

Class 4



Table View

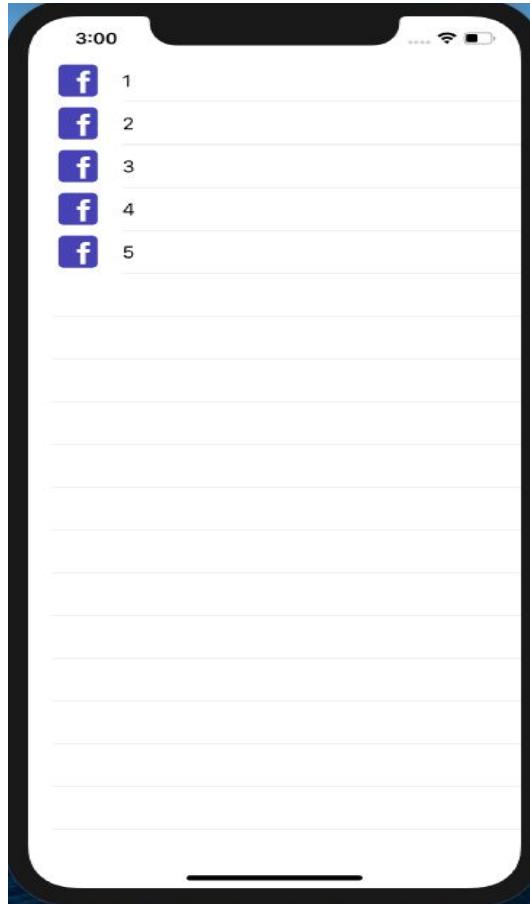


Table View

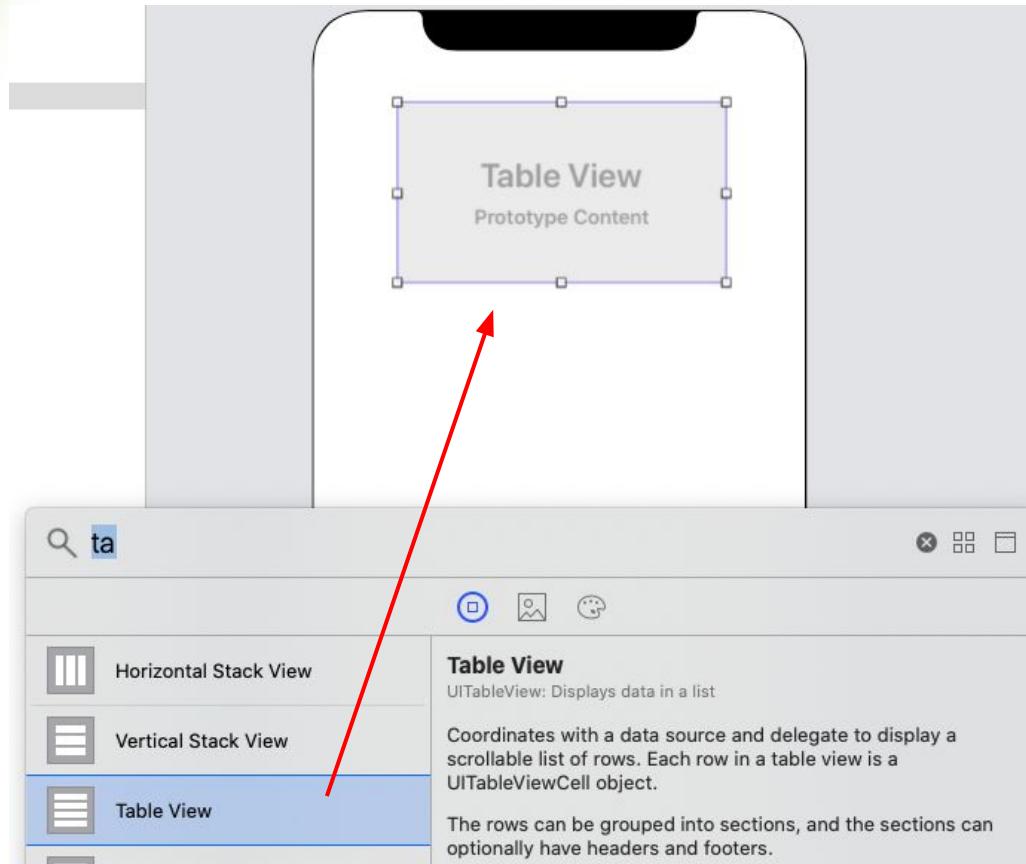


Table View

將tableView
拉到與畫面
差不多大小

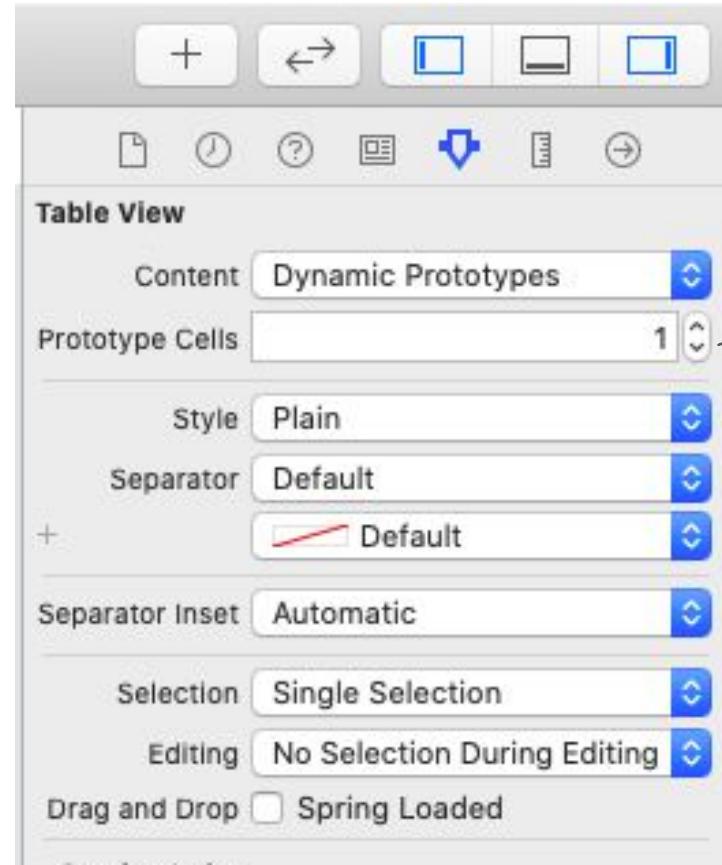
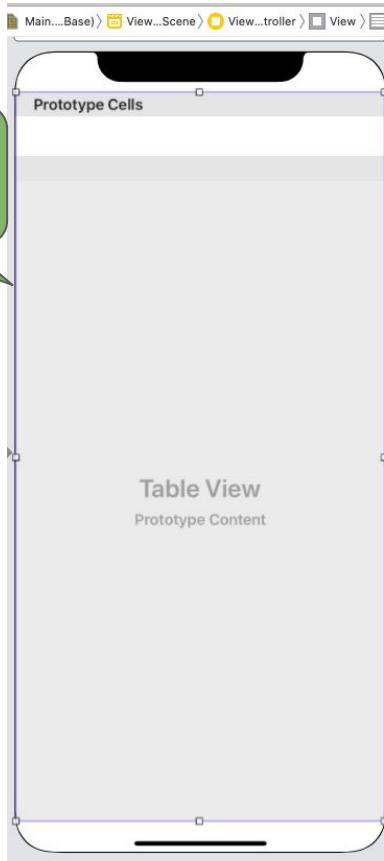


