# iOS 應用程式開發

## Week 3

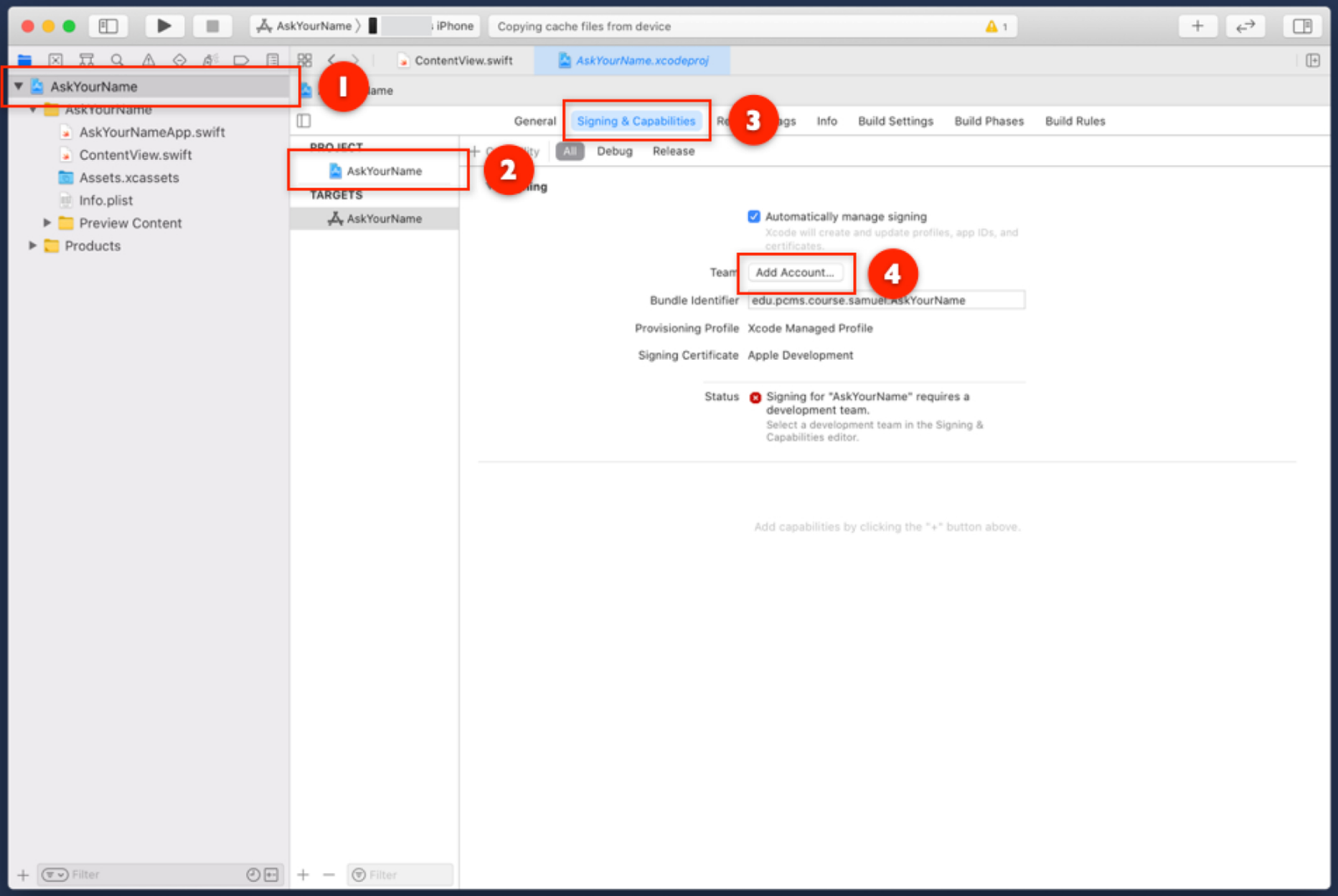
# 本週內容

- 於機器上測試編寫好的程式
  - 設定 XCode Project
  - 在機器上運行及測試程式

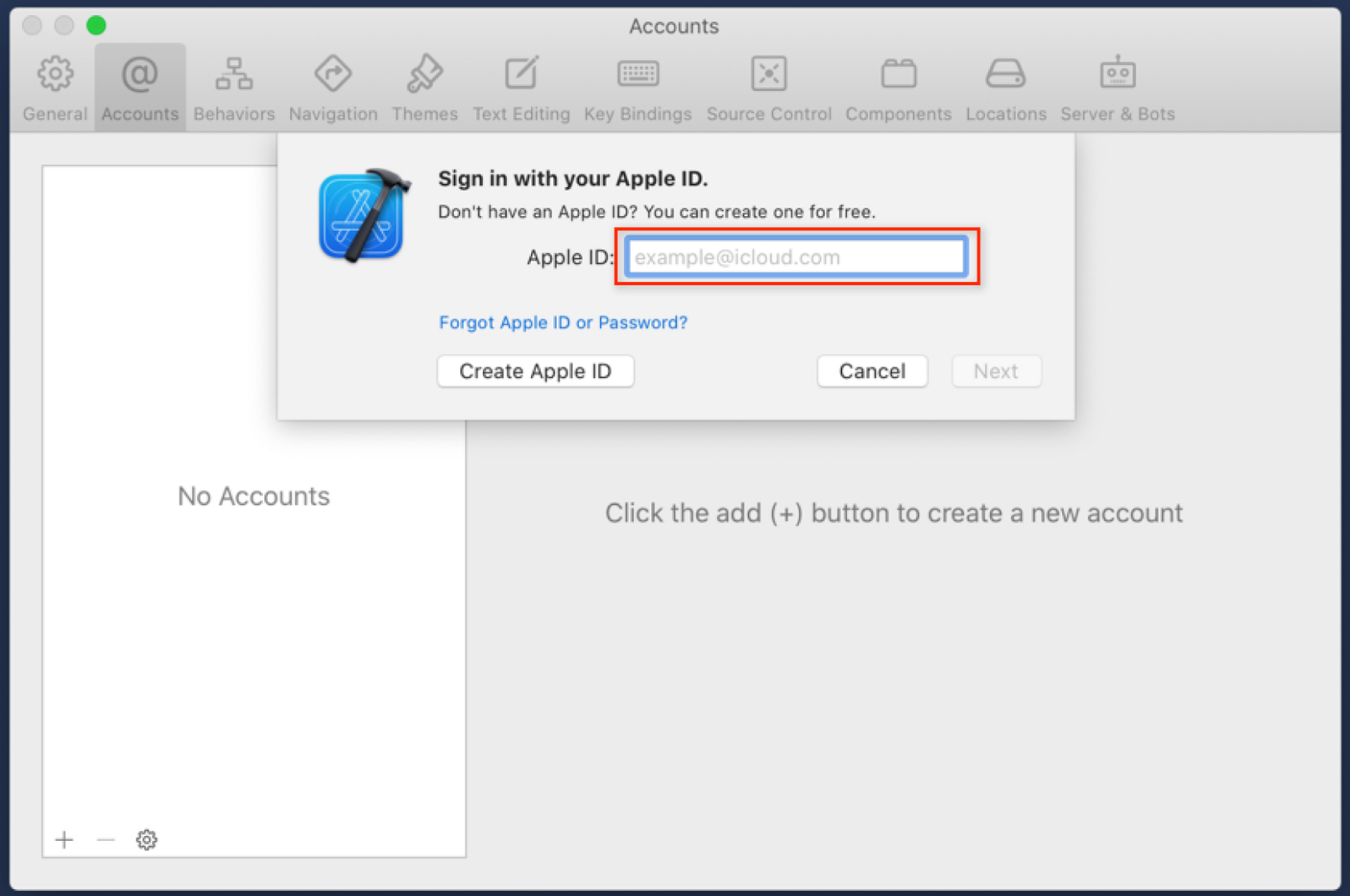
# 本週內容 (續)

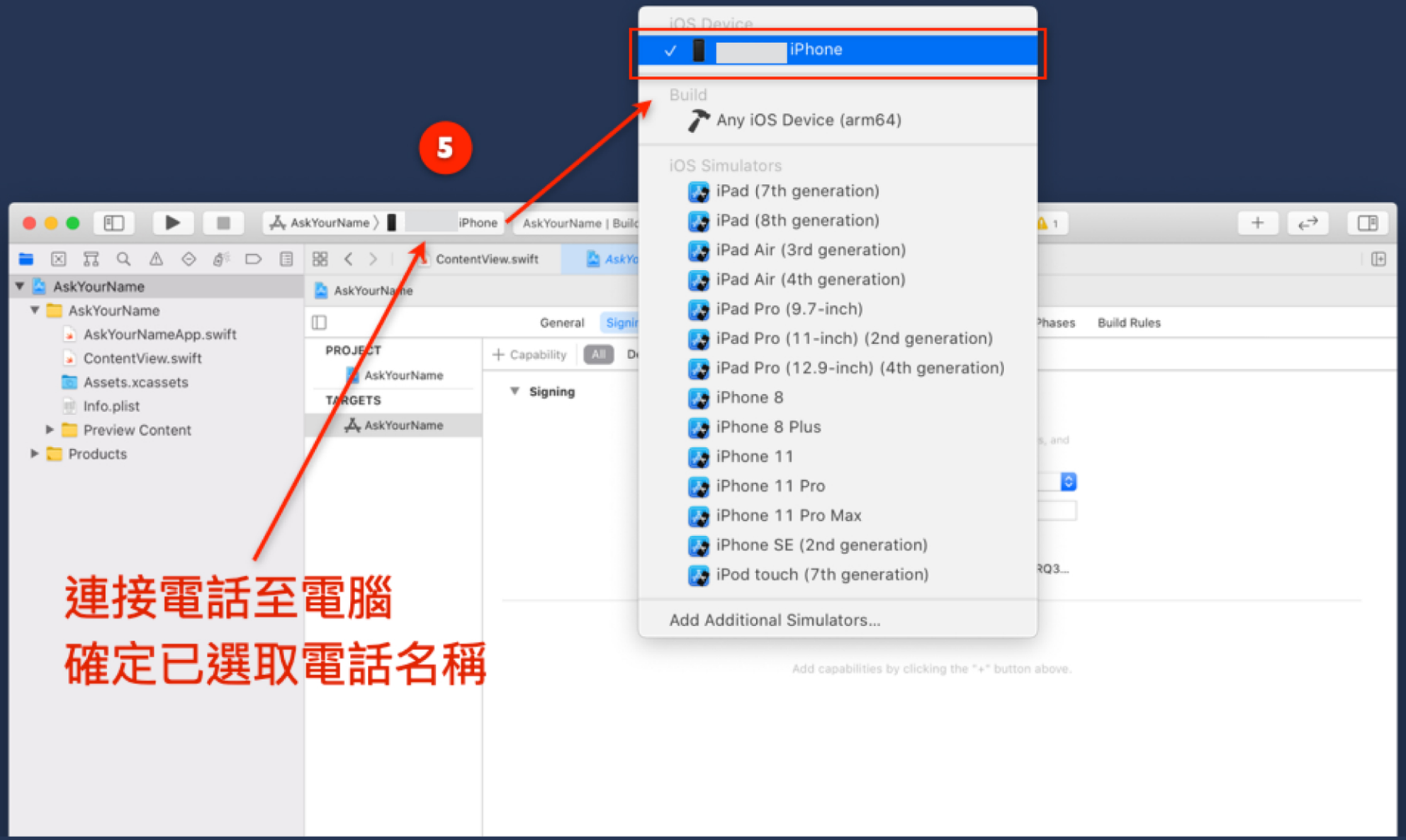
- 講解 Simple Calculator
- 定義 Function
- 甚麼是 Optional
- 甚麼是 Binding
- Classwork: Member Form

