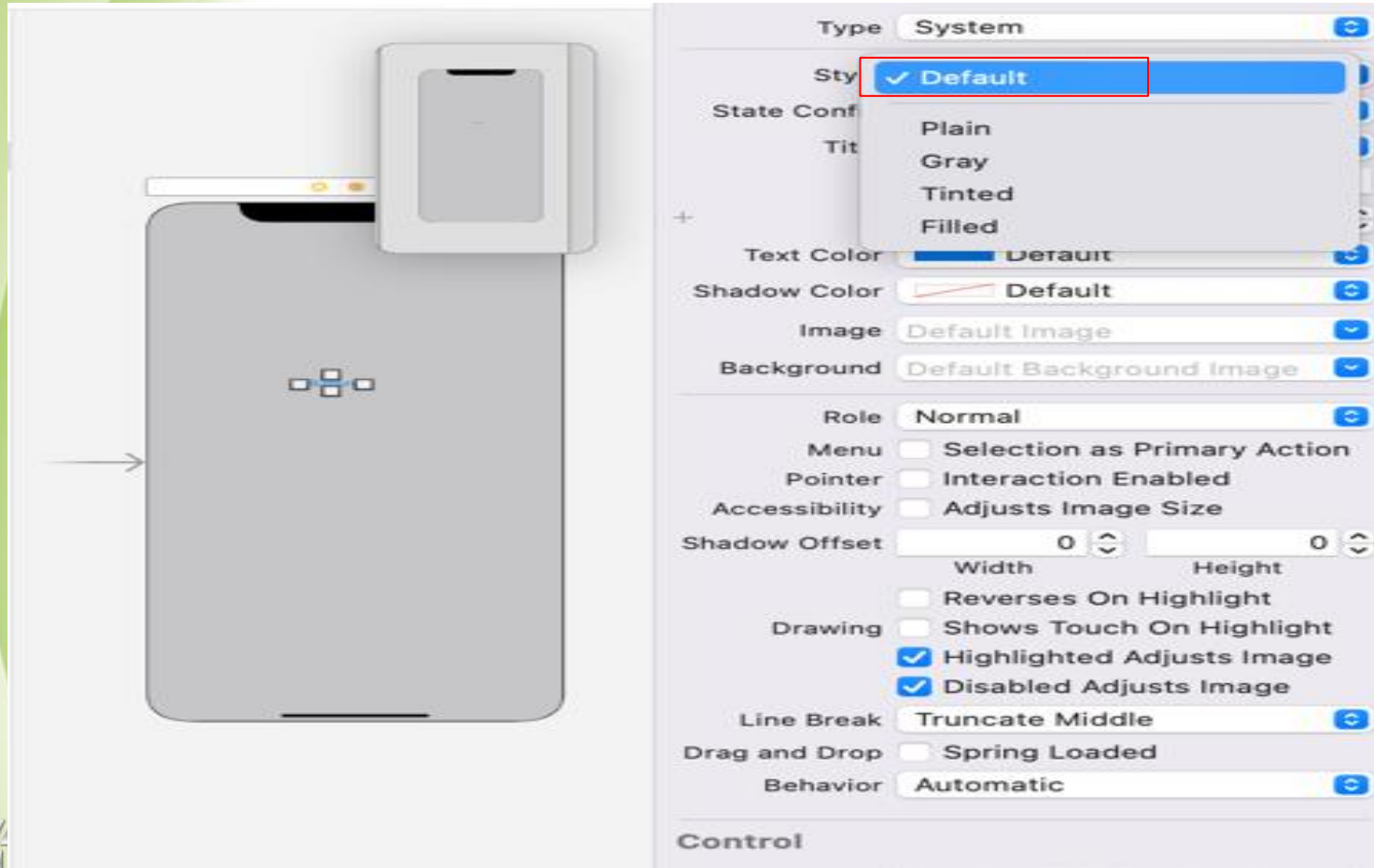


Calculator

Calculator



Calculator

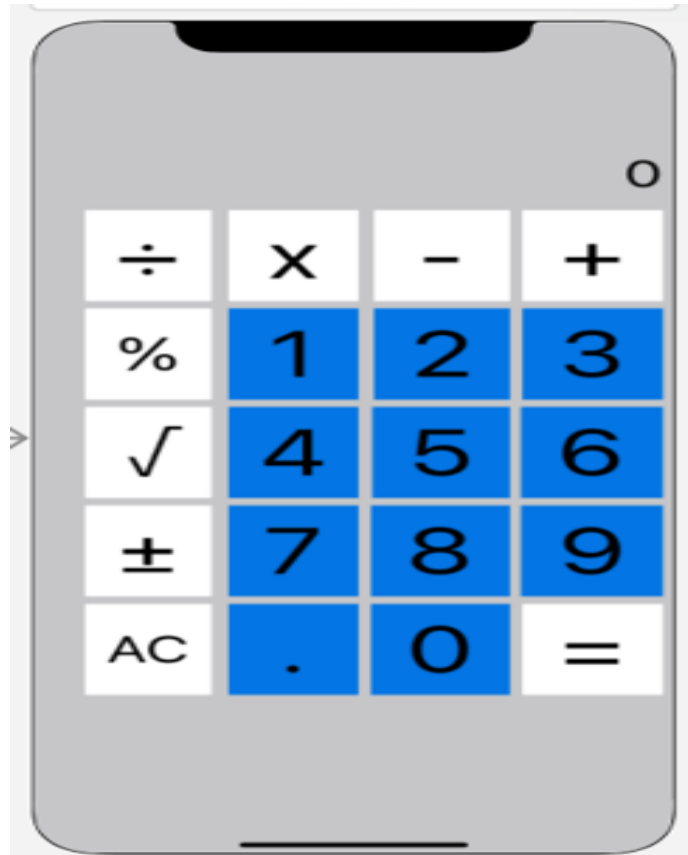


Calculator



Calculator

將計算機
layout拉
好



Calculator

```
1 //
2 // ViewController.swift
3 // Class1-2
4 //
5 // Created by nick on 2022/2/14.
6 //
```

```
7
8 import UIKit
9
```

```
10 class ViewController: UIViewController {
11
12
```

```
13     @IBOutlet weak var display: UILabel!
14     func hDigit(_ sender:
```

```
15     UIButton?) {
16         display.text = display.text + sender.currentTitle!
```

```
17     }
18     func hOperator(_ sender:
```

```
19     UIButton?) {
20         display.text = display.text + sender.currentTitle!
```

```
21     }
22     func hClear(_ sender:
```

```
23     UIButton?) {
24         display.text = ""
25     }
26     func hEquals(_ sender:
```

```
27     UIButton?) {
28         display.text = ""
