iOS 應用程式開發

Week 1

第一部分

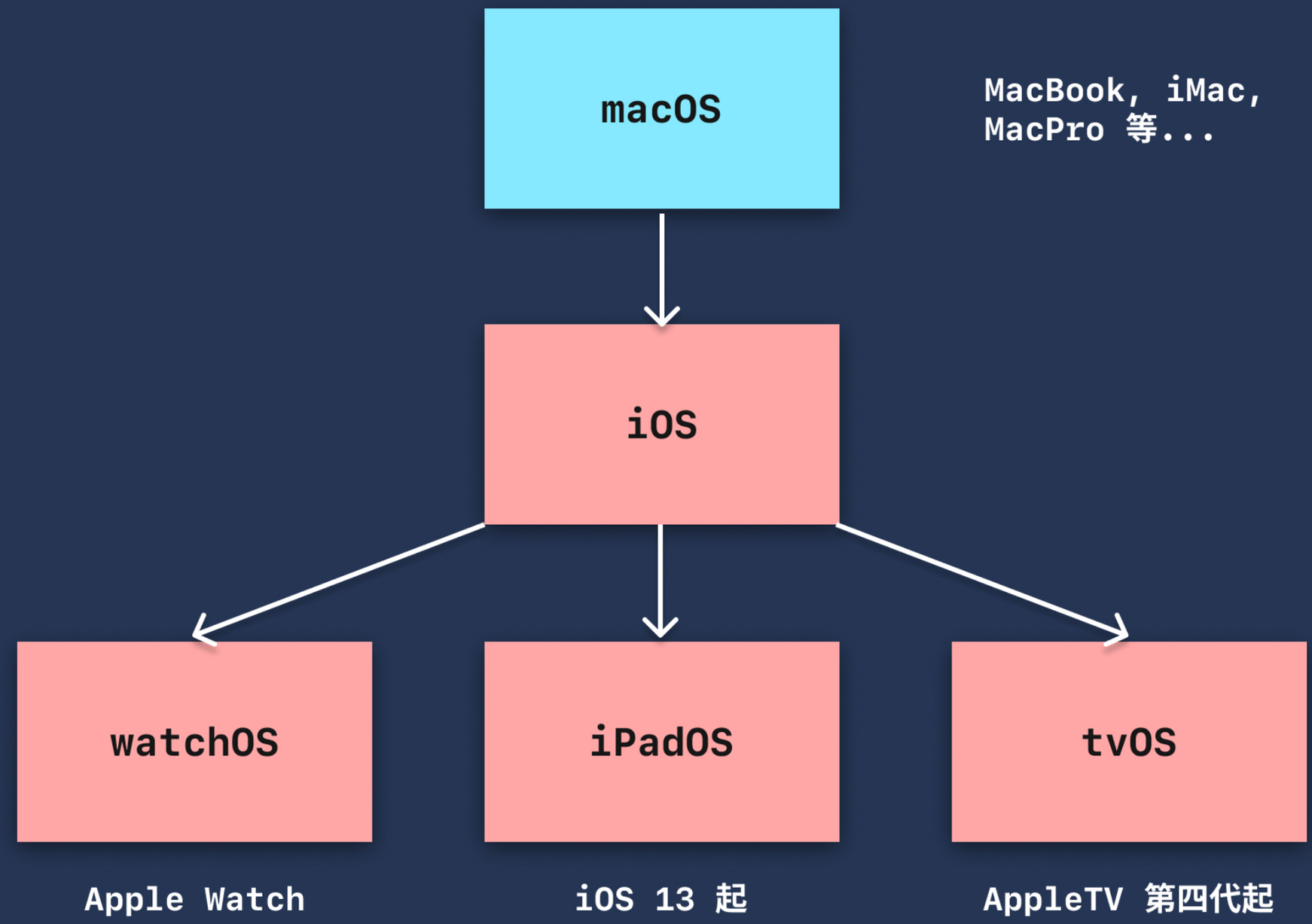
- 介紹 iOS 背景資料
- 甚麼是 App?
- 如何寫一個 App?
- 不同技術的比較
- 重要的概念: 大部分移動應用程式是為了解決特定的問題而開發的
- 基本的開發環境介紹及操作

第二部分

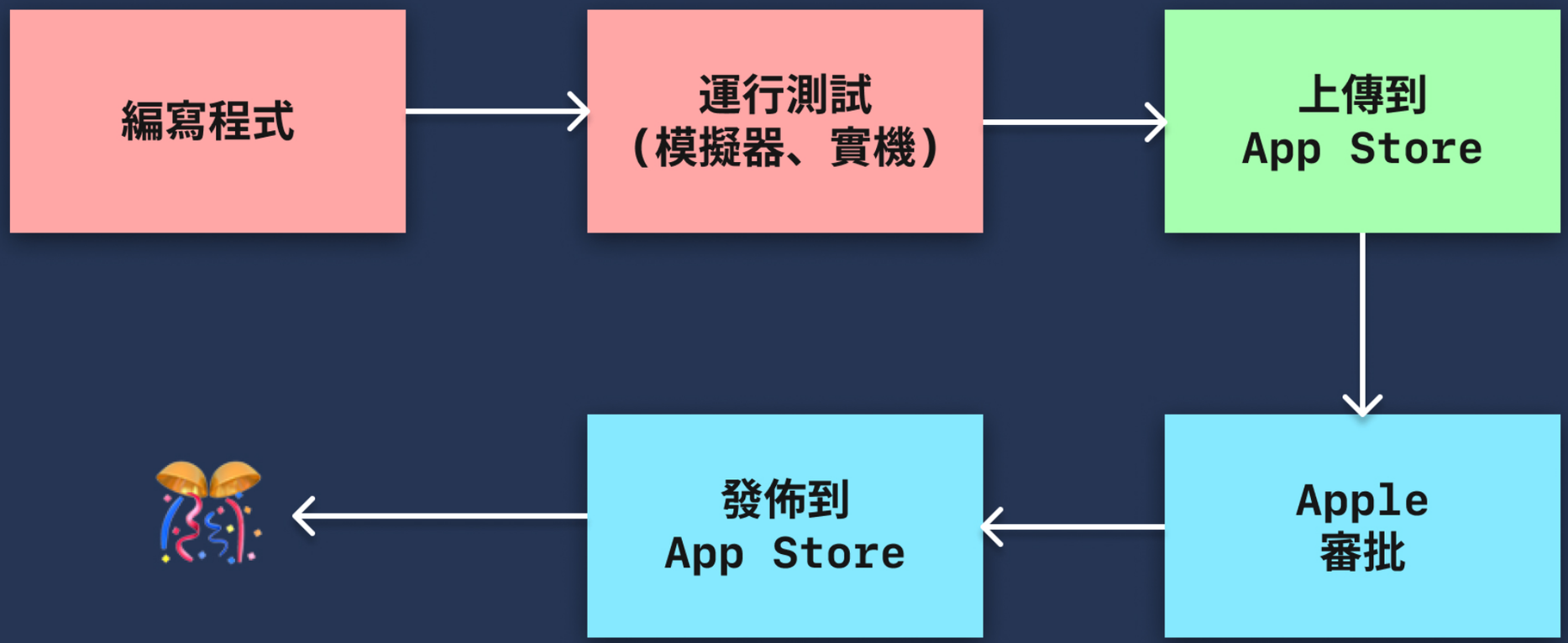
- 體驗開發軟件 Xcode，編寫第一個 App
- Swift UI 介面設計
- Swift UI vs UIKit
- UIKit 介面設計
- 加入互動功能 (輸入、計算、輸出)
- 介面製作練習

iOS 家族

iOS 的起源

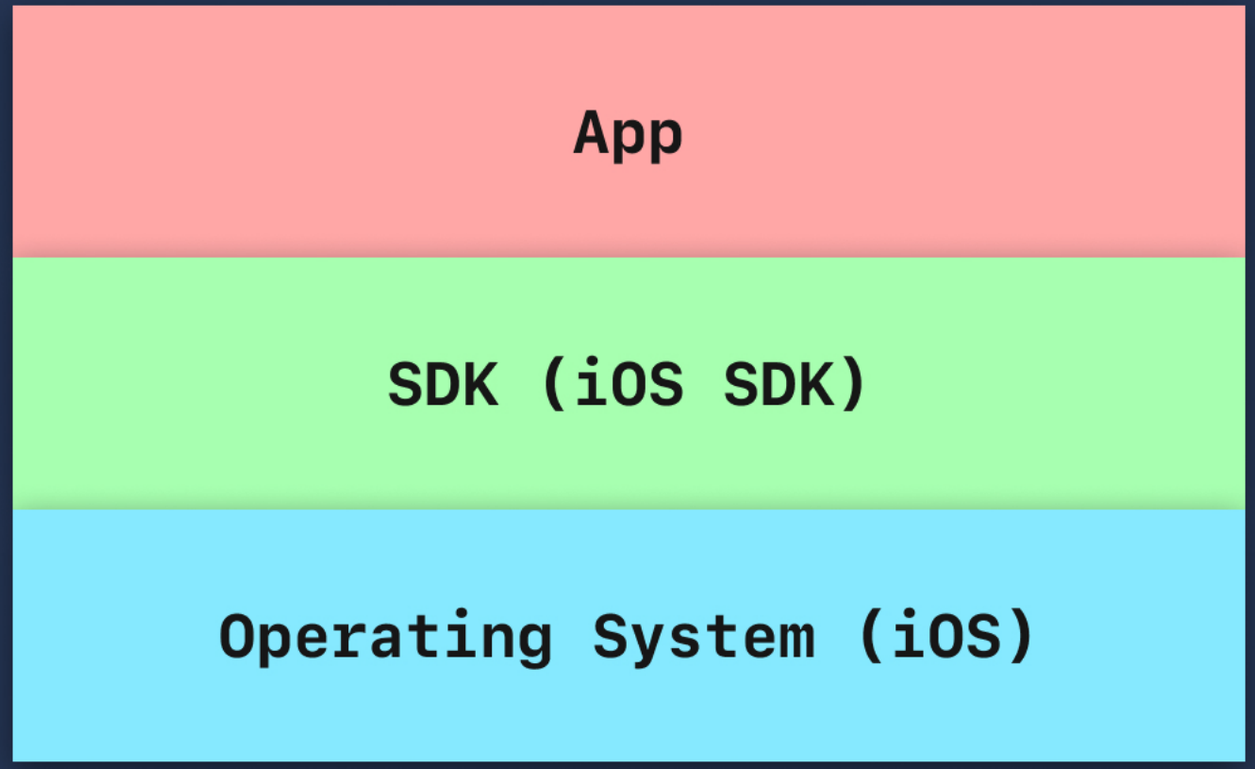

macOS
MacBook, iMac,
MacPro 等...
iOS
watchOS
iPadOS
tvOS
Apple Watch
iOS 13 起
AppleTV 第四代起

如何寫一個 App

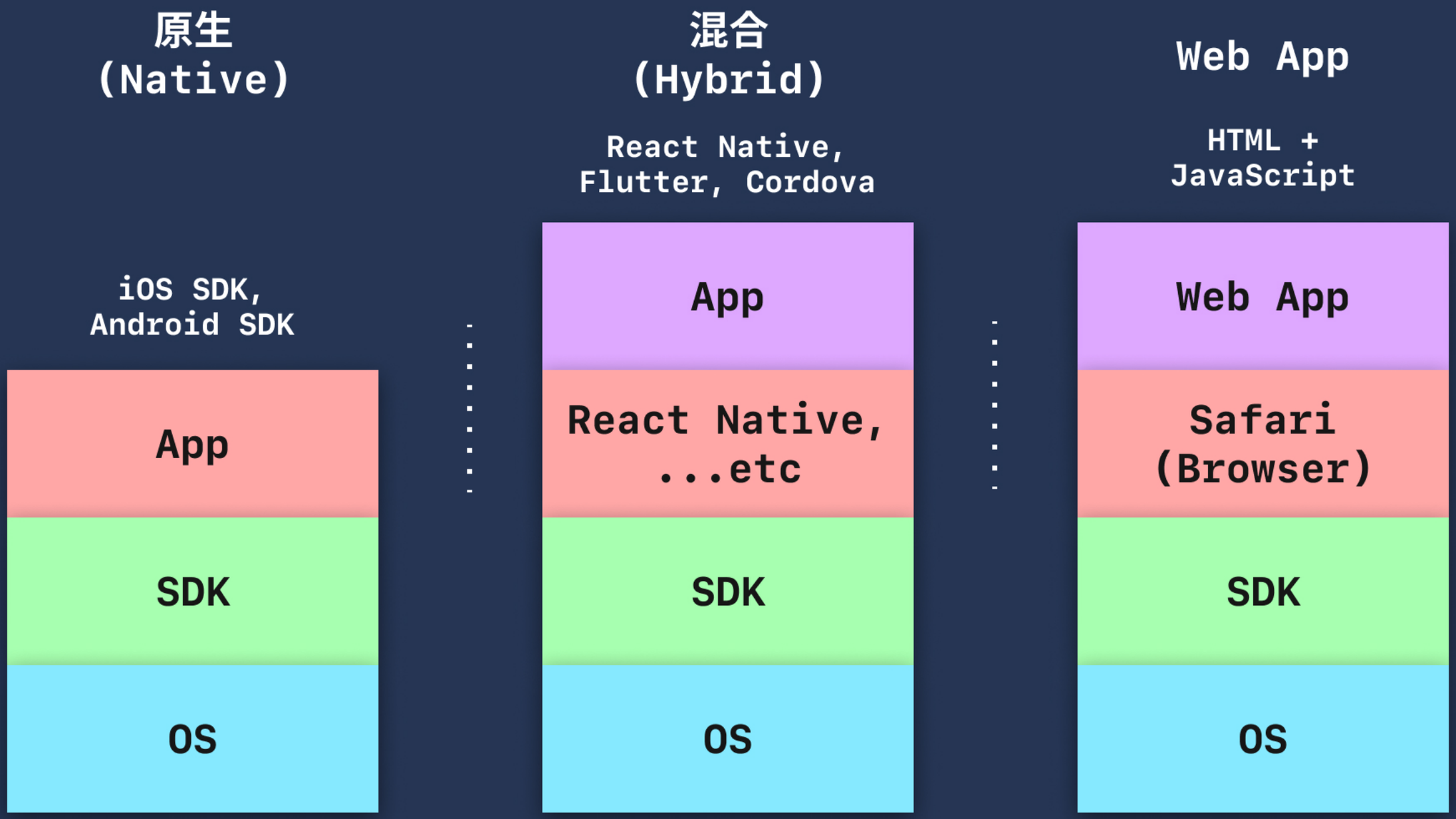


甚麼是 App?