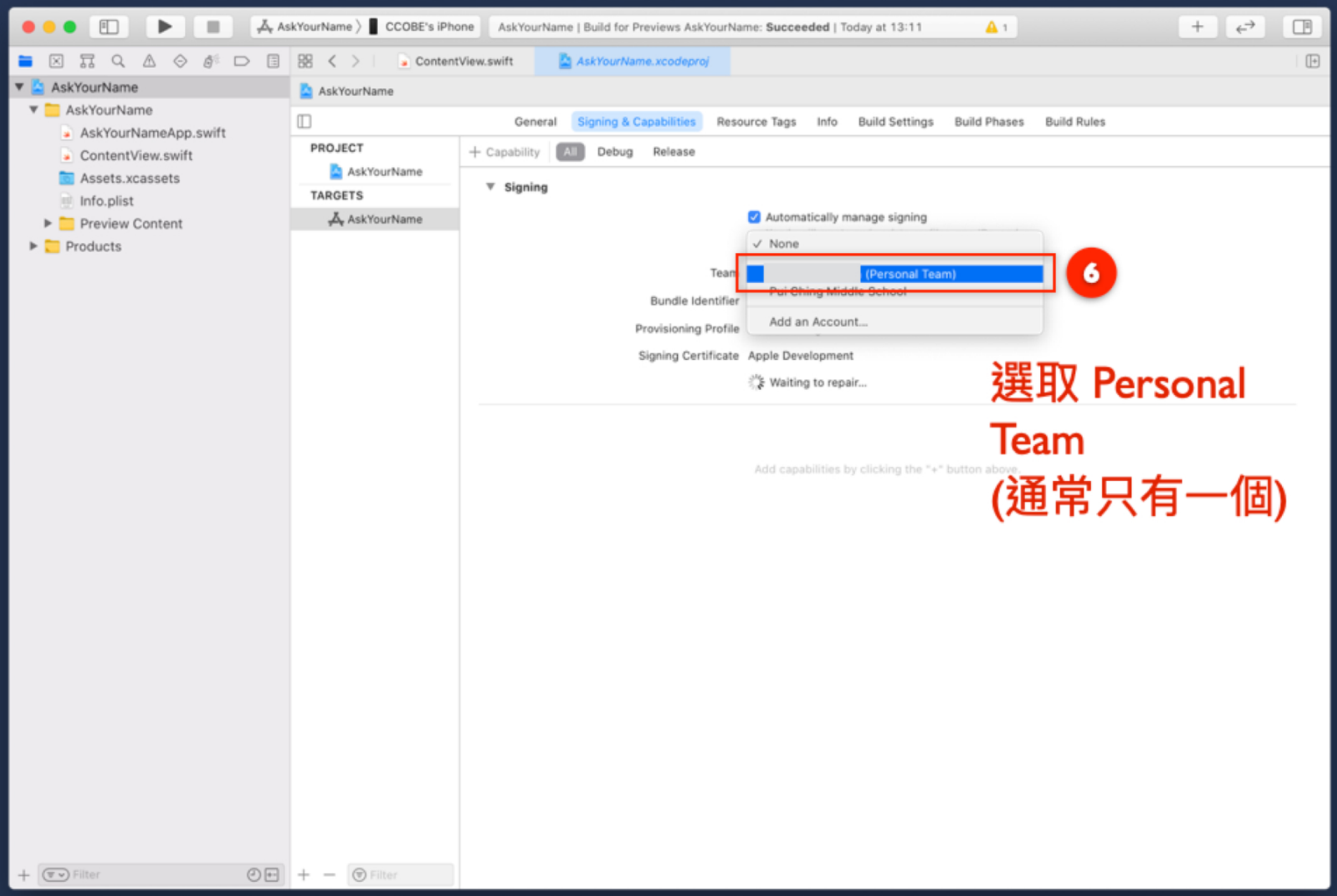
# 於機器上測試
# 編寫好的程式

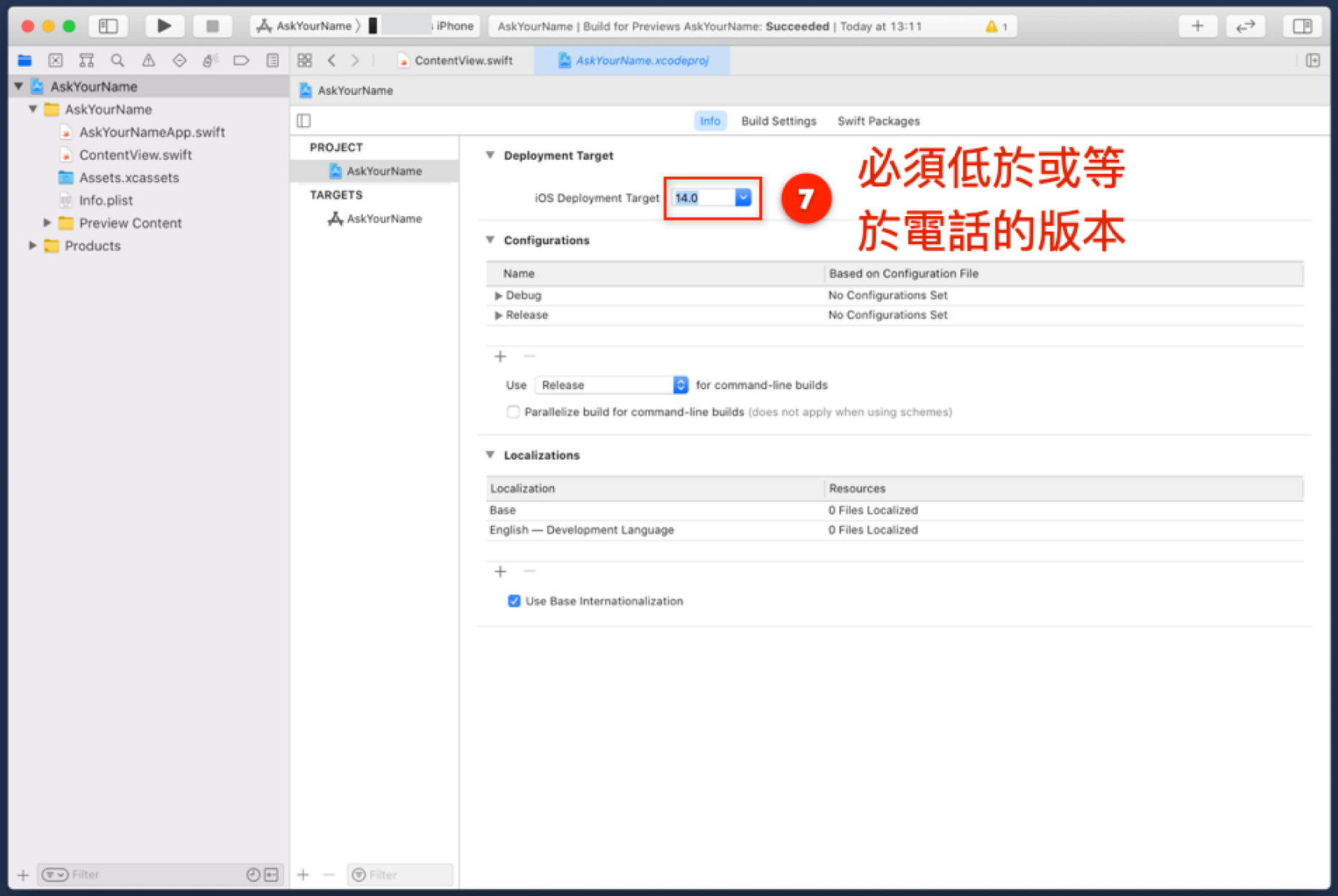

AskYourName
iPhone
Copying cache files from device
ContentView.swift
AskYourName.xcodeproj
AskYourName
AskYourNameApp.swift
ContentView.swift
Assets.xcassets
Info.plist
Preview Content
Products
1
General
Signing & Capabilities
Info
Build Settings
Build Phases
Build Rules
3
PROJECT
AskYourName
2
TARGETS
AskYourName
All
Debug
Release
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team
Add Account...
4
Bundle Identifier
edu.pcms.course.samuel.AskYourName
Provisioning Profile
Xcode Managed Profile
Signing Certificate
Apple Development
Status
Signing for "AskYourName" requires a development team.
Select a development team in the Signing & Capabilities editor.
Add capabilities by clicking the "+" button above.
Filter
Filter

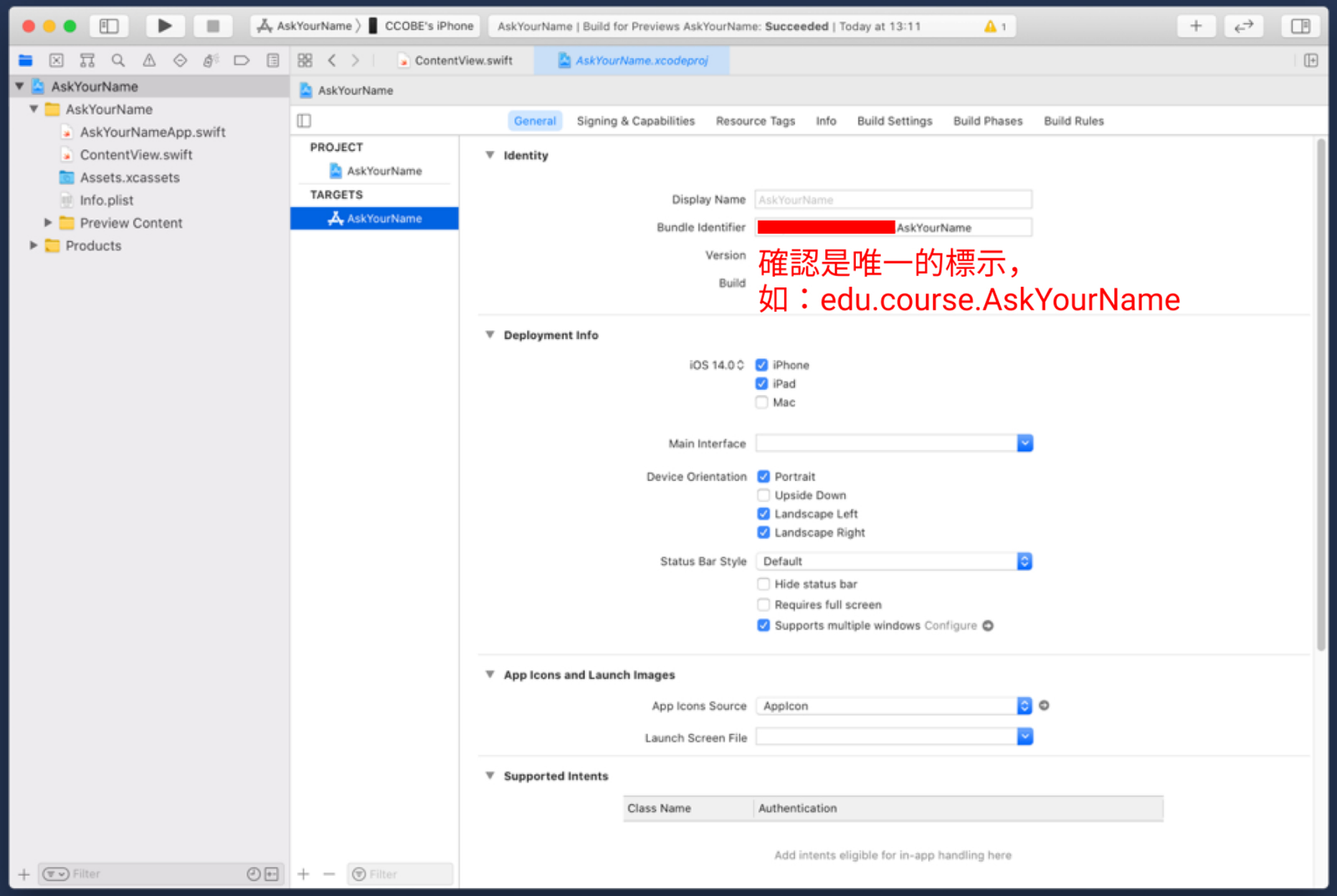
# 輸入 Apple ID (平常使用 AppStore 的戶口都可以)



