

# iOS 應用程式開發

## Week 2

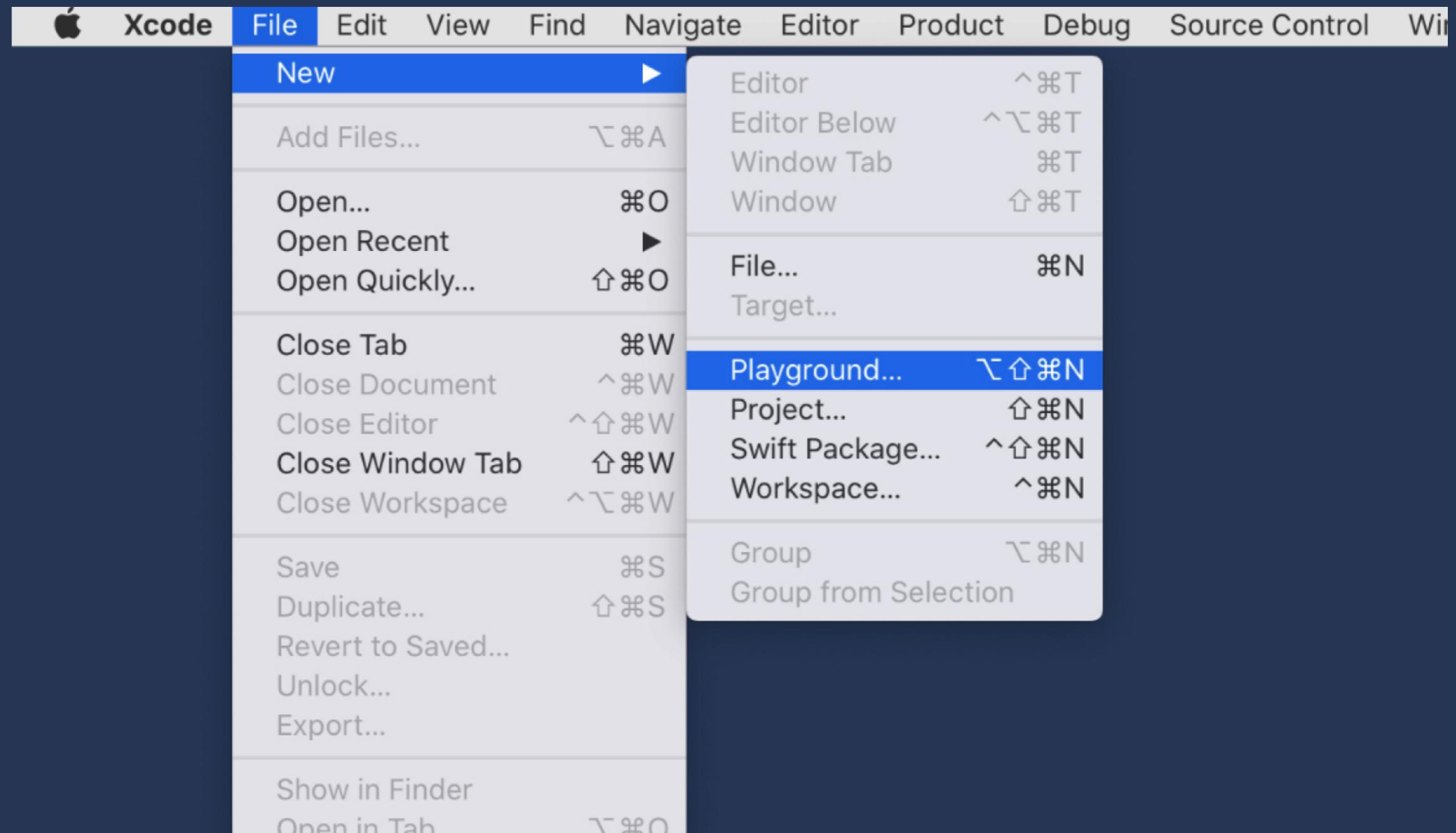
# 本週內容

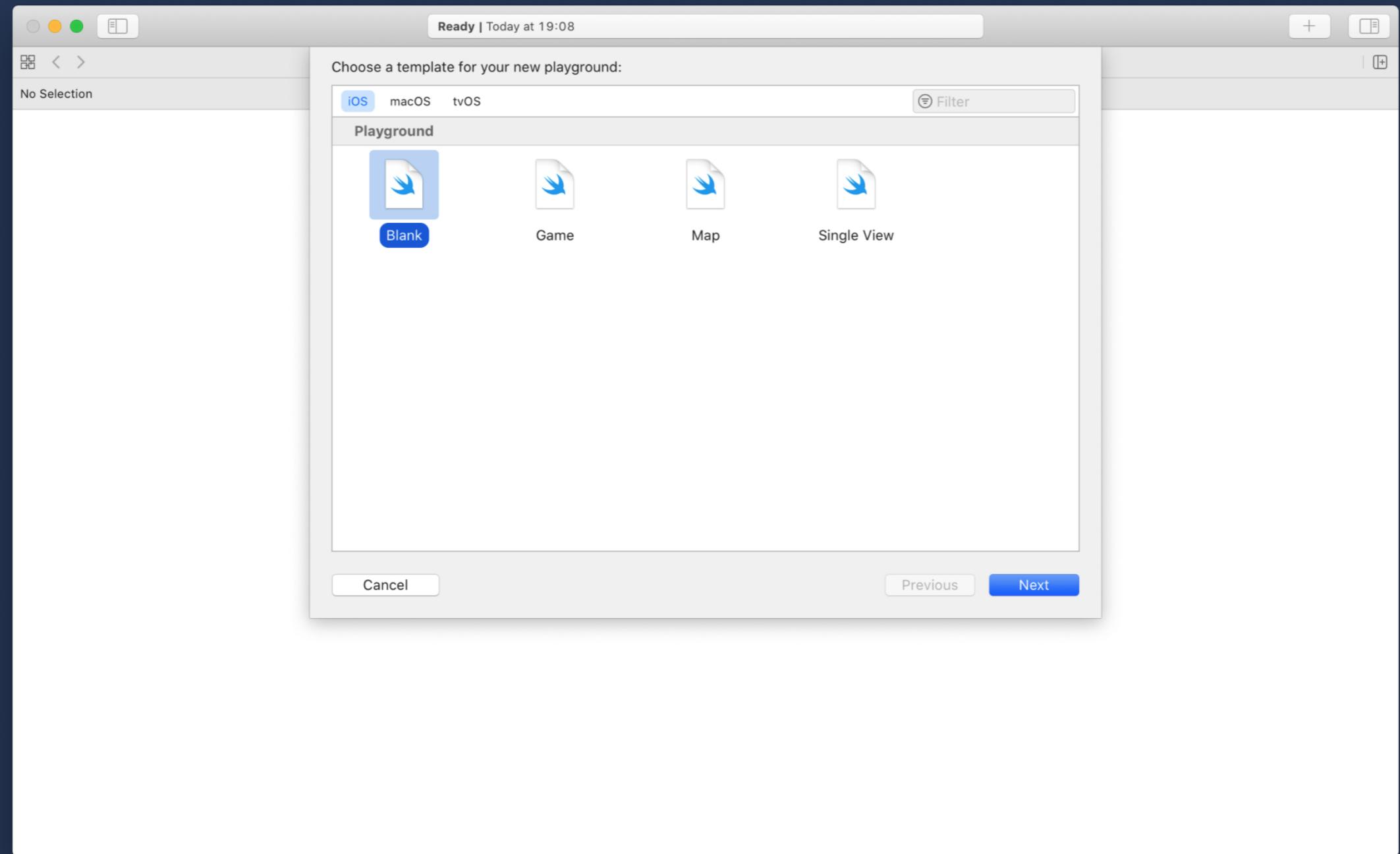
- Swift 語言簡單介紹
- 範例講解
  - Calculator 介面
  - Ask Your Name
  - Classwork: Simple Calculator