- App 全寫 Application (應用程式)
- Software Development Kit (SDK; 程式開發包)



如何寫一個程式


原生
(Native)
iOS SDK,
Android SDK
App
SDK
OS
混合
(Hybrid)
React Native,
Flutter, Cordova
App
React Native,
...etc
SDK
OS
Web App
HTML +
JavaScript
Web App
Safari
(Browser)
SDK
OS

技術比較

	Native App （原生）	Hybrid App （混合技術）	Web App
執行速度	5	3–4	1–2
學習門檻	3–5	3–4	1–2*
推出市場	開發、測試 時間長	較快	最快
應用場境	遊戲、VR、AR 等需要高效能的 應用程式	休閒類遊戲、 資訊、表格、 購物應用等	資訊、內容經常變 動的應用程式

混合技術


17:34
WeChat
HEYTEA喜茶
星期二
绿色喜茶，灵感再生中
聊天
通訊錄
發現
我


17:34
HEYTEA喜茶
星期二 21:42
INSPIRATION OF GREEN
灵感再生
实验室
绿色喜茶，灵感再生中
喜爱音乐，MUSIC ON
茶茶测评 | 芝芝茗茶解构篇
茶茶与你 | 请她喝37杯喜茶
奥利奥巴士，长沙惠农站回顾
上海东方明珠店&八城新店，灵感开启
喜茶GO
关于喜茶
喜茶星球


17:34
喜茶GO
銅鑼灣希慎廣場店
距離您112.0km
更多門店信息
當季限定
喜茶製冰
芝芝水果茶
滿杯水果家族
芝芝茗茶
波波家族
熱飲推薦
純茶
加料
靈感提示
綠色喜茶
奧利奧波波冰
冷萃桂花綠
COLD BREW GREEN OSMANTHUS
當季限定
奧利奧波波冰
含乳製品
含小麥、大豆
HKD42
選規格
芝芝桃桃
含乳製品、茶
含桃子
點單
靈感舖
取茶
我的

編寫一個成功的 App...

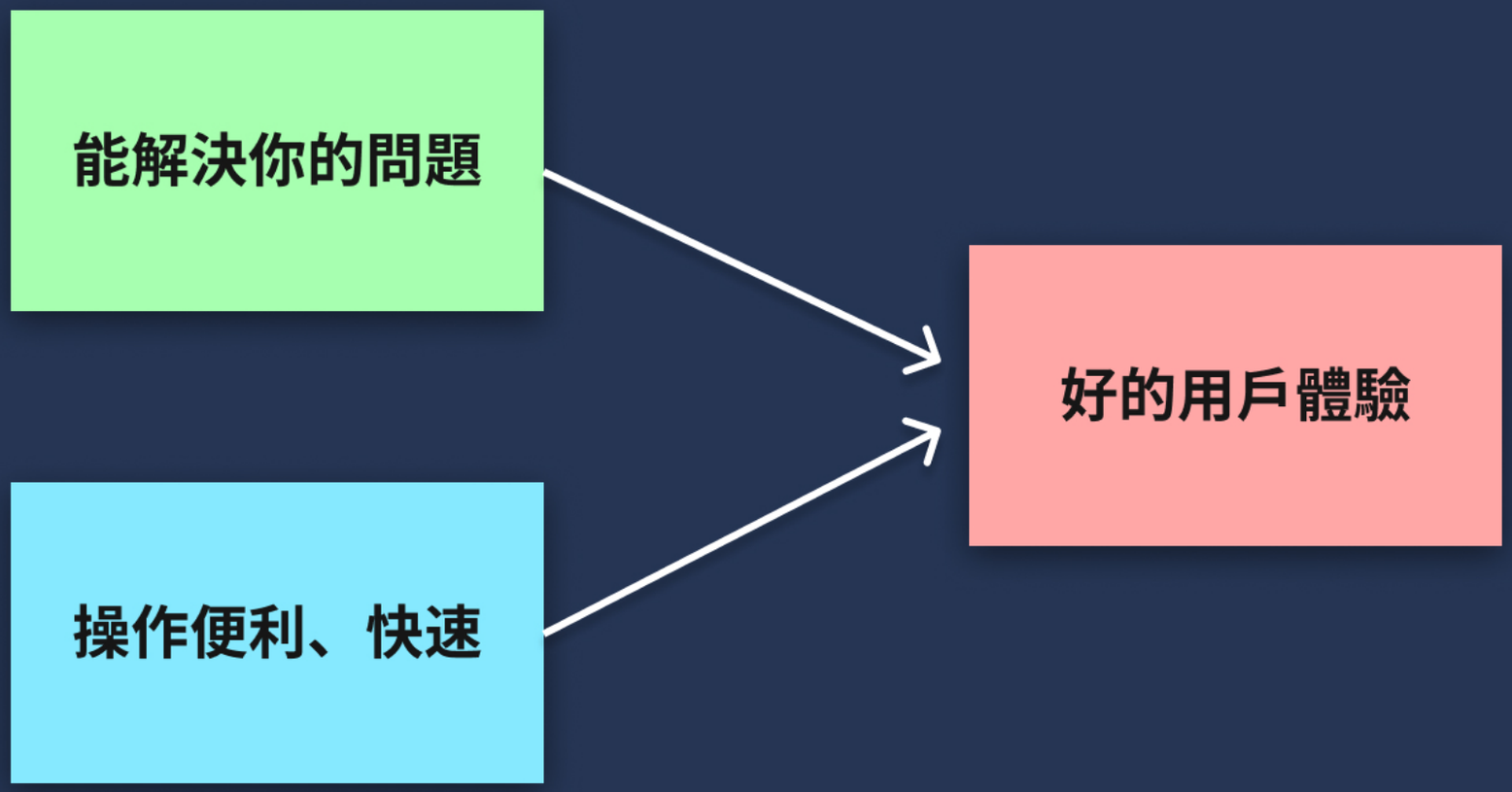
Observe

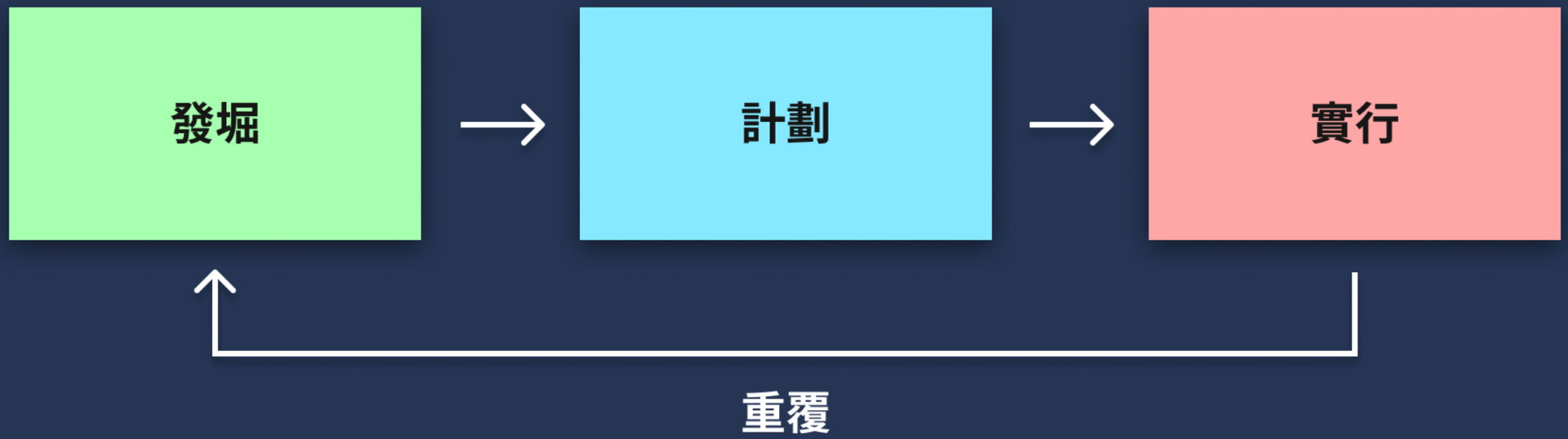
+

Creative

...觀察 + 創意, 就能開發出好的應用程式!

大部分移動應用程式是為了解決〈特定的問題〉而開發的

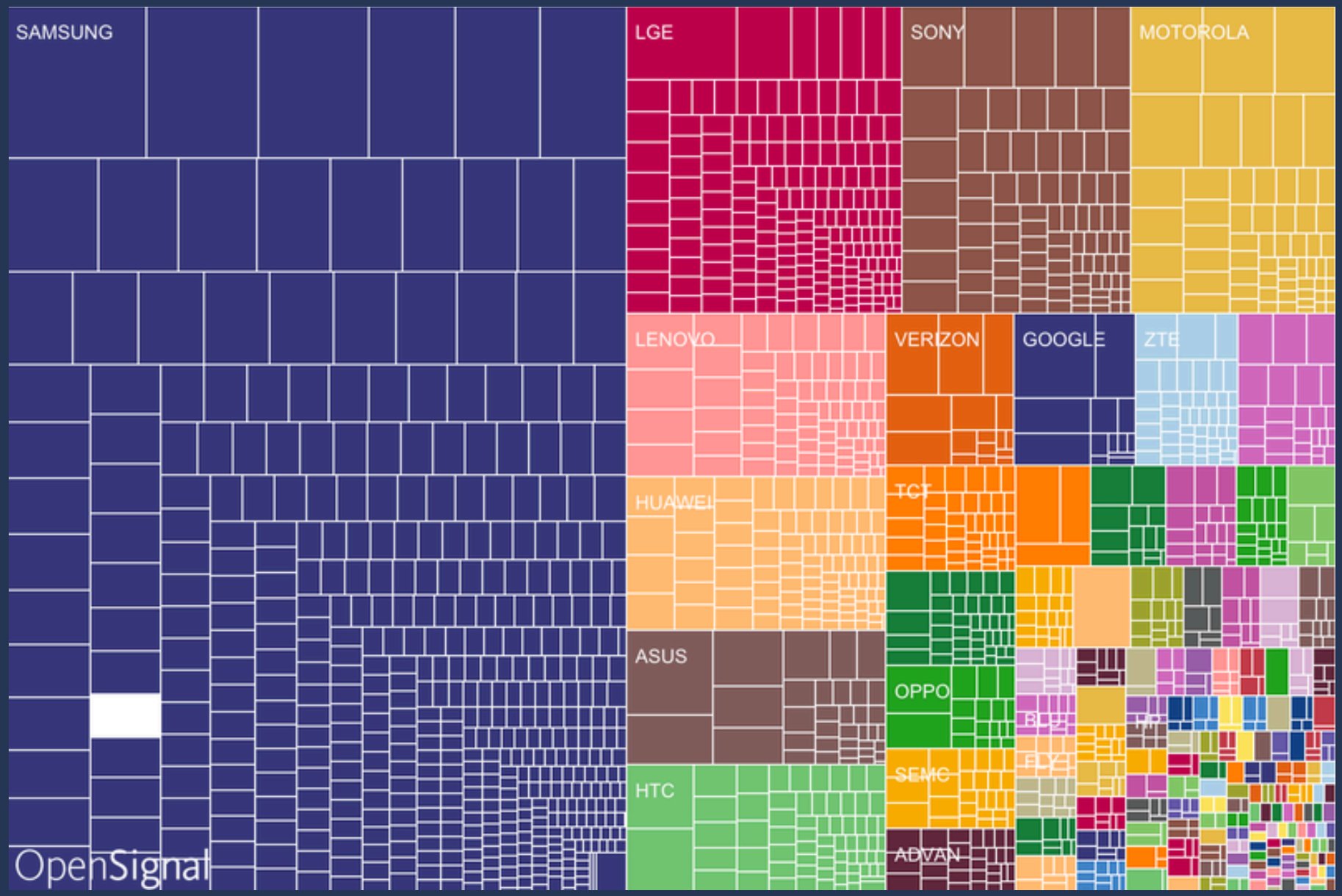

能解決你的問題
操作便利、快速
好的用戶體驗


發掘
計劃
實行
重覆

開發 iOS 應用程式的〈優勢〉？

- 能夠確保所有(或大部分)硬件都能運行同一個應用程式, 因為 iOS 的運行平台在 Apple 的控制之內 (iPhone / iPod / iPad)
- 簡便的發佈渠道 – App Store
- iOS 的產品都能夠連接互聯網, 而 iPhone 更是所有智能手機中上網普及程度最高
- 所有應用程式均會經過審批通過才能發佈，保障質素

Android Fragmentation [1]



[1] 來源 OpenSignal http://opensignal.com/reports/2015/08/android-fragmentation/

開發 iOS 應用程式的〈弱勢〉？

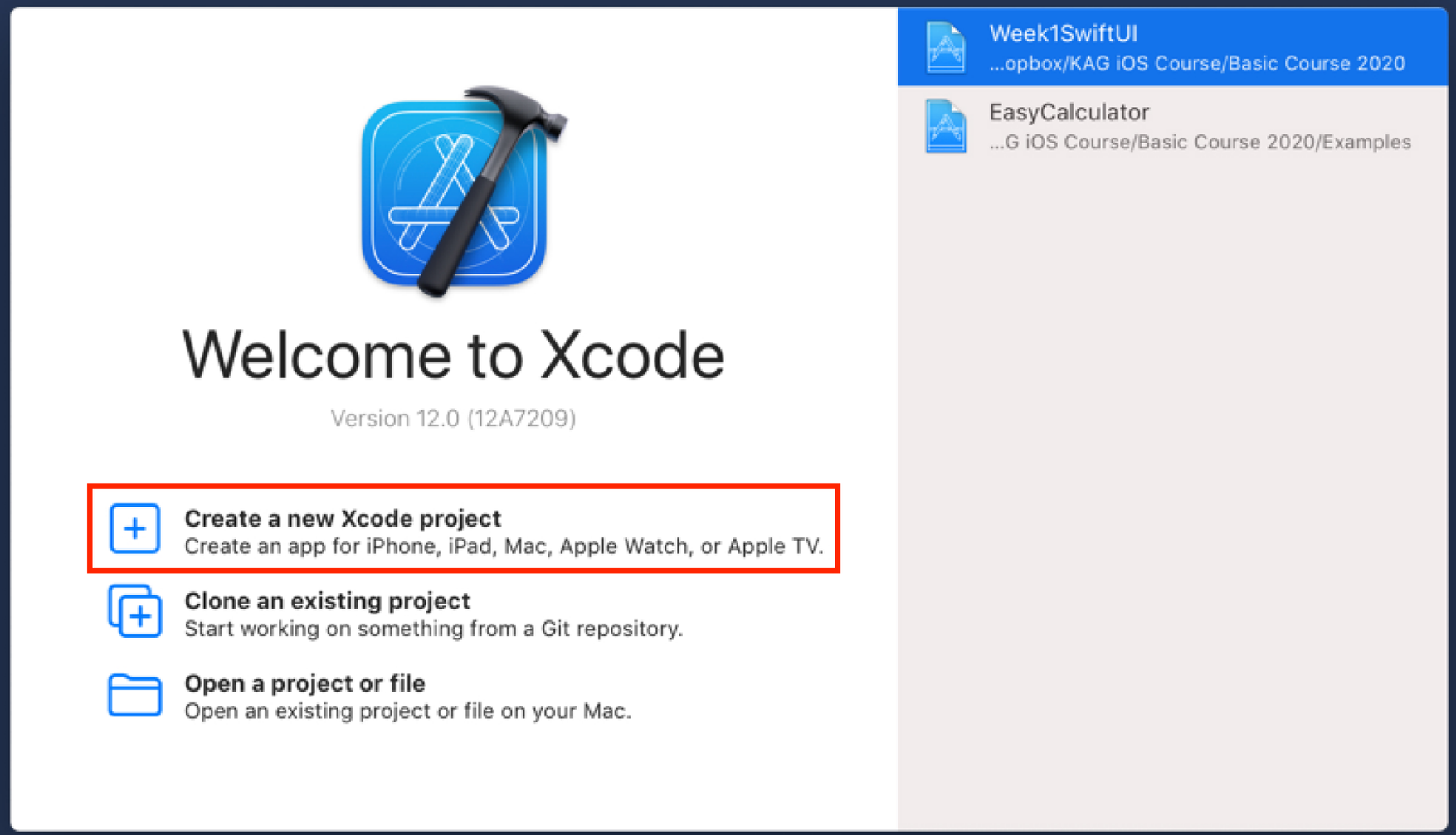
- 平台相對較封閉，程式需要經過審批才可以在 App Store 上架
- 程式審批過程較慢，長達三星期至一個月
- 不符合蘋果條款的程式會被撤銷，造成某部份的程式無法推出
- 在本機測試時並不須作審批

