

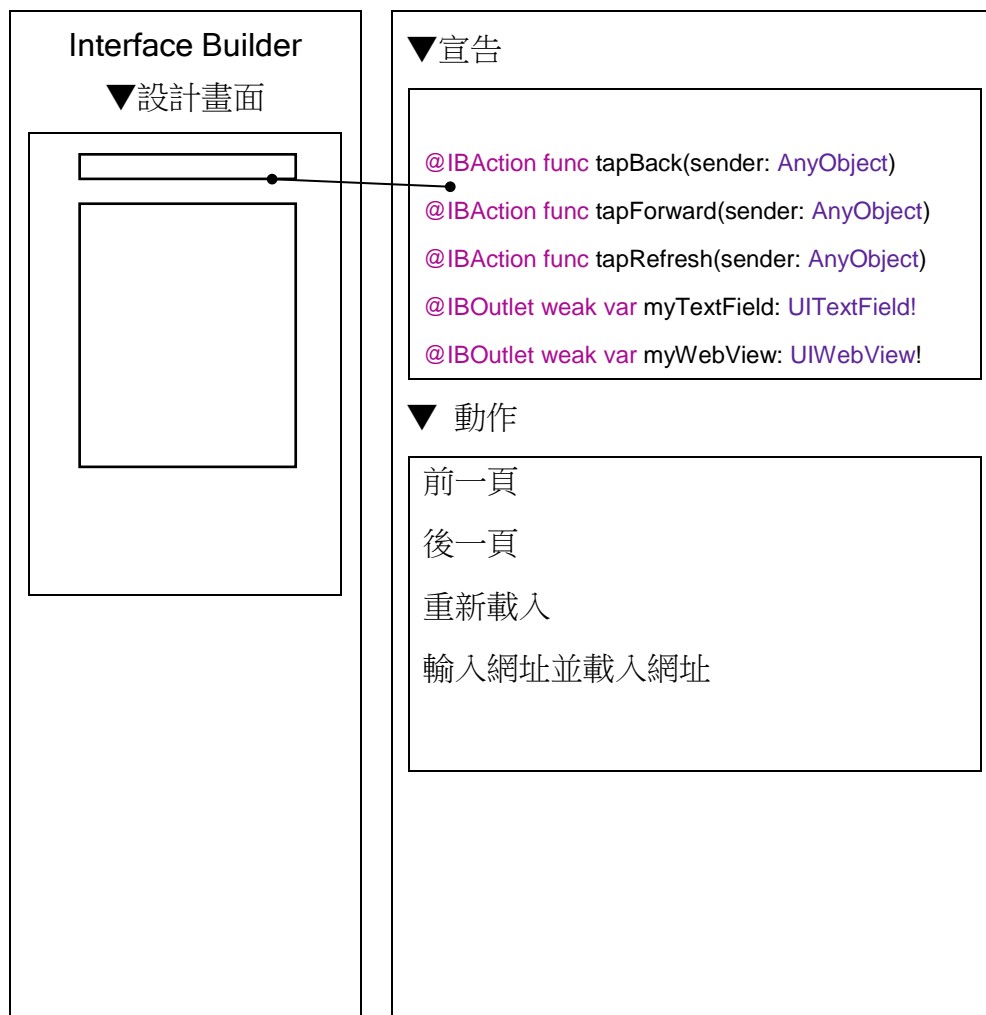
CHAPTER 8-3

Toolbar: 建立工具列

製作網頁瀏覽器

學習概念：

1. 首先用 IB 建立【UIWebView】，於上方建置【Toolbar】與【item】。
2. 接著將【item】與【UIWebView】與程式碼連結。
3. 最後在實作相關程式。
4. 。