Table View

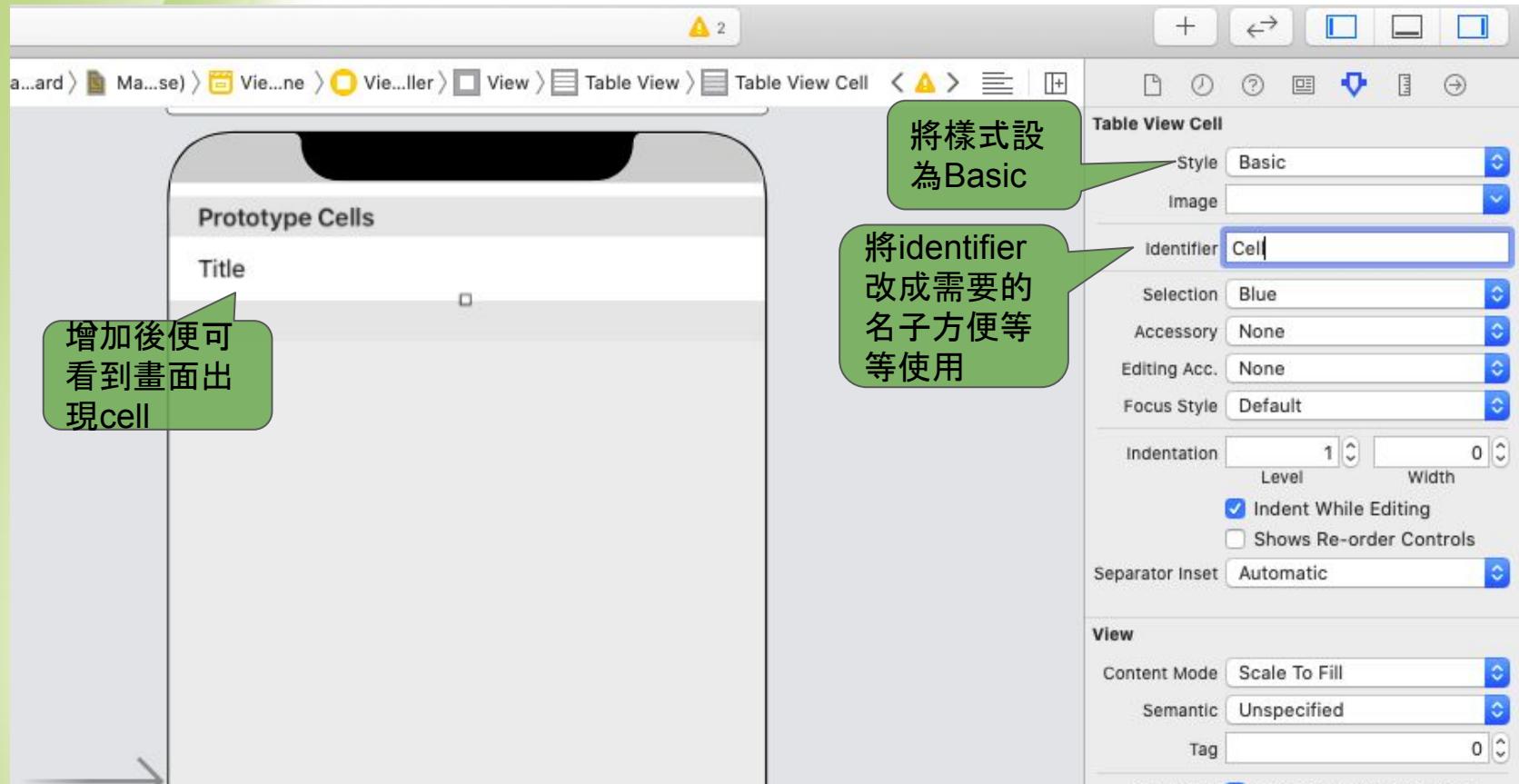


Table View

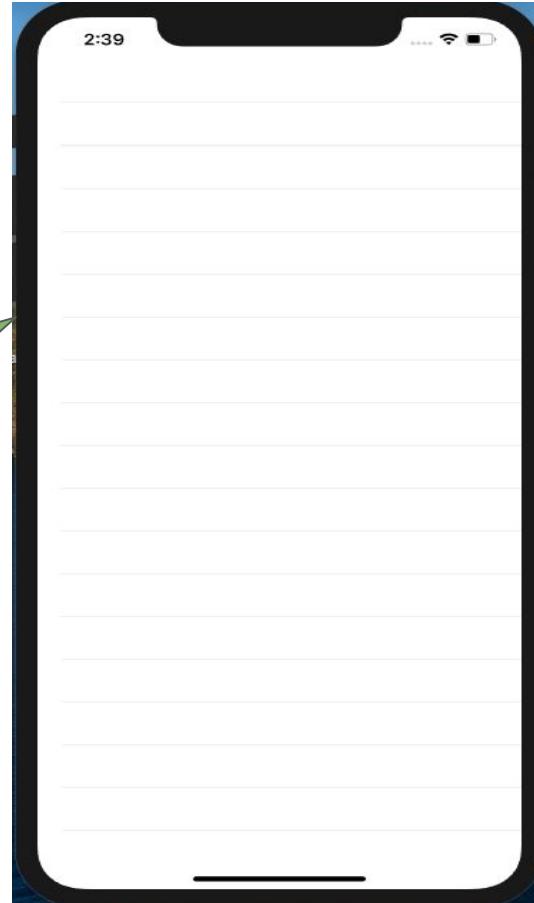


Table View

```
PRO MAX Finished running Table View on iPhone 11 Pro Max
Table View > Table View > ViewController.swift > ViewController
1 // ViewController.swift
2 // Table View
3 // Created by evan on 2020/3/23.
4 // Copyright © 2020 evan. All rights reserved.
5 //
6
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional setup after loading the view.
16     }
17
18
19 }
20
```

如同 PickView
一樣需要透過
Delegate 與
DataSource 來
使用

Type 'ViewController' does not conform to protocol
'UITableViewDataSource'

