

# Syntax intro



# xcode



# Playground



## Welcome to Xcode

Version 11.3.1 (11C504)



### Get started with a playground

Explore new ideas quickly and easily.



### Create a new Xcode project

Create an app for iPhone, iPad, Mac, Apple Watch, or Apple TV.



### Clone an existing project

Start working on something from a Git repository.



Calculator  
~/Desktop



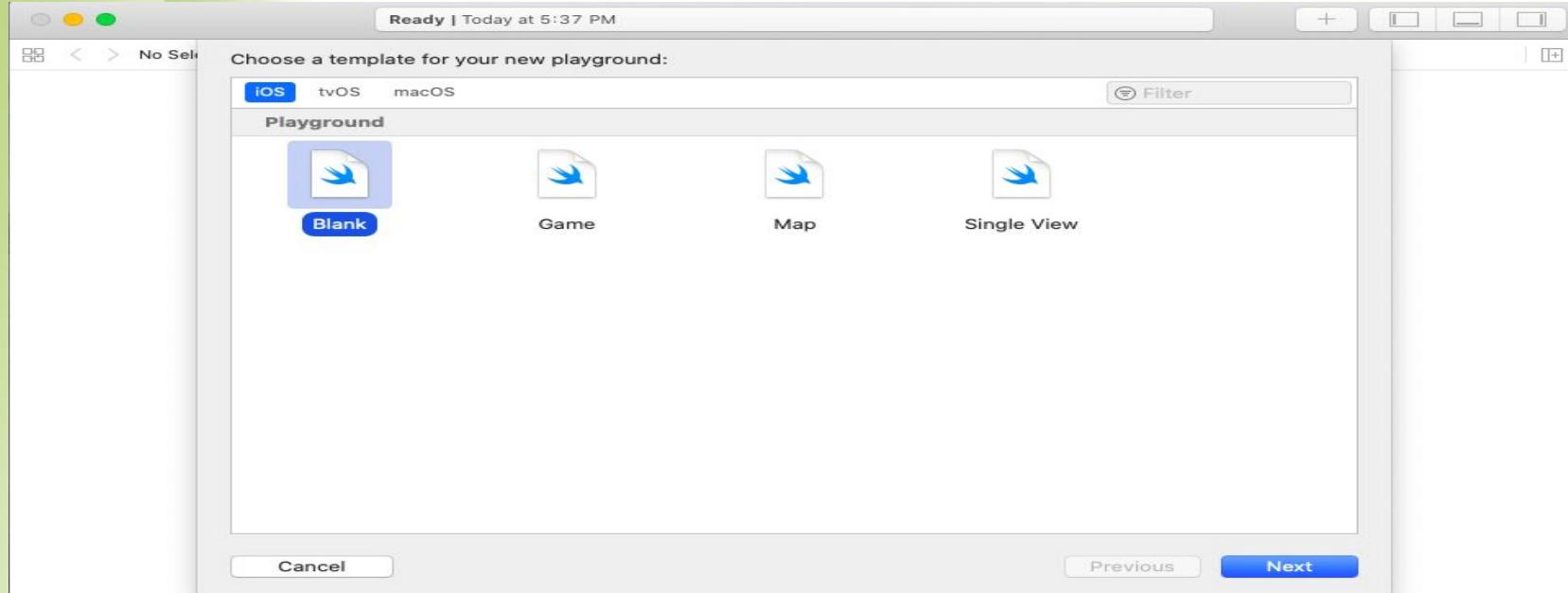
FirstApp  
~/Trash



Calculator  
~/Trash

Open another project...

# Playground



# Variable Declaration

```
1 import UIKit  
2  
3 var str = "Hi"  
4  
5 let str1 = "Hello"  
6  
7 str = "Oh"  
8  
9 str1 = "Oh"
```

var 告訴為變數  
let 告訴為常數

變數初始化值後  
能再次變更值，  
而常數不能

按此執行程式

```
1 import UIKit  
2  
3  
4 var str: String  
5  
6  
7 var str1: String = "Hi"  
8  
9  
10 var x = 1.0, y: Double = 2.0, z = 3.0  
11  
12  
13  
14
```

變數與常數宣  
告可加或不加  
型態，也可以  
選擇是否要給  
初值

連續宣告，變數  
之間使用逗點  
隔開

# Print

```
1 import UIKit  
2  
3  
4 var str: String = "Hi"  
5 let str1: String = "Jack"  
6  
7  
8 print ("HI Jack")  
9 print ("HI " + str1)  
10 print ("\((str) Jack")  
11  
12  
13 var str2 = str + " " + str1  
14 print (str2)
```

假如要print變數或常數可以使用'+'或'\'

String 能用其他String來串接



16

# Control flow

```
1 import UIKit  
2  
3 var condition: Bool = true  
4 var condition2: Int = 1  
5  
6 if condition {  
7  
8 }else if condition2 {  
9  
10 }else {  
11  
12 }
```

基本的if else 用法, 但condition部分只能是Bool型態

```
1 import UIKit  
2  
3 var n = 3  
4 switch n{  
5 case 1:  
6     break  
7 case 2:  
8     print("2")  
9 default:  
10    break  
11 }
```

switch 中每個case中一定要有程式碼, 如果不需要執行的內容可以使用break跳過

不像C/C++ 每個case最後不一定要加break也會自己跳出switch

# Function

```
var Height = 1.7 //variable  
var Weight :Double = 45  
let BMI : Double //constant
```

函式的回傳型態

```
func BMICalculator (height: Double, weight: Double) -> Double {  
    let BMI = weight / pow(height, 2)  
    return BMI  
}  
BMI = BMICalculator(height: Height, weight: Weight)  
print ("YourBMI = \(BMI)")
```

呼叫函式時  
參數要對應  
宣告時的標  
籤

# Optional

```
1 import UIKit  
2  
3 let possibleNumber = "123"  
4 let convertedNumber = Int(possibleNumber)  
5  
6  
7 let possibleNumber2: String? = nil  
8 let possibleNumber3: String? = "456"  
9  
10  
11 print(type(of: convertedNumber))  
12 print(convertedNumber)
```

假如字串是數字就  
能夠正常轉為Int

一般的Int是  
無法設為nil  
的因此會被  
設為Optional  
此資料型態

也可以自行宣告為  
Optional資料型態,  
主要是在一般型態  
後面加上'?'

Optional<Int>  
Optional(123)

```
1 import UIKit  
2  
3 let possibleNumber = "Hello"  
4 let convertedNumber = Int(possibleNumber)  
5  
6  
7 print(type(of: convertedNumber))  
8
```

假如字串不是數字  
就會回傳nil

使用type能知道目標  
變數的型態

Optional就好比一個  
箱子可能  
有東西也  
可能沒有,  
要打開才  
能知道

Optional<Int>  
nil

# Optional

```
1 import UIKit  
2  
3 let possibleNumber = "123"  
4 let convertedNumber = Int(possibleNumber)  
5  
6 print(convertedNumber!)  
7  
8  
9
```

我們可以在變數後面加上'!'來將Optional解開取得裡面的資料

```
1 import UIKit  
2  
3 let possibleNumber = "hello"  
4 let convertedNumber = Int(possibleNumber)  
5  
6 print(convertedNumber!) ! error: Execution was i  
7  
8  
9
```

但假如解開後發現是nil在某些地方會出錯

# Optional

```
1 import UIKit  
2  
3 let possibleNumber = "123"  
4 let convertedNumber = Int(possibleNumber)  
5  
6 if let Number = convertedNumber{  
7     print(Number)  
8 } else {  
9     print("無法轉換為數字")  
10 }
```



另一種方式是用if let 生成常數去接收Optional這樣比較能避免出錯

# Hello World



## Welcome to Xcode

Version 9.2 (9C40b)



### Get started with a playground

Explore new ideas quickly and easily.