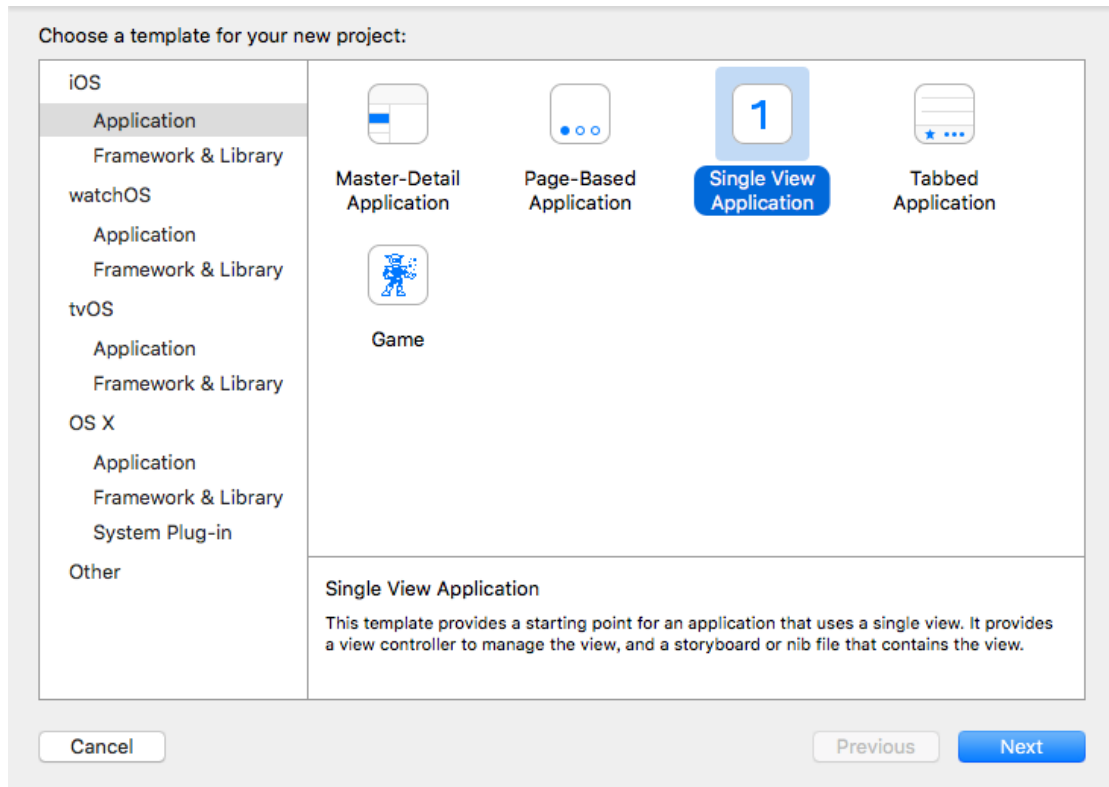


【執行結果】

當 App 執行後，會自動於【網頁視窗】上開啟設定的網頁。

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.08-3**，選擇裝置為 **iPad**

Choose options for your new project:

Product Name:

Organization Name:

Organization Identifier:

Bundle Identifier:

Language:

Devices:

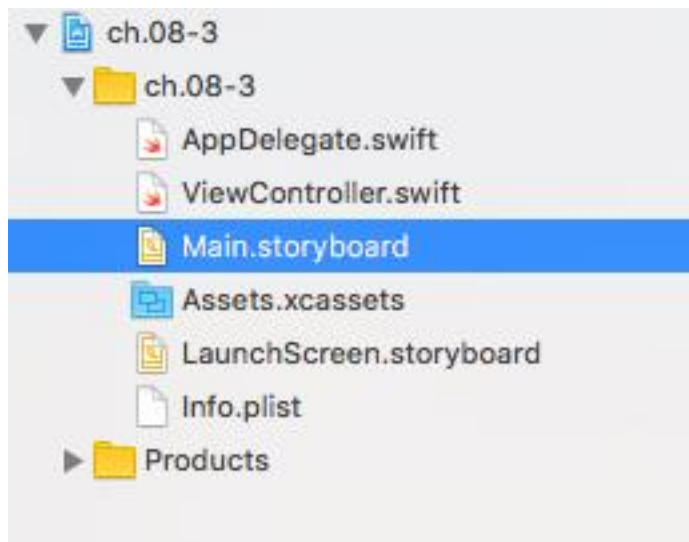
☐ Use Core Data

☐ Include Unit Tests

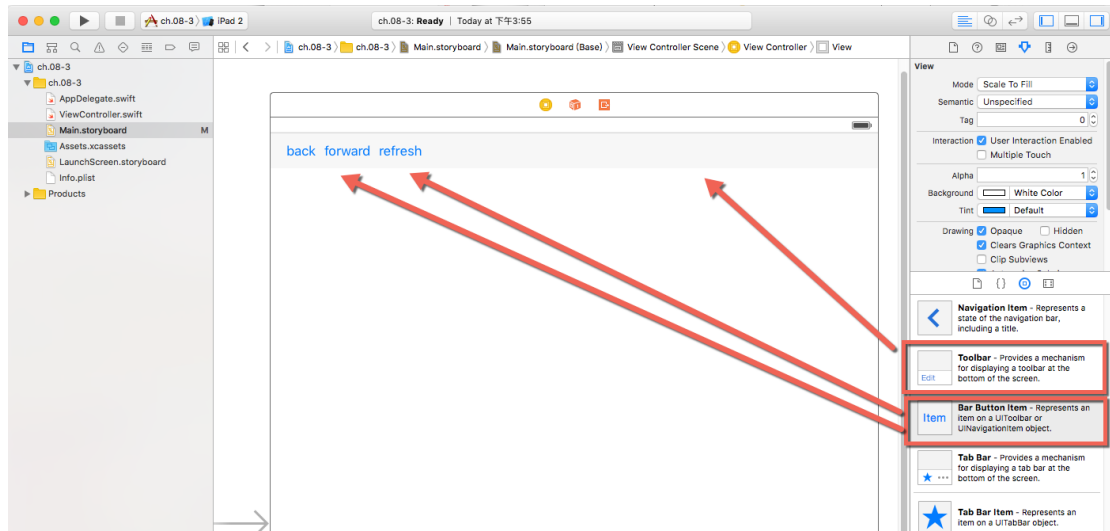
☐ Include UI Tests

Step.2

選取【Main.storyboard】



從【物件庫】中拖曳【Toolbar】到畫面上方，並在多加上兩個【Bar Button Item】。
將此三個 item 分別命名為 back， forward， Refresh。



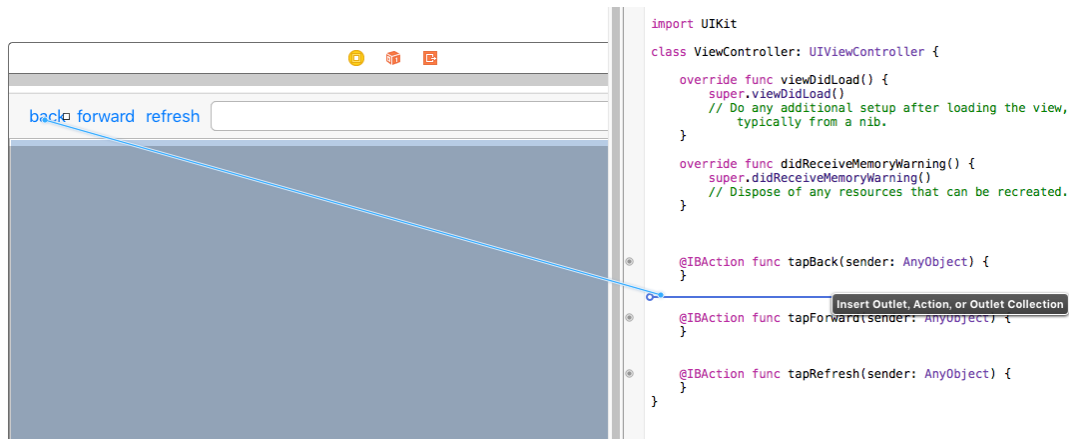
在上方處再放置一個 **【TextField】**，下方畫面放置 **【Web View】**



Step.3.