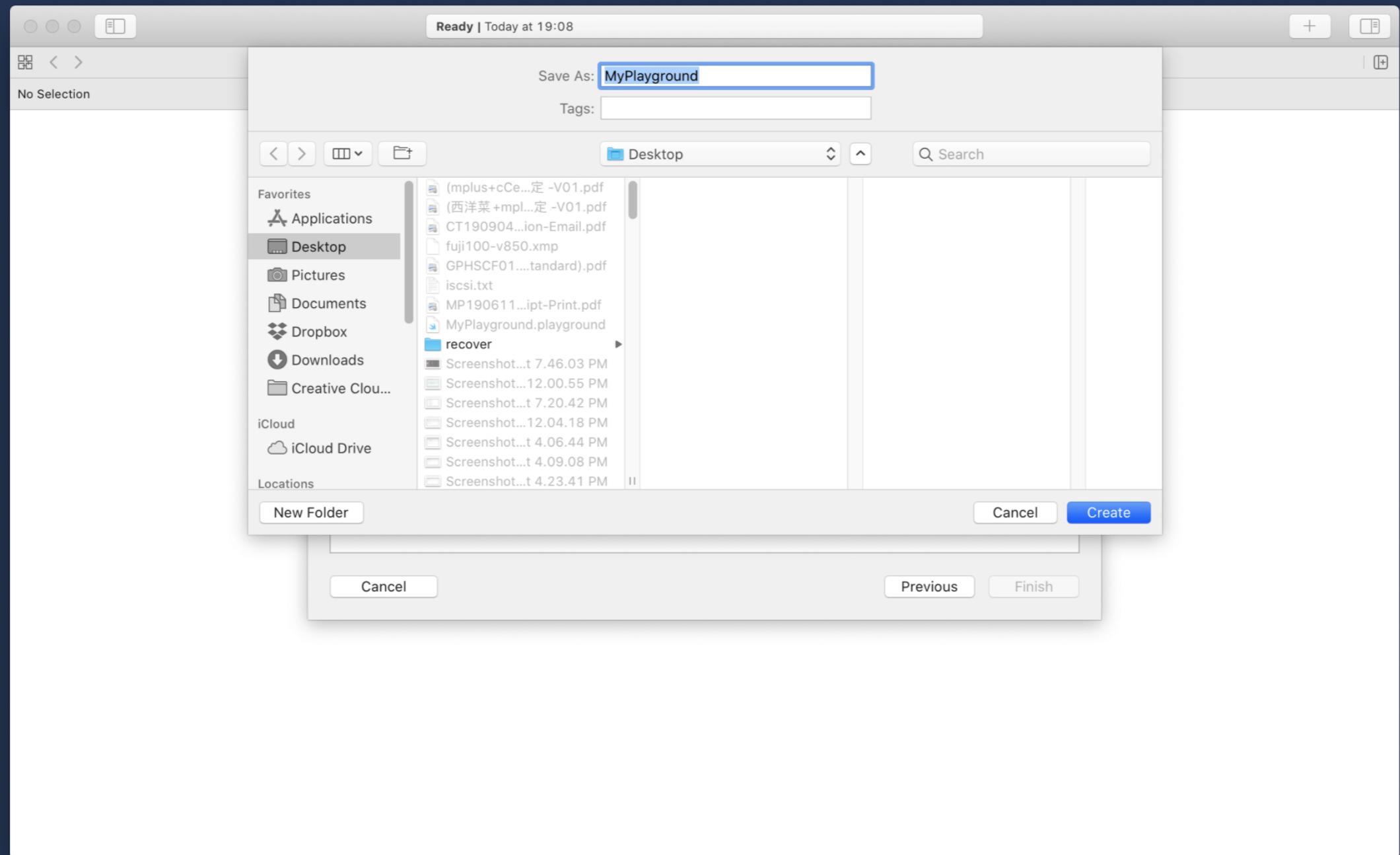
# iOS 的程式語言

- Objective-C (??? 至現在)
- Swift (2014 年發表)
- 新開發的 iOS 程式大部分以 Swift 為主
- 較大型的 iOS 的程式仍然以 Objective C 語言為主