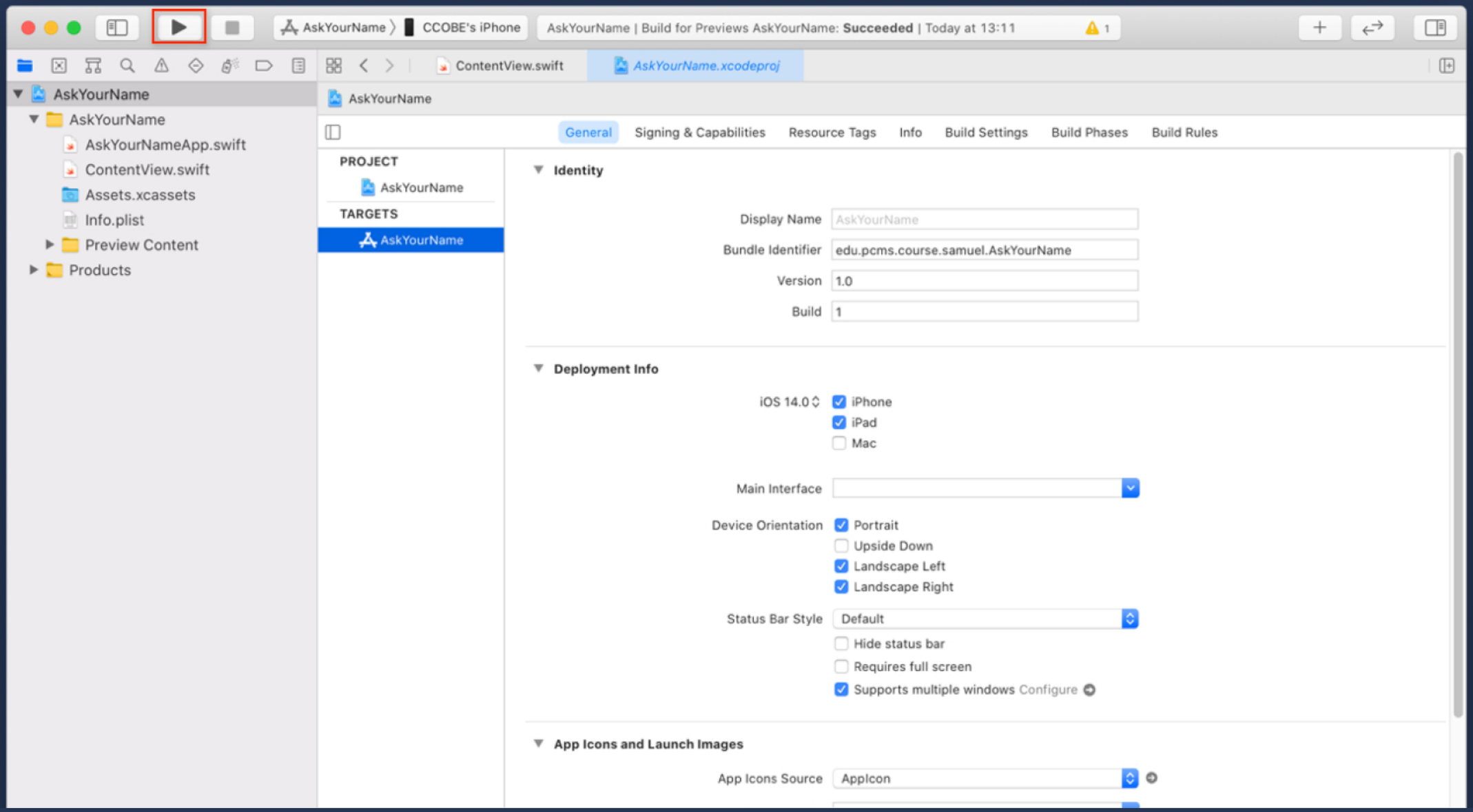

5
AskYourName
iPhone
AskYourName | Build
ContentView.swift
AskYourName
AskYourName
AskYourNameApp.swift
ContentView.swift
Assets.xcassets
Info.plist
Preview Content
Products
General
Signing
PROJECT
AskYourName
TARGETS
AskYourName
Capability
All
Signing
Phases
Build Rules
iOS Device
iPhone
Build
Any iOS Device (arm64)
iOS Simulators
iPad (7th generation)
iPad (8th generation)
iPad Air (3rd generation)
iPad Air (4th generation)
iPad Pro (9.7-inch)
iPad Pro (11-inch) (2nd generation)
iPad Pro (12.9-inch) (4th generation)
iPhone 8
iPhone 8 Plus
iPhone 11
iPhone 11 Pro
iPhone 11 Pro Max
iPhone SE (2nd generation)
iPod touch (7th generation)
Add Additional Simulators...
Add capabilities by clicking the "+" button above.
連接電話至電腦
確定已選取電話名稱


AskYourName 〉 CCOBE's iPhone
AskYourName | Build for Previews AskYourName: Succeeded | Today at 13:11
ContentView.swift
AskYourName.xcodeproj
AskYourName
AskYourNameApp.swift
ContentView.swift
Assets.xcassets
Info.plist
Preview Content
Products
General
Signing & Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
TARGETS
Capability
All
Debug
Release
Signing
Automatically manage signing
None
Team
(Personal Team)
Bundle Identifier
Add an Account...
Provisioning Profile
Signing Certificate
Apple Development
Waiting to repair...
6
選取 Personal Team
(通常只有一個)
Add capabilities by clicking the "+" button above.
Filter
Filter

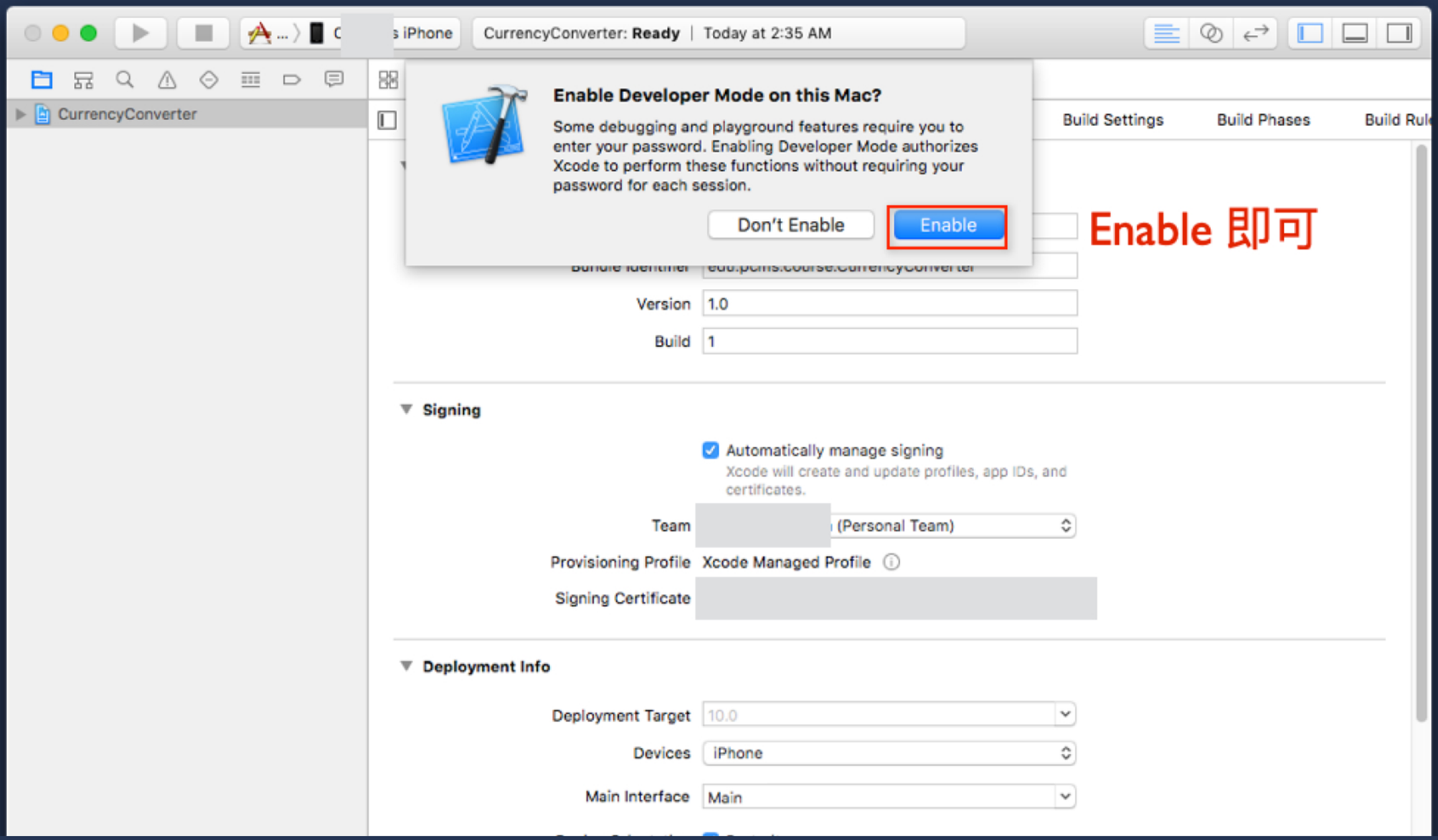

AskYourName
iPhone
AskYourName | Build for Previews AskYourName: Succeeded | Today at 13:11
ContentView.swift
AskYourName.xcodeproj
AskYourName
AskYourName
AskYourNameApp.swift
ContentView.swift
Assets.xcassets
Info.plist
Preview Content
Products
Info
Build Settings
Swift Packages
PROJECT
AskYourName
TARGETS
AskYourName
Deployment Target
iOS Deployment Target
14.0
7
必須低於或等
於電話的版本
Configurations
Name
Based on Configuration File
Debug
No Configurations Set
Release
No Configurations Set
Use
Release
for command-line builds
Parallelize build for command-line builds (does not apply when using schemes)
Localizations
Localization
Resources
Base
0 Files Localized
English — Development Language
0 Files Localized
Use Base Internationalization
Filter
Filter


AskYourName 〉 CCOBE's iPhone
AskYourName | Build for Previews AskYourName: Succeeded | Today at 13:11
ContentView.swift
AskYourName.xcodeproj
AskYourName
AskYourName
AskYourNameApp.swift
ContentView.swift
Assets.xcassets
Info.plist
Preview Content
Products
General
Signing & Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
AskYourName
TARGETS
AskYourName
Identity
Display Name
AskYourName
Bundle Identifier
AskYourName
Version
Build
確認是唯一的標示，
如：edu.course.AskYourName
Deployment Info
iOS 14.0
iPhone
iPad
Mac
Main Interface
Device Orientation
Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar Style
Default
Hide status bar
Requires full screen
Supports multiple windows Configure
App Icons and Launch Images
App Icons Source
AppIcon
Launch Screen File
Supported Intents
Class Name
Authentication
Add intents eligible for in-app handling here
Filter
Filter

# 執行程式


AskYourName 〉 CCOBE's iPhone
AskYourName | Build for Previews AskYourName: Succeeded | Today at 13:11
ContentView.swift
AskYourName.xcodeproj
AskYourName
AskYourName
AskYourNameApp.swift
ContentView.swift
Assets.xcassets
Info.plist
Preview Content
Products
AskYourName
General
Signing & Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
AskYourName
TARGETS
AskYourName
Identity
Display Name
AskYourName
Bundle Identifier
edu.pcms.course.samuel.AskYourName
Version
1.0
Build
1
Deployment Info
iOS 14.0
iPhone
iPad
Mac
Main Interface
Device Orientation
Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar Style
Default
Hide status bar
Requires full screen
Supports multiple windows
Configure
App Icons and Launch Images
App Icons Source
AppIcon

# 可能的問題1


CurrencyConverter: Ready | Today at 2:35 AM
CurrencyConverter
Enable Developer Mode on this Mac?
Some debugging and playground features require you to enter your password. Enabling Developer Mode authorizes Xcode to perform these functions without requiring your password for each session.
Don't Enable
Enable
Enable 即可
Build Settings
Build Phases
Version
1.0
Build
1
Signing
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team
(Personal Team)
Provisioning Profile
Xcode Managed Profile
Signing Certificate
Deployment Info
Deployment Target
10.0
Devices
iPhone
Main Interface
Main