Do you want to add protocol stubs? **Fix**

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

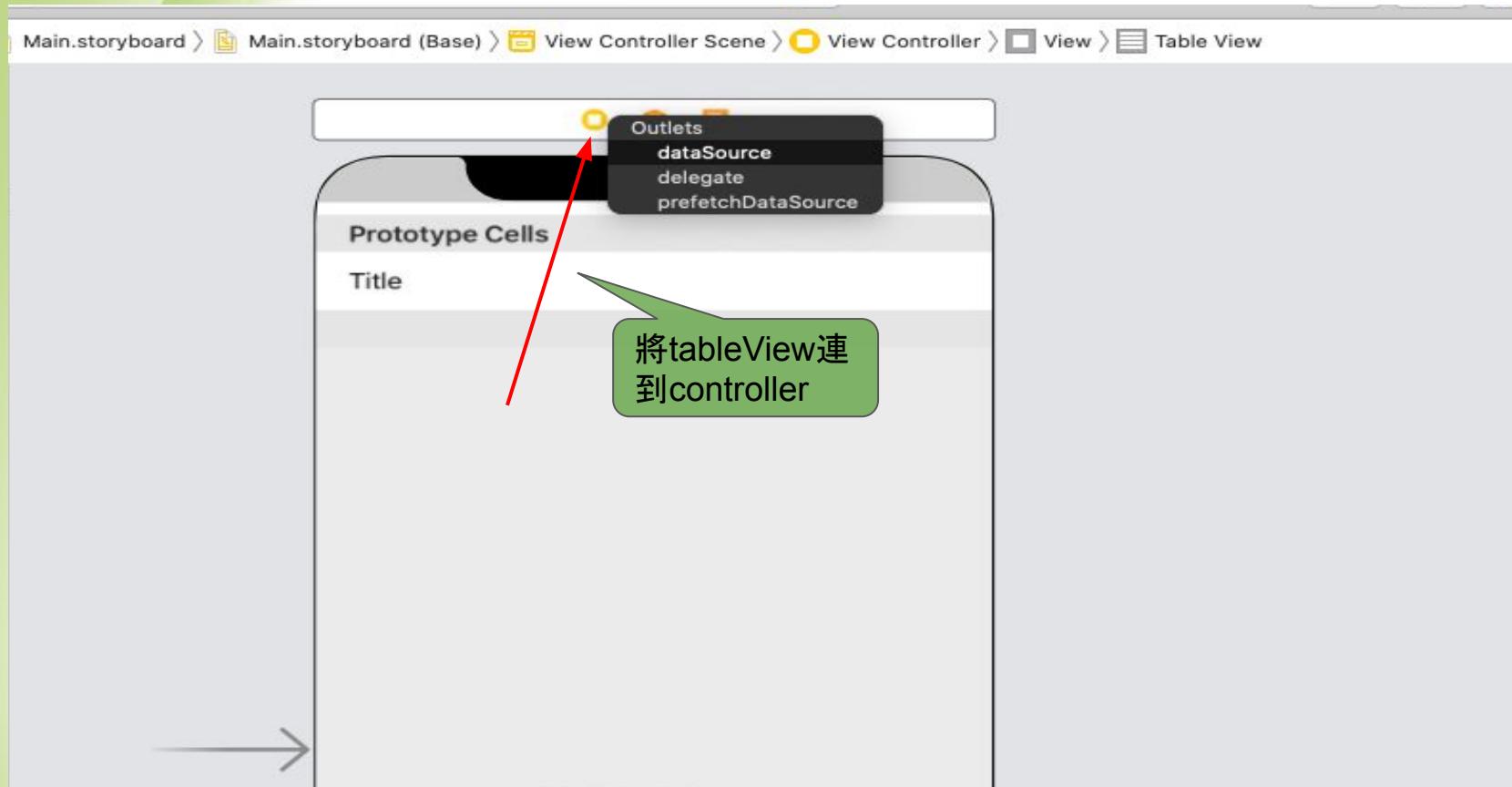


Table View

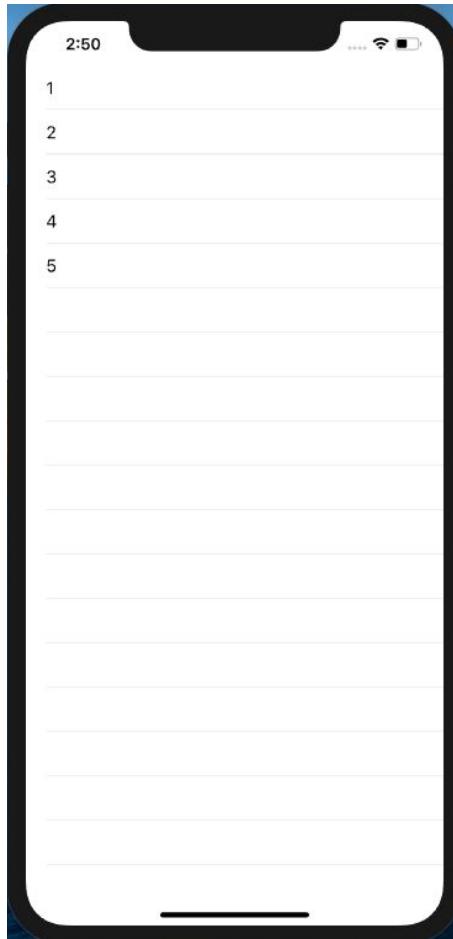


Table View

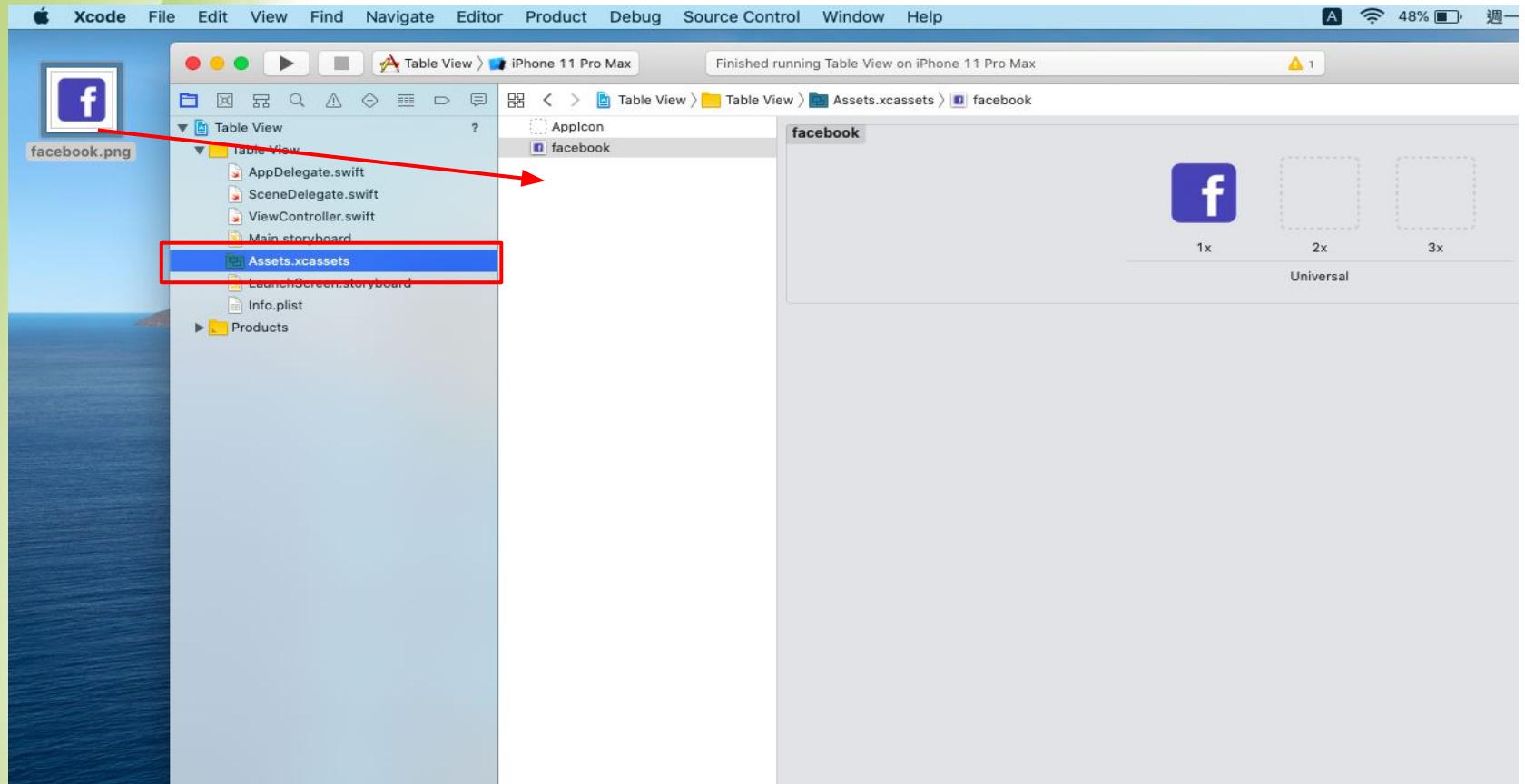


Table View

```