# 使用 Playground



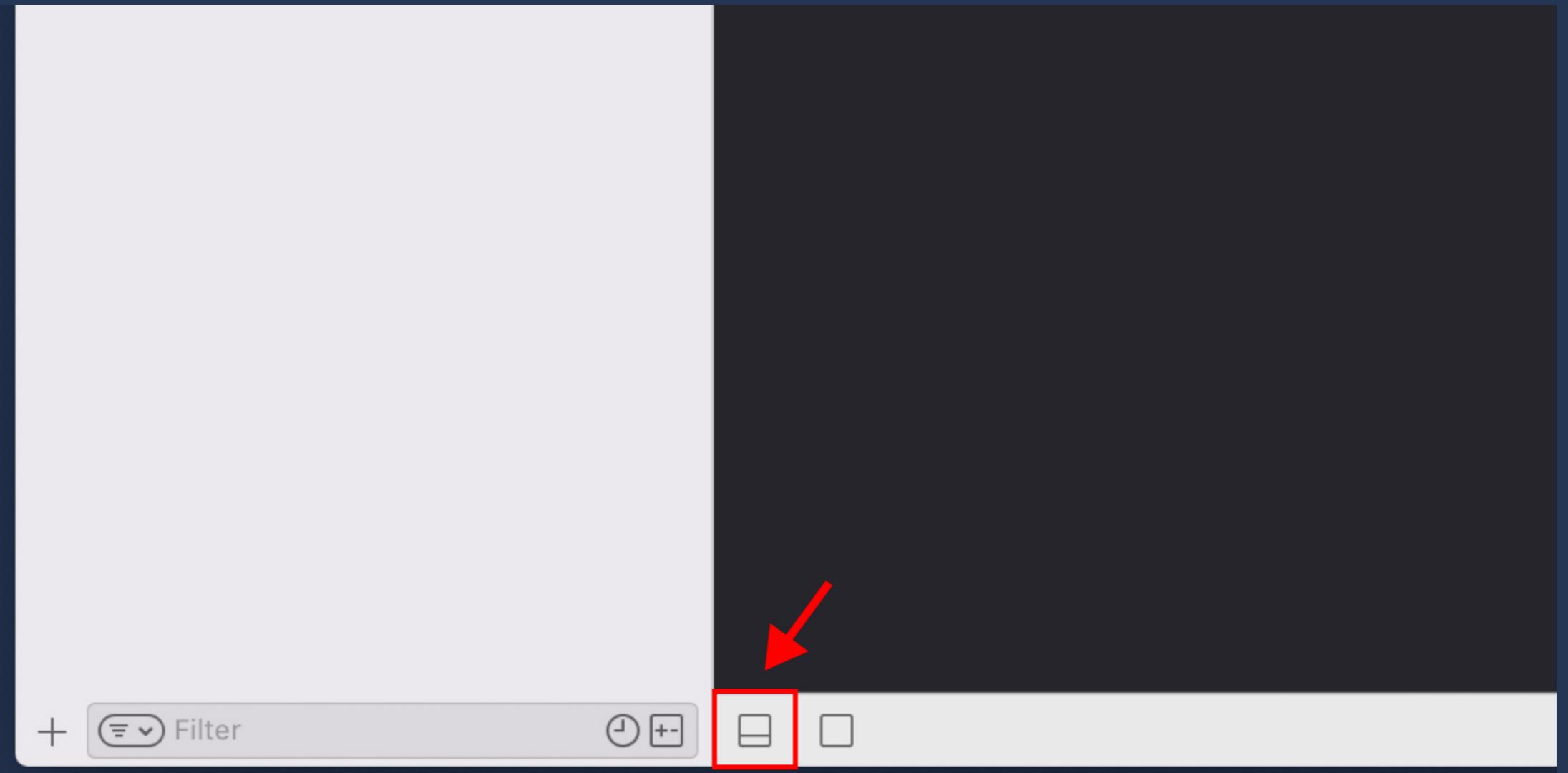


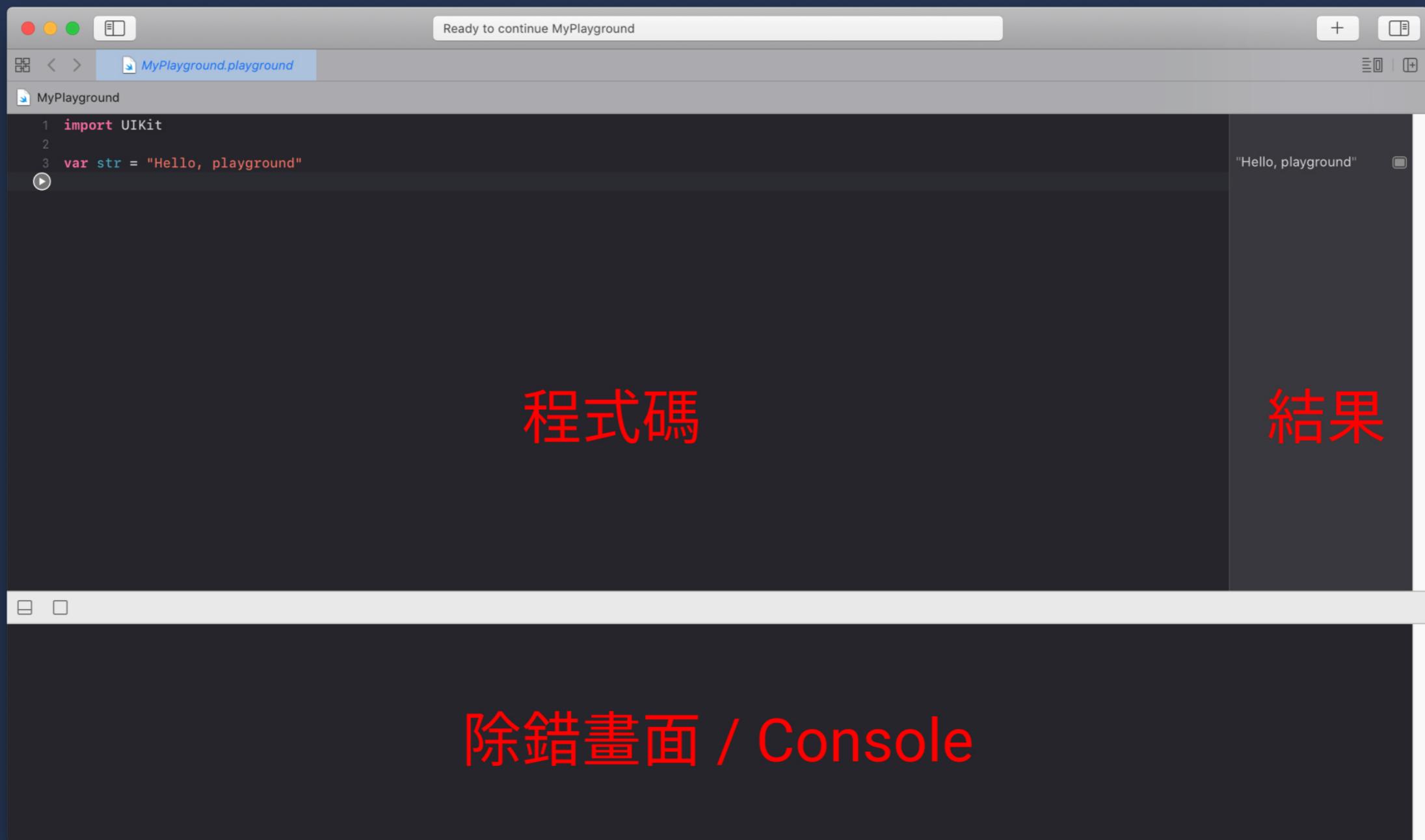


The screenshot shows the Xcode playground interface. The title bar says "Ready to continue MyPlayground". The left sidebar shows a project named "MyPlayground" with "Sources" and "Resources" folders. The main area contains a playground file "MyPlayground.playground" with the following code:

```
1 import UIKit
2
3 var str = "Hello, playground"
```

To the right of the code editor is a dark panel labeled "Hello, playground" which displays the output of the code: "Hello, playground". Red text overlays are present: "程式碼" (Code) is positioned above the code editor, and "結果" (Result) is positioned above the output panel.





A screenshot of the Xcode playground interface. The main area shows a Swift script named "MyPlayground" with the following code:

```
1 import UIKit
2
3 var str = "Hello, playground"
4
5 print("Hello, This is iOS Basic Course!!")
```

To the right of the code, there is a preview pane displaying the output of the print statement: "Hello, playground". Below the main editor, a smaller preview pane shows the full output: "Hello, This is iOS Basic Course!!\n". A large red arrow points from the text "試試打一段" (try it) to the line of code "print("Hello, This is iOS Basic Course!!")".

# 試試打入...

```
print("....")
print("a\nb\nc\n")
```

A screenshot of an Xcode playground window titled "MyPlayground.playground". The title bar includes standard OS X window controls and a status message "Ready to continue MyPlayground". The main editor area shows the following Swift code:

```
1 import UIKit
2
3 print("a\nb\nc\n")
```

The output pane on the right displays the result of the print statement: "a\nb\nc\n". Below the editor, a sidebar contains the letters "a", "b", and "c" followed by a cursor character "|".

# 試試打入...

1+5

5-2

8\*4

32/4

# 試試打入...

2 + 6

2+6

2+ 6 // 出現錯誤

2 +6 // 出現錯誤