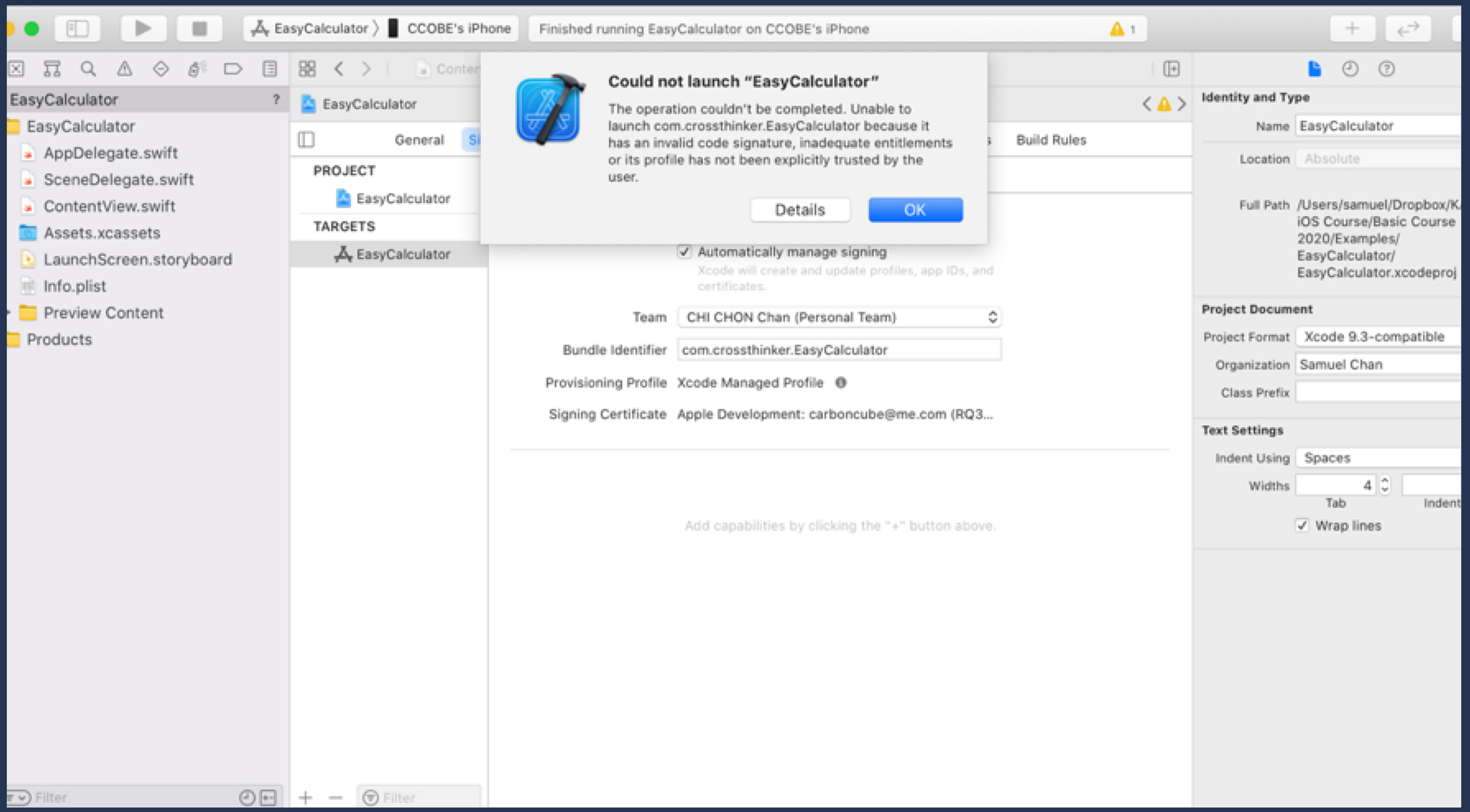
## 可能的問題2

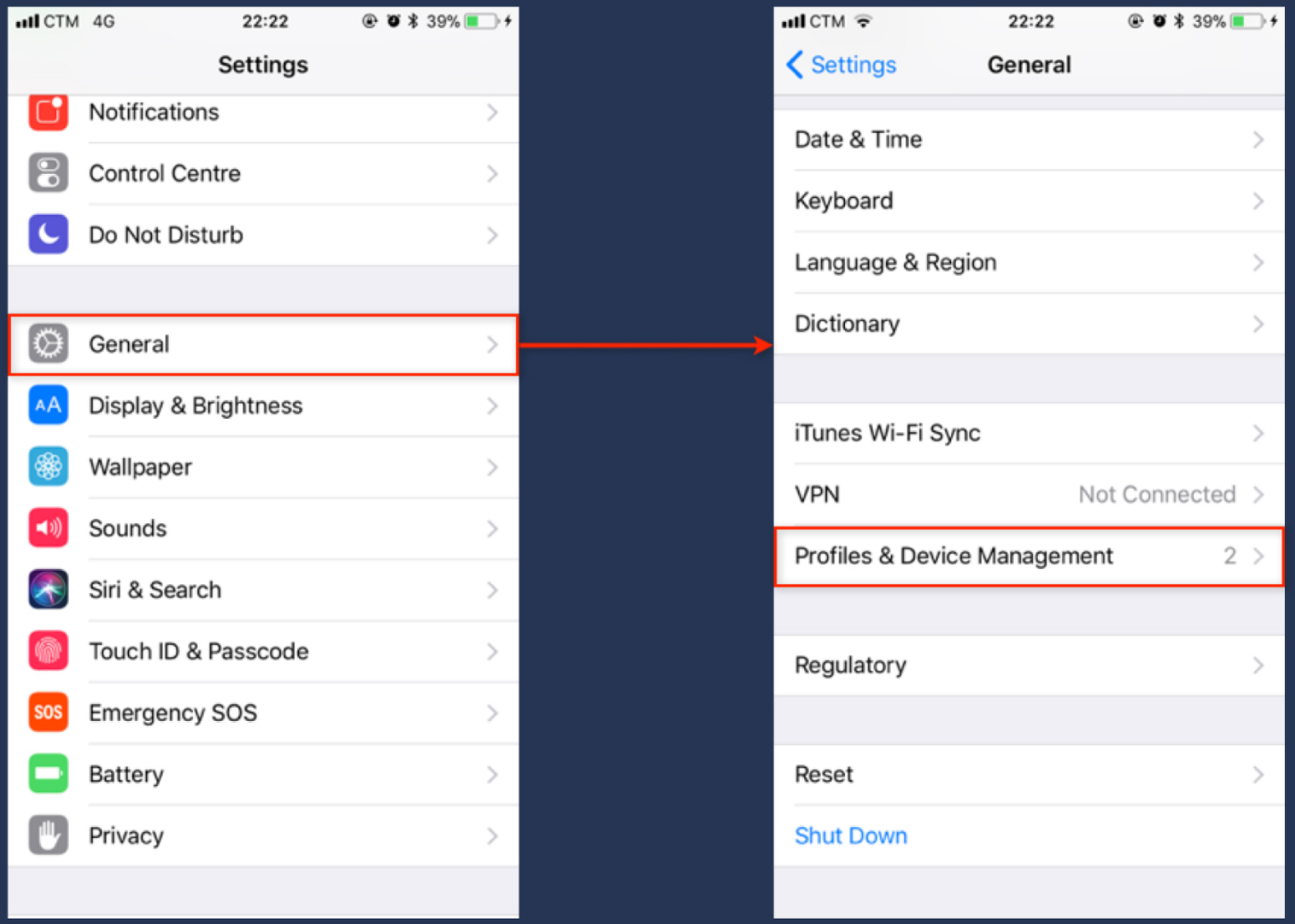
電話版本太新，只要更新 XCode 即可

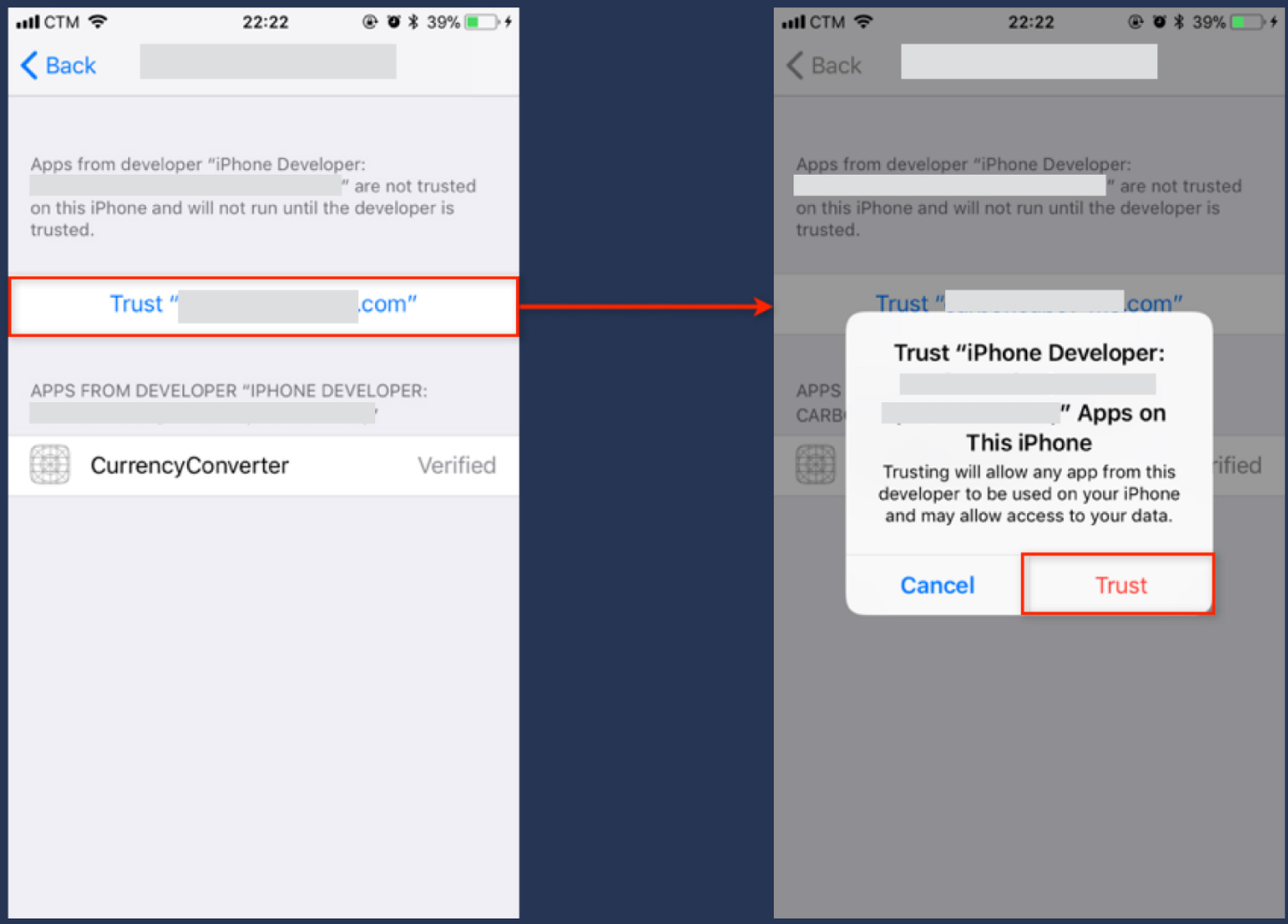
## 可能的問題3

電話需要解鎖畫面或插入電話時需要按"信任"

# 可能的問題 4


EasyCalculator 〉 CCOBE's iPhone
Finished running EasyCalculator on CCOBE's iPhone
EasyCalculator
EasyCalculator
AppDelegate.swift
SceneDelegate.swift
ContentView.swift
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Preview Content
Products
Could not launch "EasyCalculator"
The operation couldn't be completed. Unable to launch com.crossthinker.EasyCalculator because it has an invalid code signature, inadequate entitlements or its profile has not been explicitly trusted by the user.
Details
OK
General
PROJECT
EasyCalculator
TARGETS
EasyCalculator
Build Rules
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team
CHI CHON Chan (Personal Team)
Bundle Identifier
com.crossthinker.EasyCalculator
Provisioning Profile
Xcode Managed Profile
Signing Certificate
Apple Development: carboncube@me.com (RQ3...
Add capabilities by clicking the "+" button above.
Identity and Type
Name
EasyCalculator
Location
Absolute
Full Path
/Users/samuel/Dropbox/K
iOS Course/Basic Course
2020/Examples/
EasyCalculator/
EasyCalculator.xcodeproj
Project Document
Project Format
Xcode 9.3-compatible
Organization
Samuel Chan
Class Prefix
Text Settings
Indent Using
Spaces
Widths
4
Tab
Indent
Wrap lines
Filter
Filter