< > Table View > Table View > ViewController.swift > tableView(_:cellForRowAt:)

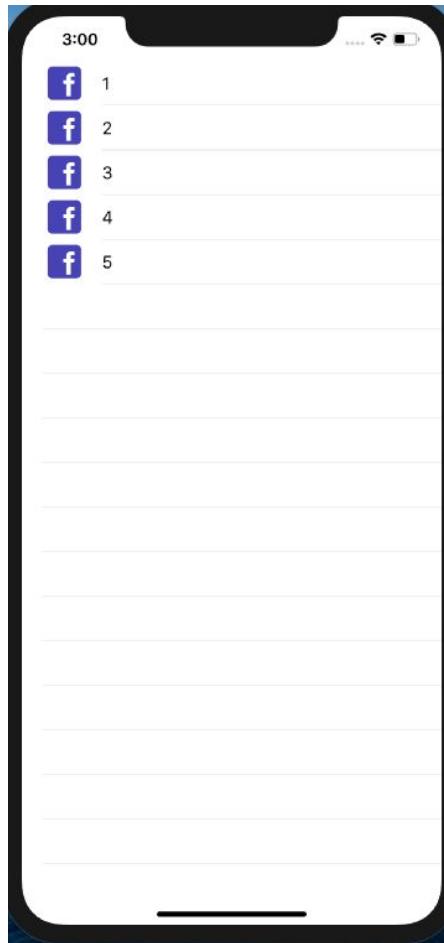
7 // 
8 
9 import UIKit
10 
11 class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
12     var names = ["1", "2", "3", "4", "5"]
13 
14     func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