### Create a new Xcode project

Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.



### Clone an existing project

Start working on something from an SCM repository.



### Segue

~/Desktop/2018 iOS/class03-1



### AlertWithInput

~/Desktop/2018 iOS/class02-2



### Alert

~/Desktop/2018 iOS/class02-2



### Date Picker

~/Desktop/2018 iOS/class02-2



### Hello world

~/Desktop/2018 iOS/class01-1



### class02-1

~/Desktop/2018 iOS/class02-1

Open another project...

# Hello World

Choose a template for your new project:

iOS watchOS tvOS macOS Cross-platform

**Application**

 Single View App	 Game	 Augmented Reality App	 Document Based App	 Master-Detail App
 Tabbed App	 Sticker Pack App	 iMessage App		

**Framework & Library**

 Framework	 Static Library	 Metal Library		
--	---	---	--	--

# Hello World

Choose options for your new project:

Product Name:

Team:

Organization Name:

Organization Identifier:

Bundle Identifier:

Language:  

User Interface:  

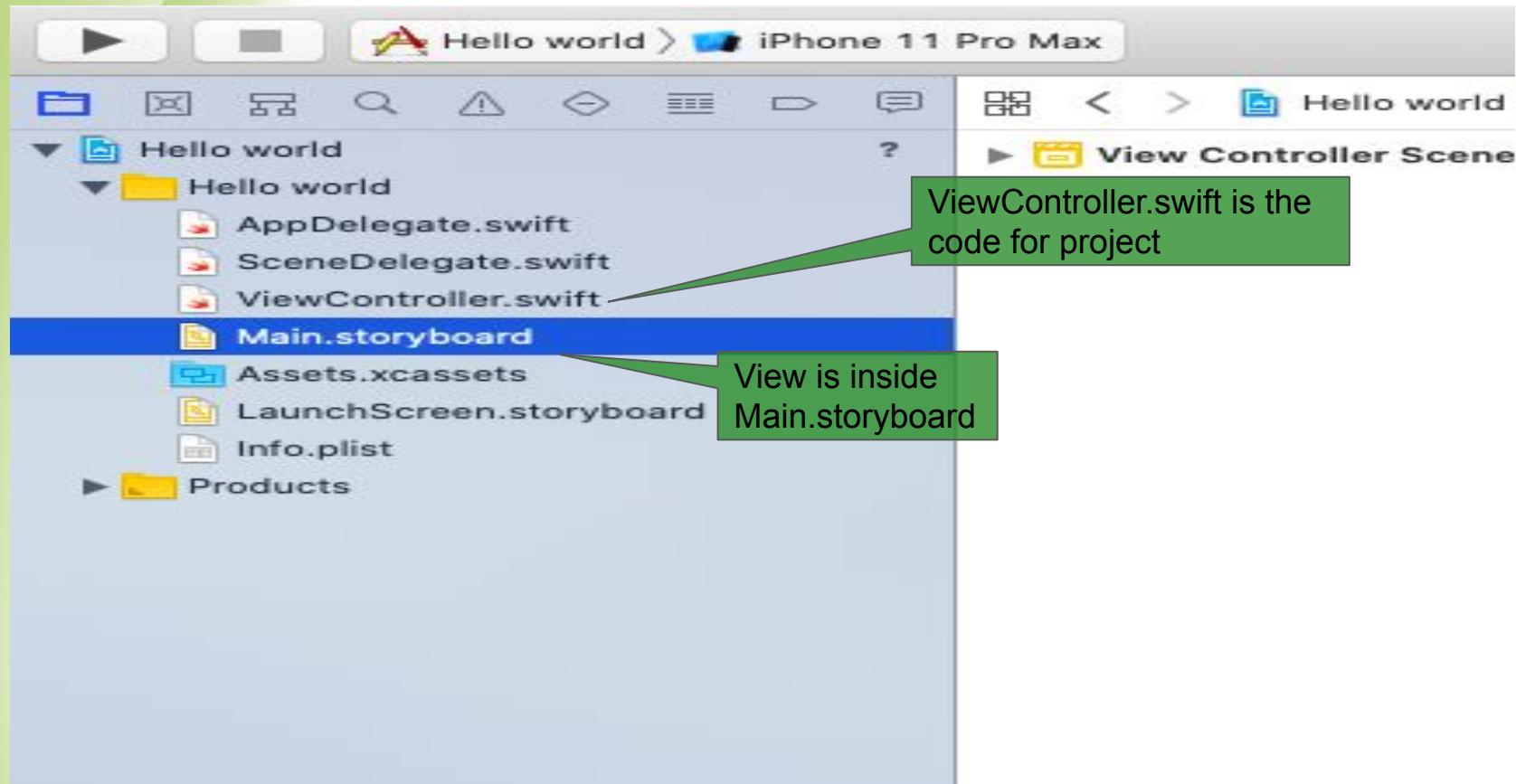
- Use Core Data
- Use CloudKit
- Include Unit Tests
- Include UI Tests