CTM 4G
22:22
39%
Settings
Notifications
Control Centre
Do Not Disturb
General
Display & Brightness
Wallpaper
Sounds
Siri & Search
Touch ID & Passcode
Emergency SOS
Battery
Privacy
CTM
22:22
39%
Settings
General
Date & Time
Keyboard
Language & Region
Dictionary
iTunes Wi-Fi Sync
VPN
Not Connected
Profiles & Device Management
2
Regulatory
Reset
Shut Down


CTM
22:22
39%
Back
Apps from developer "iPhone Developer:
" are not trusted on this iPhone and will not run until the developer is trusted.
Trust "
.com"
APPS FROM DEVELOPER "IPHONE DEVELOPER:
CurrencyConverter
Verified
Trust "iPhone Developer:
" Apps on This iPhone
Trusting will allow any app from this developer to be used on your iPhone and may allow access to your data.
Cancel
Trust

# 上一週 Classwork: Simple Calculator

# @State

```
struct ContentView: View {
...
    @State var firstNum: String = ""
    @State var secondNum: String = ""
    @State var result: String = "0"
...
}
```

# 最基本做法

```
Button(action: {
    if let firstValue = Int(firstNum) {
        if let secondValue = Int(secondNum){
            let resultValue = firstValue + secondValue
            result = "\(resultValue)"
        }
    }
}, label: {
    Image(systemName: "plus")
})
    .font(.title)
    .frame(width: 44.0, height: 44.0)
    .background(Color.blue)
    .cornerRadius(22.0)
    .foregroundColor(.white)
```

# 抽取這一段：使用 ButtonStyle (LabelStyle ...etc)

```
Button(action: {
    ...
}, label: {
    Image(systemName: "plus")
})
    .font(.title)
    .frame(width: 44.0, height: 44.0)
    .background(Color.blue)
    .cornerRadius(22.0)
    .foregroundColor(.white)
```

# ButtonStyle (LabelStyle ...etc)

放在最下面

```
struct ActionButtonStyle : ButtonStyle {
    func makeBody(configuration: Configuration) -> some View {
        configuration.label
            .font(.title)
            .frame(width: 44.0, height: 44.0)
            .background(configuration.isPressed ? Color.red : Color.blue)
            .cornerRadius(22.0)
            .foregroundColor(.white)
    }
}
```

# Function

## 定義

```
func functionName(
    parameterA: String,
    parameterB: Int,
    parameterC: String
) -> Void {
    // ...執行內容
}
```

## 使用

```
functionName(
    parameterA: "1234",
    parameterB: 123,
    parameterC: "abcd"
)
```

# Function (第二種方式)

## 定義 (留意)

```
func functionName(
    _ parameterA: String,
    _ parameterB: Int,
    _ parameterC: String
) -> Void {
    // ...執行內容
}
```

## 使用 (留意沒有 label)

```
functionName("1234", 123, "abcd")
```

# 加減乗除

```
enum Operation {
    case Add
    case Subtract
    case Multiply
    case Divide
}

func commit(_ firstNum: String, _ secondNum: String, _ operation: Operation) -> Void {
    if let firstValue = Int(firstNum) {
        if let secondValue = Int(secondNum) {
            var resultValue : Int
            switch operation {
                case .Add:
                    resultValue = firstValue + secondValue
                case .Subtract:
                    resultValue = firstValue - secondValue
                case .Multiply:
                    resultValue = firstValue * secondValue
                case .Divide:
                    resultValue = firstValue / secondValue
            }
            result = "\(resultValue)"
        }
    }
}
```

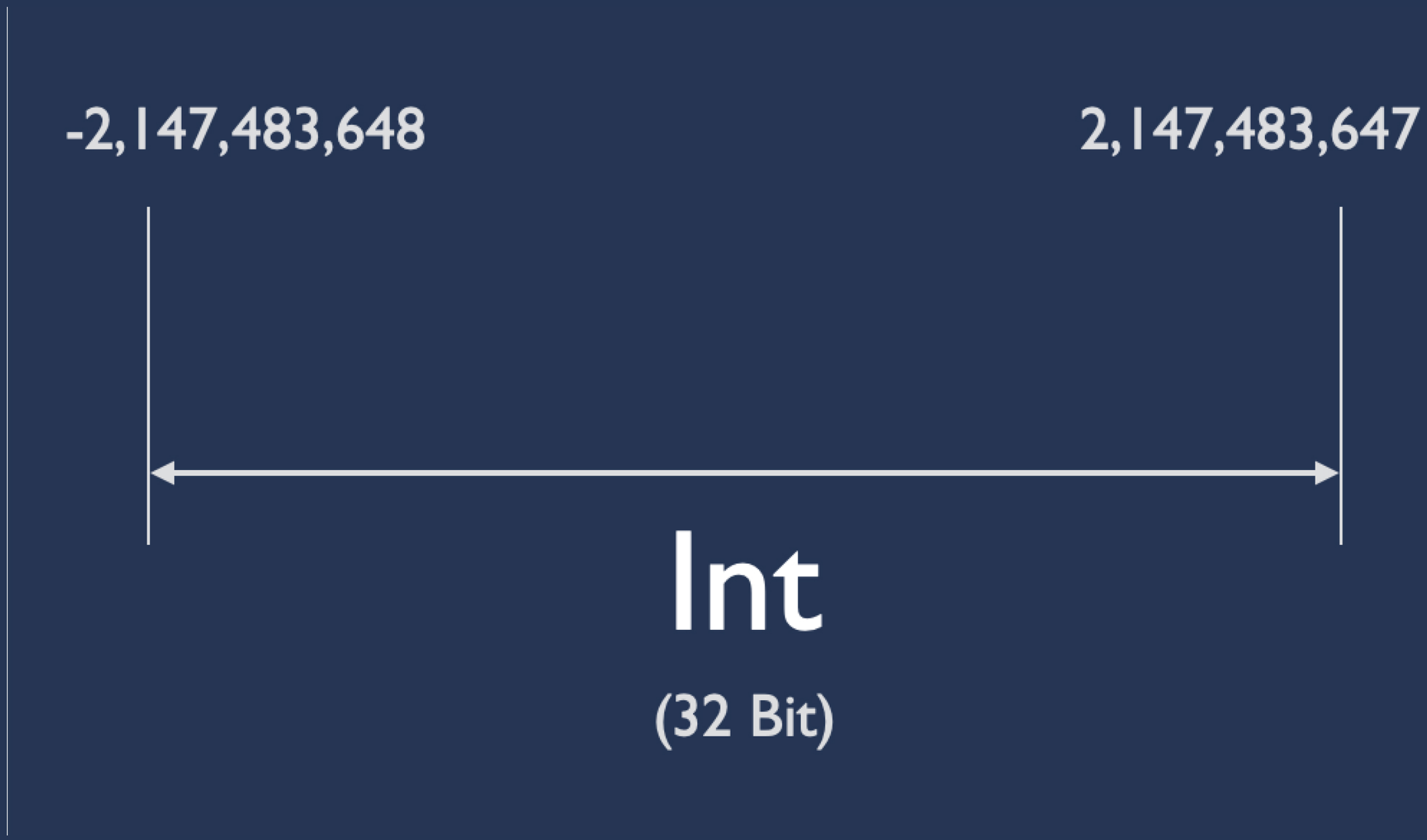
# Custom View

```
struct CustomButton<Content: View>: View {
    let handler: () -> Void // 執行的內容 (Closure)
    let content: () -> Content // 建立 content 的方法 (Closure)

    var body: some View {
        Button(action: handler, label: content)
            .font(.title)
            .frame(width: 44.0, height: 44.0)
            .background(Color.blue)
            .cornerRadius(22.0)
            .foregroundColor(.white)
    }
}
```

# 使用 Custom View

```
CustomButton(handler: {
    // 執行的內容 (Closure)
    commit(firstNum, secondNum, .Add)
}, content: {
    // 建立 content 的方法 (Closure)
    Image(systemName: "plus")
})
```

# 甚麼是 Optional


-2,147,483,648
2,147,483,647
Int
(32 Bit)

String?

Int?

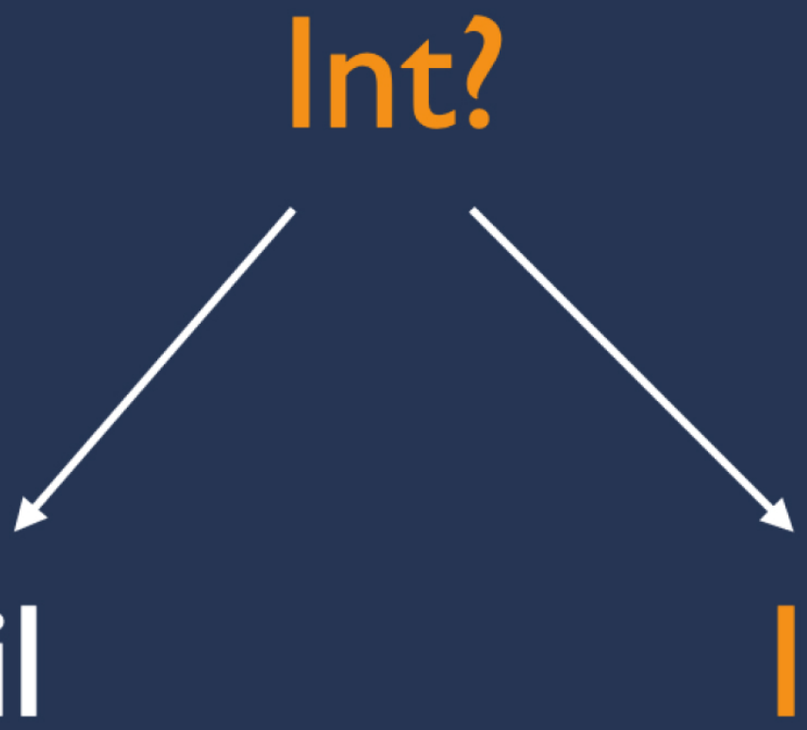
Optional

String

Int

Int(number1) → Int?

Int() 方法產生了 Int? 類型數值




Int? 有兩種可能性，一是 nil (即沒有值)，二是有效的數值 (-2,147,483,648 - 2,147,483,647) (32 位元)

if let xxx = Int? {

這種方法確保我們得到有效的數值

nil

Int

}

```
if let number1Value = Int(number1)
```

Int ← Int?

# String 轉 Int 中使用 if let

```
if let firstValue = Int(firstNum) {
    // 第一個數值能轉換，繼續執行
    if let secondValue = Int(secondNum){
        // 第二個數值能轉換，繼續執行
    }
}
```

# String 轉 Int 中使用 guard let

```
guard let firstValue = Int(firstNum) else {
    // 第一個數值不能能轉換，繼續這一句
    print("first num error")
    // 後面句字不再繼續執行
    return
}
guard let secondValue = Int(secondNum) else {
    // 第二個數值不能能轉換，繼續這一句
    print("second num error")
    // 後面句字不再繼續執行
    return
}
```

# 為甚麼要用 Optional?

# 假設沒有 Optional

var temp

天文台傳來資料 “24”

```
temp = Int(tempFromWeatherStation)
if temp <= 0 {    // 24 度不會發送訊息
    發送天氣寒冷訊息
}
```

# 假設無法表示的值都變成 0

var temp

天文台傳來資料 “error”

```
temp = Int(tempFromWeatherStation)    // 變成 0

if temp <= 0 {

    發送天氣寒冷訊息    //😵

}
```