15         return names.count
16     }
17 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

將圖案新增
到欄位上

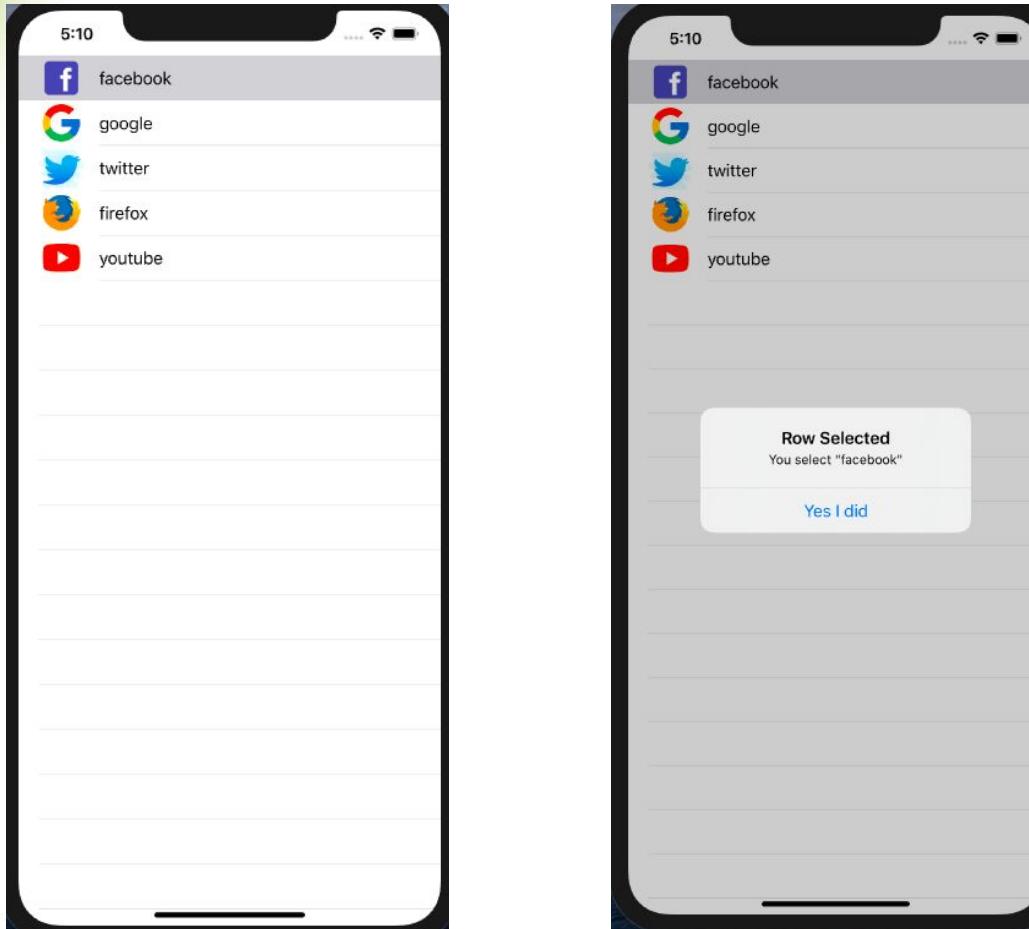


有可用的更新項目
您要立即安裝更新項目嗎？

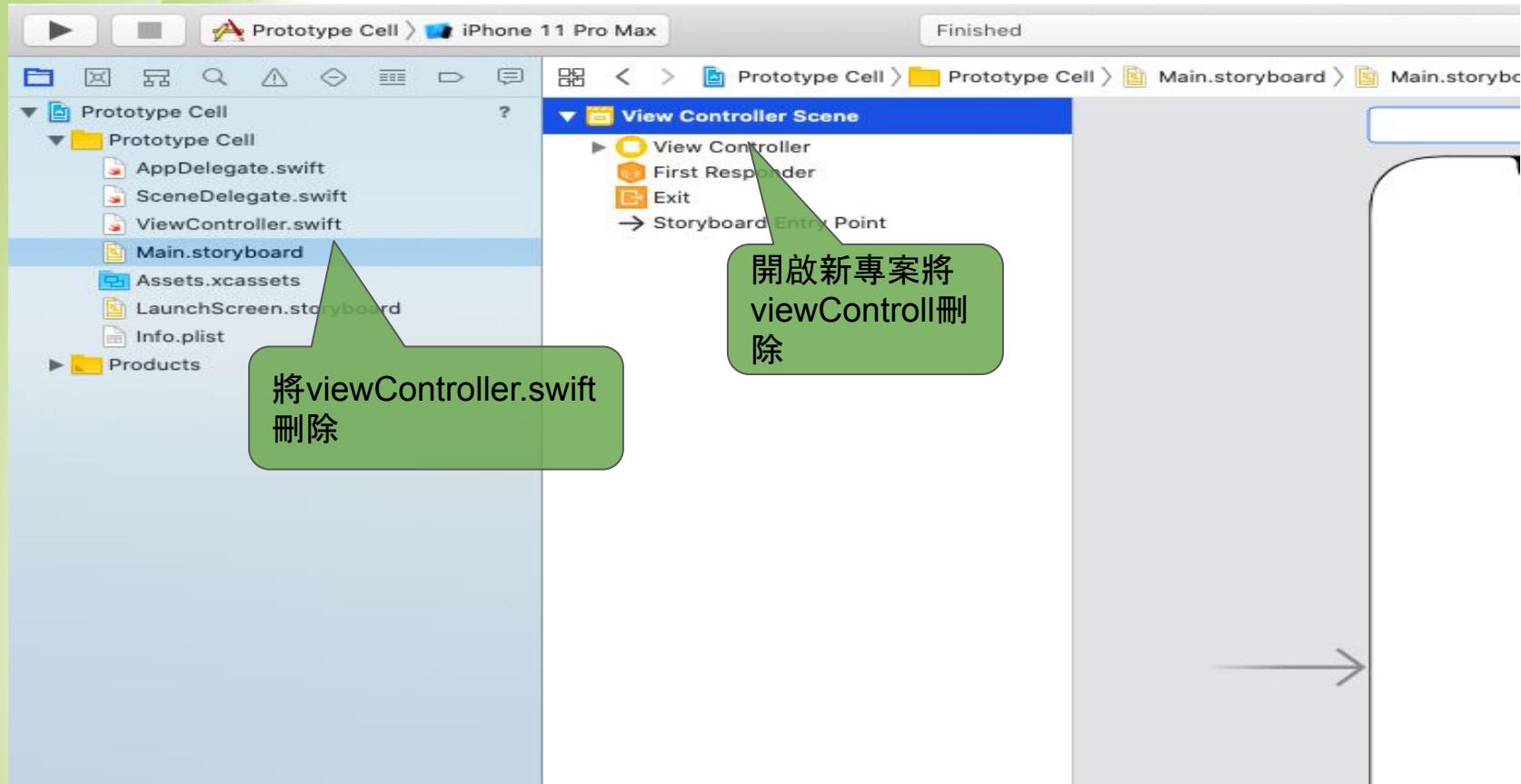
Table View



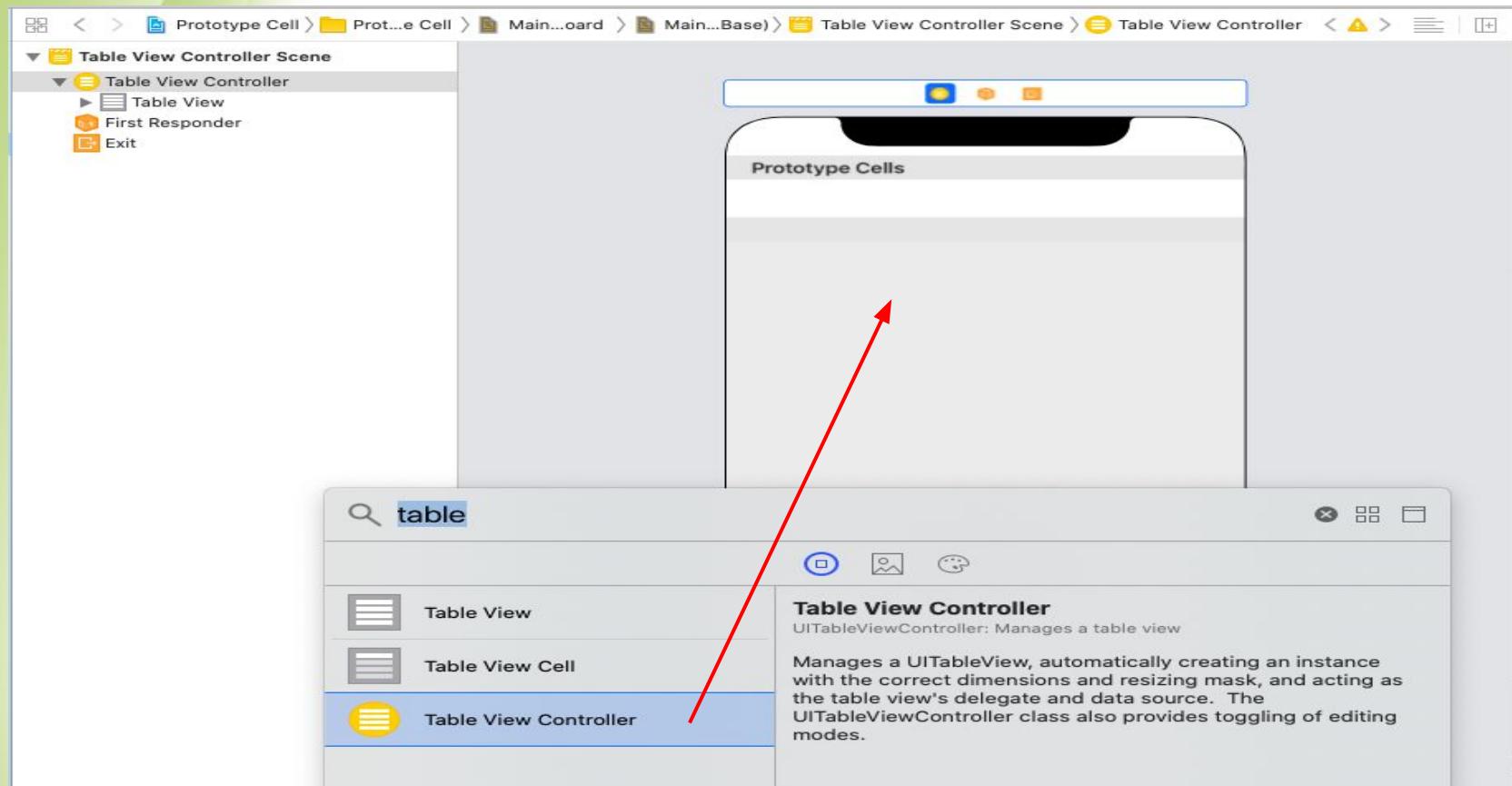
TableViewController