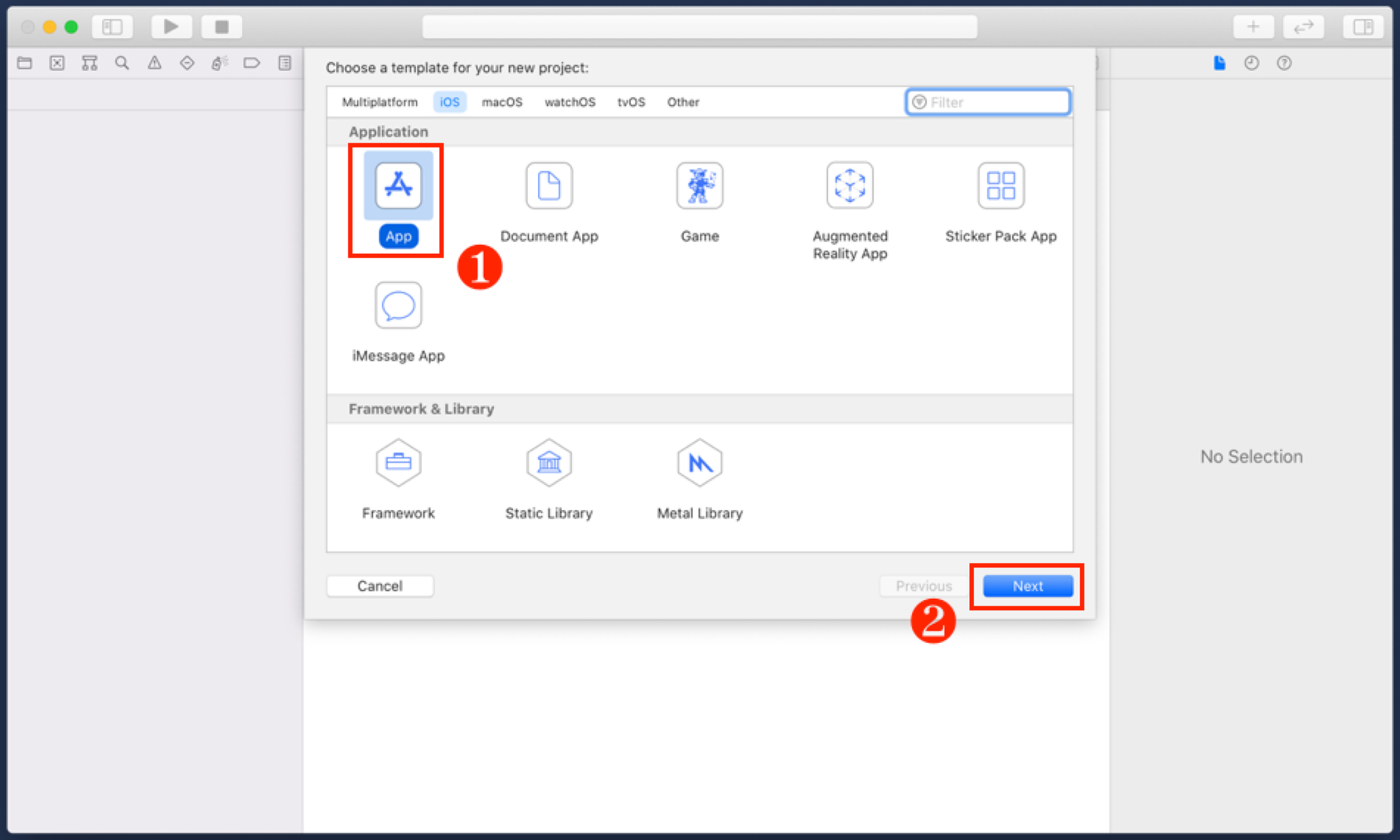
基本的開發環境介紹及演示、實際操作

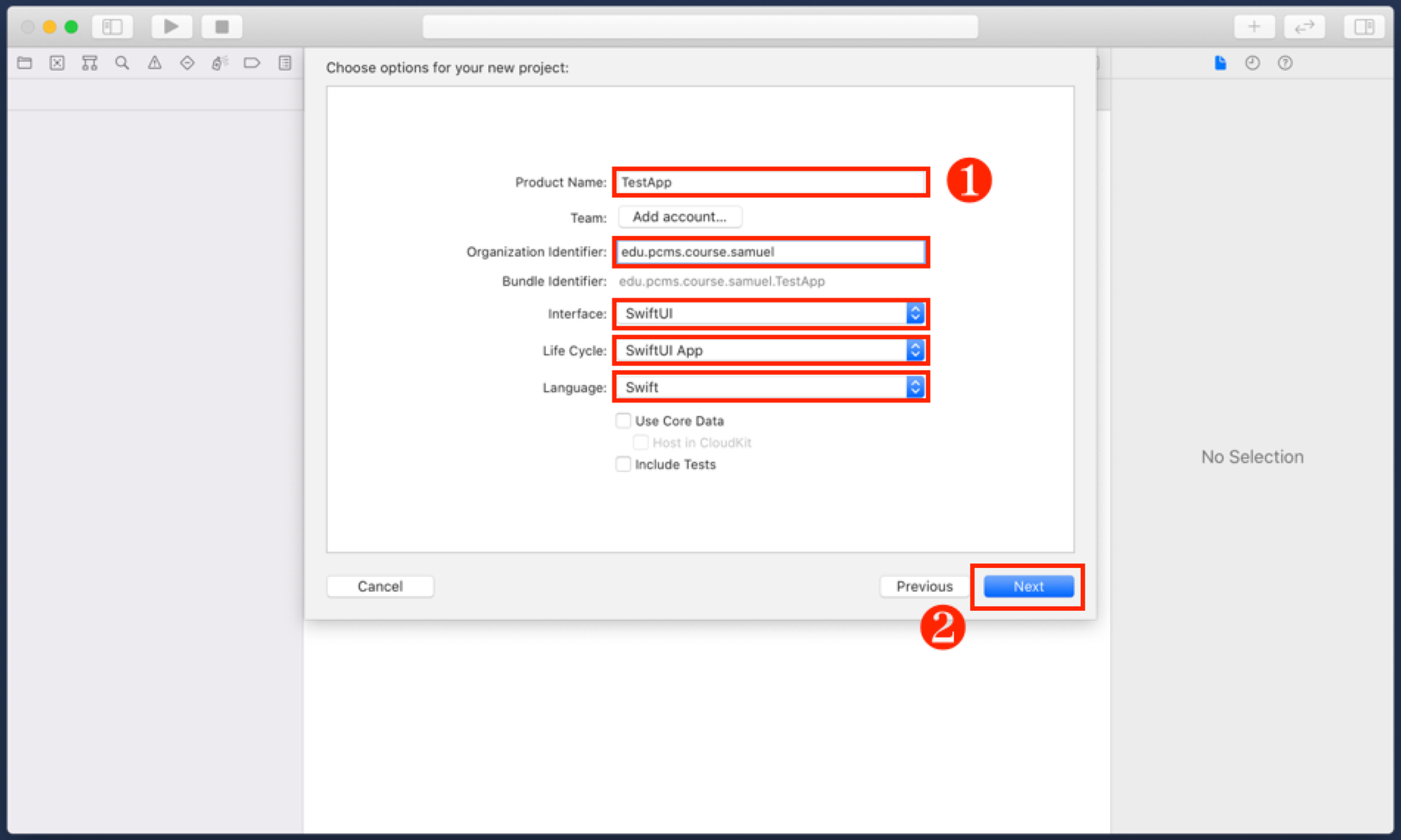
基本的開發環境介紹

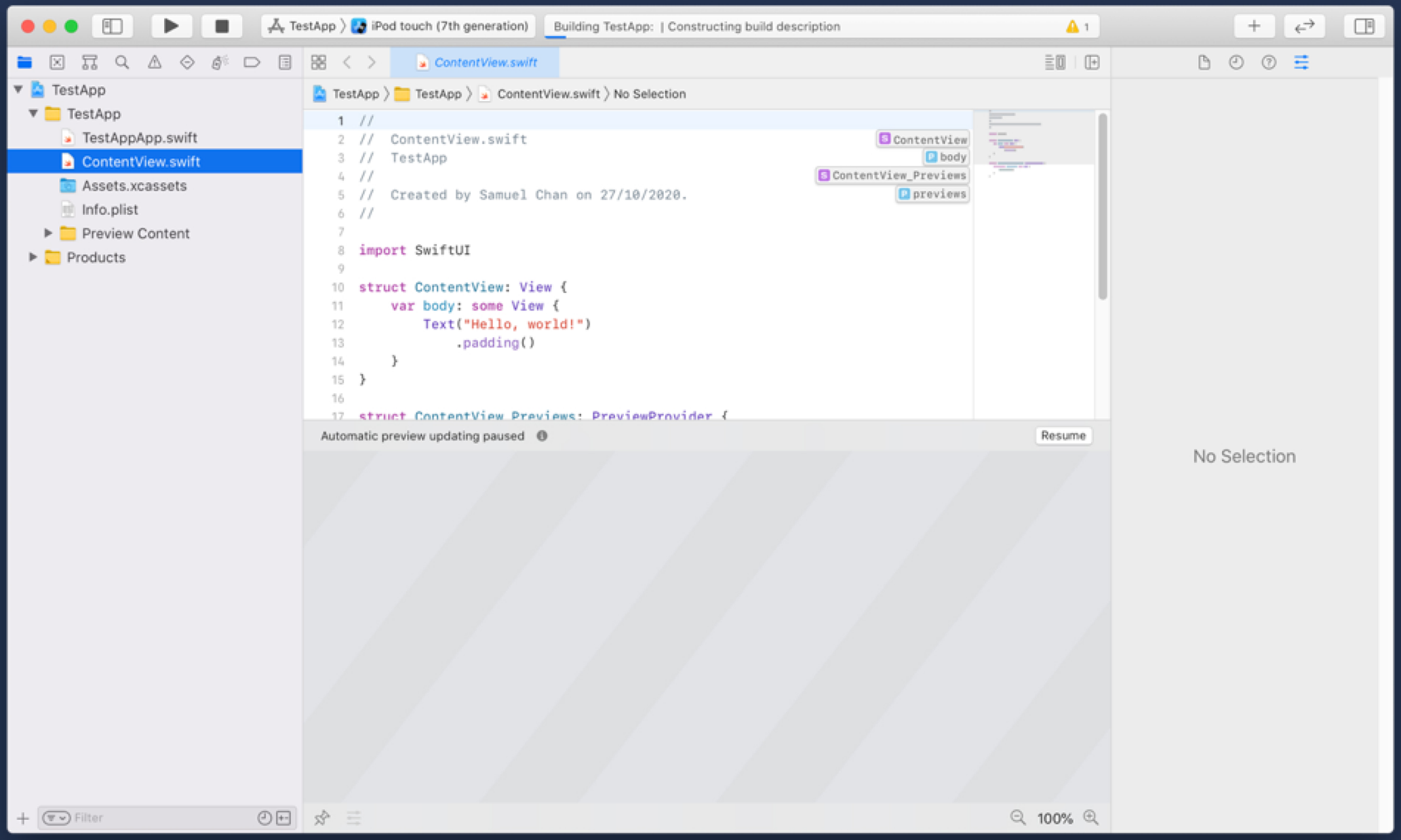
- macOS (Mac OS) 是一個 UNIX 系統
- iOS 是 macOS 的精簡版本
- Finder 是 macOS 內的檔案管理程式
- Mac ⌘ 相對於 Windows Ctrl
- ⌘+C = 複制 / ⌘+V = 貼上 /⌘+X = 剪下