- (1) 將三個**【 Bar Button Item】**與程式連結，分別命名 `tapBack`，`tapForward`，`tapRefresh`。

在點選【Bar Button Item】時，要點選到按鈕上出現一個小白方塊，這時才是點選到此按鈕。



(2) 將【Text Field】與程式連結，命名 myTextField

(3) 將【Image View】與程式連結，命名 myImageView

Step.4

(1) web 的上一頁、下一頁以及重新整理，已經有內建函式，只需將此函式放置在各個按鈕中。

```
@IBAction func tapBack(sender: AnyObject) {  
    myWebView.goBack()  
}
```

```
@IBAction func tapForward(sender: AnyObject) {  
    myWebView.goForward()  
}
```

```
@IBAction func tapRefresh(sender: AnyObject) {  
    myWebView.reload()  
}
```

(2) 必須使用到 TextField 的函式 textFieldShouldReturn，因此必須於 ViewController 中加入 UITextFieldDelegate。

```
class ViewController: UIViewController, UITextFieldDelegate
```

(3) 並且於程式執行時告知 `myTextField` 的作用要應用在此程式上。

```
myTextField.delegate = self
```

(4) 撰寫按下 `Enter` 鍵後，讀取網頁的程式。

```
func textFieldShouldReturn(textField: UITextField) -> Bool {
```

```
    let myURL = NSURL (string: myTextField.text!)
```

```
    let myData = NSData(contentsOfURL: myURL!)
```

```
