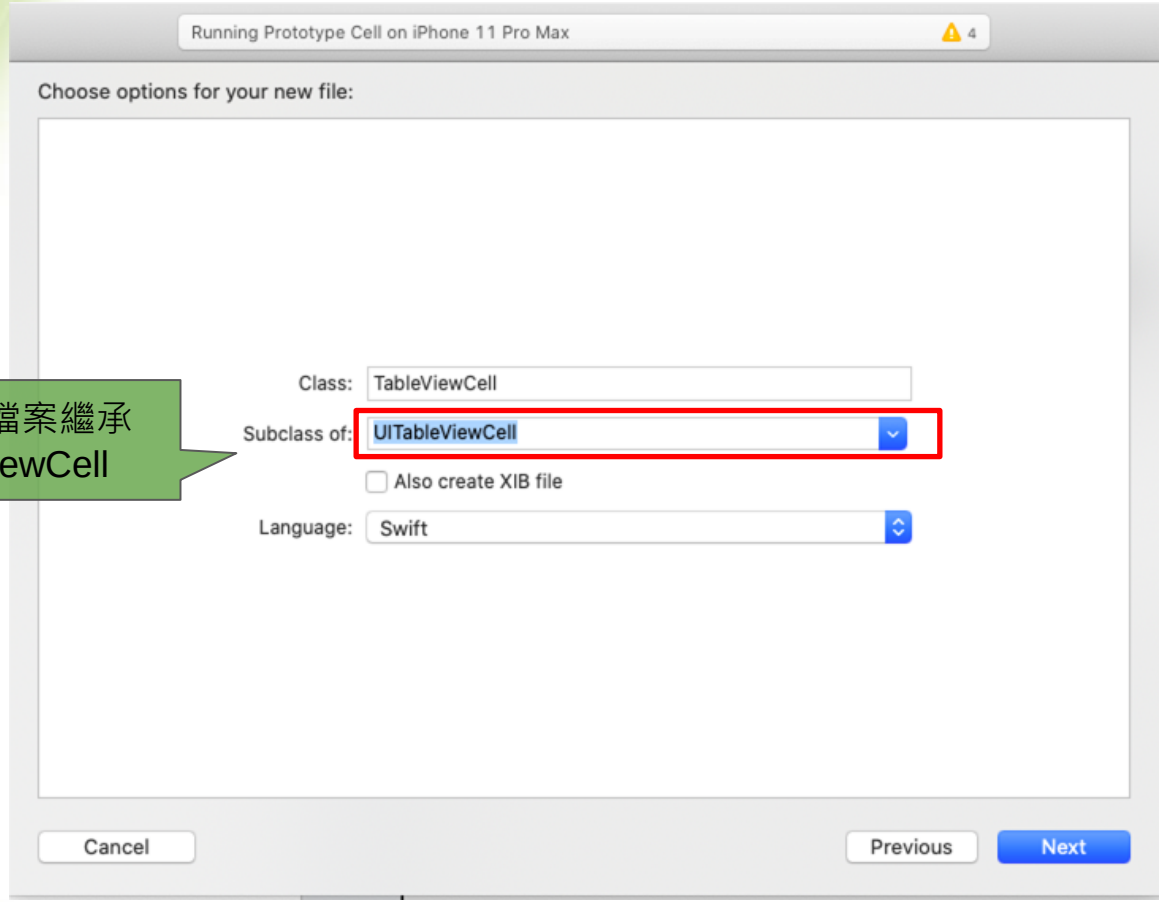






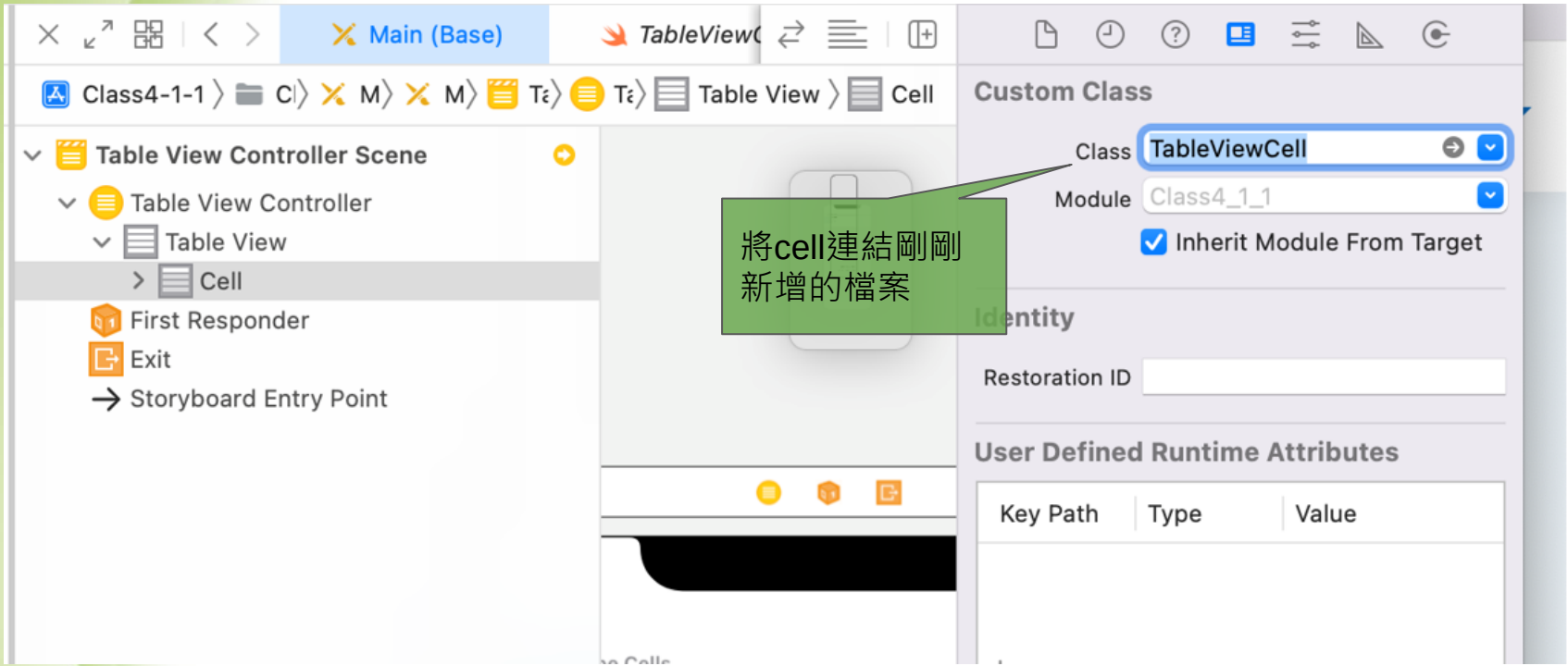


TableViewController



新增新的檔案繼承
UITableViewCell

TableViewController



TableViewController

```
import UIKit

class TableViewCell: UITableViewCell {
    @IBOutlet weak var thumbnail: UIImageView!
    @IBOutlet weak var nameLabel: UILabel!
    @IBOutlet weak var locaitionLabel: UILabel!
    @IBOutlet weak var typeLabel: UILabel!
```

將cell裡的元件連結到TableViewCell裡面

TableViewController

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath) as! TableViewCell
    cell.nameLabel?.text = names[indexPath.row]
    cell.thumbnail?.image = UIImage(named: imgs[indexPath.row])
    return cell
}
```

將cell類別轉成TableViewCell

取得cell裡的原件