The screenshot shows an Xcode playground window titled "MyPlayground" with the file "MyPlayground.playground" selected. The code editor contains the following Swift code:

```
1 import UIKit
2
3 2 + 6
4
5 2+6
6
7 2+ 6
8
9 2_+6
```

Two error messages are displayed in the status bar:

- "2 ✘ '+' is not a postfix unary operator"
- "Consecutive statements on a line must be separated by ';' "

The playground output window at the bottom shows the following errors:

```
error: MyPlayground.playground:7:3: error: consecutive statements on a line must be separated by ';'
2+ 6
^
;

error: MyPlayground.playground:9:2: error: consecutive statements on a line must be separated by ';'
2 _+6
^
;
```

# 試試打入...

22 / 7

22.0 / 7.0

350 / 5 + 2

350 / (5 + 2)

A screenshot of the Xcode playground interface. The title bar says "Ready to continue MyPlayground". The file tab shows "MyPlayground.playground". The code editor contains the following Swift code:

```
1 import UIKit
2
3 22 / 7
4
5 22.0 / 7.0
6
7 // ---
8
9 350 / 5 + 2
10
11 350 / (5 + 2)
```

The right panel displays the results of the calculations:

Result	Description
3	Integer division of 22 by 7
3.142857142857143	Floating-point division of 22.0 by 7.0
72	Integer division of 350 by 5, then adding 2
50	Floating-point division of 350 by the result of (5 + 2)

22 / 7 = 3 Integer 整數

22.0 / 7.0 = 3.142857142857143  
↑  
Double / Float 浮點數

# 試試打入...

```
let anInteger:Int = 3
```

```
let aDouble:Double = 3.142857142857143
```

```
print(anInteger)
```

```
print(aDouble)
```

A screenshot of an Xcode playground window titled "MyPlayground.playground". The playground contains the following Swift code:

```
1 import UIKit
2
3 let anInteger:Int = 3
4 let aDouble:Double = 3.142857142857143
5
6 print(anInteger)
7 print(aDouble)|
```

The right pane shows the output of the print statements:

```
3
3.142857142857143
"3\n"
"3.142857142857143\n"
```

The bottom-left corner of the playground window shows a preview of the output, displaying the integers 3 and 3.142857142857143.