    let myImage = UIImage (data: myData!)
```

```
    myImageView.image = myImage
```

```
    return true
```

```
}
```

```

import UIKit

class ViewController: UIViewController, UITextFieldDelegate {

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

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

    @IBAction func tapBack(sender: AnyObject) {
        myWebView.goBack()
    }

    @IBAction func tapForward(sender: AnyObject) {
        myWebView.goForward()
    }

    @IBAction func tapRefresh(sender: AnyObject) {
        myWebView.reload()
    }

    @IBOutlet weak var myTextField: UITextField!

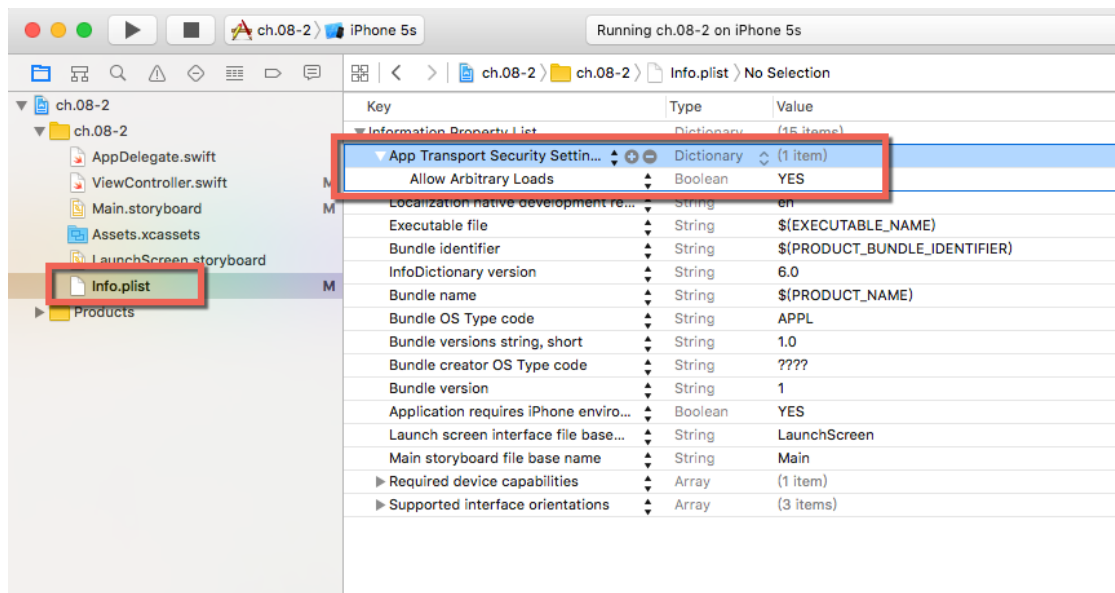
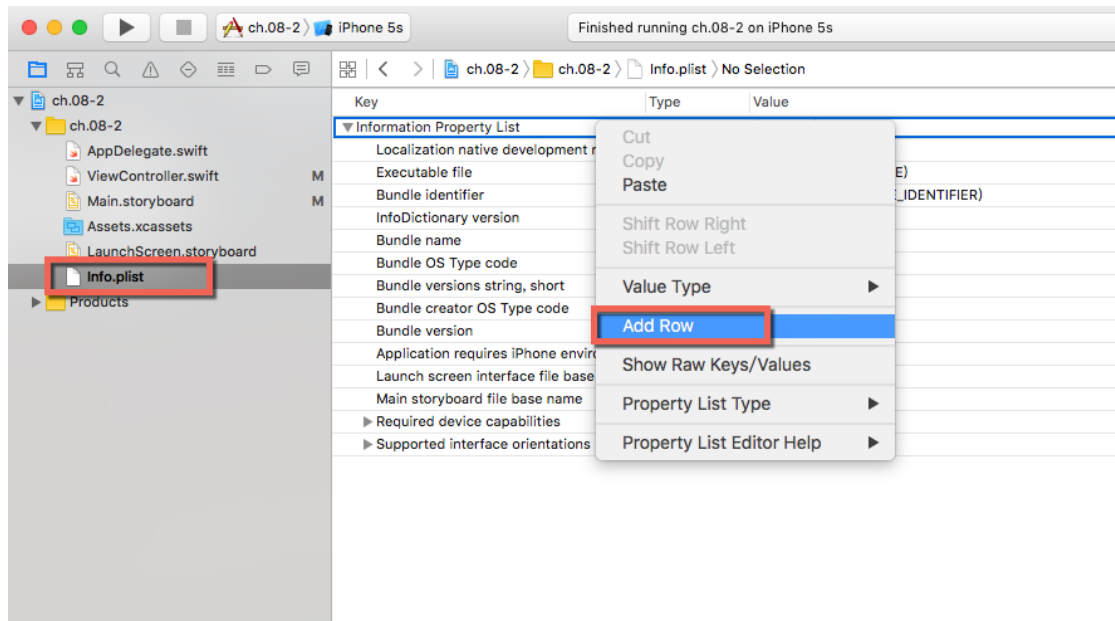
    @IBOutlet weak var myWebView: UIWebView!

    func textFieldShouldReturn(textField: UITextField) -> Bool {
        let myURL = NSURL (string: myTextField.text!)
        let myURLRequest = NSURLRequest (URL: myURL!)
        myWebView.loadRequest(myURLRequest)
        return true
    }
}