# Hello World

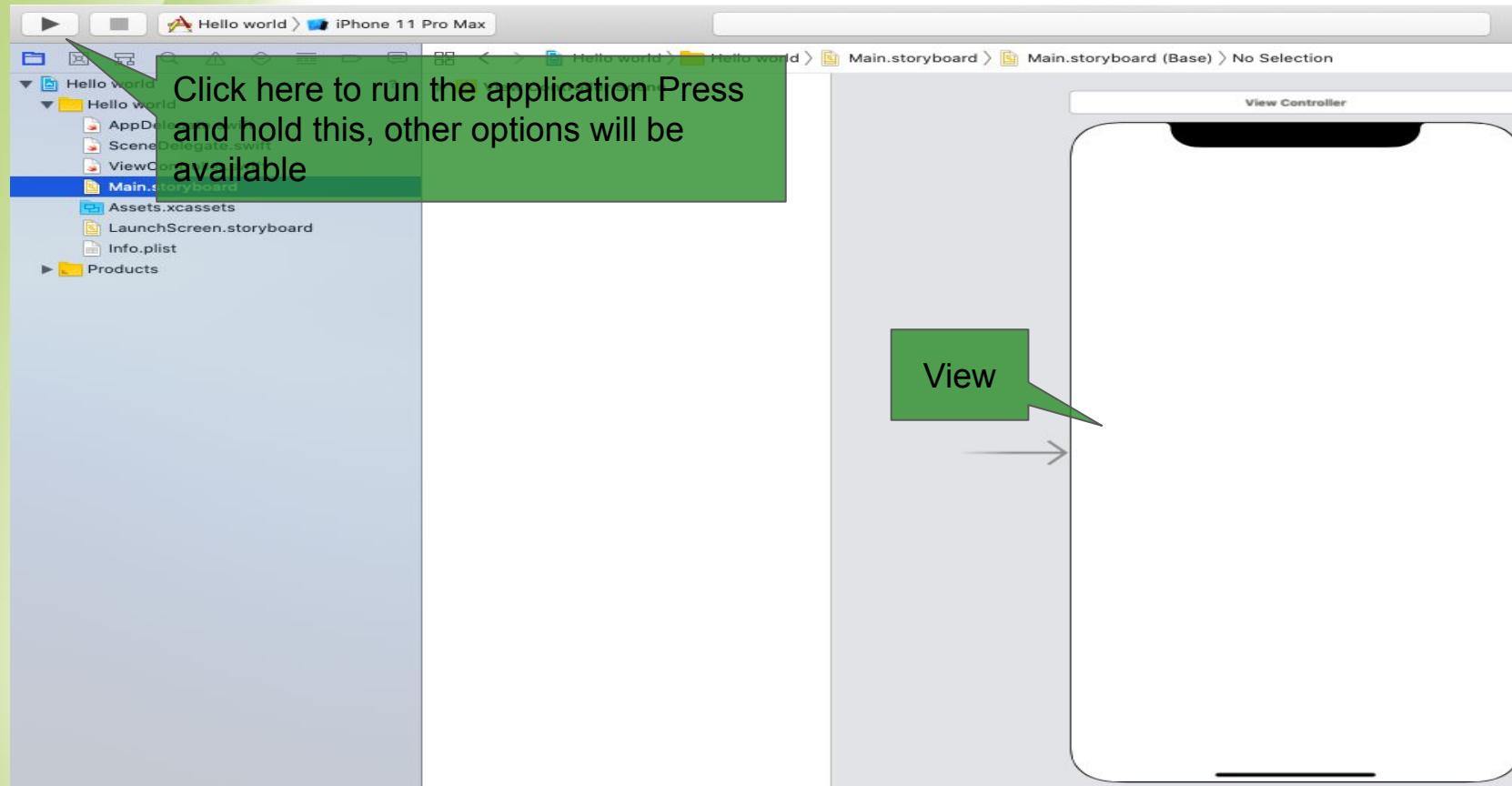
## ▼ Deployment Info

Target	Device
iOS 13.2	<input checked="" type="checkbox"/> iPhone <input checked="" type="checkbox"/> iPad <input type="checkbox"/> Mac (requires macOS 10.15)
Main Interface	<input type="text" value="Main"/>
Device Orientation	<input checked="" type="checkbox"/> Portrait <input type="checkbox"/> Upside Down <input checked="" type="checkbox"/> Landscape Left <input checked="" type="checkbox"/> Landscape Right
Status Bar Style	<input type="text" value="Default"/> <input type="checkbox"/> Hide status bar <input type="checkbox"/> Requires full screen <input type="checkbox"/> Supports multiple windows