29     }
30 }
```

```
31
```

```
32
```

```
33
```

```
34
```

```
35
```

```
36
```

```
37
```

```
38
```

```
39
```

```
40
```

```
41
```

```
42
```

```
43
```

```
44
```

將數字與小數點按鈕連接到
viewController裡的
同個Action func

View Controller Scene

View Controller

View

Safe Area

2

3

4

5

6

8

0

9

7

1

.

√

+

-

x

÷

%

±

AC

=

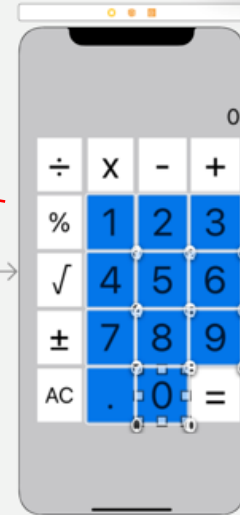
Display

First Responder

Exit

→ Storyboard Entry Poi...

Filter



iPhone 11

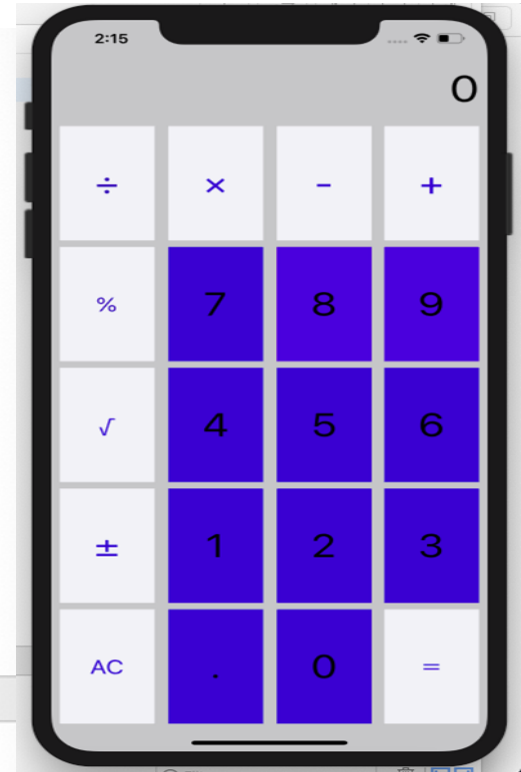
Calculator

```
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBAction func touchDigit(_ sender: UIButton) {
14         let digit = sender.currentTitle!
15         print("\(digit) was touched")
16     }
17     override func viewDidLoad() {
18         super.viewDidLoad()
19         // Do any additional setup after loading the view.
20     }
21
22 }
23
24
25
```