```

Step.5

在 Info.plist 中添加 AppTransport Security Settings，並於項下添加 Allow Arbitrary Loads 類型為 Boolean,將值改為 YES



Step.6

執行程式，即可完成簡易網頁瀏覽器。



iPhone SE

一小部，邁進一大步。

[進一步了解 >](#)
[觀看 keynote 發表會 >](#)



iPhone 6 S
3D Touch、1200 萬像素照片、4K 影片，
強大功能，集於一身。



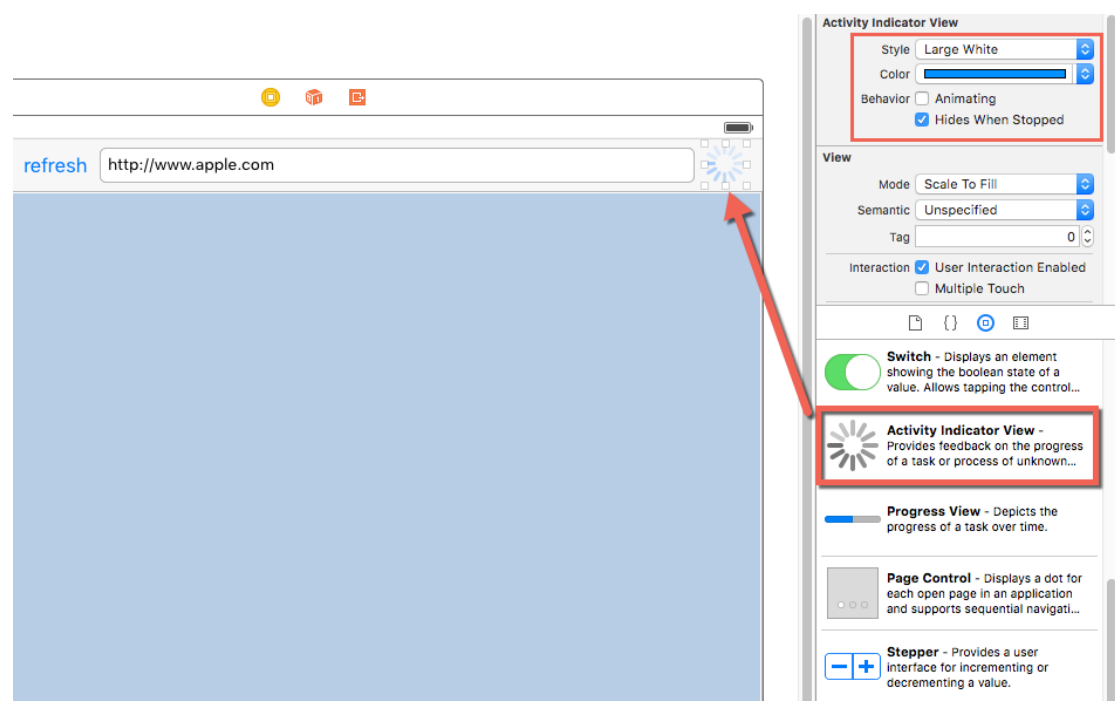
///



活用【web View】進階

在上例中，若網頁載入時間過久，會誤以為沒有載入成功，因此希望能夠有一個載入的動畫顯示。此外若點選網頁中其他的超連結切換到別頁面，網址列並未跟著修改。

本單元將介紹使用 **Activity Indicator View** 來顯示載入等待的圖示動畫。



(1) 必須使用到 **WebView** 的函式，因此必須於 **ViewController** 中加入 **UIWebViewDelegate**。

```
class ViewController: UIViewController, UITextFieldDelegate,  
UIWebViewDelegate {
```

(2) 並且於程式執行時告知 **myWebView** 的作用要應用在此程式上。

```
myWebView.delegate = self
```

(3) 必須使用到 **webViewDidStartLoad** 與 **webViewDidFinishLoad** 函式，

來得知開始載入網頁以及網頁載入完成的事件。

```
func webViewDidStartLoad(webView: UIWebView) {  
    myActivity.startAnimating()  
    myActivity.hidden = false  
}  
  
func webViewDidFinishLoad(webView: UIWebView) {  
    myActivity.stopAnimating()  
    myActivity.hidden = true  
}
```

若希望點選其他連結時，會自動更新網址列上的網址，則需要透過 `request` 去取得現在的位置，再將位置轉換為字串並放置於網址列。

```
let currentRequest = myWebView.request  
let currentURL = currentRequest?.URL  
myTextField.text = currentURL?.absoluteString
```

```

import UIKit

class ViewController: UIViewController, UITextFieldDelegate, UIWebViewDelegate {

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

        myTextField.delegate = self
        myWebView.delegate = self
    }

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

    @IBAction func tapBack(sender: AnyObject) {
        myWebView.goBack()
    }

    @IBAction func tapForward(sender: AnyObject) {
        myWebView.goForward()
    }

    @IBAction func tapRefresh(sender: AnyObject) {
        myWebView.reload()
    }

    @IBOutlet weak var myTextField: UITextField!
    |
    @IBOutlet weak var myWebView: UIWebView!

    func textFieldShouldReturn(textField: UITextField) -> Bool {
        let myURL = NSURL (string: myTextField.text!)
        let myURLRequest = NSURLRequest (URL: myURL!)
        myWebView.loadRequest(myURLRequest)
        return true
    }

    @IBOutlet weak var myActivityIndicator: UIActivityIndicatorView!

    func webViewDidStartLoad(webView: UIWebView) {
        myActivityIndicator.startAnimating()
        myActivityIndicator.hidden = false
    }

    func webViewDidFinishLoad(webView: UIWebView) {
        myActivityIndicator.stopAnimating()
        myActivityIndicator.hidden = true

        let currentRequest = myWebView.request
        let currentURL = currentRequest?.URL
        myTextField.text = currentURL?.absoluteString
    }
}

```