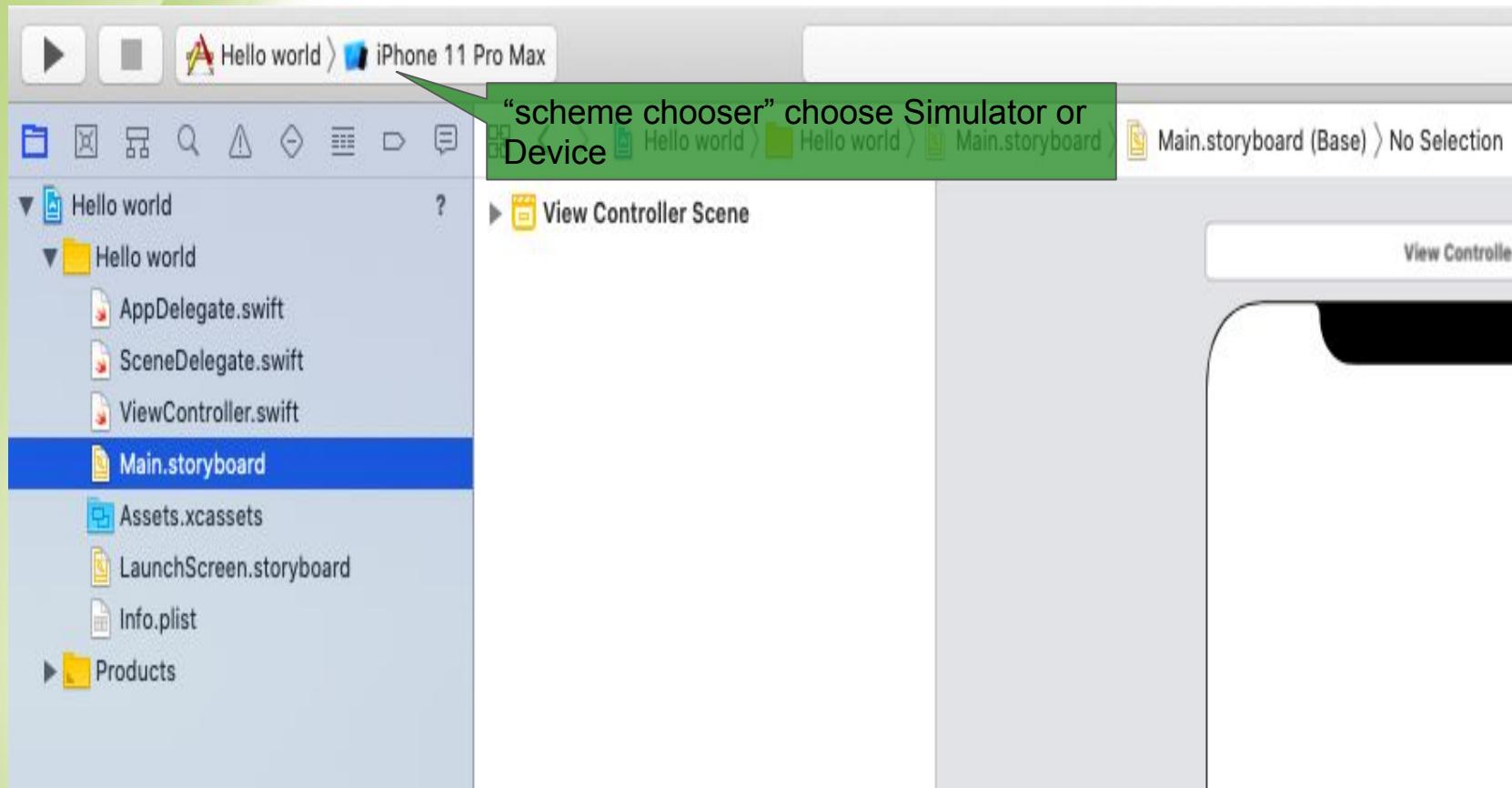
# Hello World



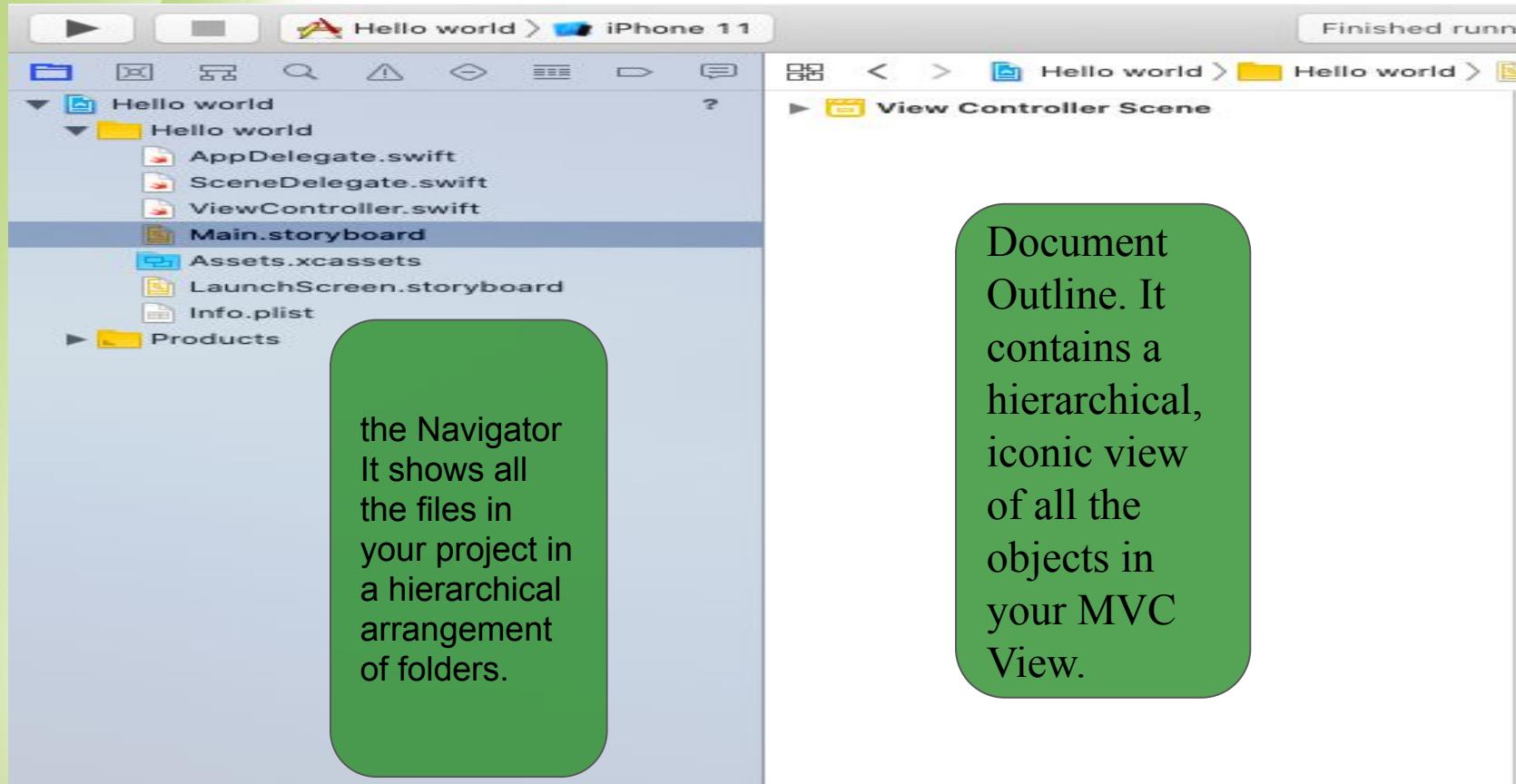
# Hello World



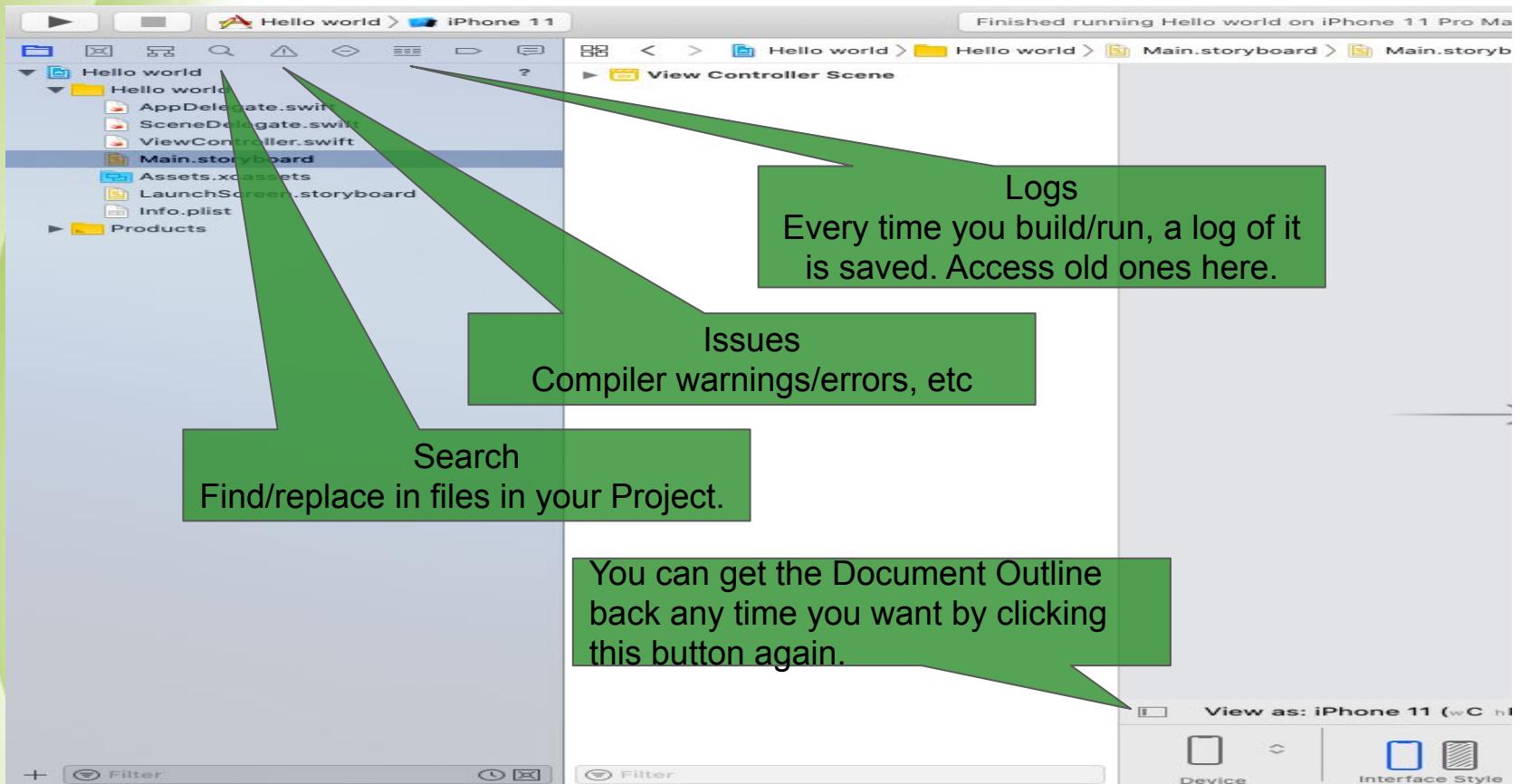
# Hello World



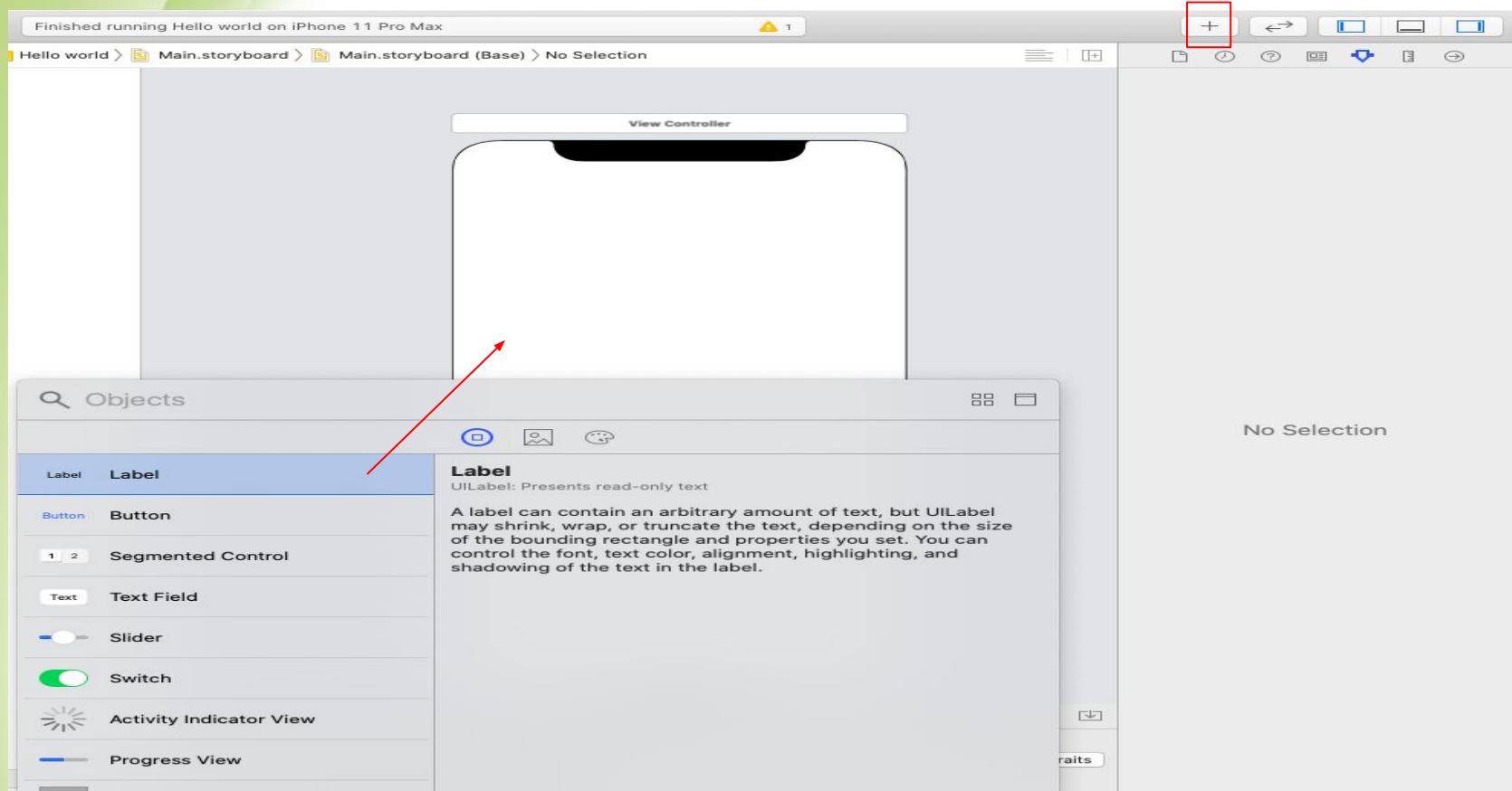
# Hello World



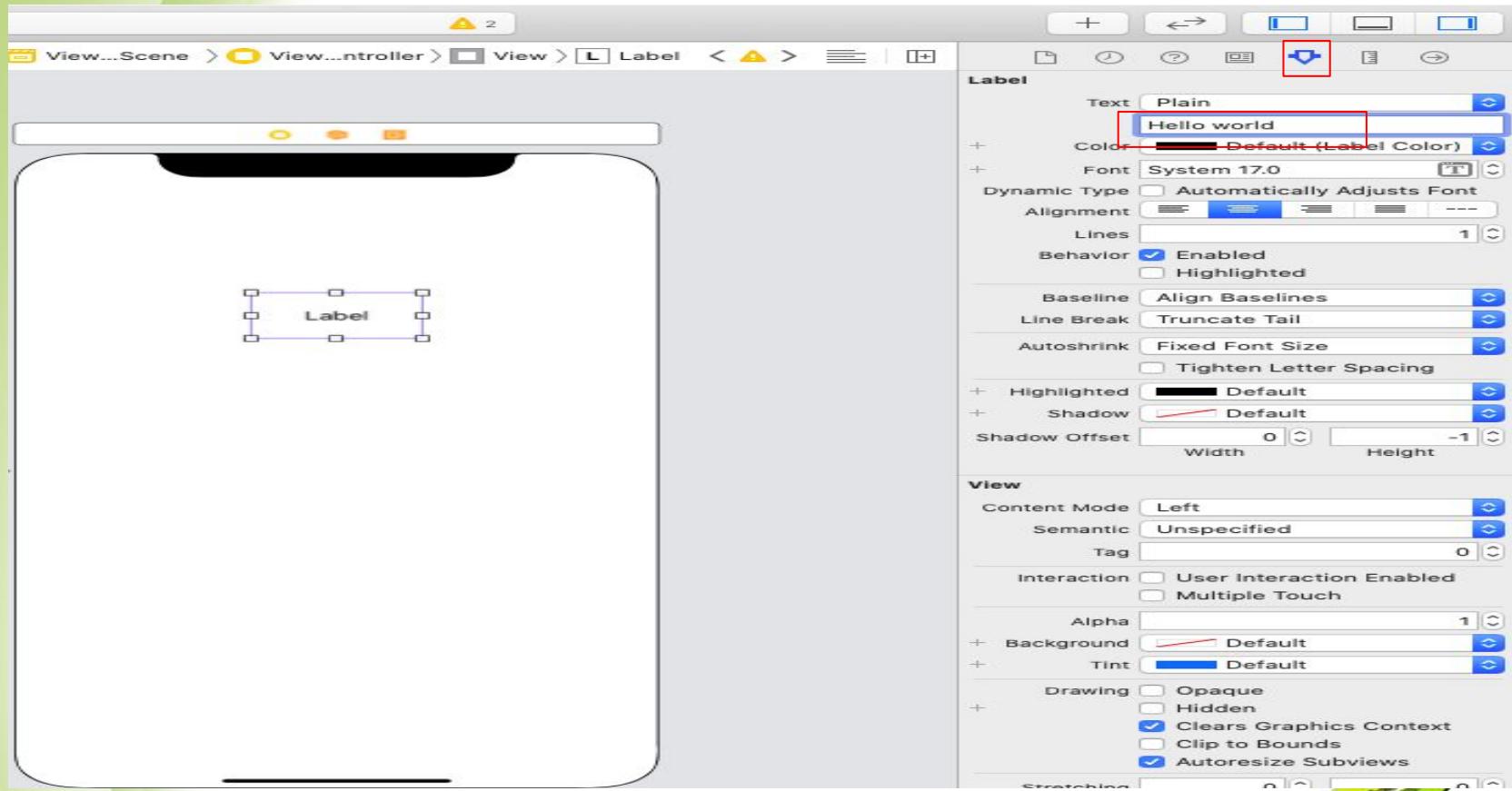
# Hello World



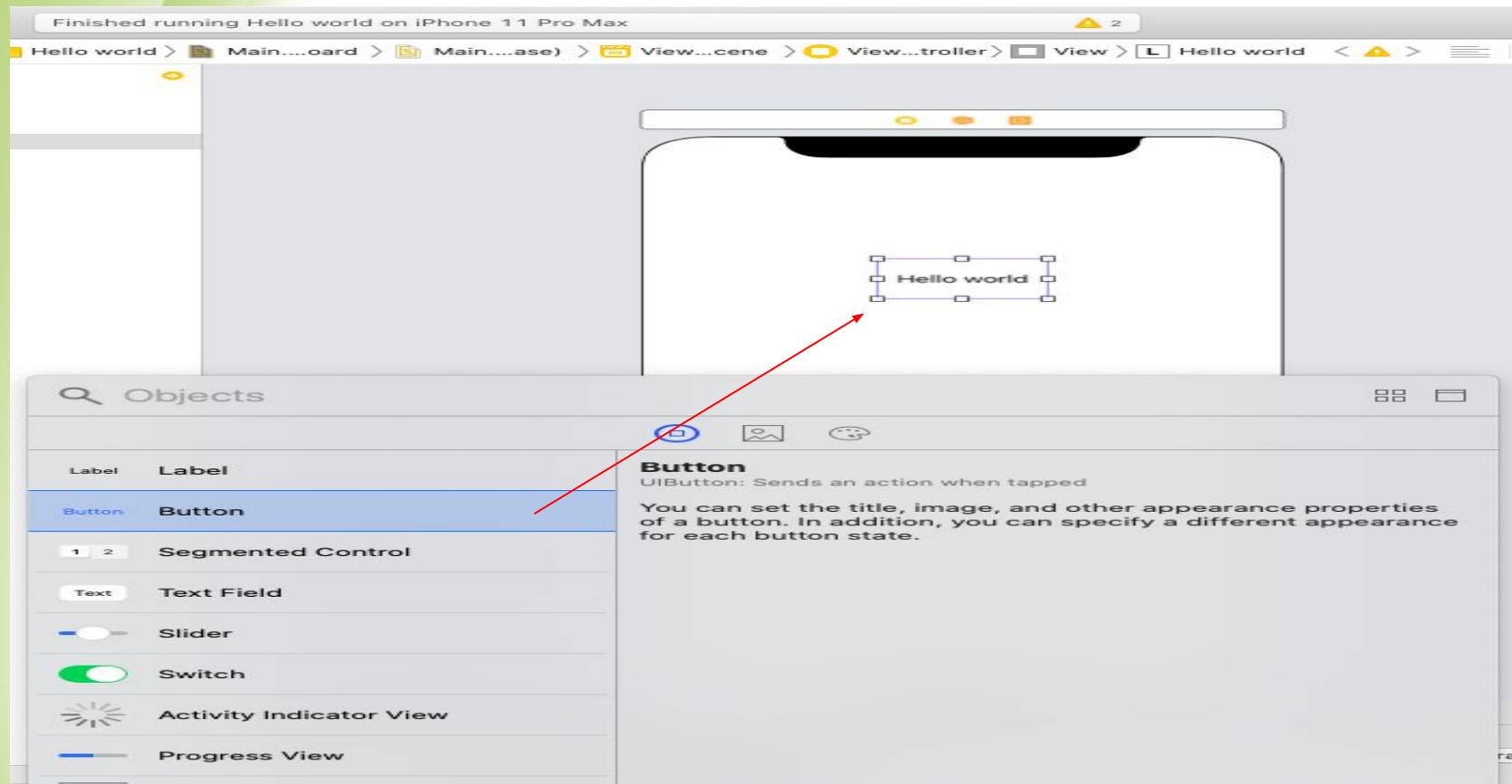
# Hello World



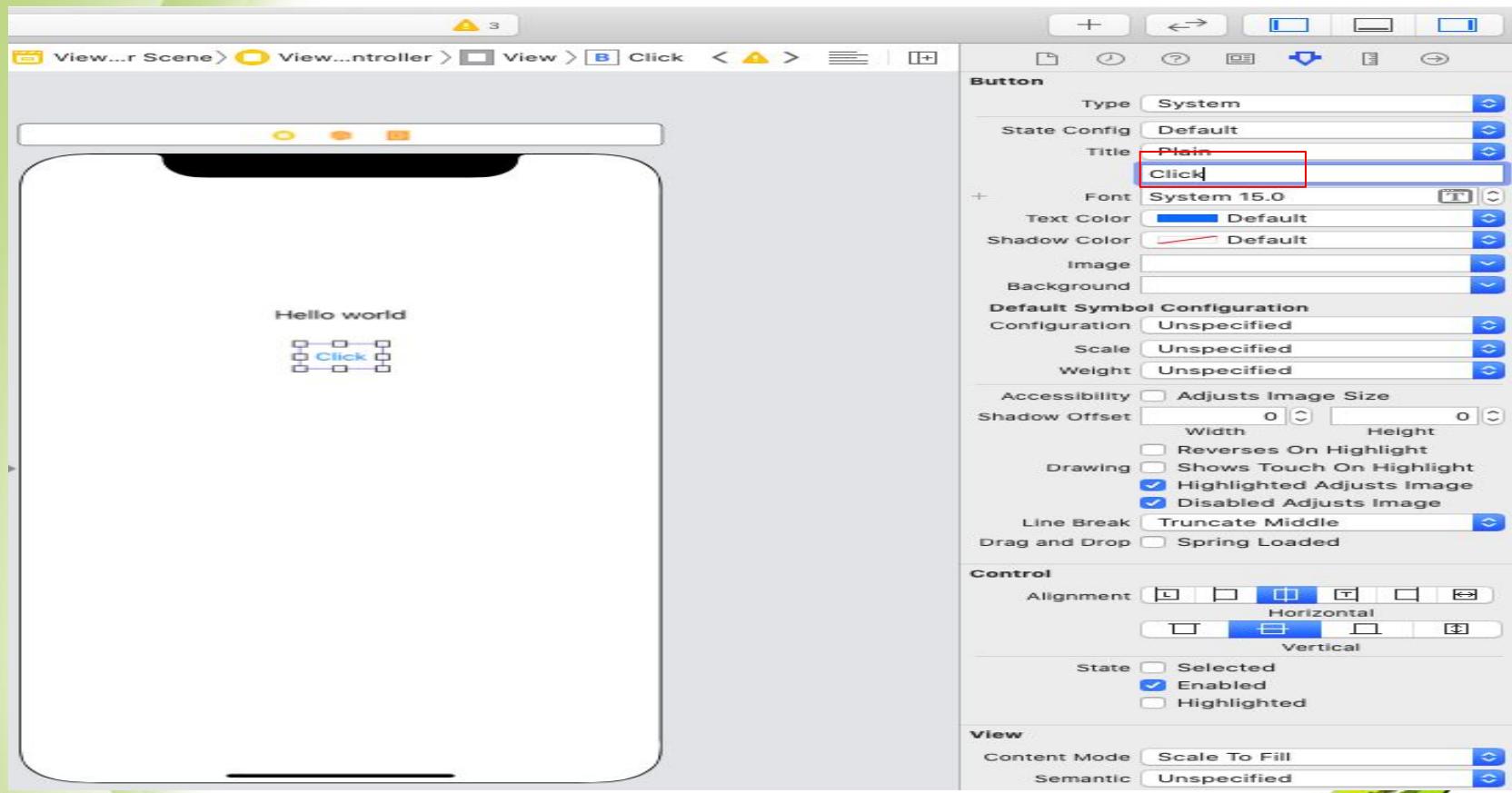
# Hello World



# Hello World



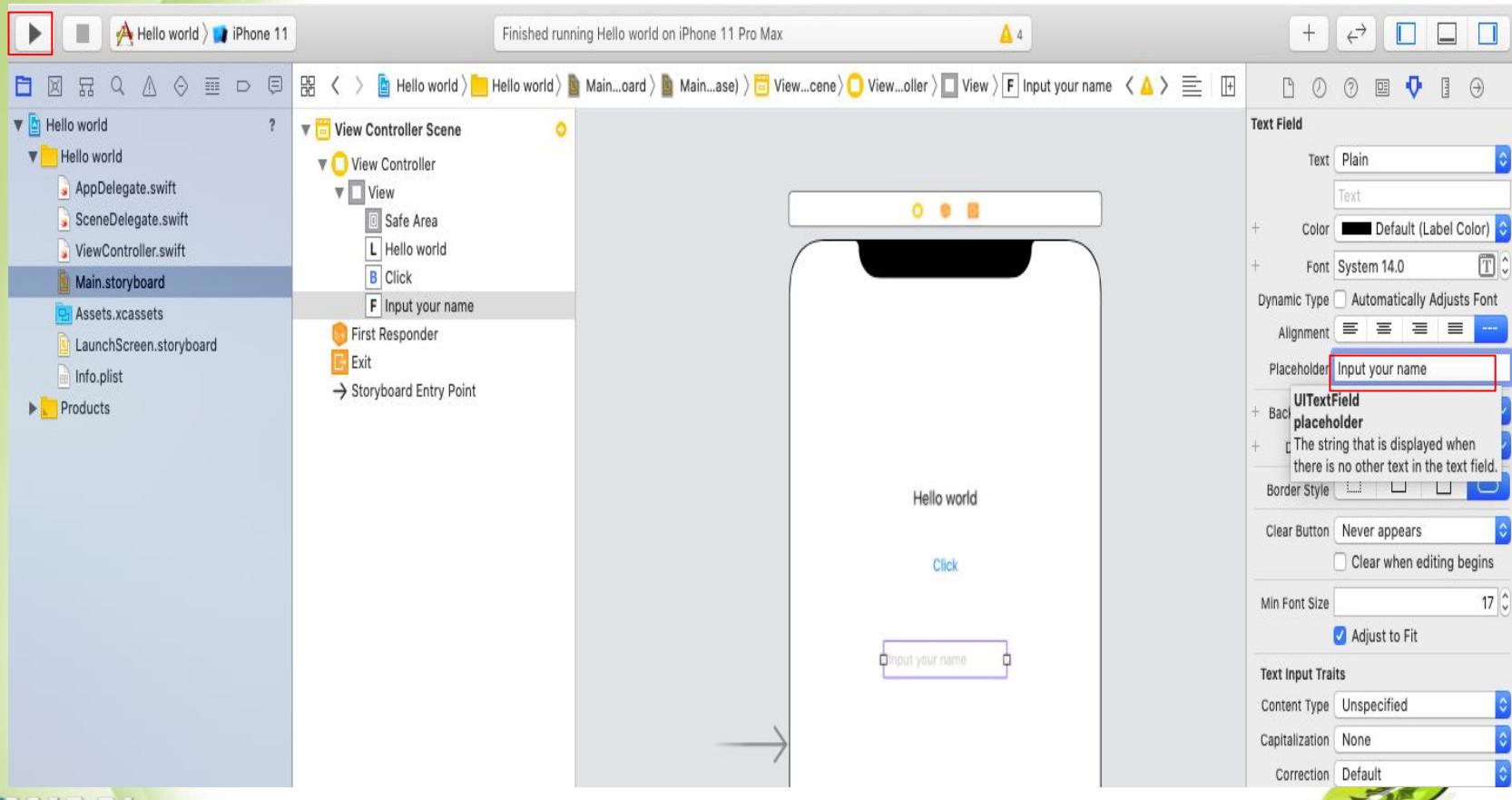
# Hello World



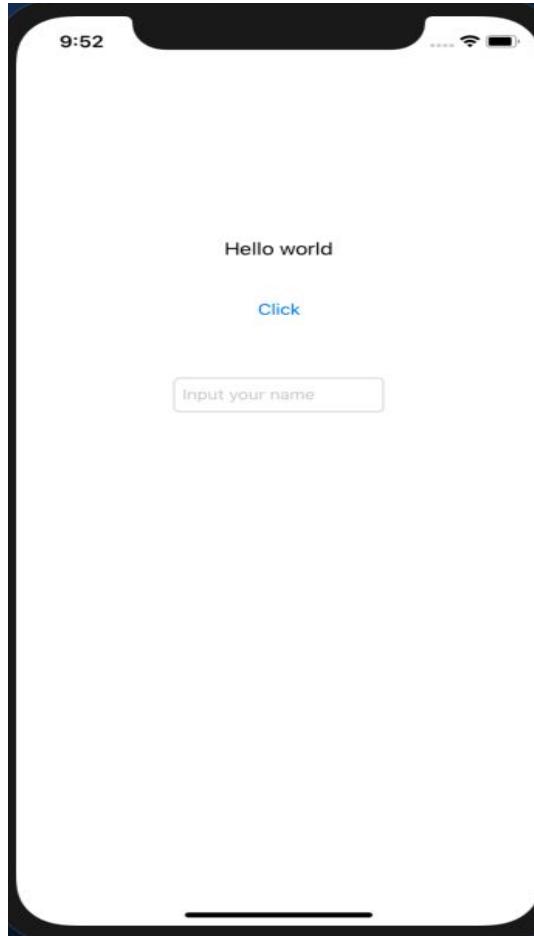
# Hello World



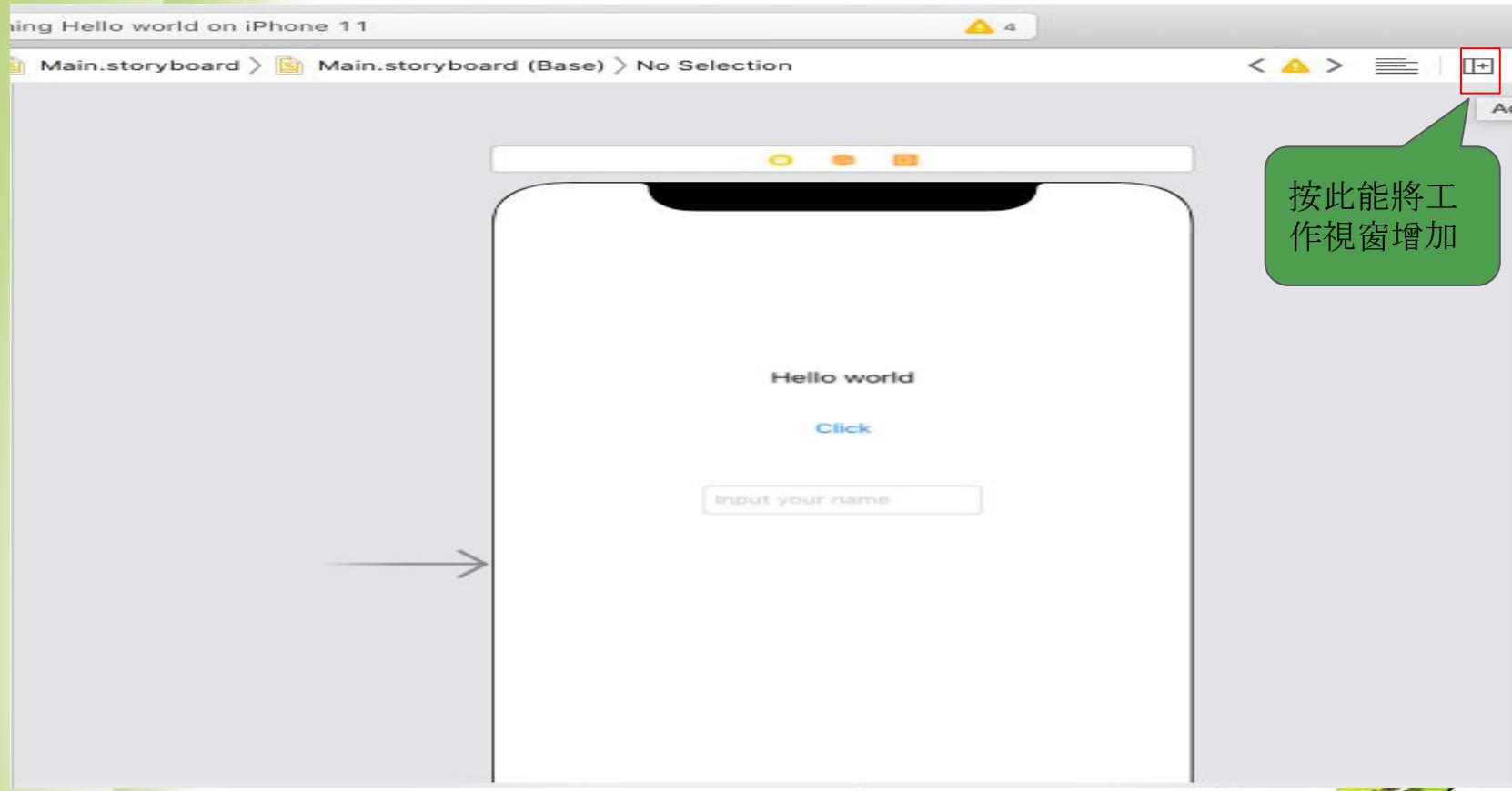
# Hello World



# Hello World



# Hello World



# Hello World

The screenshot shows the Xcode interface with a project named "Hello world". The left sidebar displays the project structure with "ViewController.swift" selected. The main editor shows the Swift code for the ViewController:

```
1 // ViewController.swift
2 // Hello world
3 // Created by evan on 2020/2/26.
4 // Copyright © 2020 evan. All rights reserved.
5 //
6 import UIKit
7
8