# 常量 / 常數 (Constant)

`let 名稱: 類型 = 內容`

如：`let anInteger:Int = 3`

- 用途：如 `pi = 3.14159`，定義後不會改變的常數，方便在程式碼各處使用

# 試試打入...

```
var age:Int = 5  
age = 6  
print(age)
```

A screenshot of an Xcode playground window titled "MyPlayground.playground". The title bar includes standard OS X window controls and a status message "Ready to continue MyPlayground". The main area shows a Swift script named "MyPlayground" with the following code:

```
1 import UIKit
2
3 var age:Int = 5
4 age = 6
5 print(age)
```

The output pane on the right displays the results of the print statement:

```
5
6
"6\n"
```

The bottom pane shows the value "6" in the console.

# 變量 / 變數 (Variable)

`var` 名稱: 類型 = 內容

如：`var age:Int = 5`

- 用途：用作暫存，定義變量後，在程式其他地方可以再改變其內容

# 將剛才的程式碼改成...

```
let age:Int = 5  
age = 6 // 錯誤  
print(age)
```

A screenshot of an Xcode playground window titled "MyPlayground.playground". The playground file is named "MyPlayground". The code editor contains the following Swift code:

```
1 import UIKit
2
3 let age:Int = 5
4 age = 6
5 print(age)
```

The line "age = 6" is highlighted in red, indicating a syntax error. A tooltip message "Cannot assign to value: 'age' is a 'let' constant" appears next to the error. The output pane at the bottom shows the result of the code execution: "6".

# 關於名稱

建議的命名方法：

- anInteger
- myCurrentAge
- theAddressOfPuiChing

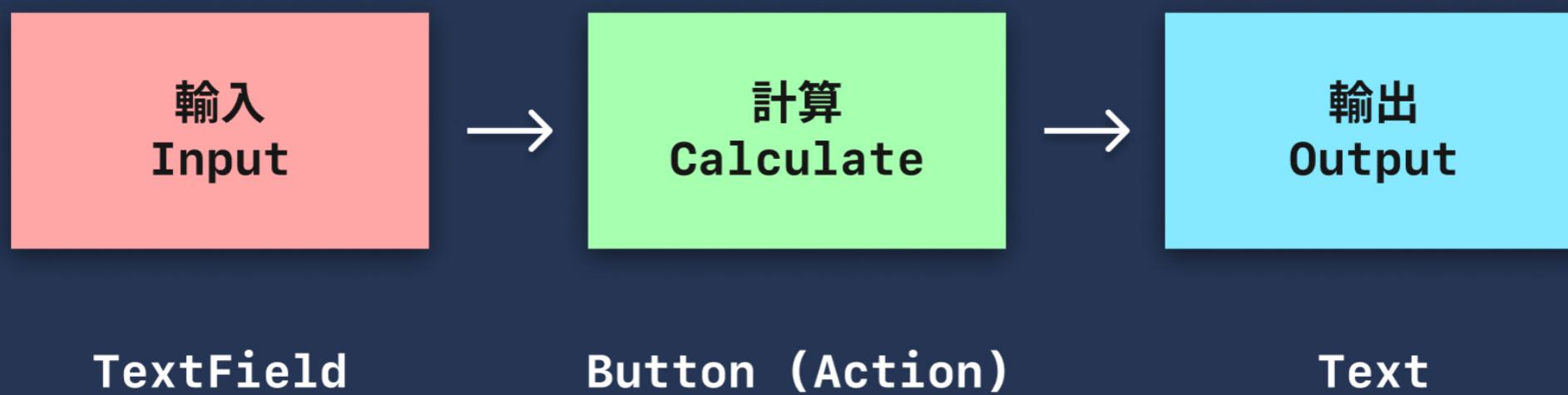
沒有錯但不是慣用的方式：

- myemailaddress
- name

# 小結

- `print()`
- `+, -, *, /`
- `let` 常量 / 常數 (Constant)
- `var` 變量 / 變數 (Variable)
- 關於常量 / 變量名稱

# 範例: Ask Your Name

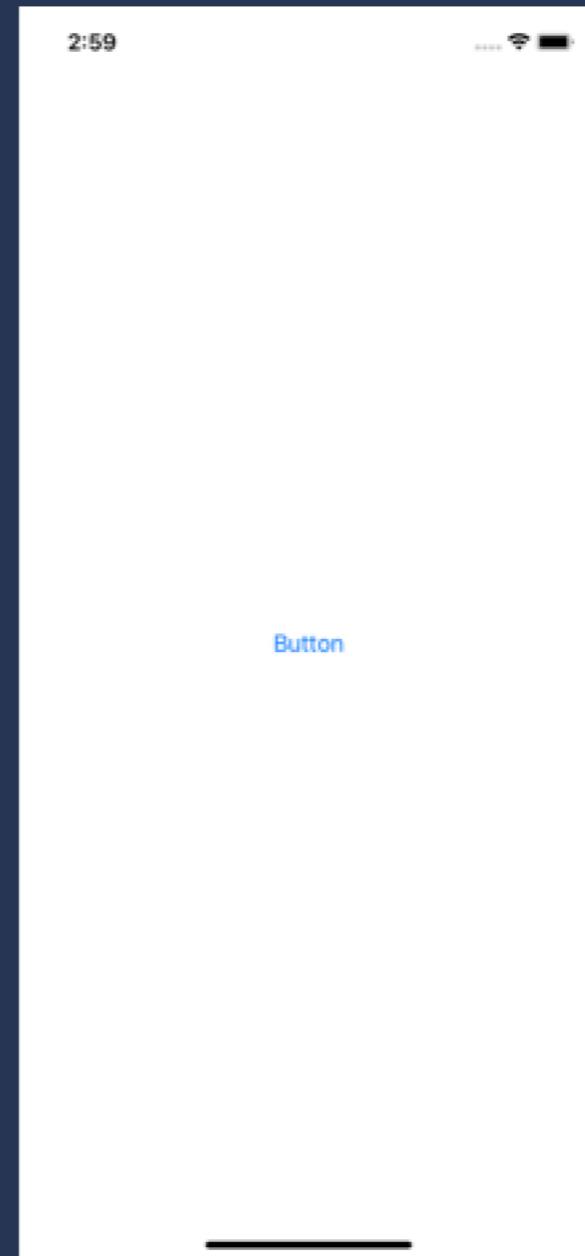


# 如何通過按鈕觸發一些動作

```
import SwiftUI

struct ContentView: View {
    var body: some View {
        Button(action: {
            // 做一些動作
        }, label: {
            Text("Button")
        })
    }
}
```

\* 試試 print?