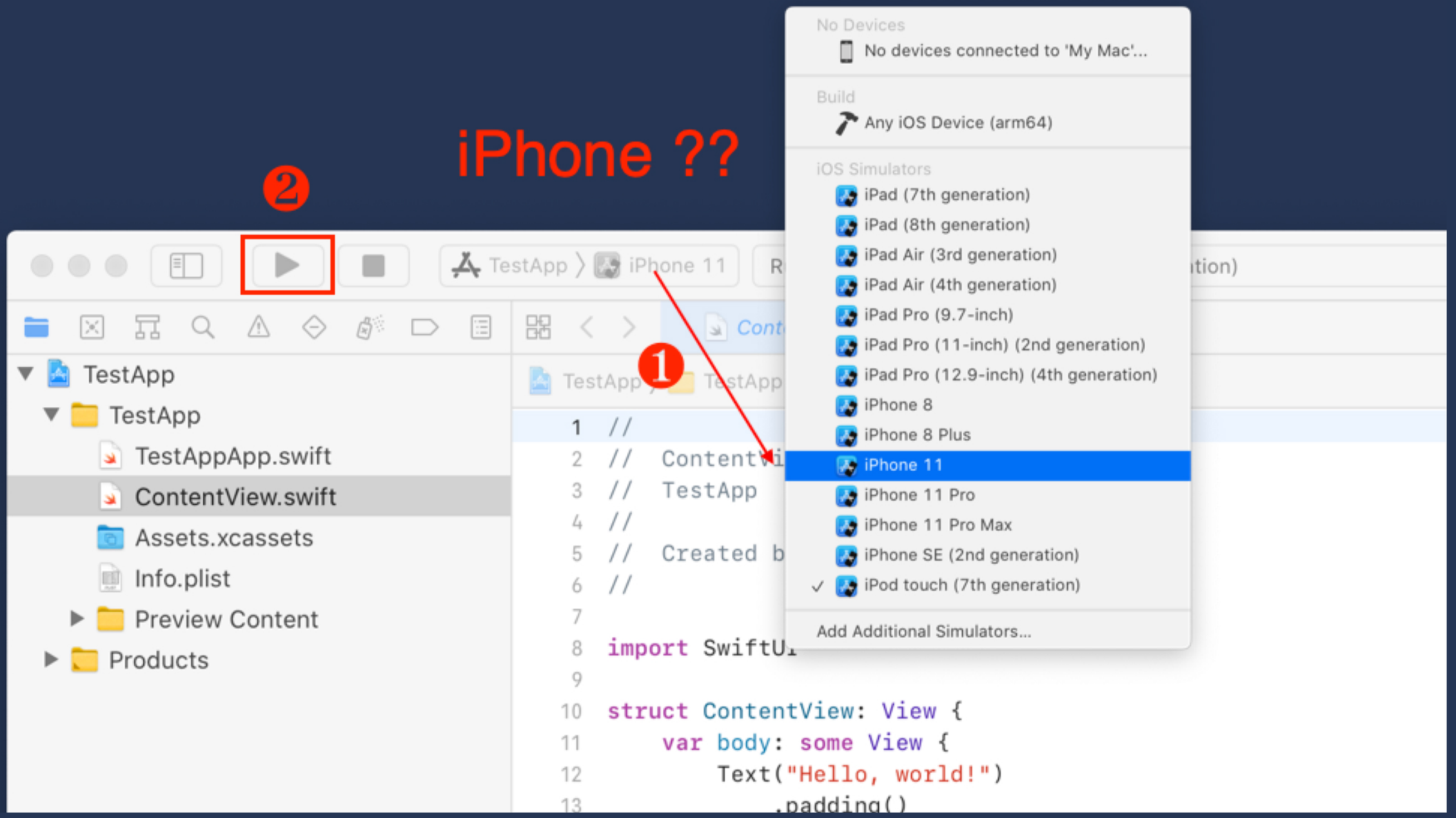
建立第一個應用程式

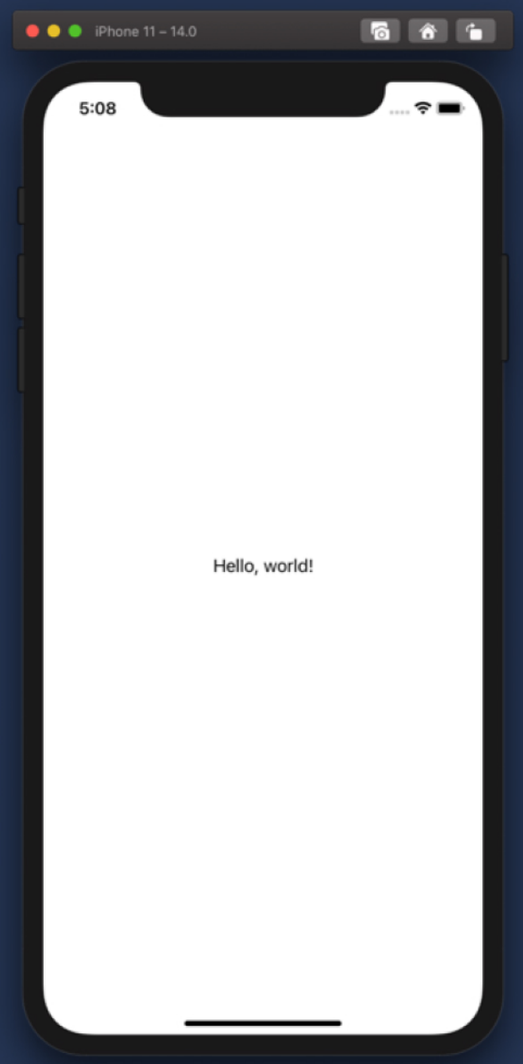

Week1SwiftUI
...opbox/KAG iOS Course/Basic Course 2020
EasyCalculator
...G iOS Course/Basic Course 2020/Examples
Welcome to Xcode
Version 12.0 (12A7209)
Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch, or Apple TV.
Clone an existing project
Start working on something from a Git repository.
Open a project or file
Open an existing project or file on your Mac.


Choose a template for your new project:
Multiplatform
iOS
macOS
watchOS
tvOS
Other
Filter
Application
App
Document App
Game
Augmented Reality App
Sticker Pack App
iMessage App
1
Framework & Library
Framework
Static Library
Metal Library
Cancel
Previous
Next
2
No Selection

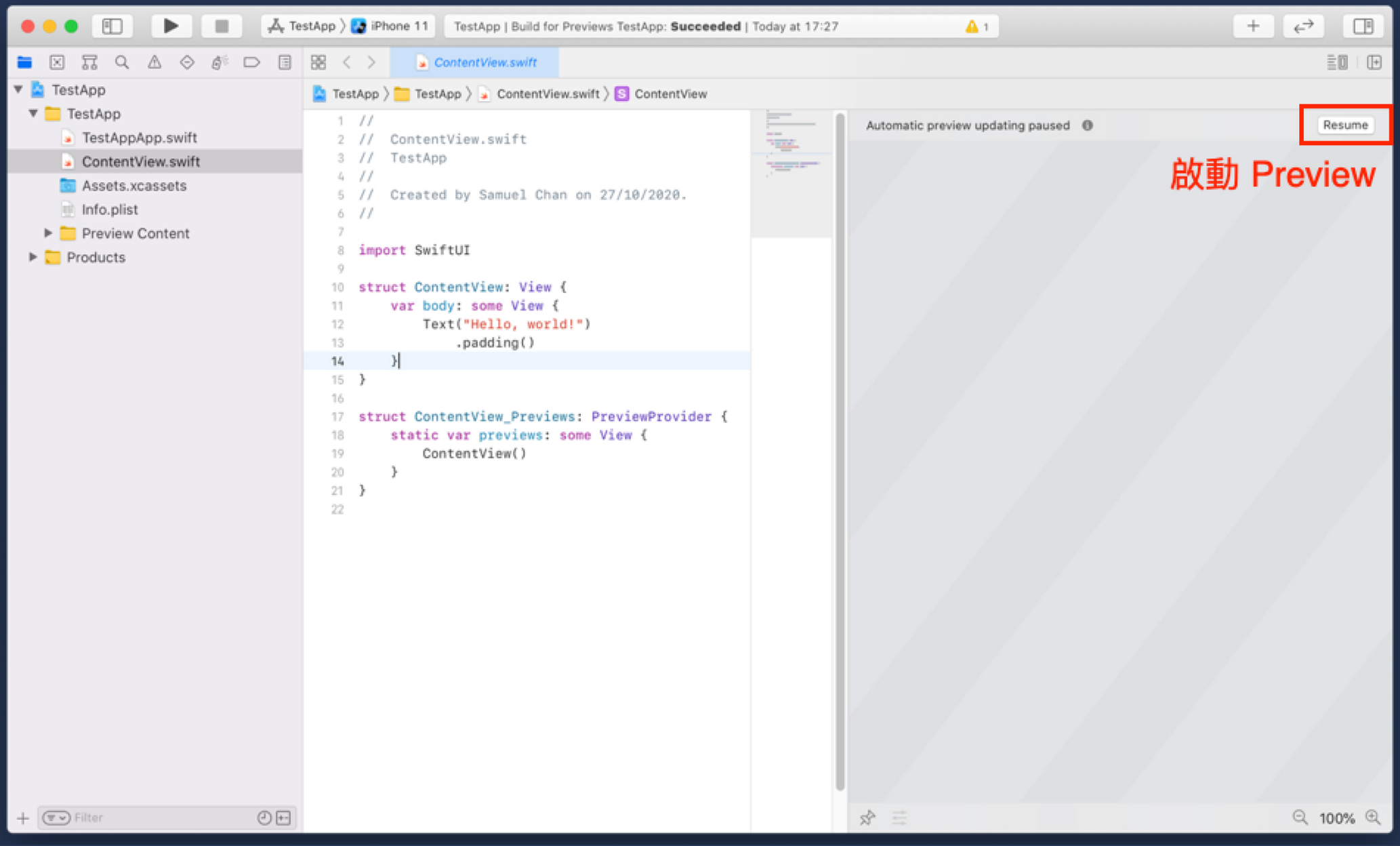

Choose options for your new project:
Product Name: TestApp
1
Team: Add account...
Organization Identifier: edu.pcms.course.samuel
Bundle Identifier: edu.pcms.course.samuel.TestApp
Interface: SwiftUI
Life Cycle: SwiftUI App
Language: Swift
Use Core Data
Host in CloudKit
Include Tests
Cancel
Previous
Next
2
No Selection

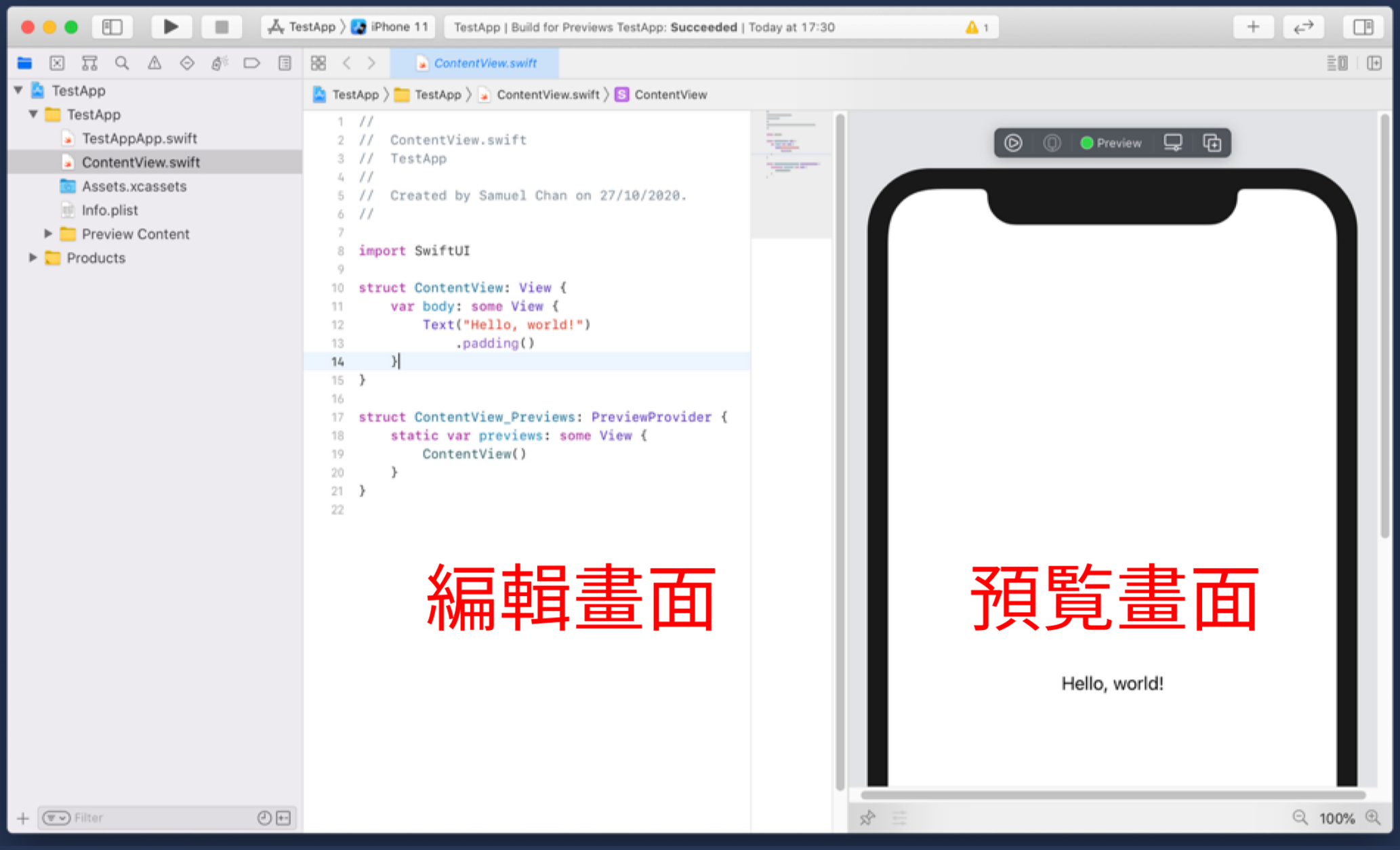

TestApp
iPod touch (7th generation)
Building TestApp: | Constructing build description
1
ContentView.swift
TestApp
TestApp
TestAppApp.swift
ContentView.swift
Assets.xcassets
Info.plist
Preview Content
Products
TestApp > TestApp > ContentView.swift > No Selection
1 //
2 // ContentView.swift
3 // TestApp
4 //
5 // Created by Samuel Chan on 27/10/2020.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11 var body: some View {
12 Text("Hello, world!")
13 .padding()
14 }
15 }
16
17 struct ContentView_Previews: PreviewProvider {
ContentView
body
ContentView_Previews
previews
Automatic preview updating paused
Resume
No Selection
Filter
100%