9 class ViewController: UIViewController {
10
11     override func viewDidLoad() {
12         super.viewDidLoad()
13         // Do any additional setup after loading the view.
14     }
15
16 }
17
18
19 }
20
21 }
```

The right side of the interface shows the iPhone 11 simulator. The screen displays the text "Hello world" and "Click" above a text input field with the placeholder "Input your name". At the bottom, there is a button labeled "View as: iPhone 11 (wC hR)".

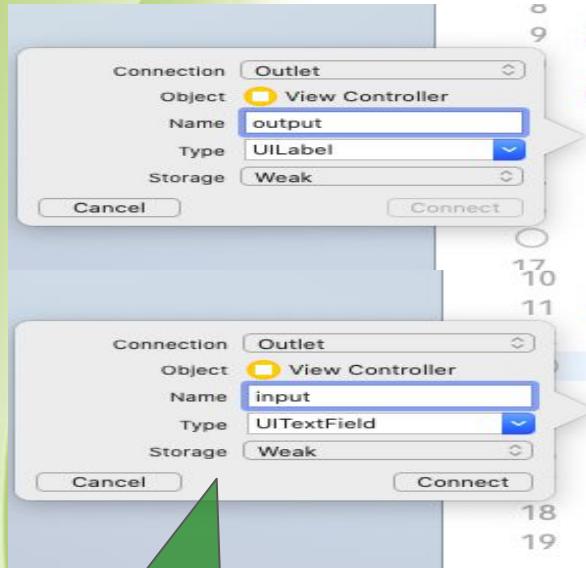
# Hello World

The screenshot shows the Xcode interface with two main panes. The left pane displays the `ViewController.swift` file, which contains the following code:

```
1 // ViewController.swift
2 // Hello world
3 //
4 //
5 // Created by evan on 2020/2/26.
6 // Copyright © 2020 evan. All rights
7 // reserved.
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13
14
15
16     override func viewDidLoad() {
17         super.viewDidLoad()
18         // Do any additional setup after
19             // loading the view.
20     }
21
22 }
```