# 如何獲得 **TextField** 的內容？

```
import SwiftUI

struct ContentView: View {
    @State var yourName = ""

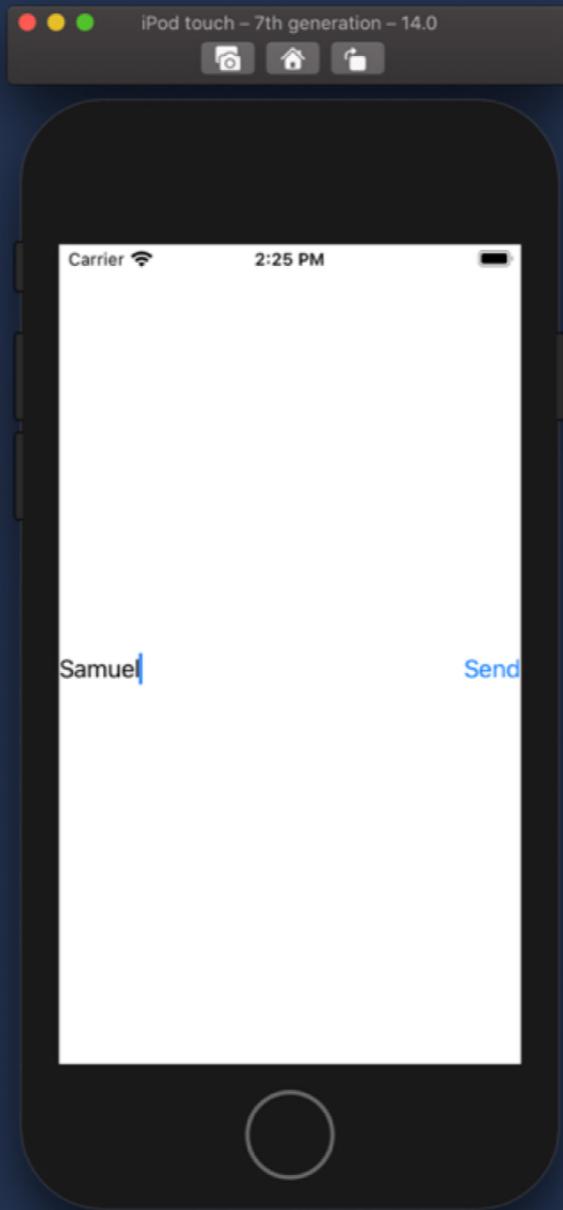
    var body: some View {
        TextField("Your name here", text: $yourName)
    }
}
```

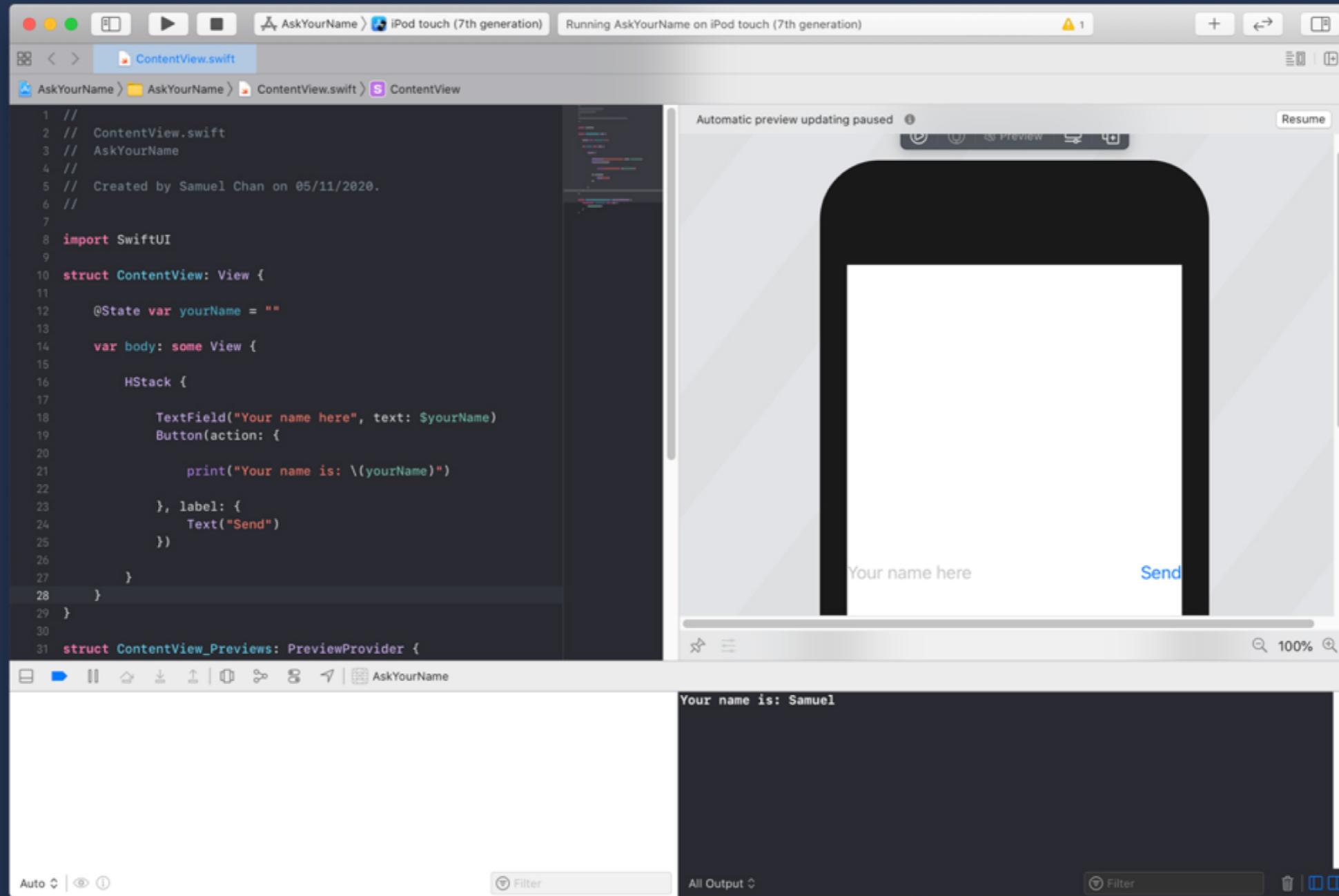
\* 使用 `$yourName` 將 `TextField` 和變量連結起來  
(Two-way Binding)