我們可以透過sender取得按鈕的名字

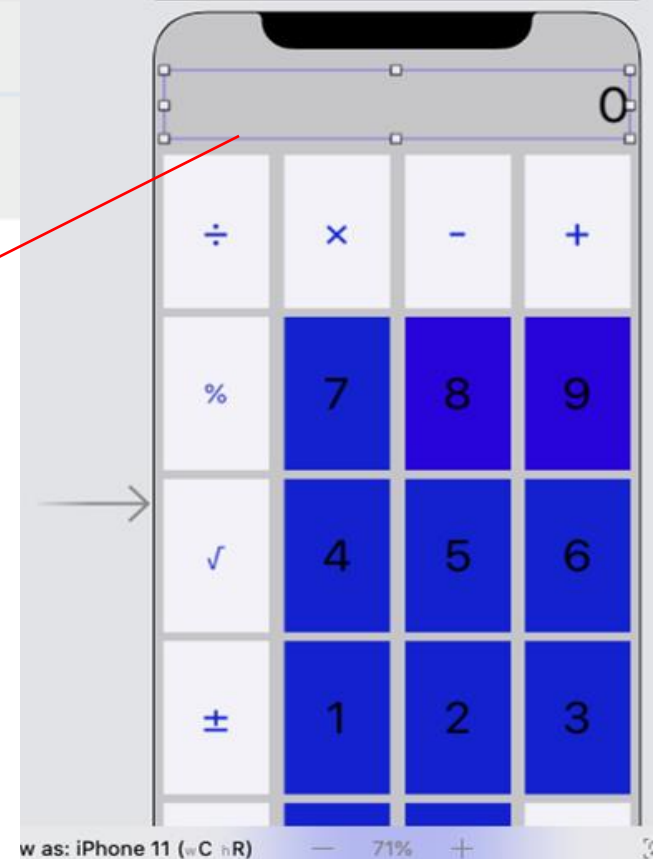
當我們按下
按鈕時就會
顯示我們按
了哪個按鈕

```
5 was touched
8 was touched
8 was touched
2 was touched
```



Calculator

```
1 //
2 // ViewController.swift
3 // Class1-2
4 //
5 // Created by nick on 2022/2/14.
6 //
7
8 import UIKit
9
10 class ViewController: UIViewController {
11
12
13     @IBOutlet weak var display: UILabel!
14     @IBAction func touchDigit(_ sender: UIButton) {
15         let digit = sender.currentTitle!
16         print("\(digit) was touched")
17     }
18     override func viewDidLoad() {
19         super.viewDidLoad()
20         // Do any additional setup after loading the
21         // view.
22     }
23 }
```



Calculator

```
1 //  
2 // ViewController.swift  
3 // Class1-2  
4 //  
5 // Created by nick on 2022/2/14.  
6 //  
7  
8 import UIKit  
9  
10 class ViewController: UIViewController {  
11  
12     @IBOutlet weak var display: UILabel!  
13     @IBAction func touchDigit(_ sender: UIButton) {  
14         let digit = sender.currentTitle!  
15         let textCurrentlyInDisplay = display.text!  
16         display.text = textCurrentlyInDisplay + digit  
17         print("\(digit) was touched")  
18     }  
19 }
```