# 為甚麼要用 Optional?

- 提升程式碼的安全性 (Safe)
- 編譯器 (Compiler) 可以提早檢測出錯誤，減少出錯的機會

# Binding

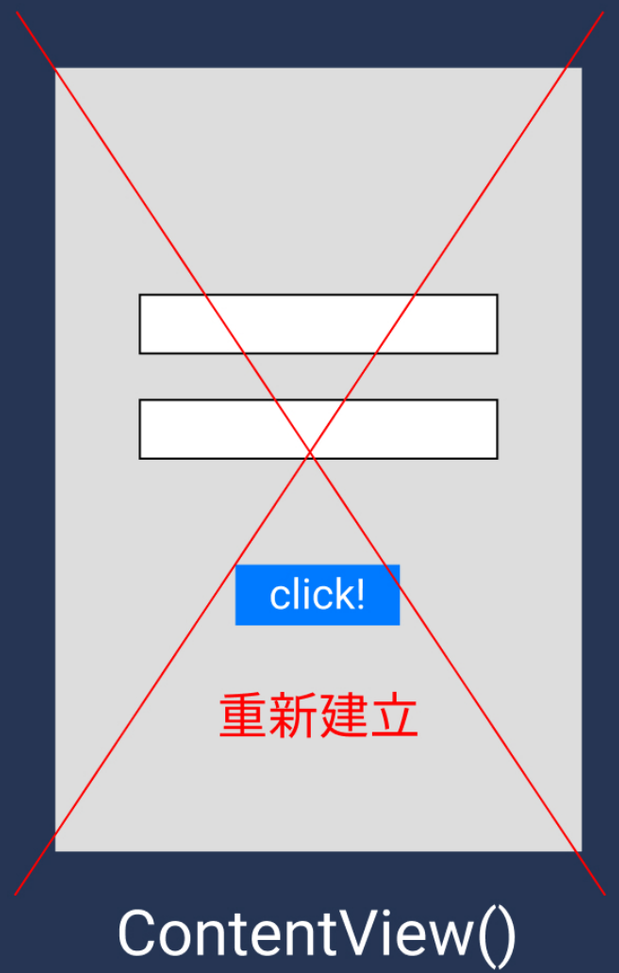
# 為甚麼要用 @State?

```
struct ContentView: View {
    var body: some View {
        VStack {
            TextField()
            TextField()
            Button()
        }
    }
}
```


ContentView()

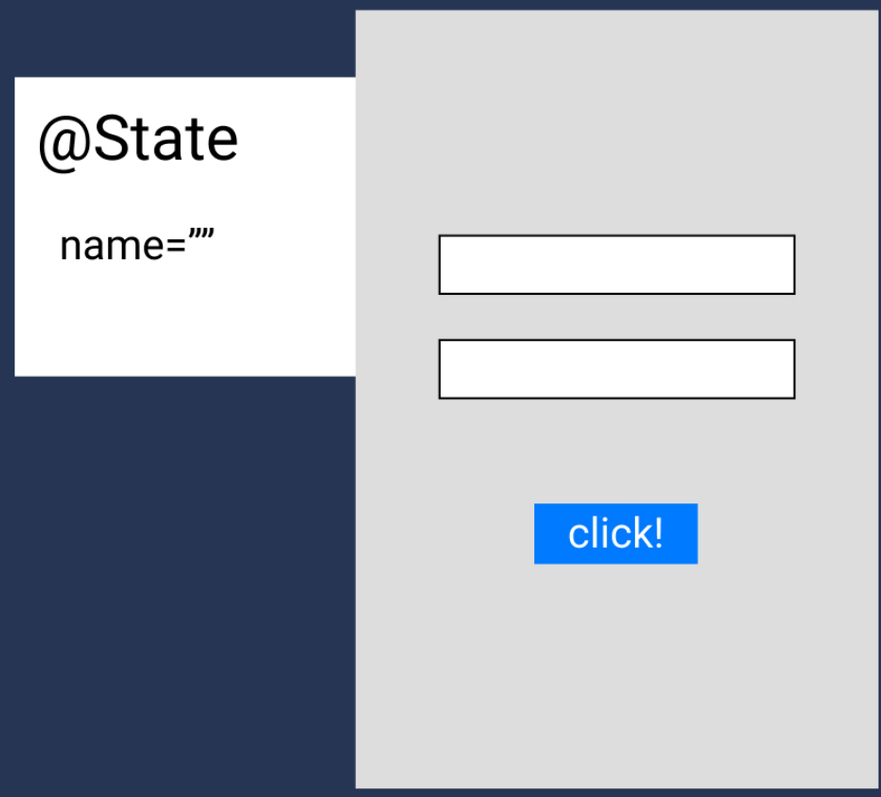
# 基本上建立後不會改變，如果需要加減東西，需要重新建立

```
struct ContentView: View {
    var body: some View {
        VStack {
            TextField()
            TextField()
            Text()
            Button()
        }
    }
}
```



# 使用 @State 加入變量，@State 自動將變量抽出來，記錄在某個地方

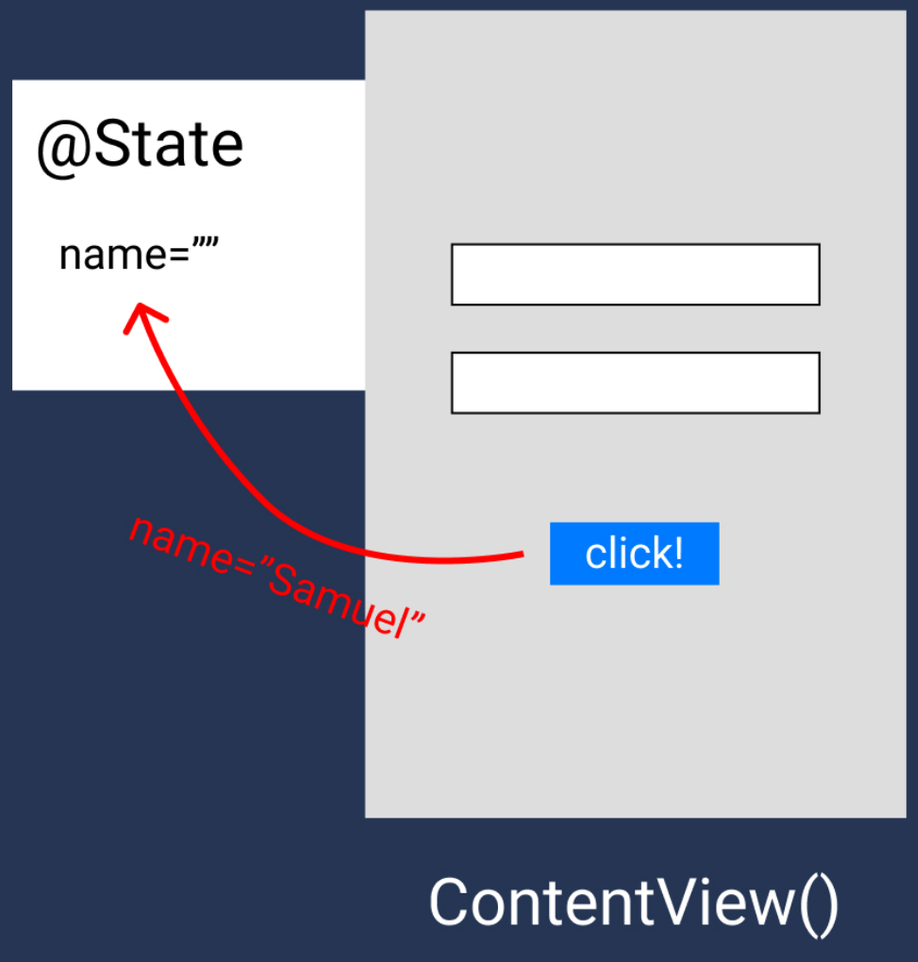
```
struct ContentView: View {
    @State var name = ""

    var body: some View {
        VStack {
            TextField()
            TextField()
            Text(name)
            Button() {
                name = "Samuel"
            }
        }
    }
}
```




ContentView()

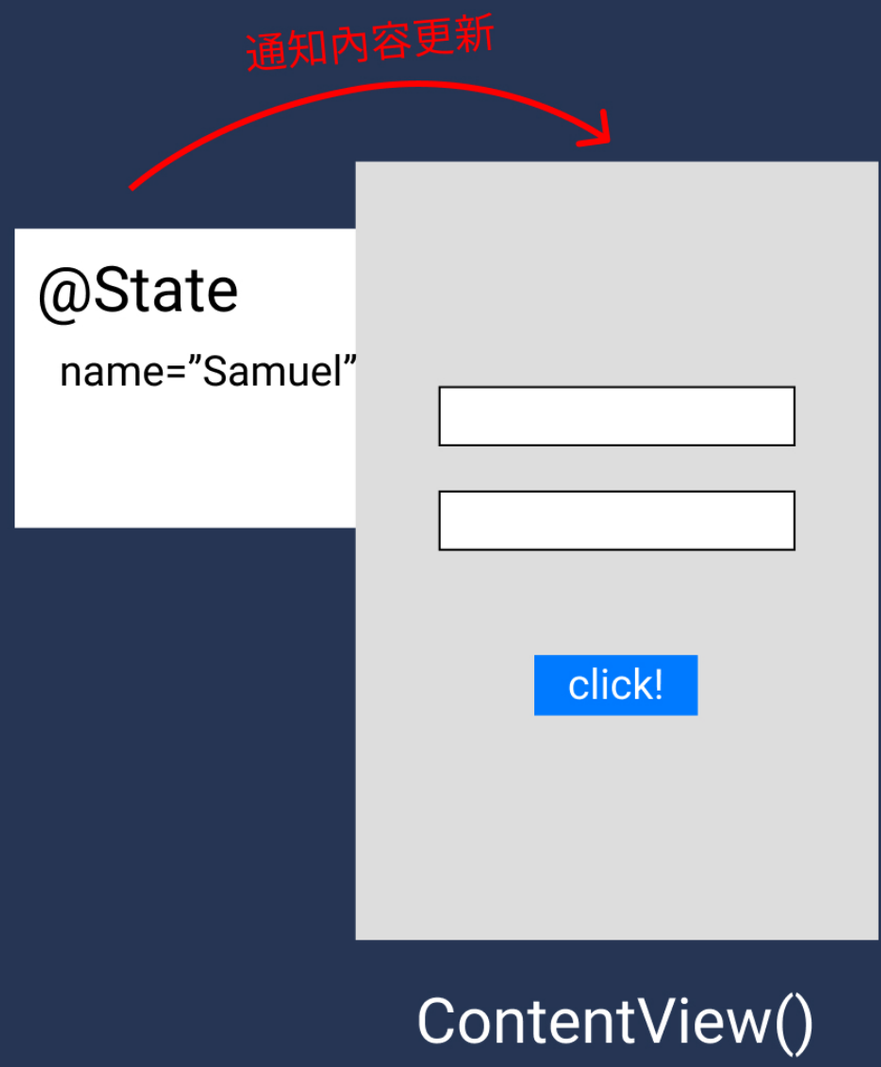
# 修改 @State 內的變量

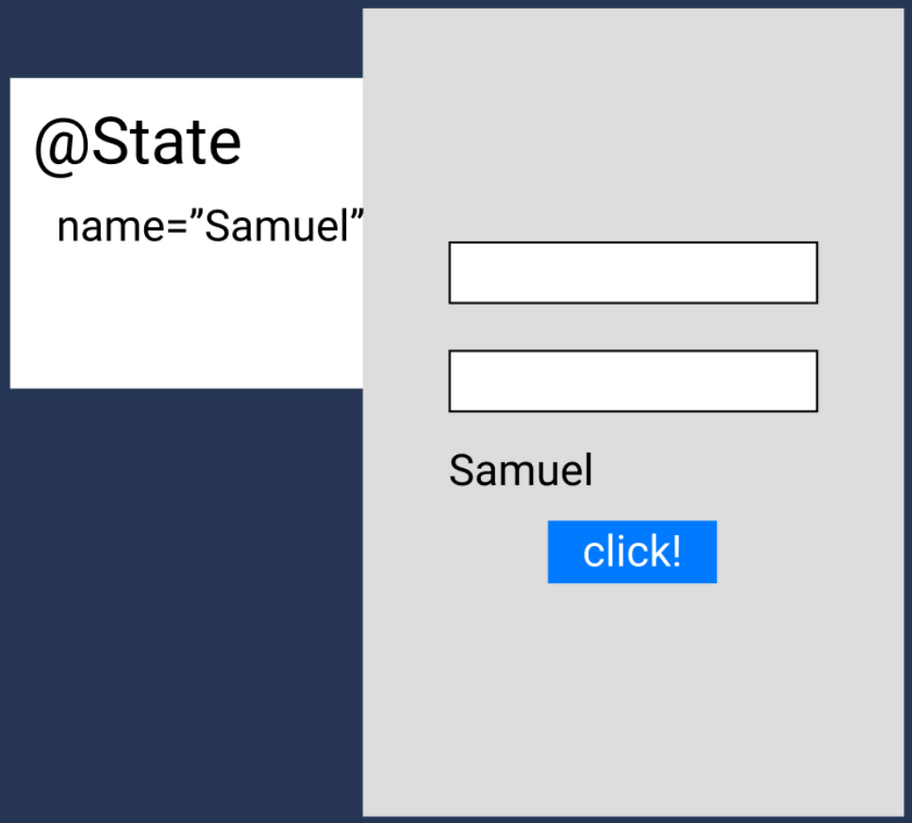
```
struct ContentView: View {
    @State var name = ""

    var body: some View {
        VStack {
            TextField()
            TextField()
            Text(name)
            Button() {
                name = "Samuel"
            }
        }
    }
}
```