# 如何使用 yourName?

```
import SwiftUI

struct ContentView: View {
    @State var yourName = ""
    var body: some View {
        HStack {
            TextField("Your name here", text: $yourName)
            Button(action: {
                print("Your name is: \(yourName)") // 將內容打印到 Console
            }, label: {
                Text("Send")
            })
        }
    }
}
```





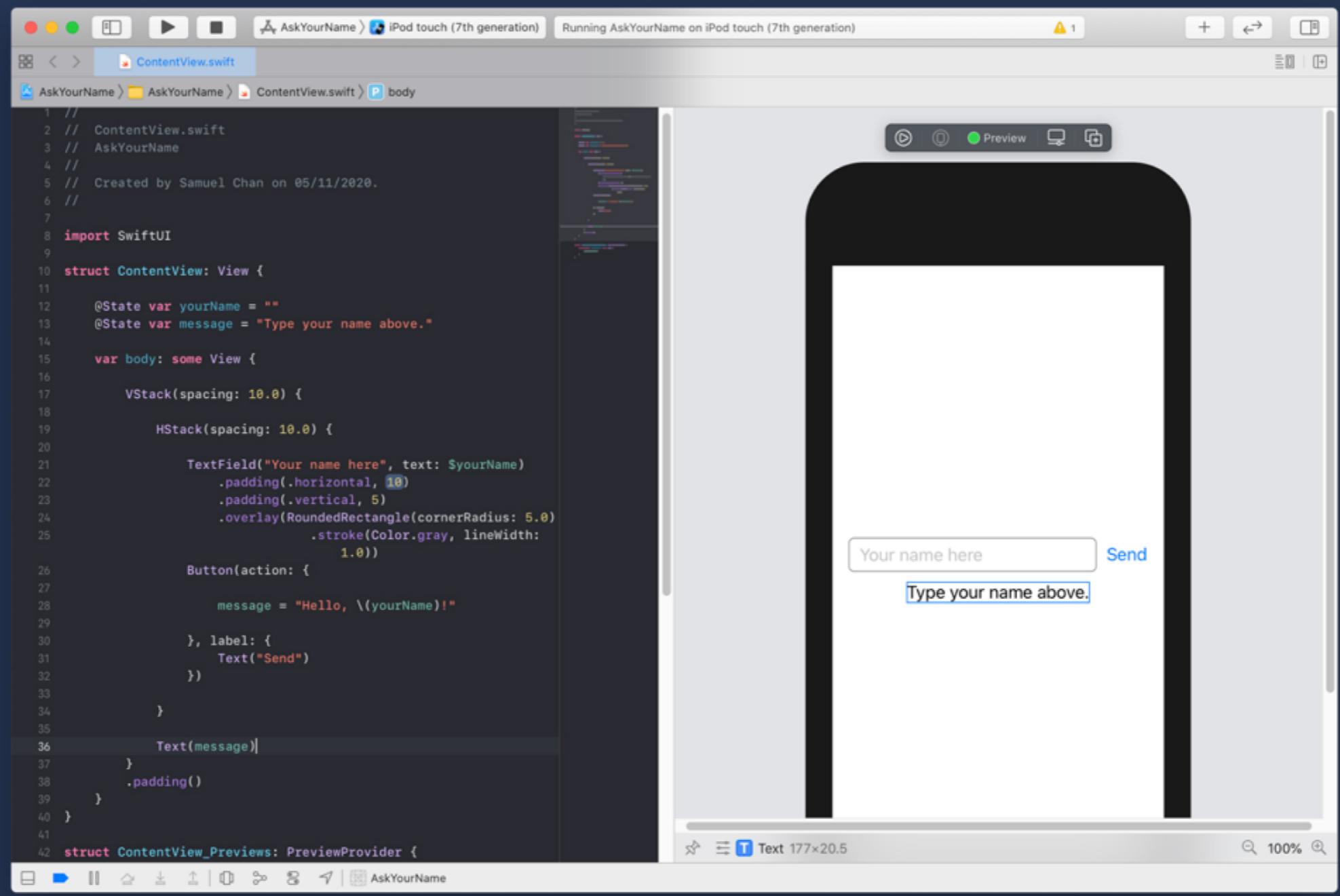
# 如何改變畫面的內容？

```
struct ContentView: View {
    @State var yourName = ""
    @State var message = "Type your name above."

    var body: some View {
        VStack {
            HStack {
                TextField("Your name here", text: $yourName)
                Button(action: {
                    message = "Hello, \(yourName)!"
                }, label: {
                    Text("Send")
                })
            }
            Text(message)
        }
    }
}
```