當每次點擊時先宣告一個常數將當前顯示數字存起來再將新按的數字與當前的串接並顯示

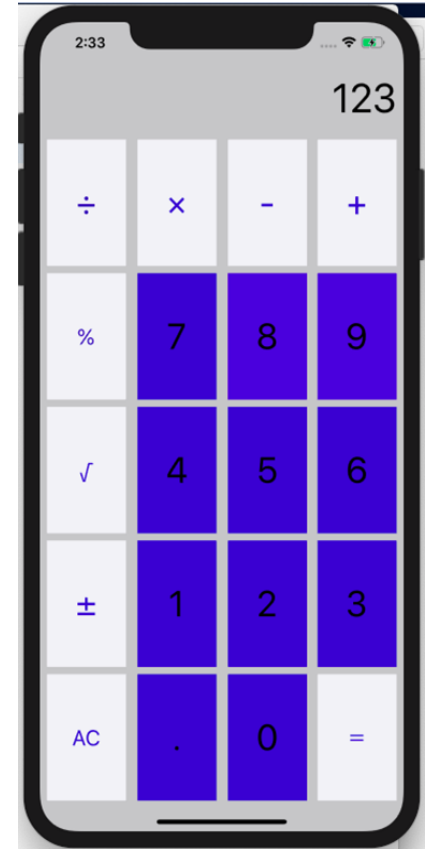


Calculator

```
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var display: UILabel!
14
15     var InTheMiddleOfTyping = false
16
17     @IBAction func touchDigit(_ sender: UIButton) {
18         let digit = sender.currentTitle!
19         if InTheMiddleOfTyping{
20             let textCurrentlyInDisplay = display.text!
21             display.text = textCurrentlyInDisplay + digit
22         } else{
23             display.text = digit
24             InTheMiddleOfTyping = true
25         }
26     }
27
28     override func viewDidLoad() {
29         super.viewDidLoad()
30         // Do any additional setup after loading the view.
31
32
33     }
34 }
```

用判斷式來判斷是否為輸入新數字是的話就蓋掉原本的數字，並把布林值改為true

宣告一個布林值來記錄是否有輸入第一個新數字



Calculator

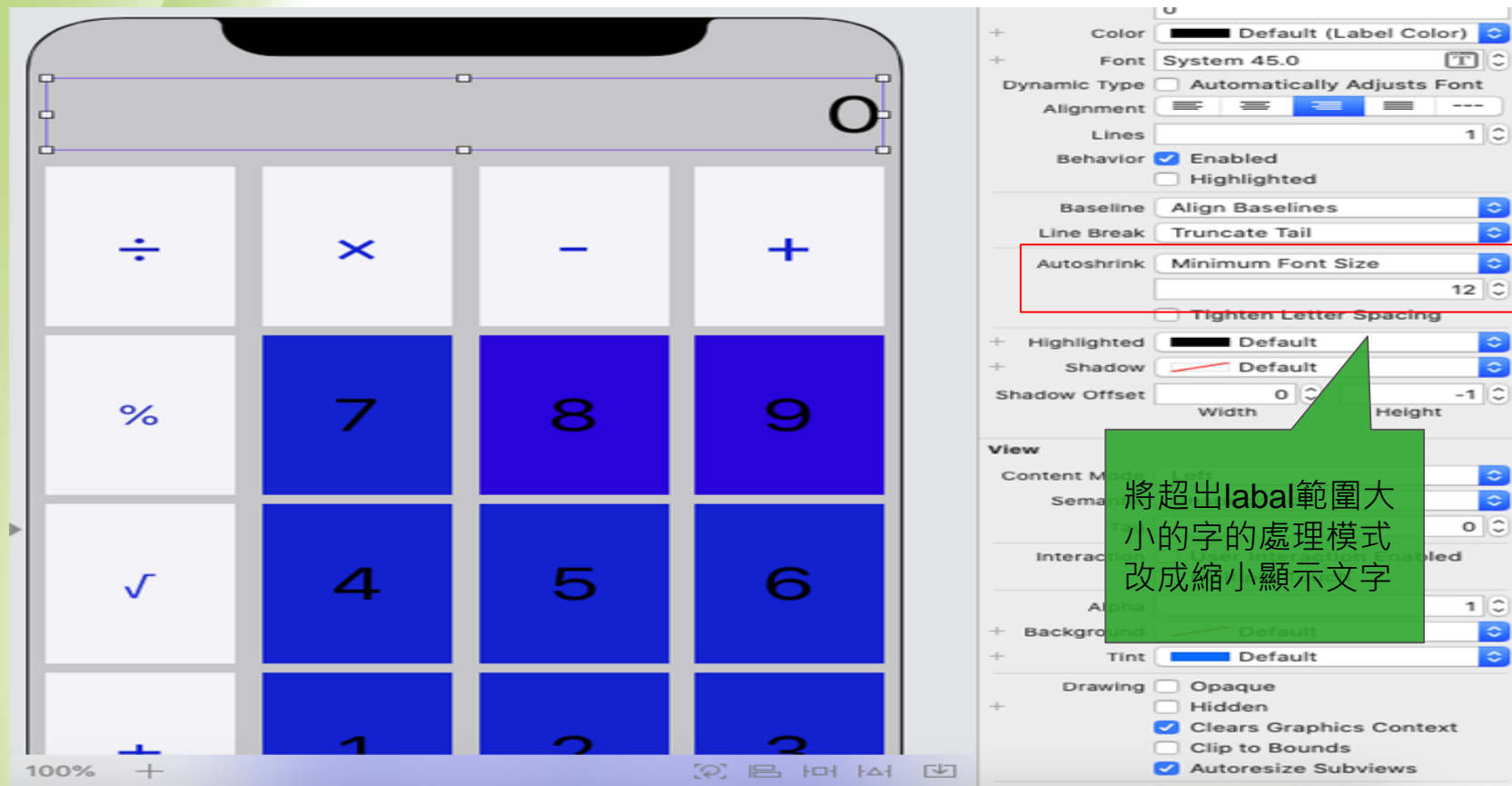
```
12
13  @IBOutlet weak var display: UILabel!
14
15  var InTheMiddleOfTyping = false
16  var Isfloating = false
17
18  Action func touchDigit(_ sender: UIButton) {
19      let ditgit = sender.currentTitle!
20
21      if ditgit == "." {
22          InTheMiddleOfTyping = true
23      }
24
25      if !(Isfloating && ditgit == ".") {
26          if InTheMiddleOfTyping{
27              let textCurrentlyInDisplay = display.text!
28              display.text = textCurrentlyInDisplay + ditgit
29          } else {
30              display.text = ditgit
31              InTheMiddleOfTyping = true
32          }
33      }
34      if ditgit == "." {
35          Isfloating = true
36      }
37  }
38 }
```