# @State 通知 ContentView 內容改變，需要重新建立(部分)畫面

```
struct ContentView: View {
    @State var name = ""

    var body: some View {
        VStack {
            TextField()
            TextField()
            Text(name)
            Button() {
                name = "Samuel"
            }
        }
    }
}
```

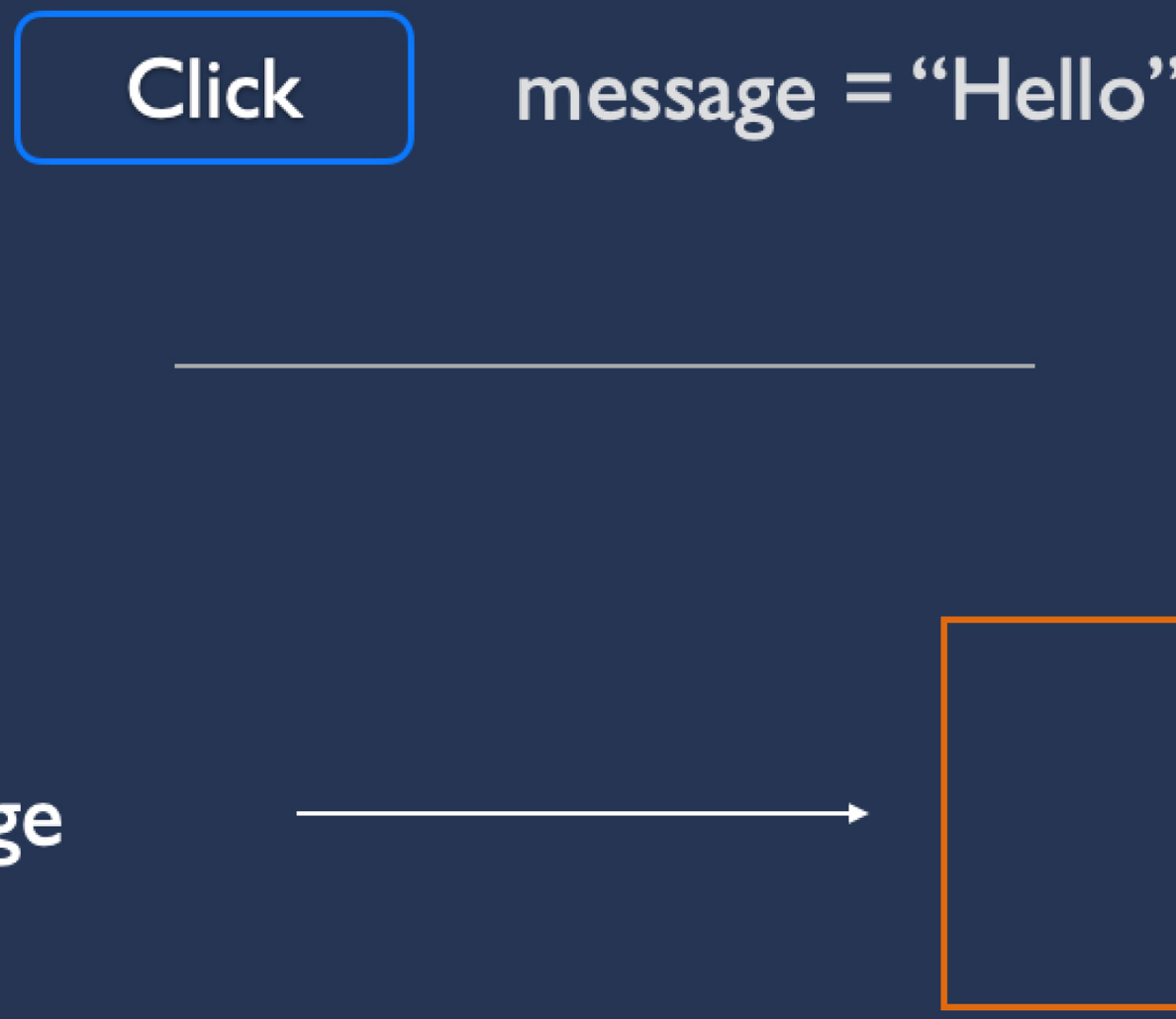





ContentView()

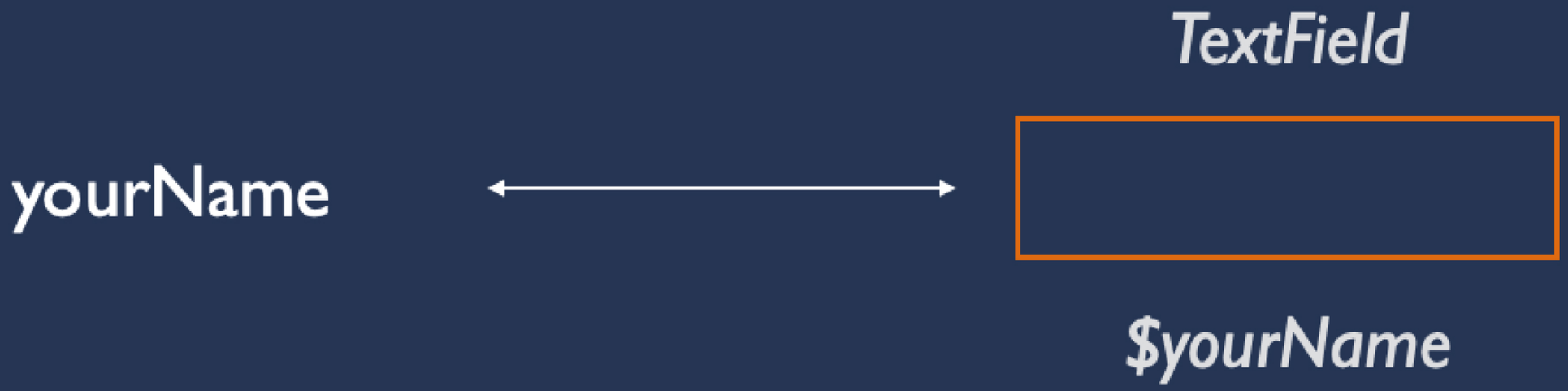
# One-way Binding

# One-way Binding



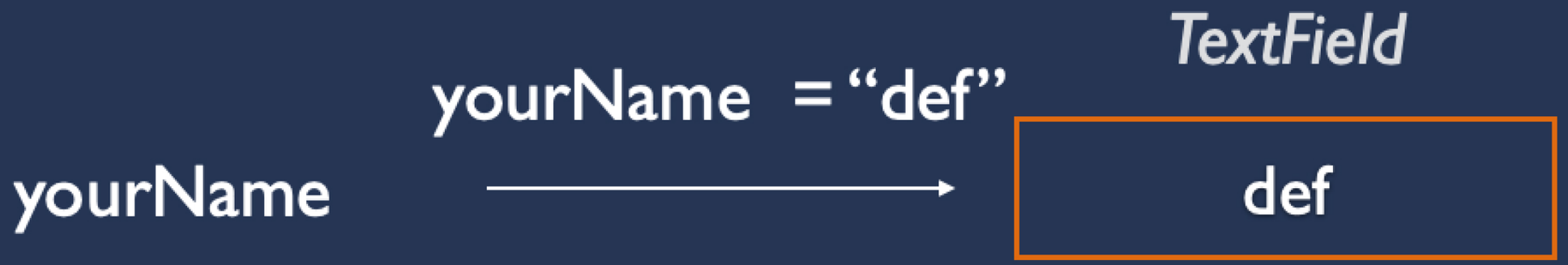
```swift
@State var yourName = ""
@State var message = "Type your name above."

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()
}
```