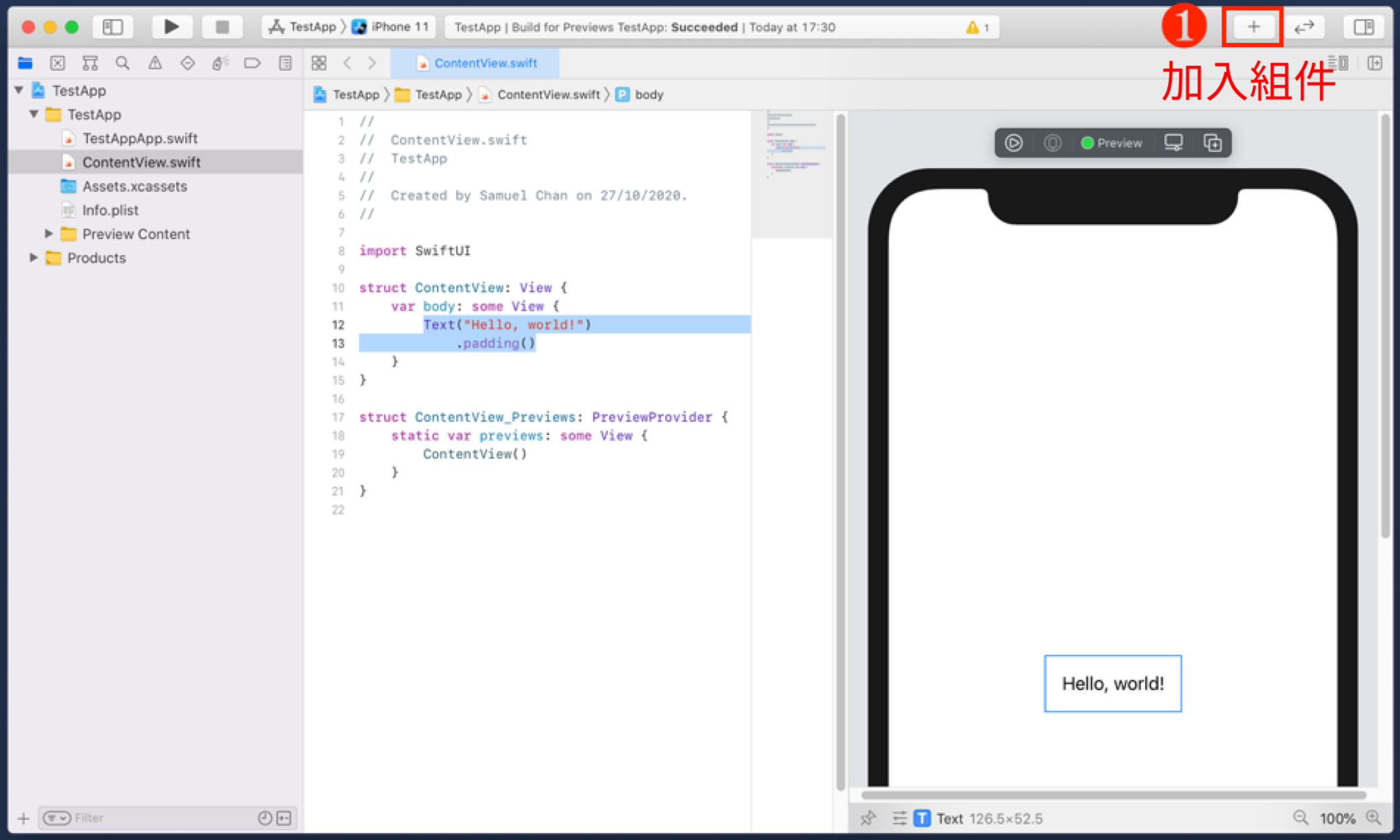

iPhone ??
2
1
TestApp
iPhone 11
TestApp
TestApp
TestAppApp.swift
ContentView.swift
Assets.xcassets
Info.plist
Preview Content
Products
No Devices
No devices connected to 'My Mac'...
Build
Any iOS Device (arm64)
iOS Simulators
iPad (7th generation)
iPad (8th generation)
iPad Air (3rd generation)
iPad 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...
1 //
2 //
3 // TestApp
4 //
5 //
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11 var body: some View {
12 Text("Hello, world!")
13 .padding()


iPhone 11 – 14.0
5:08
Hello, world!

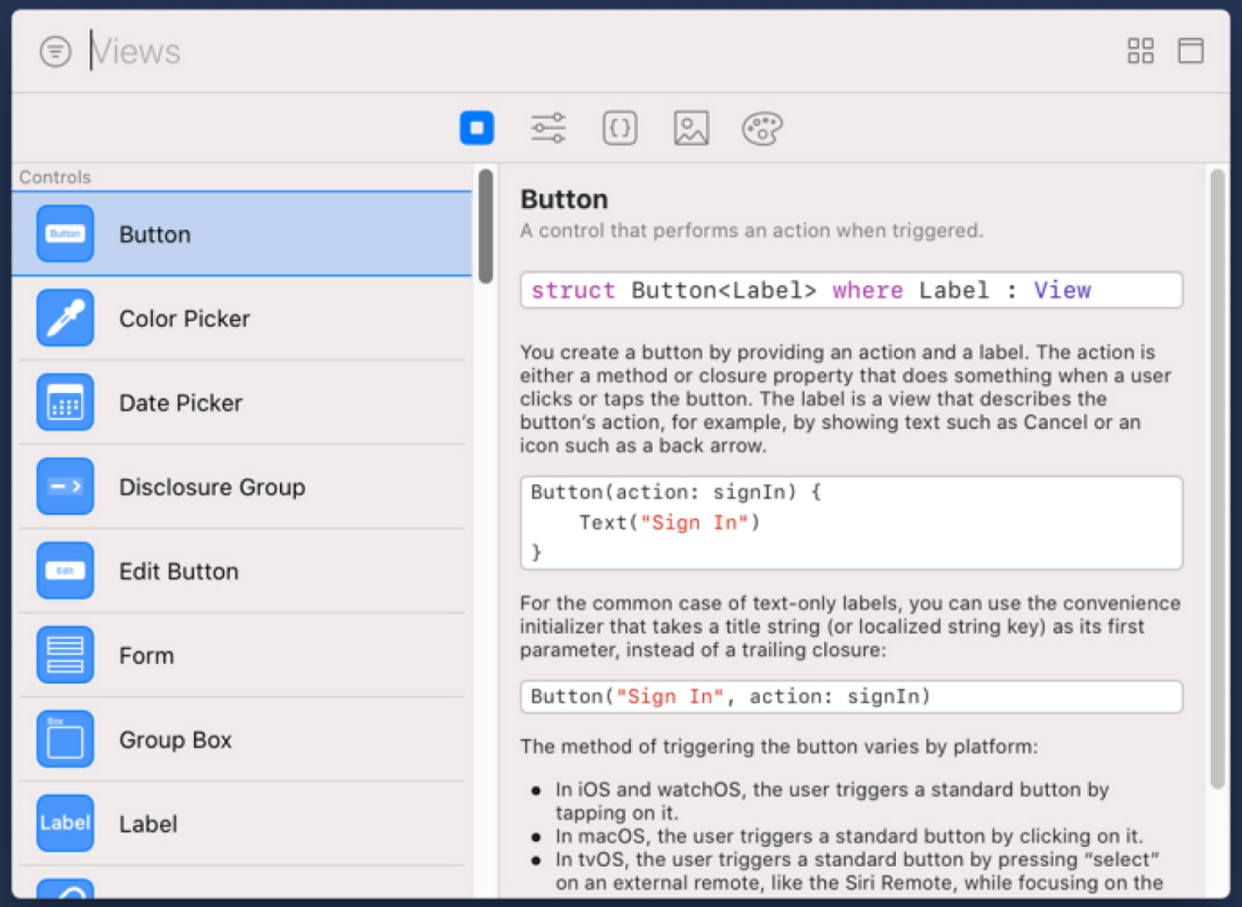
Swift UI


TestApp 〉 iPhone 11
TestApp | Build for Previews TestApp: Succeeded | Today at 17:27
ContentView.swift
TestApp
TestApp
TestAppApp.swift
ContentView.swift
Assets.xcassets
Info.plist
Preview Content
Products
TestApp 〉 TestApp 〉 ContentView.swift 〉 ContentView
1 //
2 // ContentView.swift
3 // TestApp
4 //
5 // Created by Samuel Chan on 27/10/2020.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11 var body: some View {
12 Text("Hello, world!")
13 .padding()
14 }
15 }
16
17 struct ContentView_Previews: PreviewProvider {
18 static var previews: some View {
19 ContentView()
20 }
21 }
22
Automatic preview updating paused
Resume
啟動 Preview
Filter
100%


TestApp
iPhone 11
TestApp | Build for Previews TestApp: Succeeded | Today at 17:30
ContentView.swift
TestApp
TestApp
TestAppApp.swift
ContentView.swift
Assets.xcassets
Info.plist
Preview Content
Products
TestApp > TestApp > ContentView.swift > ContentView
//
// ContentView.swift
// TestApp
//
// Created by Samuel Chan on 27/10/2020.
//
import SwiftUI
struct ContentView: View {
var body: some View {
Text("Hello, world!")
.padding()
}
}
struct ContentView_Previews: PreviewProvider {
static var previews: some View {
ContentView()
}
}
編輯畫面
Preview
Hello, world!
預覽畫面
100%




```swift
//
//  ContentView.swift
//  TestApp
//
//  Created by Samuel Chan on 27/10/2020.
//

import SwiftUI

struct ContentView: View {
    var body: some View {
        Text("Hello, world!")
            .padding()
    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```


Views
Controls
Button
Color Picker
Date Picker
Disclosure Group
Edit Button
Form
Group Box
Label
Label
Button
A control that performs an action when triggered.
struct Button<Label> where Label : View
You create a button by providing an action and a label. The action is either a method or closure property that does something when a user clicks or taps the button. The label is a view that describes the button's action, for example, by showing text such as Cancel or an icon such as a back arrow.
Button(action: signIn) {
Text("Sign In")
}
For the common case of text-only labels, you can use the convenience initializer that takes a title string (or localized string key) as its first parameter, instead of a trailing closure:
Button("Sign In", action: signIn)
The method of triggering the button varies by platform:
In iOS and watchOS, the user triggers a standard button by tapping on it.
In macOS, the user triggers a standard button by clicking on it.
In tvOS, the user triggers a standard button by pressing "select" on an external remote, like the Siri Remote, while focusing on the

選取要加入的組件

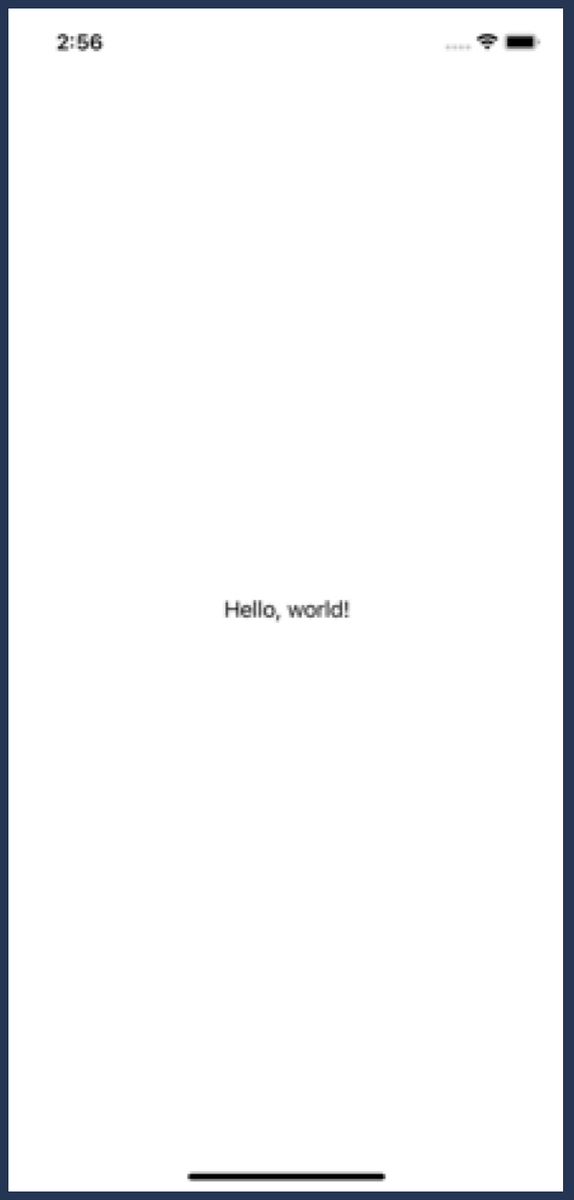
開始加入組件

```
import SwiftUI

struct ContentView: View {
    var body: some View {

        // 在這裏加入內容

    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

Text

TestApp.swift

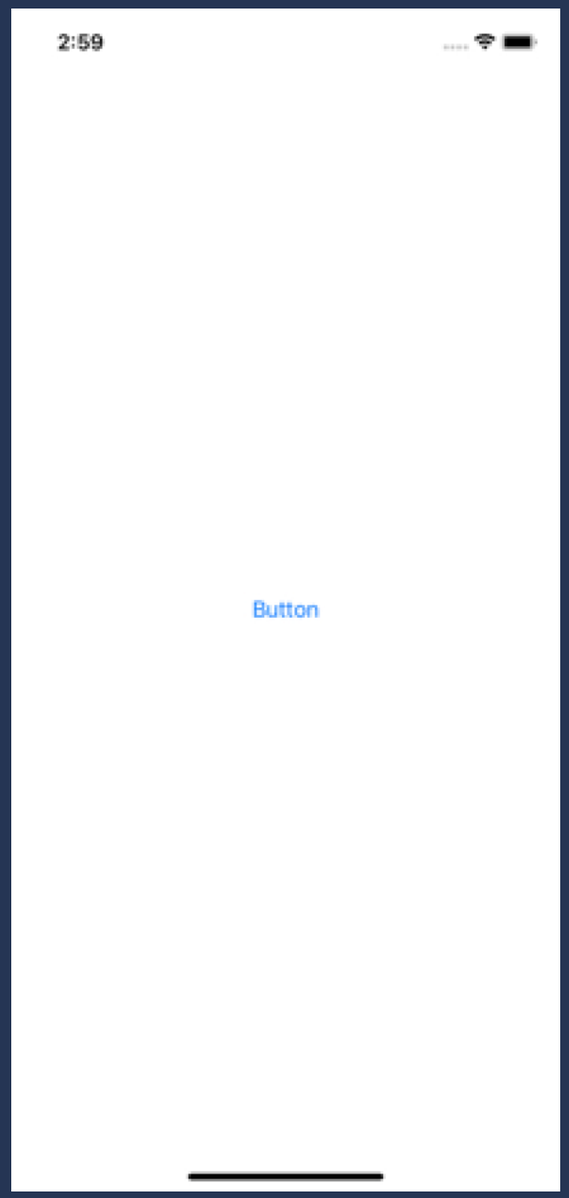
```
import SwiftUI

struct ContentView: View {
    var body: some View {

        Text("Hello, world!")

    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```



Button

TestApp.swift

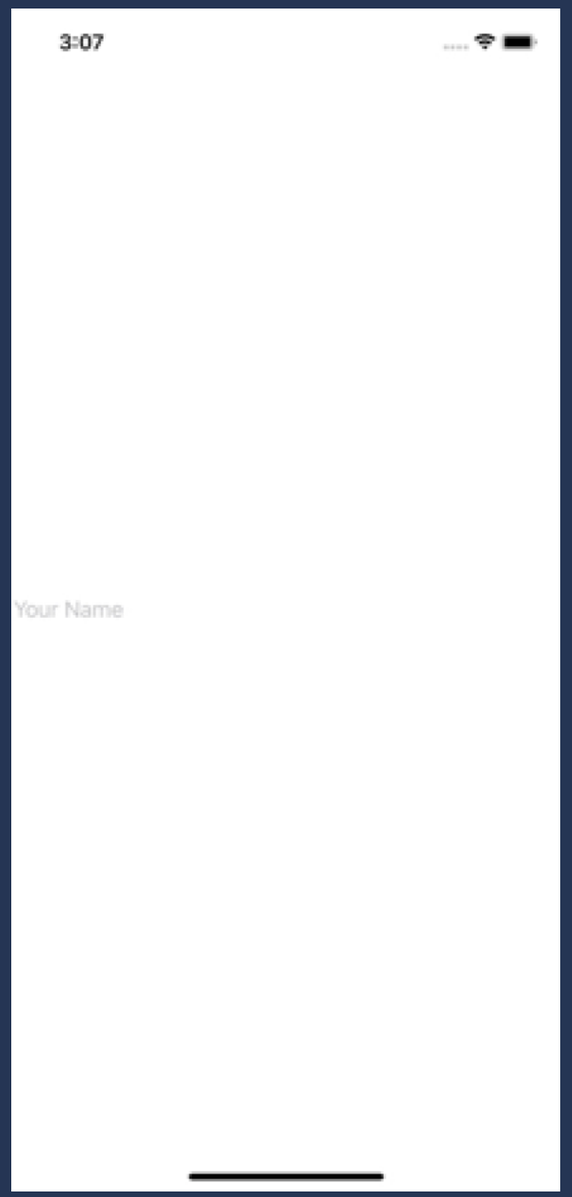
```
import SwiftUI

struct ContentView: View {
    var body: some View {

        Button(action: {}, label: {
            Text("Button")
        })