TableViewController



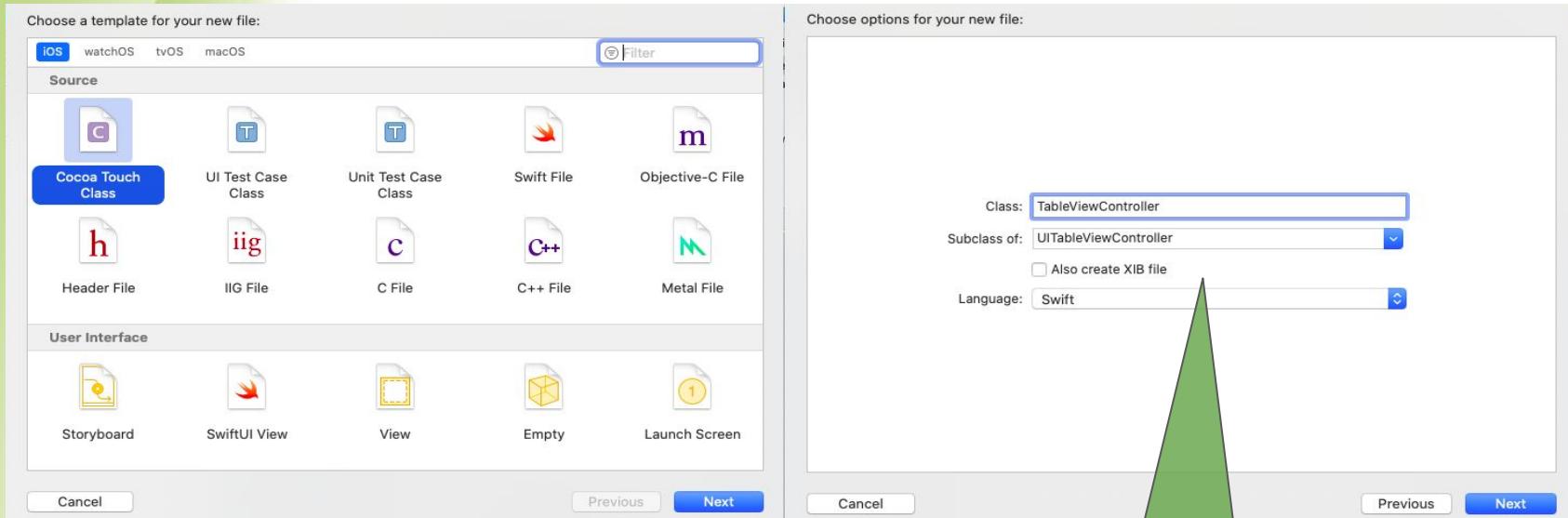
TableViewController



TableViewController



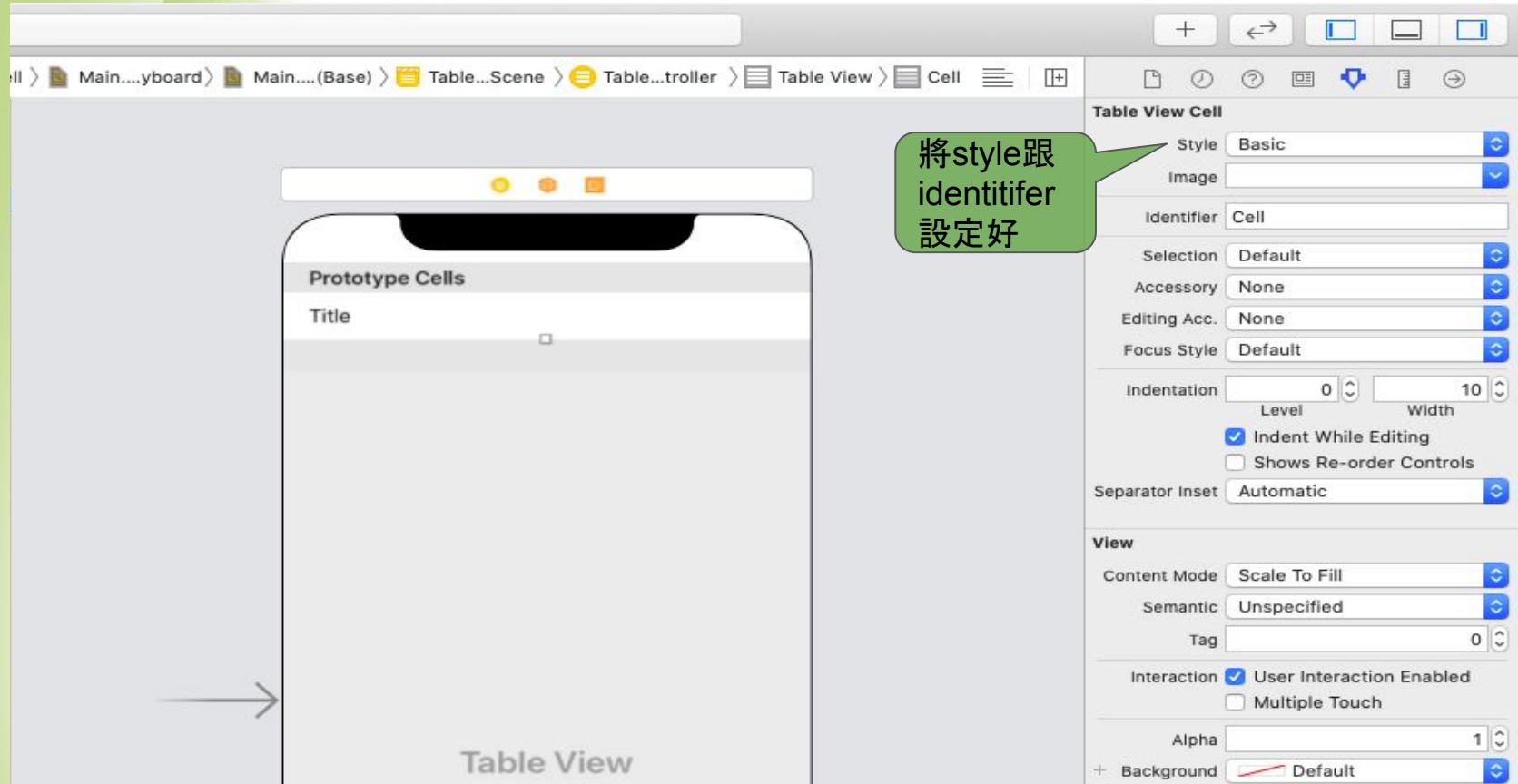
TableViewController



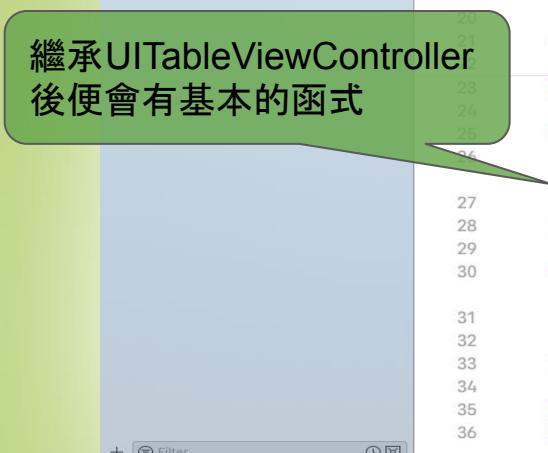
創建檔案將它繼承
UITableViewController

將class
設置好

TableViewController



TableViewController