A red arrow points from the line `super.viewDidLoad()` in the code to the storyboard on the right. The storyboard shows a single view controller with a button labeled "Hello world". A callout bubble with a green background and white text says: "點選並按住 control拖移至左側". The storyboard also shows a text input field with placeholder text "Input your name".

# Hello World



```
import UIKit  
  
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
  
    class ViewController: UIViewController {  
  
        @IBOutlet weak var output: UILabel!  
  
        override func viewDidLoad() {  
            super.viewDidLoad()  
            // Do any additional setup after  
            // loading the view.  
    }  
  
}
```

將label與  
TextField連結到  
viewcontroller

連結後會產生初  
這兩條宣告

```
class ViewController: UIViewController {  
  
    @IBOutlet weak var output: UILabel!  
    @IBOutlet weak var input: UITextField!
```

```
override func viewDidLoad() {
```

# Hello World



```
class ViewController: UIViewController {  
  
    @IBOutlet weak var output: UILabel!  
    @IBOutlet weak var input: UITextField!  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after  
        loading the view.  
    }  
}
```

```
@IBAction func BtnClick(_ sender: UIButton)  
{  
17  
}
```

產生出  
Click fuc

# Hello World

```
8  
9 import UIKit  
10  
11 class ViewController: UIViewController {  
12  
13     @IBOutlet weak var output: UILabel!  
14     @IBOutlet weak var input: UITextField!  
15     @IBAction func BtnClick(_ sender: UIButton) {  
16         output.text = "Hello, \(input.text)"  
17     }  
18     override func viewDidLoad() {  
19         super.viewDidLoad()  
20         // Do any additional setup after loading the view.  
21     }  
22  
23 }  
24  
25  
26
```

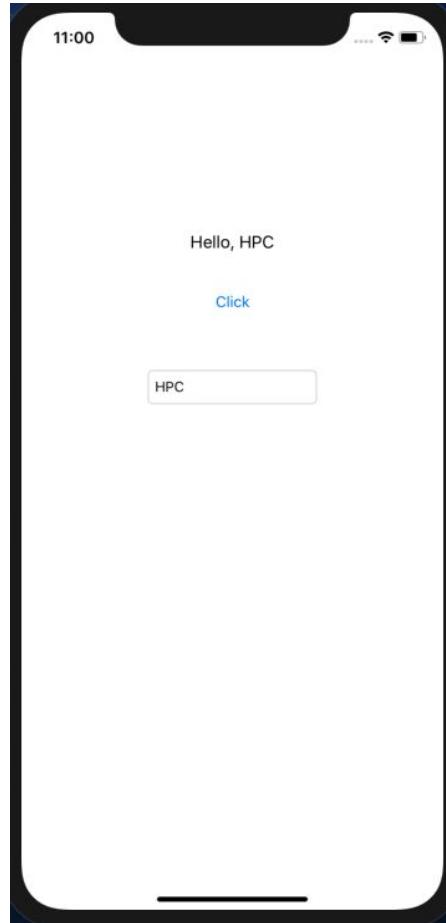
將label的顯示  
"Hello, TextField的文字"

此處的  
input.text為  
Optional因  
此要將他拆  
開

# Hello World

```
 9 import UIKit  
10  
11 class ViewController: UIViewController {  
12  
    @IBOutlet weak var output: UILabel!  
    @IBOutlet weak var input: UITextField!  
    @IBAction func BtnClick(_ sender: UIButton) {  
        output.text = "Hello, \(input.text!)"  
    }  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the  
        // view.  
    }  
}
```

# Hello World



# Hello World

```
8  
9 import UIKit  
10  
11 class ViewController: UIViewController {  
12  
13     @IBOutlet weak var output: UILabel!  
14     @IBOutlet weak var input: UITextField!  
15     @IBAction func BtnClick(_ sender: UIButton) {  
16         let controller = UIAlertController(title: "您還未輸入名字！",  
17             message: "請輸入名字", preferredStyle: .alert)  
18         let okAction = UIAlertAction(title: "OK", style: .default, handler:  
19             nil)  
20         controller.addAction(okAction)  
21  
22         if input.text! != "" {  
23             output.text = "Hello, \(input.text!)"  
24         } else {  
25             present(controller, animated: true, completion: nil)  
26         }  
27     }  
28 }
```

先宣告一個  
UIAlert, title  
決定訊息標  
題 message  
決定訊息內  
容



做出一個  
Alert的互  
動按紐

將訊息顯示  
在畫面

# Hello World