    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```



TextField

TestApp.swift

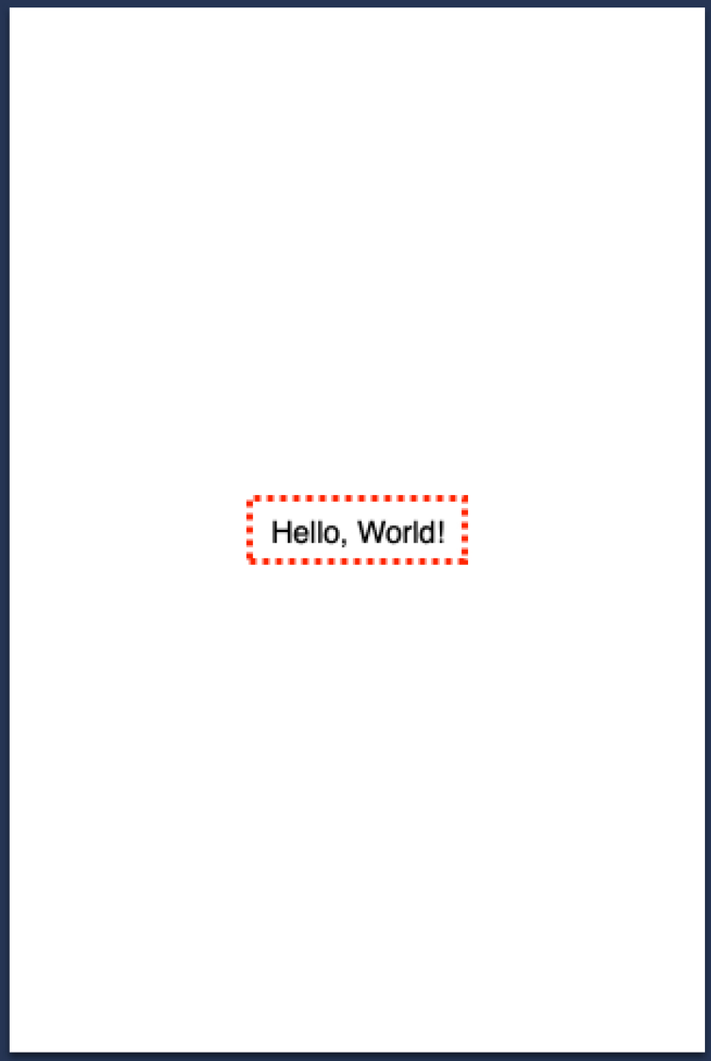
```
import SwiftUI

struct ContentView: View {

    @State var myName:String = ""

    var body: some View {

        TextField("Your Name", text: $myName)

    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

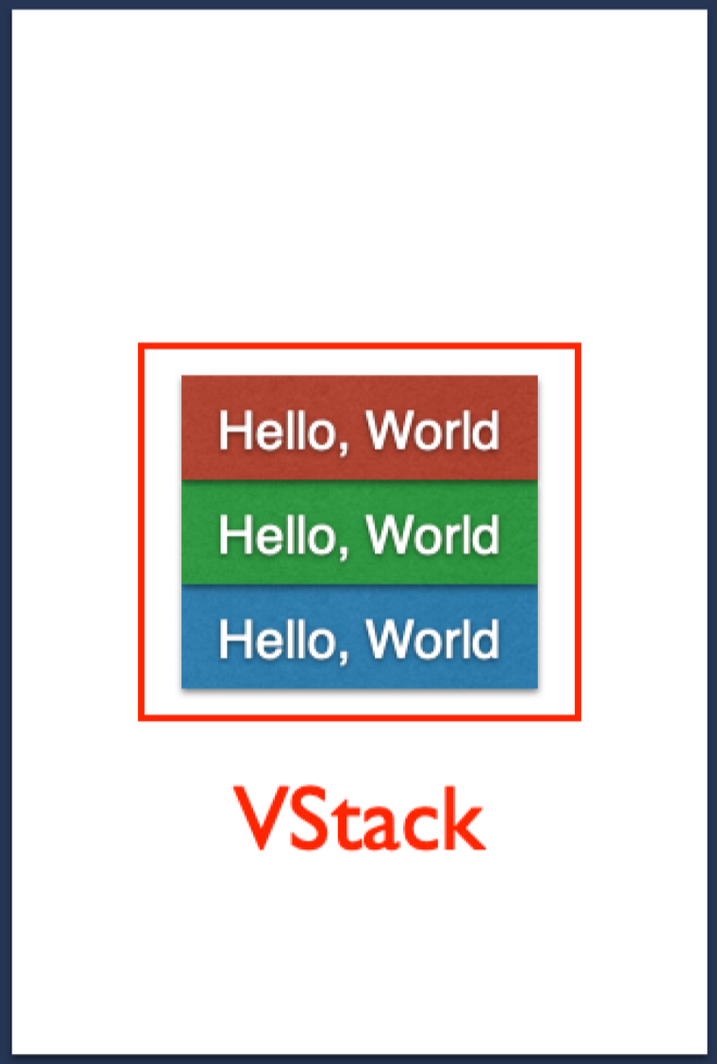


Layout:
HStack, VStack, Spacer

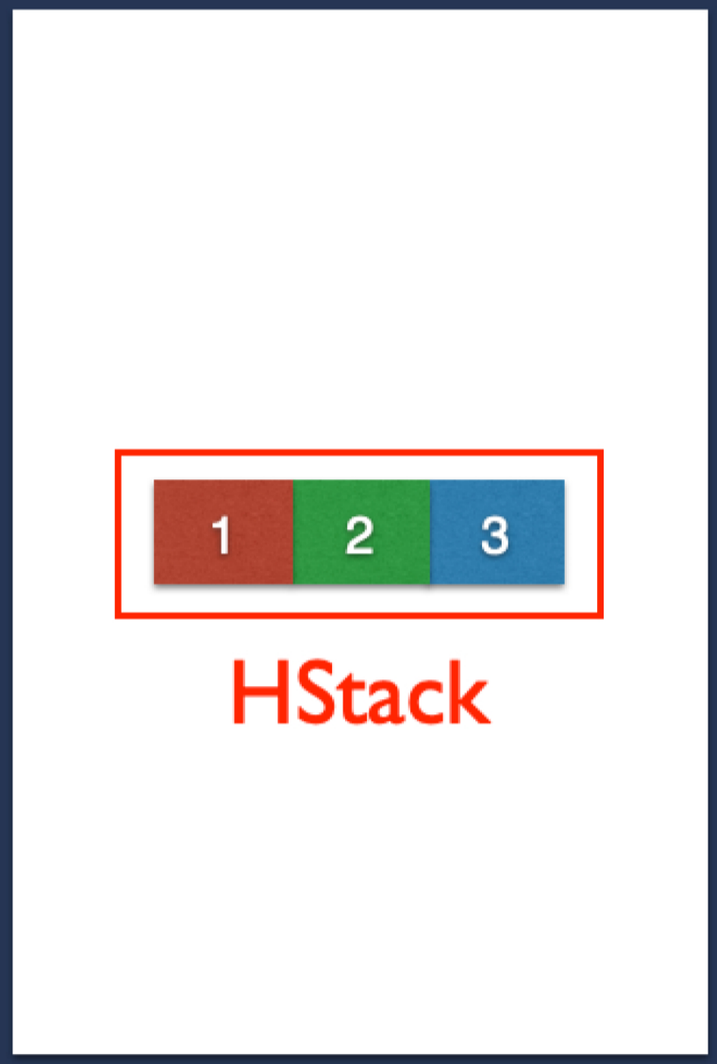
```swift

struct ContentView: View {
    var body: some View {
        Text("Hello, world!")
            .padding()
    }
}

```

一般而言，只容許一個組件存在於 View 內，如需要多個組件，就需要使用 HStack, VStack





V = Vertical 垂直

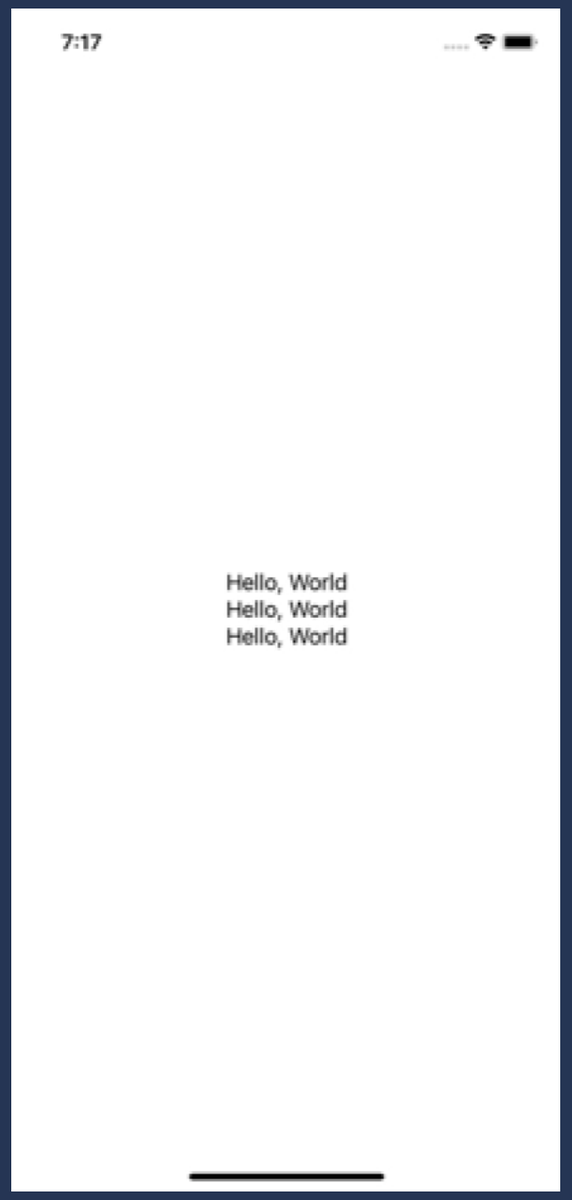


H = Horizontal 横置

VStack

TestApp.swift

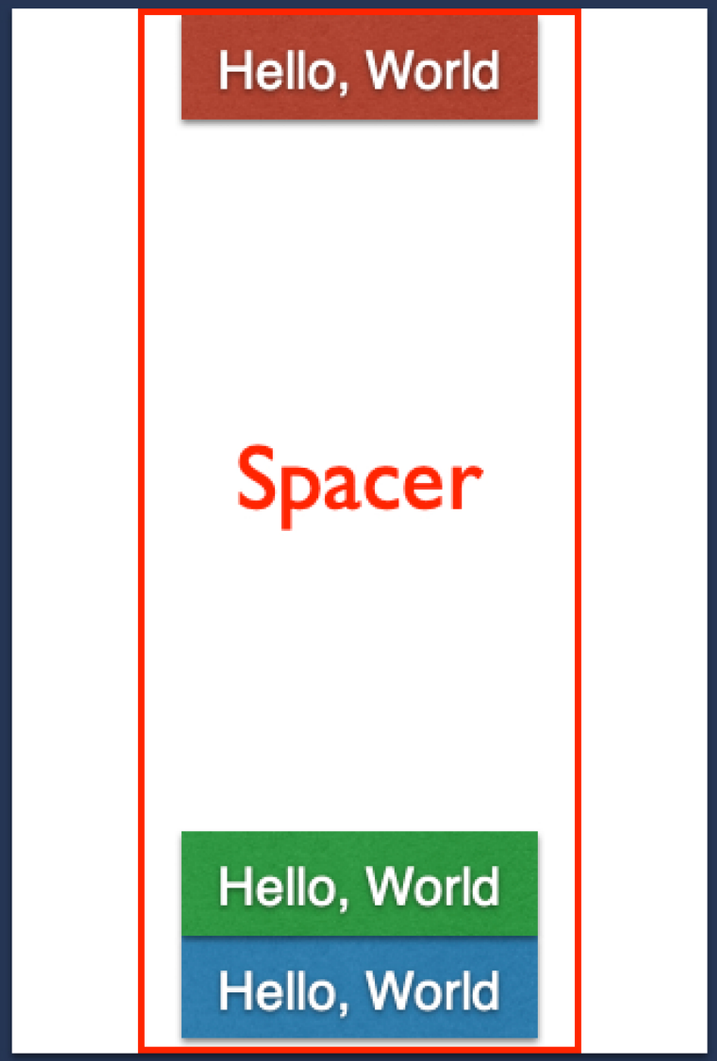
```
import SwiftUI

struct ContentView: View {

    var body: some View {

        VStack {
            Text("Hello, World")
            Text("Hello, World")
            Text("Hello, World")
        }