```
//
import UIKit
class TableViewController: UITableViewController {
    override func viewDidLoad() {
        super.viewDidLoad()

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

        // Uncomment the following line to display an Edit button in the
        // navigation bar for this view controller.
        // self.navigationItem.rightBarButtonItem = self.editButtonItem
    }

    // MARK: - Table view data source

    override func numberOfSections(in tableView: UITableView) -> Int {
        // #warning Incomplete implementation, return the number of
        // sections
        return 0
    }

    override func tableView(_ tableView: UITableView,
        numberOfRowsInSection section: Int) -> Int {
        // #warning Incomplete implementation, return the number of rows
        return 0
    }

    /*
    override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    }
    */
}
```

繼承 UITableViewController
後便會有基本的函式

No Selection

TableViewController

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

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

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

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
}
```

宣告名字
與圖片名
稱陣列

回傳有幾個
section

每個section
有幾個row

基本都與之前相同差別
在圖片是用陣列來決定

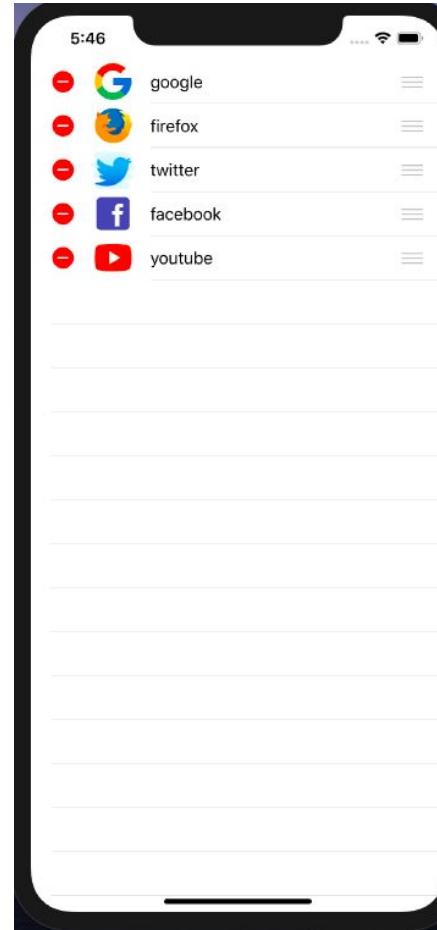
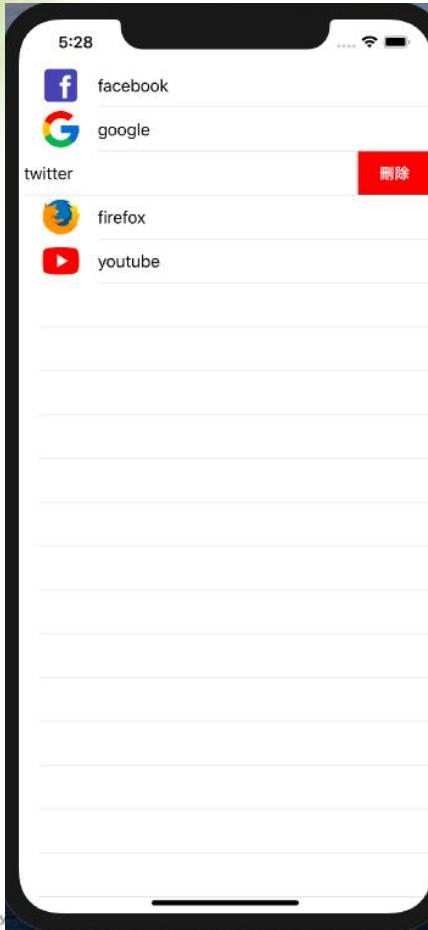
TableViewController

```
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)  
}
```

選到後便
跳出訊息

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

TableViewController



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  
        // table view.  
    }  
}
```

將元素從陣列中移除不然資料對不上會出錯

此行是將某一欄刪掉

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

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

```
92  
93     override func tableView(_ tableView: UITableView, editingStyleForRowAtIndexPath indexPath: IndexPath)  
94         -> UITableViewCellEditingStyle {  
95             return .
```

UITableViewCellEditingStyle delete
UITableViewCellEditingStyle insert
UITableViewCellEditingStyle none

① Expected identifier after '!' expression

The cell has the delete editing control; this control is a red circle with a minus sign. When you are performing an operation, you will often want to do a little preparation before enclosing a minus sign.