宣告用來記錄是否有按過小數點的布林值

當第一次按小數點時直接接在數字後而不是取代

判斷是否按過2次小數點

Calculator



Calculator

再來將+ - * / 等剩餘沒有連接的按鈕連結到viewController

先宣告兩個變數一個來記錄第一個運算元，一個來記錄目前按了哪個運算子

當我們按下運算子時便是要輸入新數字因此將布林值改為false

將第一個運算元存起來並記錄運算子符號

先將目前顯示的數字從Optional解開再將它轉為Double，但實際上它會變成Double?因此要再次解開

```
27
28     var operand1 = 0.0
29     var symbolOfOperation = ""
30
31     @IBAction func performOperation(_ sender: UIButton) {
32         let Operation = sender.currentTitle!
33         switch Operation {
34             case "AC":
35             display.text = "0"
36             InTheMiddleOfTyping = false
37             case "√":
38             let operand = Double(display.text!)!
39             display.text = String(sqrt(operand))
40             InTheMiddleOfTyping = false
41             case "+":
42             operand1 = Double(display.text!)!
43             InTheMiddleOfTyping = false
44             symbolOfOperation = "+"
45             case "-":
46             operand1 = Double(display.text!)!
47             InTheMiddleOfTyping = false
48             symbolOfOperation = "-"
49             case "x":
50             operand1 = Double(display.text!)!
51             InTheMiddleOfTyping = false
52             symbolOfOperation = "x"
53             case "÷":
54             operand1 = Double(display.text!)!
55             InTheMiddleOfTyping = false
56             symbolOfOperation = "÷"
```

Calculator

```
case "%":
    operand1 = Double(display.text!!)
    InTheMiddleOfTyping = false
    symbolOfOperation = "%"
case "±":
    let operand = Double(display.text!!)
    display.text = String(-operand)
case "=":
    if(symbolOfOperation != ""){
        let operand2 = Double(display.text!!)
        switch symbolOfOperation {
            case "+":
                display.text = String(operand1 + operand2)
            case "-":
                display.text = String(operand1 - operand2)
            case "x":
                display.text = String(operand1 * operand2)
            case "÷":
                display.text = String(operand1 / operand2)
            case "%":
                display.text = String(Int(operand1) % Int(operand2))
            default:
                break
        }
        InTheMiddleOfTyping = false
        symbolOfOperation = ""
    }
default:
    break
}
```

更改正負號
無需要輸入
新數字因此
不用改布林
值

獲取第二
個運算元

將記錄符號設為空字串

餘除不能為
Double型
態故轉為
Int