    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```



HStack

TestApp.swift

```
import SwiftUI

struct ContentView: View {

    var body: some View {

        HStack {
            Text("1")
            Text("2")
            Text("3")
        }

    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```


Hello, World
Spacer
Hello, World
Hello, World


1
2
Spacer
3

Spacer

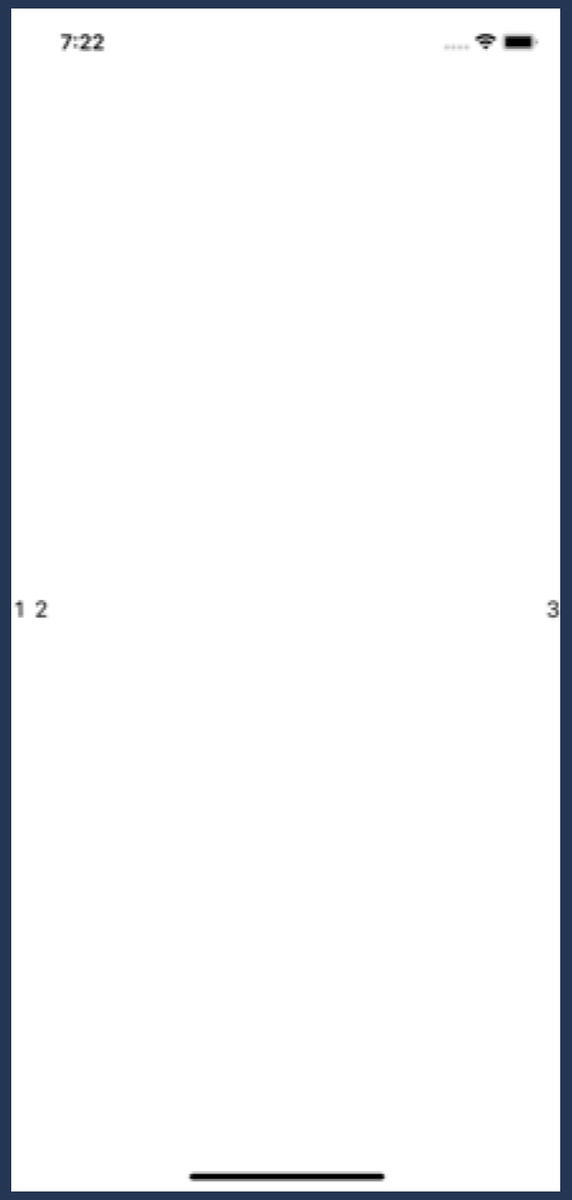
TestApp.swift

```
import SwiftUI

struct ContentView: View {

    var body: some View {

        VStack {
            Text("Hello, World")
            Spacer()
            Text("Hello, World")
            Text("Hello, World")
        }

    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

Spacer

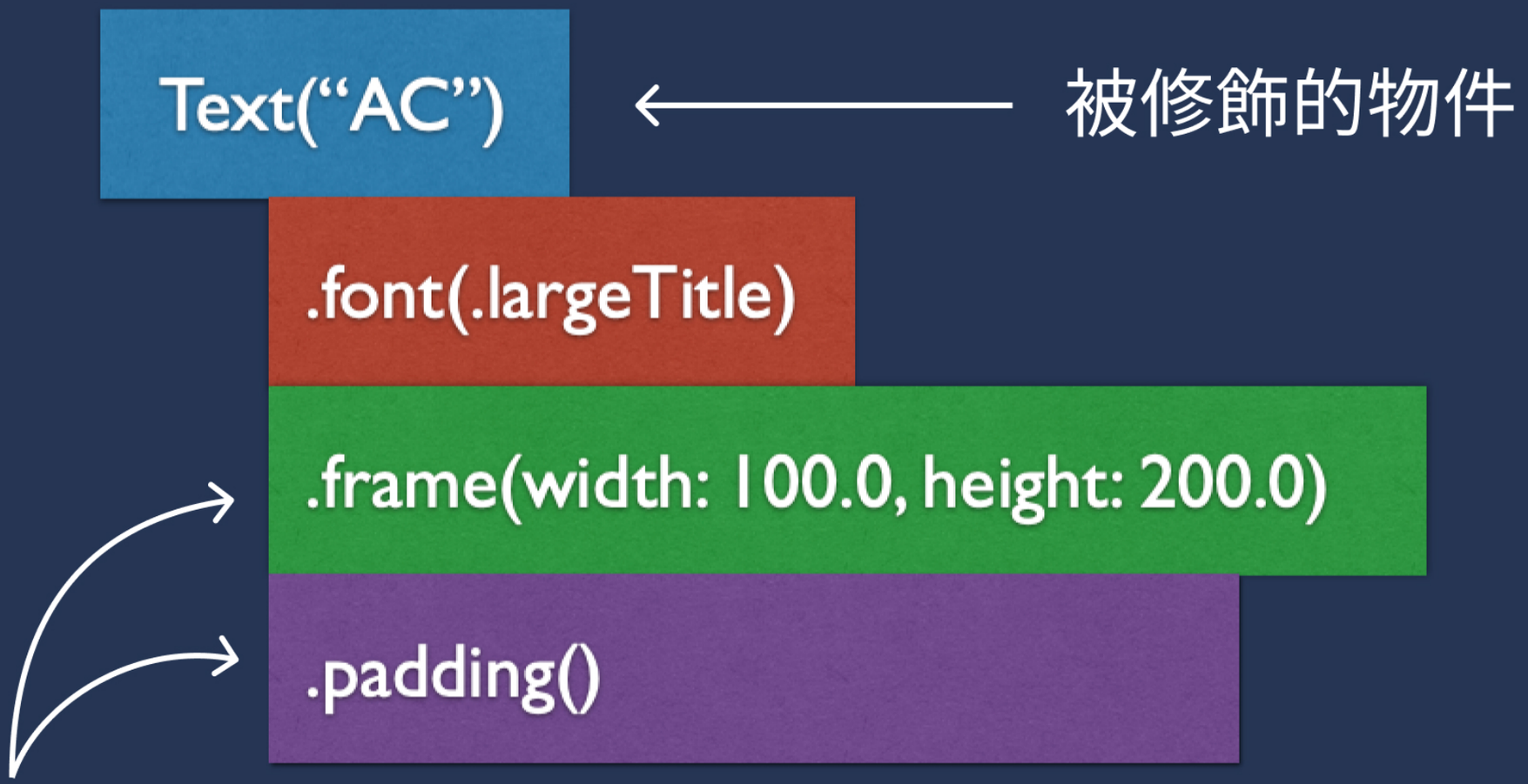
TestApp.swift

```
import SwiftUI

struct ContentView: View {

    var body: some View {

        HStack {
            Text("1")
            Text("2")
            Spacer()
            Text("3")
        }

    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```



Modifiers

用來修飾目標的組件或View



某些 Modifier 有先後次序之分，
不同次序會有不同效果

SF Symbols

Image

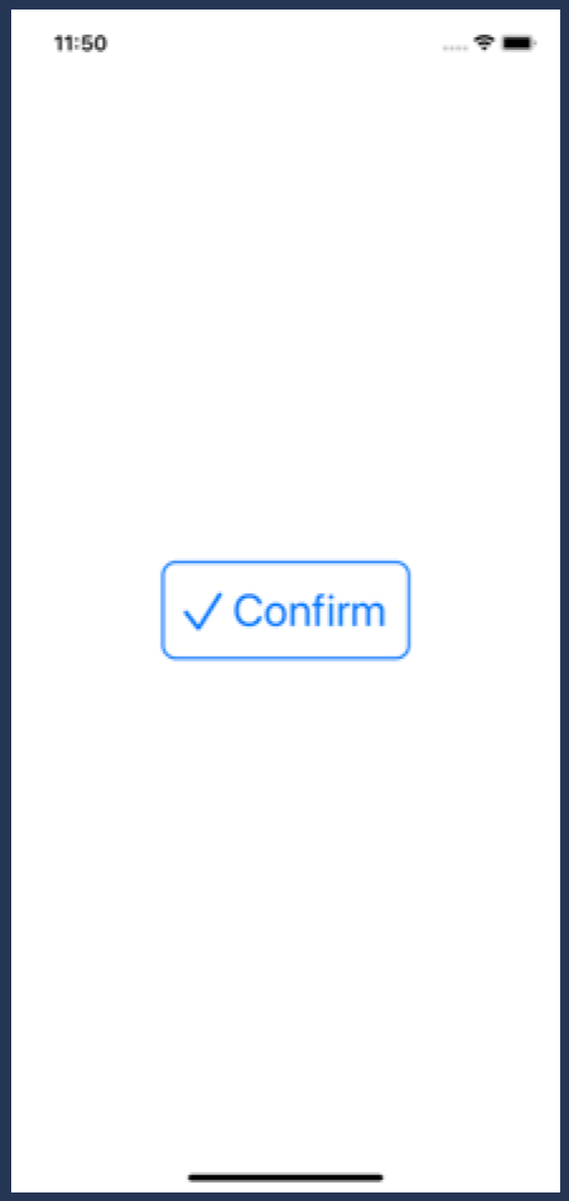
TestApp.swift

```
VStack(alignment: .center, spacing: 20) {

    Image(systemName: "divide")
    Image(systemName: "bus")
    Image(systemName: "heart")
    Image(systemName: "heart.fill")
    Image(systemName: "wind")
    Image(systemName: "wind.snow")