```
100     override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
```

此函式是選擇
編輯模式的樣
式

TableViewController



TableViewController

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

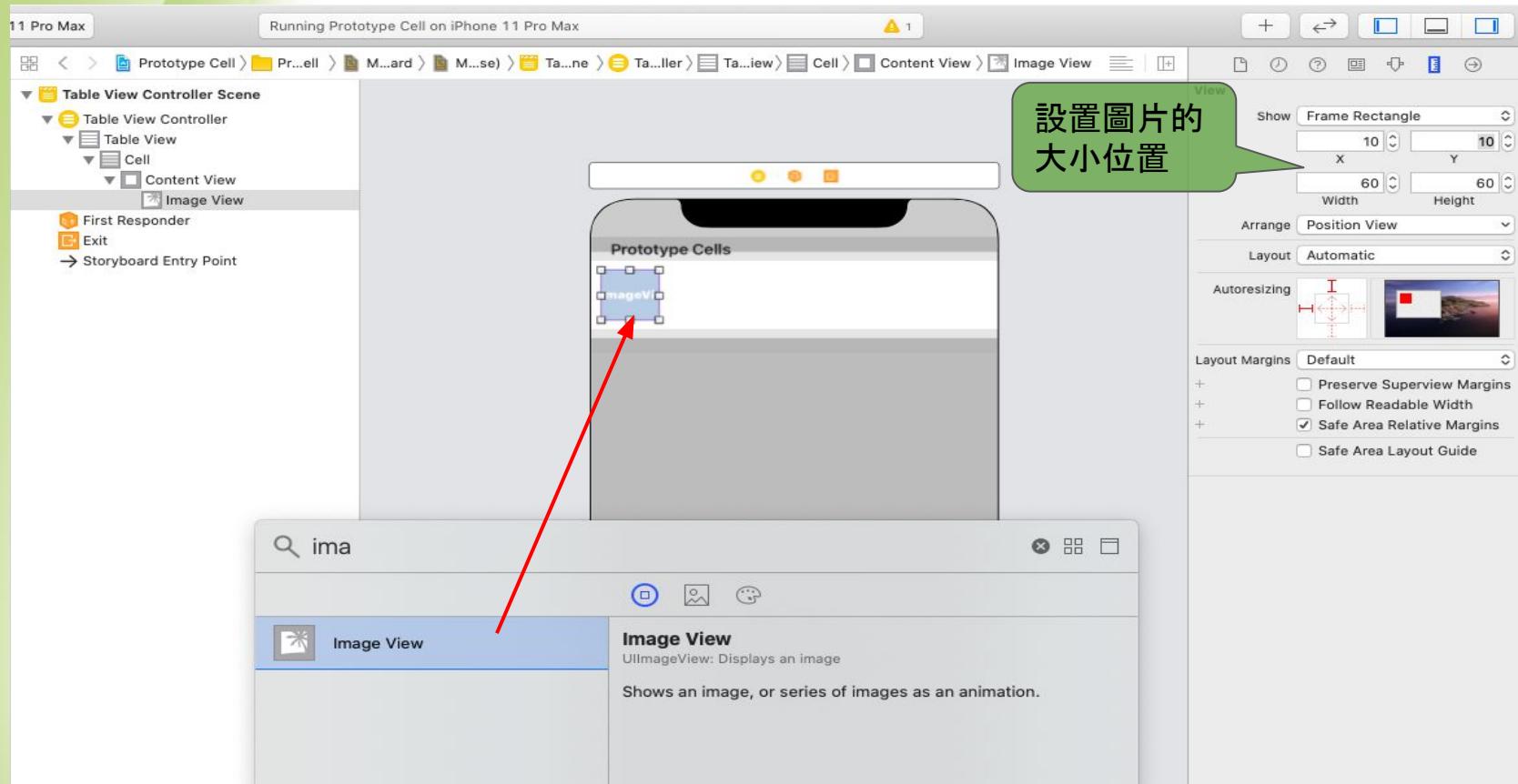
此函式是移動欄位時做的事情這裡是將陣列的元素位置排好

是否可以移動欄位

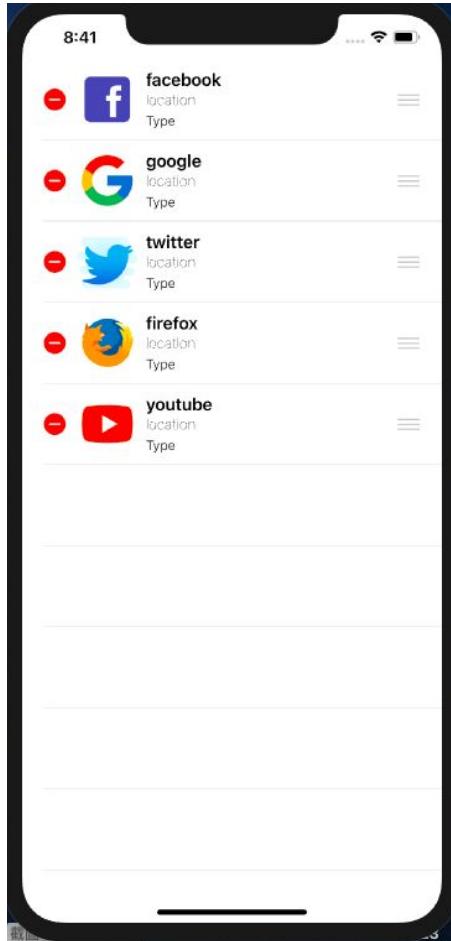
TableViewController



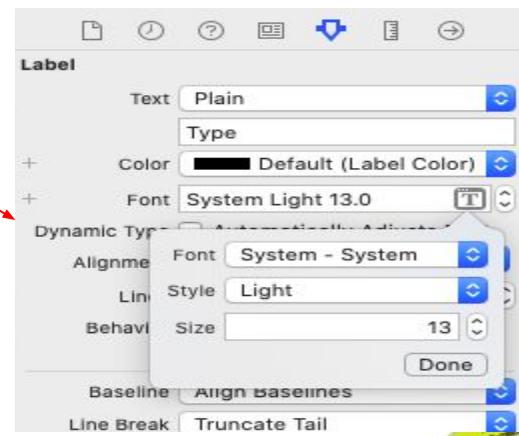
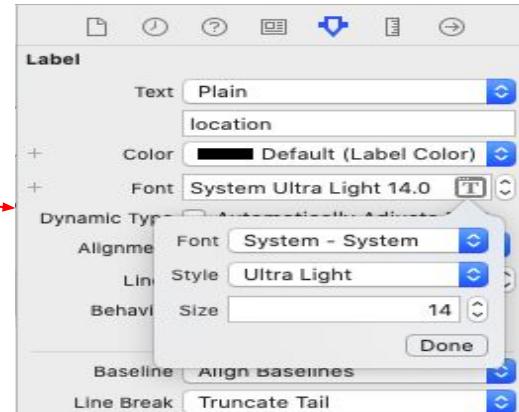
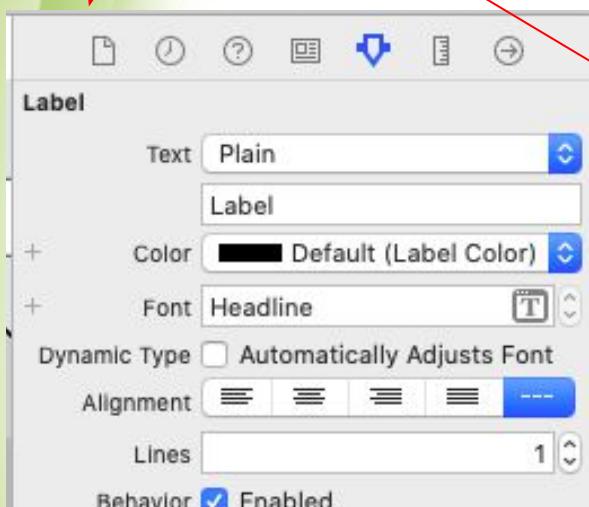
TableViewController



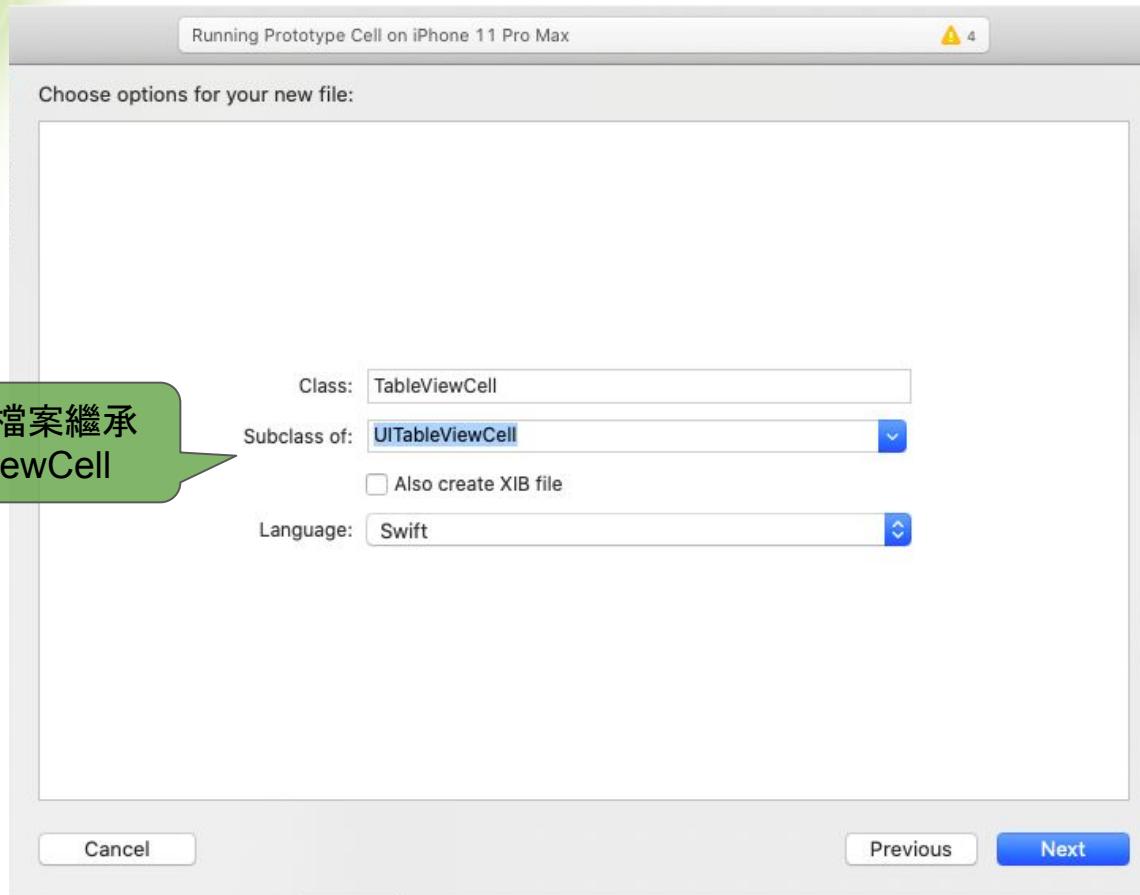
TableViewController



TableViewController



TableViewController



TableViewController

The screenshot shows the Xcode interface with the storyboard editor open. The navigation bar at the top indicates the project is for an iPhone 11 Pro Max. The storyboard path is MainStoryboard -> MainScene -> TableScene -> TableController -> Table View -> Cell. The right-hand sidebar contains the Identity Inspector, which shows the custom class is set to TableViewCell and the module is Prototype_Cell. A green callout bubble points to the 'Custom Class' section with the text: '將cell連結剛剛新增的檔案'. In the main storyboard area, a prototype cell is visible, containing an ImageView and a label. The label has the placeholder text 'Name' and 'location'. The bottom part of the screen shows the storyboard canvas with a rounded rectangle representing the table view.

將cell連結剛剛新增的檔案

Custom Class

Class: TableViewCell

Module: Prototype_Cell

Inherit Module From Target

Identity

Restoration ID:

User Defined Runtime Attributes

Key Path Type Value

Document

Label: Xcode Specific Label

Object ID: zpC-kf-8tT

Lock: Inherited - (Nothing)

Localizer Hint: Comment For Localizer

Accessibility

Accessibility: Enabled

Label: Label

Hint: Hint

TableViewController

```
import UIKit

class TableViewCell: UITableViewCell {
    @IBOutlet weak var thumbnail: UIImageView!
    @IBOutlet weak var nameLabel: UILabel!
    @IBOutlet weak var locaitonLabel: 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  
}
```

取得cell裡的原件

將cell類別轉成TableViewCell