# Two-way Binding



# Two-way Binding

# Two-way Binding

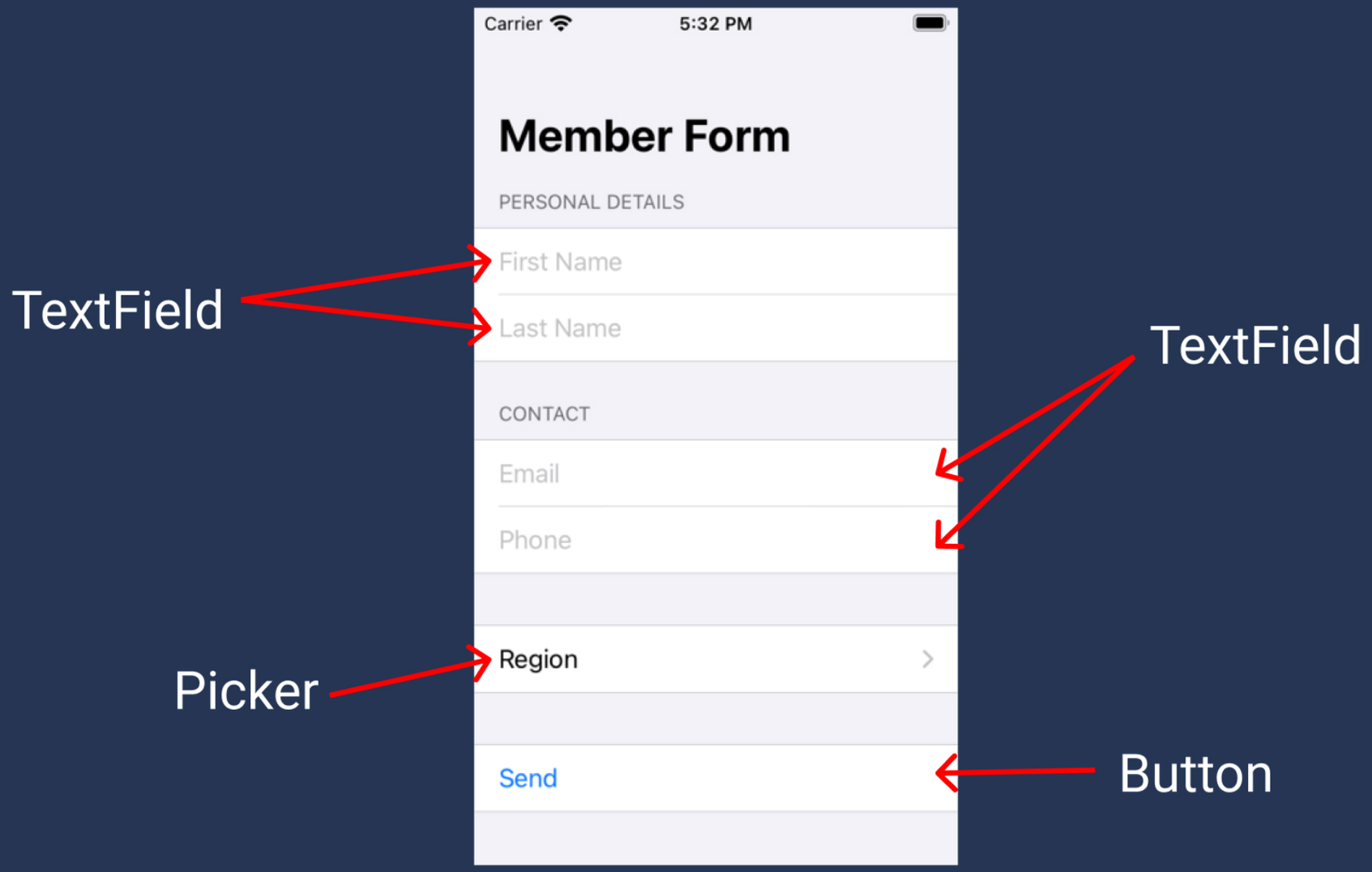



*$yourName*

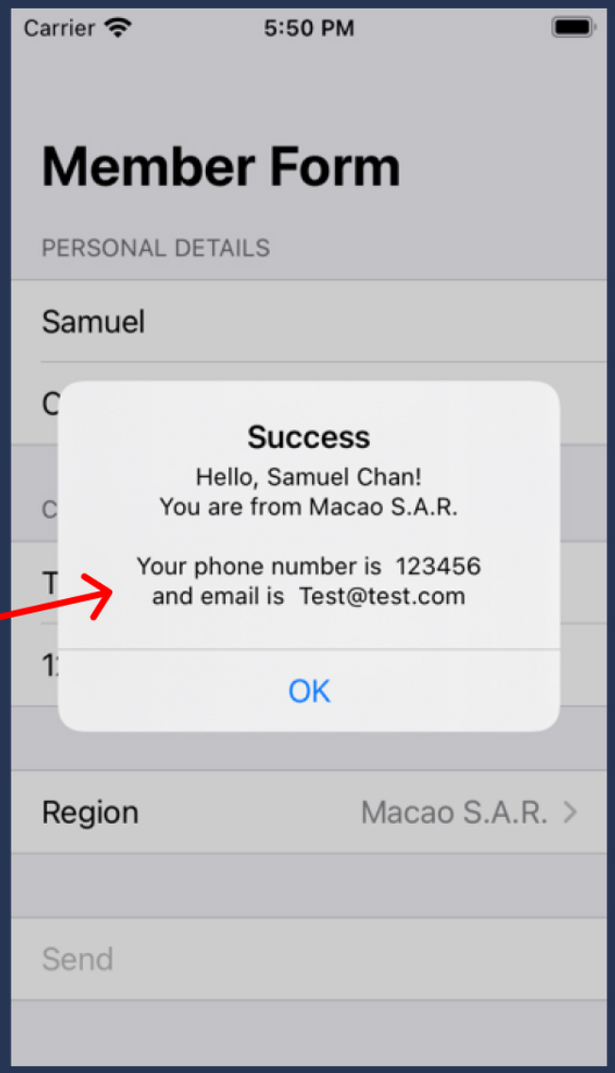
TextField 變成 “def”

# 課堂作業 Week 3 (計算分數!)

- 排版與組件使用練習。
- 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。


Carrier
5:32 PM
Member Form
PERSONAL DETAILS
First Name
Last Name
CONTACT
Email
Phone
Region
Send
TextField
TextField
Picker
Button

Alert

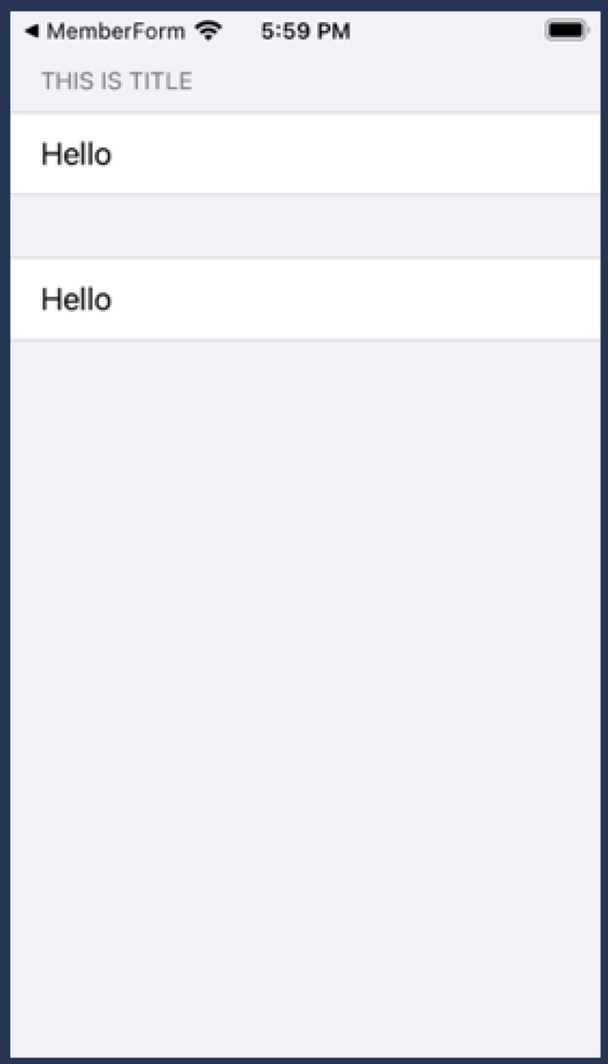

Carrier
5:50 PM
Member Form
PERSONAL DETAILS
Samuel
Success
Hello, Samuel Chan!
You are from Macao S.A.R.
Your phone number is 123456
and email is Test@test.com
OK
Region
Macao S.A.R.
Send

# 作業要求

1. 檔案命名：[你的全名].week3.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/PJYXQbc4WMZehYaThHvc
4. 到期日：2021 年 7 月 21 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

# Form / Section

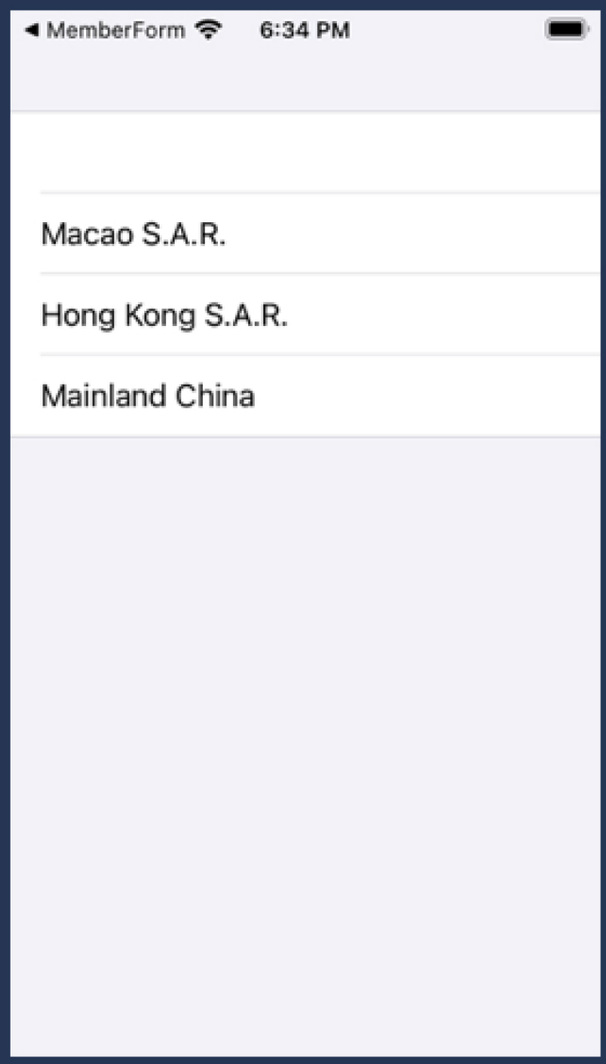
```
Form
{
    Section(header: Text("This is title"))
    {
        Text("Hello")
    }

    Section
    {
        Text("Hello")
    }
}
```



# ForEach

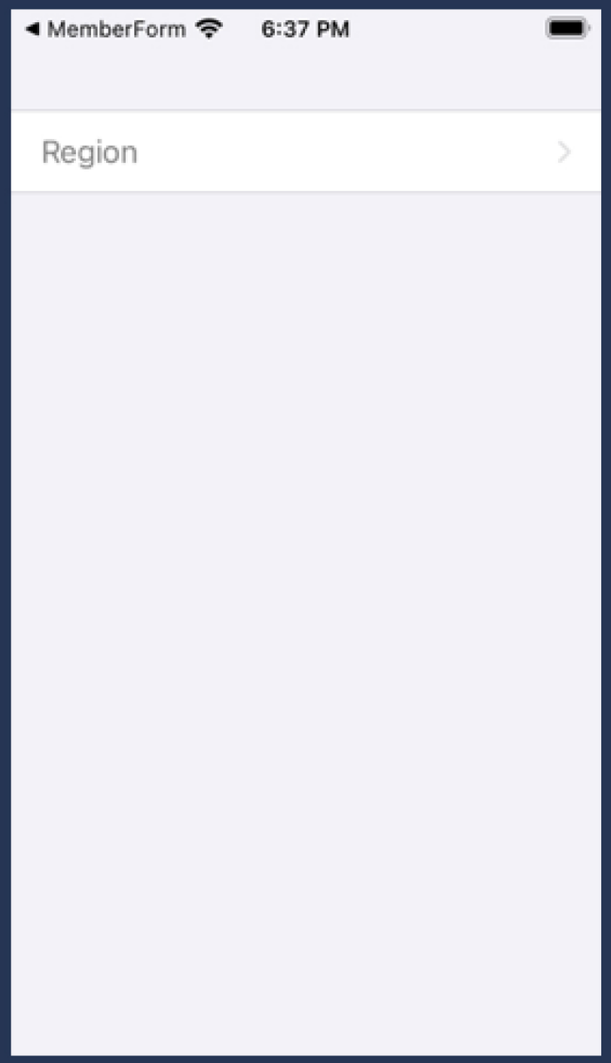
```
static let countries = [
    "",
    "Macao S.A.R.",
    "Hong Kong S.A.R.",
    "Mainland China"
]
```

```
ForEach(0 ..< Self.countries.count){ index in
    Text(Self.countries[index])
}
```



# Picker

```
Picker("Region", selection: $region){

    ForEach(0 ..< Self.countries.count){ index
in
        Text(Self.countries[index])
    }

}
```



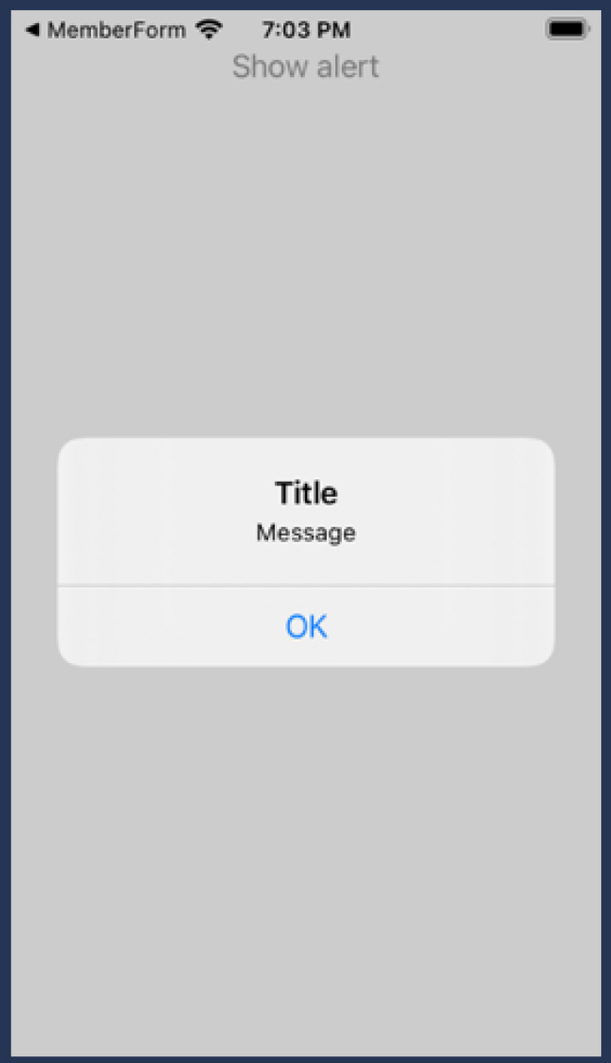
# Navigation View

```
NavigationView {

    Text("Hello World")
        .navigationTitle(Text("Hello"))

}
```

# Alert

```
...
    @State var showingAlert = false
    @State var message = ""
...
```

```
Section {
    Button(action: {

        showingAlert = true
    }, label: {
        Text("Send")
    })
    .alert(isPresented: $showingAlert){
        Alert(title: Text("Success"), message:
Text(message))
    }
}
```



# Extra: 估數字遊戲 (下一堂)

設計檔案：https://www.figma.com/file/RCNlbEk9Or5fsWqzBHNZDM/Game-Screen?node-id=0%3A1