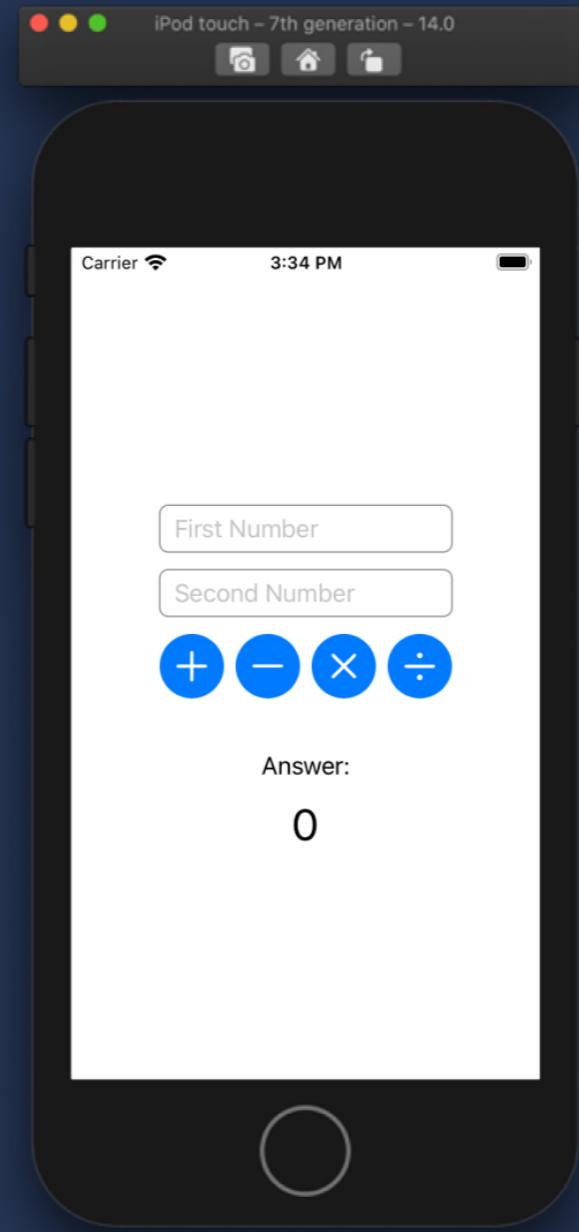
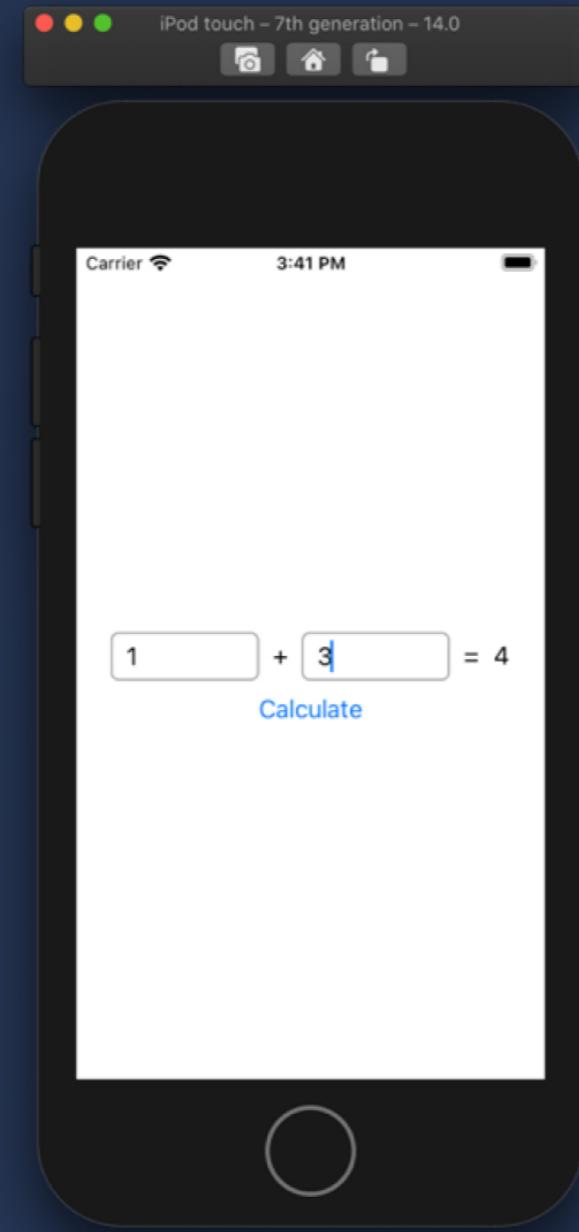
# 完整的 AskYourName

```
struct ContentView: View {  
  
    @State var yourName = ""  
    @State var message = "Type your name above."  
  
    var body: some View {  
  
        VStack(spacing: 10.0) {  
  
            HStack(spacing: 10.0) {  
  
                TextField("Your name here", text: $yourName)  
                    .padding(.horizontal, 10)  
                    .padding(.vertical, 5)  
                    .overlay(RoundedRectangle(cornerRadius: 5.0)  
                            .stroke(Color.gray, lineWidth: 1.0))  
                Button(action: {  
  
                    message = "Hello, \(yourName)!"  
                }, label: {  
                    Text("Send")  
                })  
            }  
  
            Text(message)  
        }.padding()  
    }  
}
```



# 課堂作業 Week 2 (計算分數!)

- 嘗試製作一個簡單的計算機。
- 請仔細閱讀要求，如有任何不符合要求的地方，會失去該作業的分數。



# 作業要求

1. 檔案命名：[你的全名].week2.zip
2. 檔案格式：必須將 Project 壓縮為 ZIP 檔後才上傳
3. 上傳至連結：<https://www.dropbox.com/request/dChXtukxm3uMRHhqi1wg>
4. 到期日：2021 年 7 月 14 日(下週三)晚上 11 時前
5. 評分標準：基本完成要求滿分、空白 Project 或

# 如何將 String 變成 Int

- 因為 String 無法相加 “1” ≠ 1
- 必須頁將 String 變成 Int 才能進行計算

```
if let firstValue = Int(inputOne) {  
    // ...這裏就可以使 firstValue 進行計算  
}
```

# 如何處理兩個數字？

- 一層包一層就可以了

```
if let firstValue = Int(inputOne) {  
    // ...這裏就可以使用 firstValue  
  
    if let secondValue = Int(inputTwo){  
        // ...這裏就可以使用 firstValue 和 secondValue  
    }  
}
```

# 如何將 Int 變回 String

```
result = "\(answer)"
```