}
.font(.largeTitle)
```

Image Button

TestApp.swift

```
Button(action:{}, label: {
    Label(
        title: { Text("Confirm") },
        icon: { Image(systemName: "checkmark") }
    )
    .font(.largeTitle)
    .padding()
    .overlay(
        RoundedRectangle(cornerRadius: 10.0)
                .stroke(lineWidth: 2.0)
    )
})
```



Swift UI vs UIKit

- Swift UI 是比較新的介面製作方式
- UIKit 是 iOS 一直由 MacOS 年代流傳下來的技術

- Swift UI 重於描述 "結果"
- UIKit 重於描述 "如何做"
- 兩者都是使用 Swift 的語法

Swift UI

- 較易上手
- 只描述 "結果"，程式較為簡潔
- 對於一些常用的排版會較為容易達到
- 採用了 "Reactive" 的概念

UIKit

- 使用 Storyboard 拖拉介面
- 很多時候需要指明 "如何做"，程式較為雜亂
- 使用 Auto Layout 排版，有時需要花很多功夫才能達到所想的版面
- 尤其是要同時支援 iPhone, iPad 時，需要較多工作

如何使用 UIKit 建立一個 App?

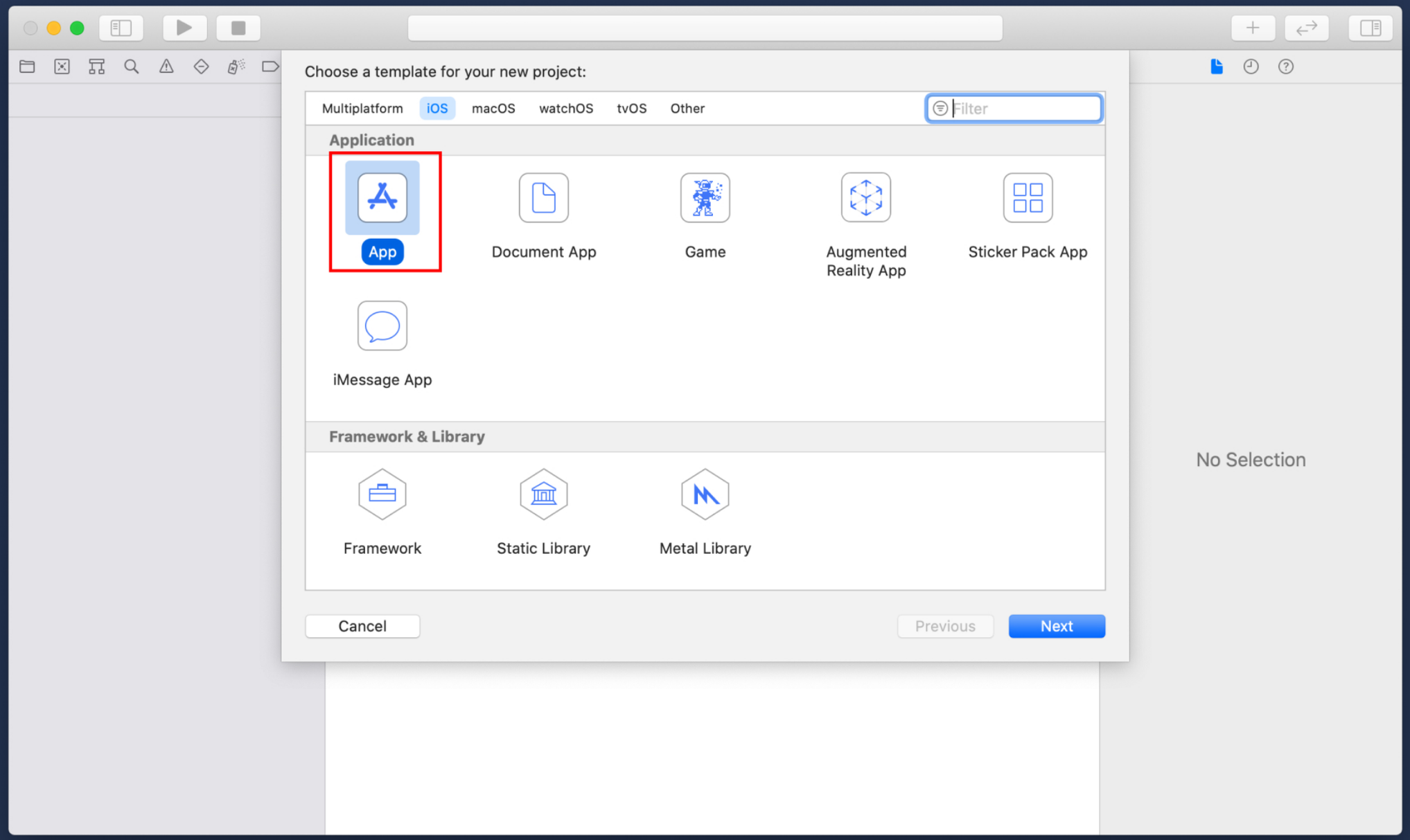
Welcome to Xcode

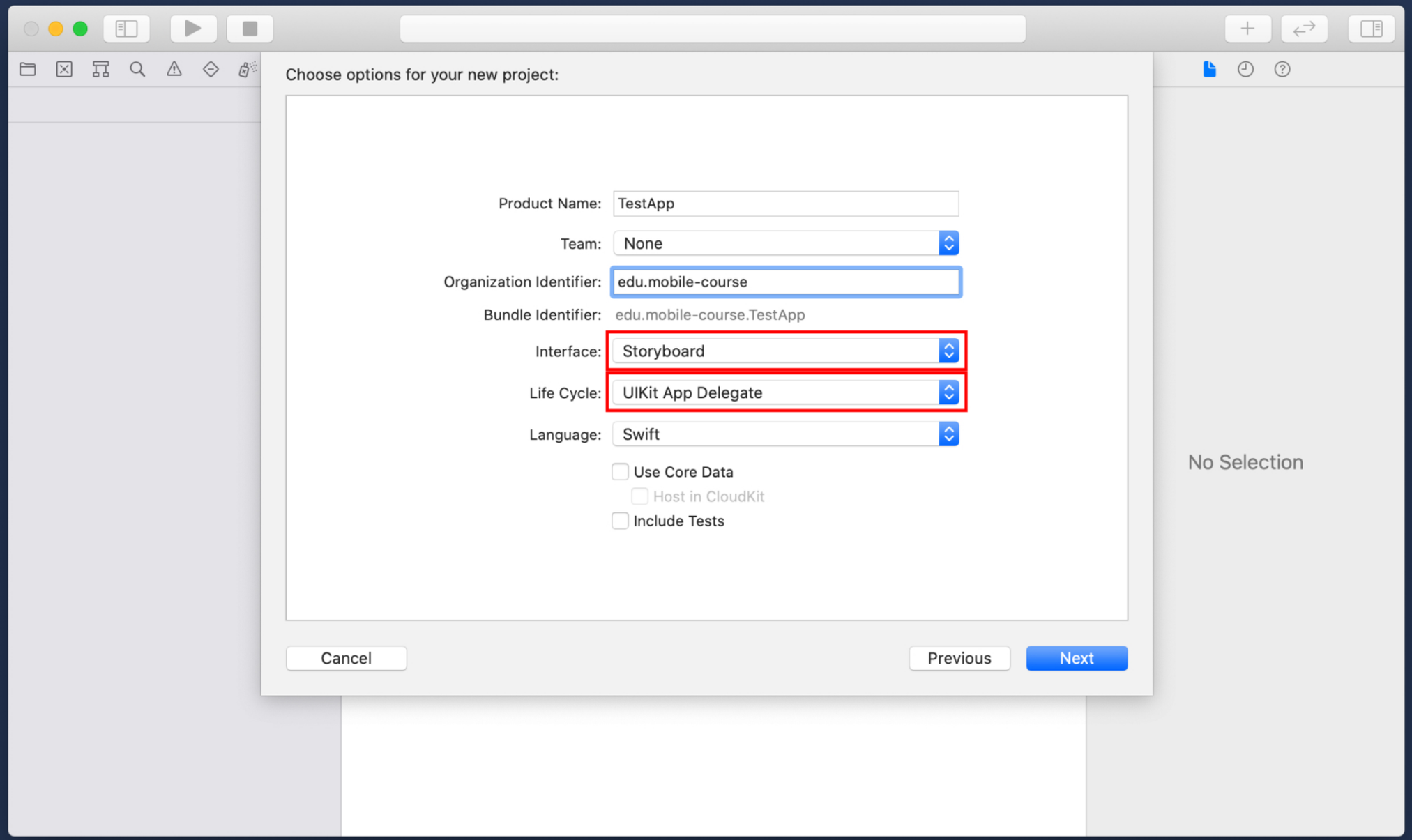
Version 12.4 (12D4e)

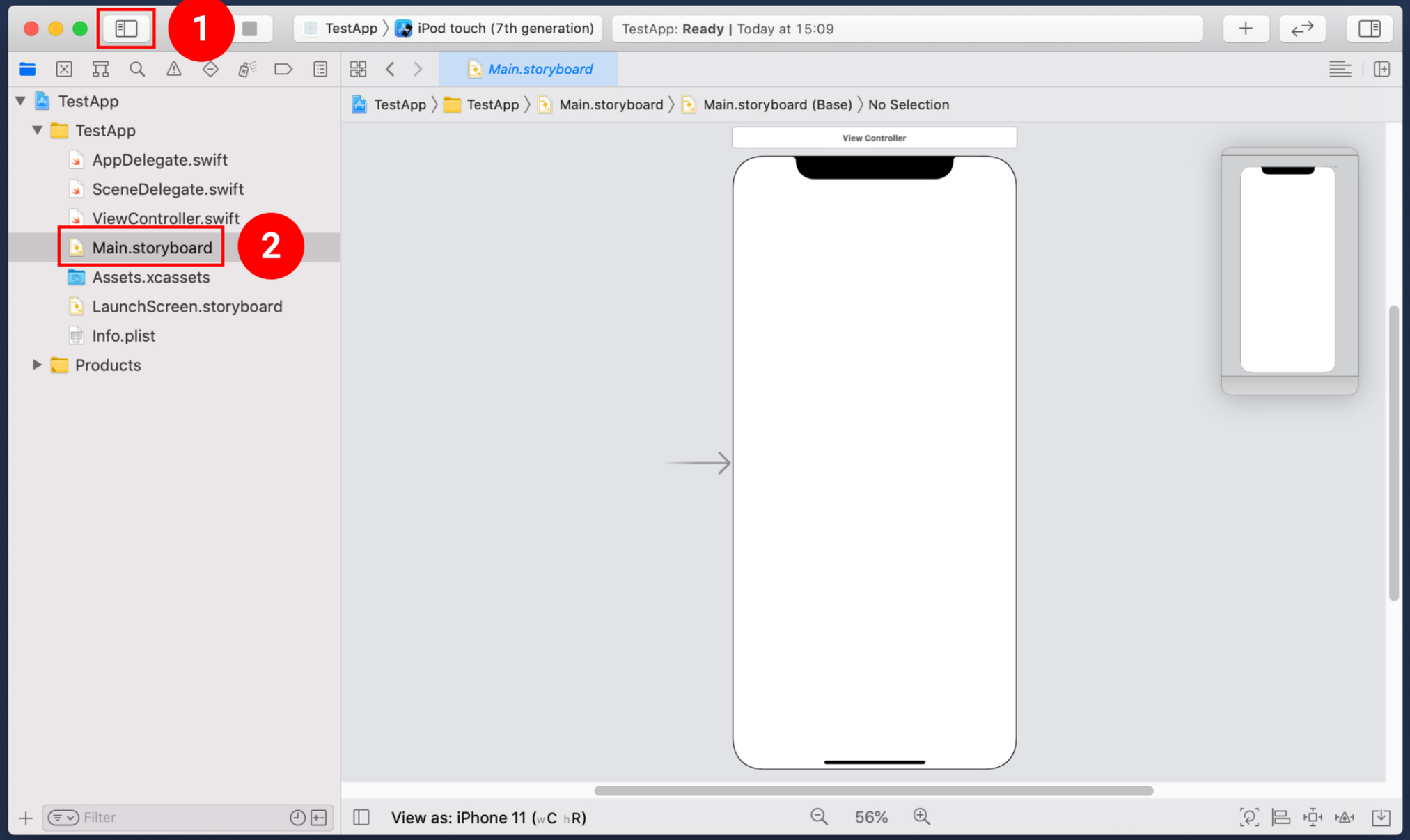
Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch, or Apple TV.

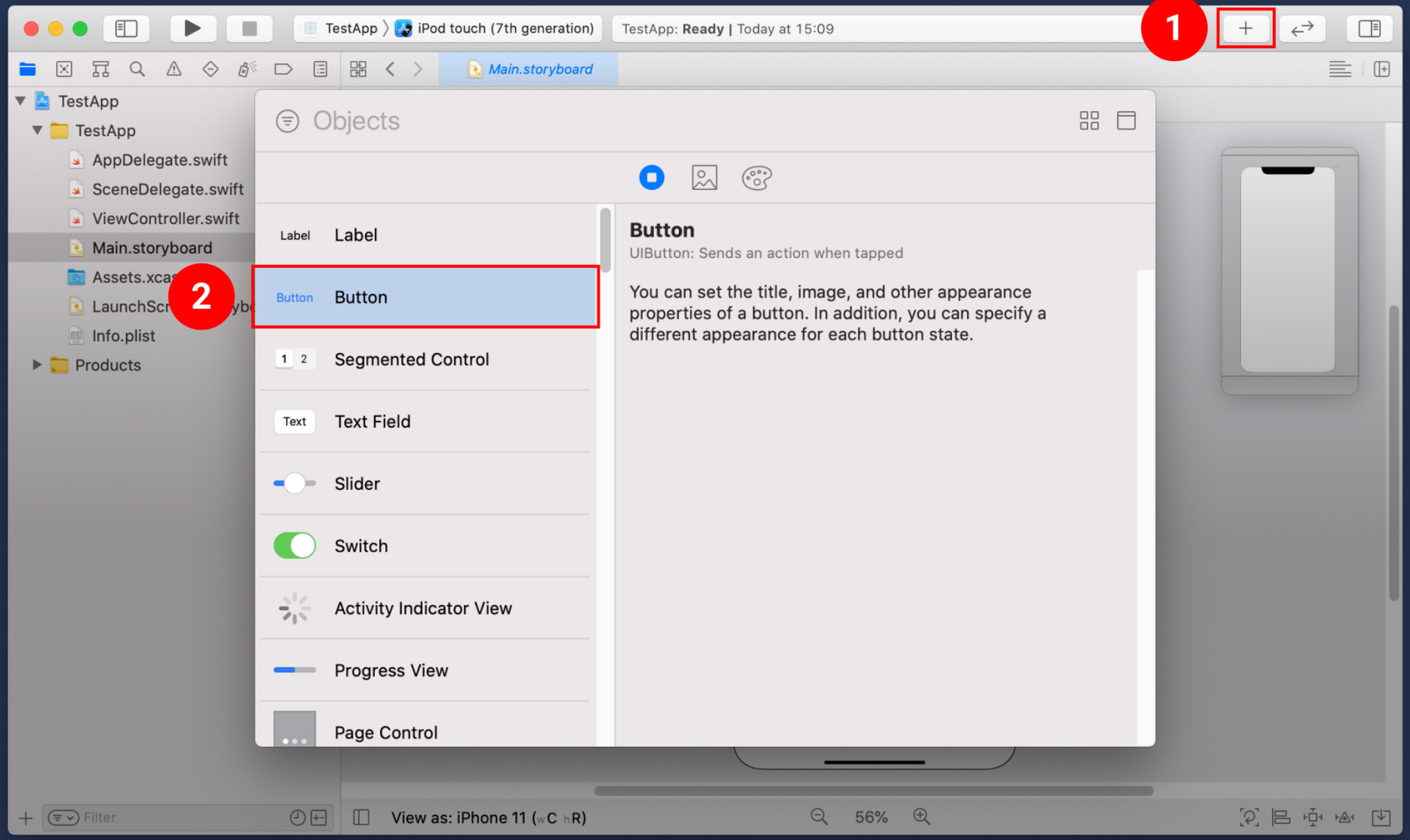
Clone an existing project
Start working on something from a Git repository.

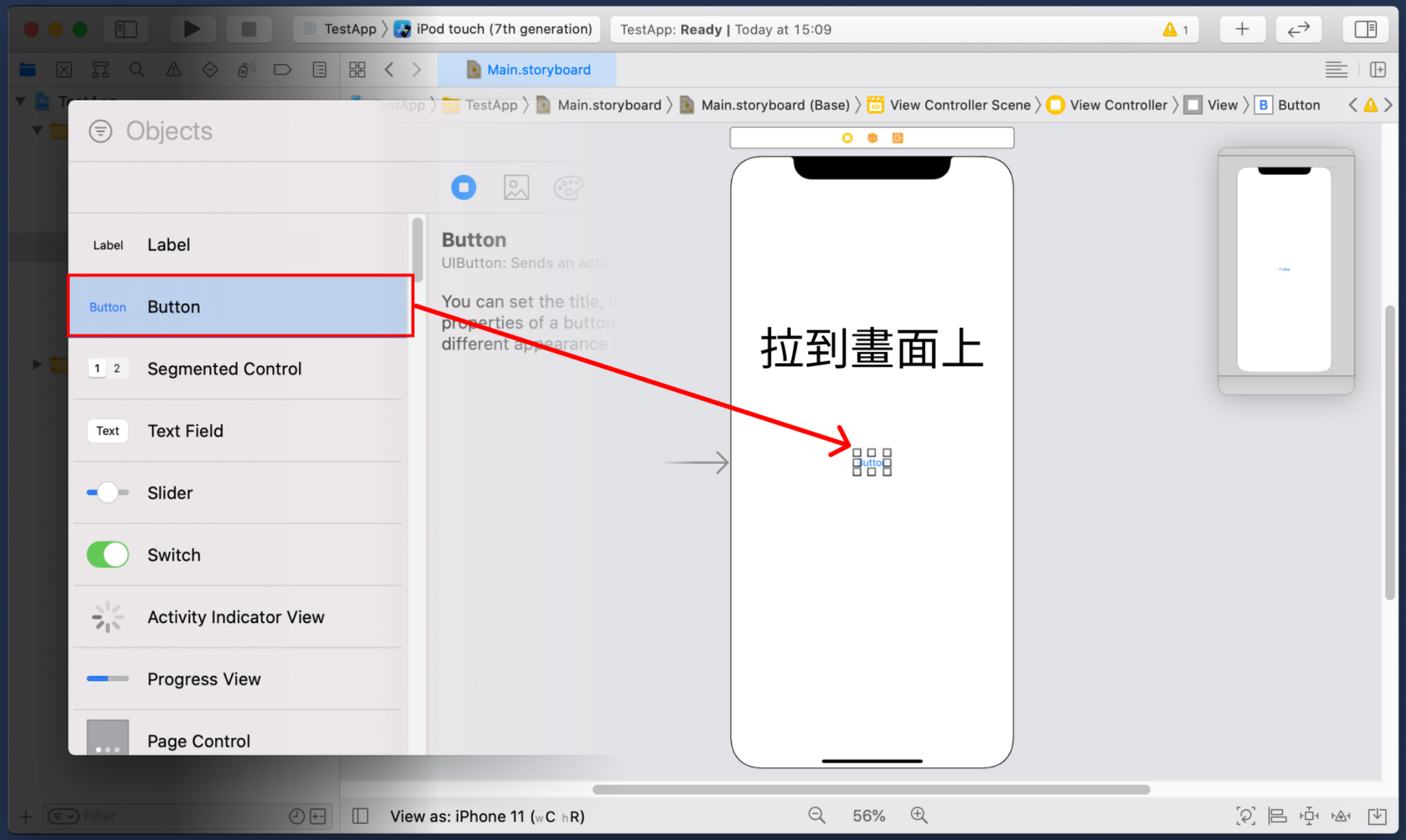
Open a project or file
Open an existing project or file on your Mac.

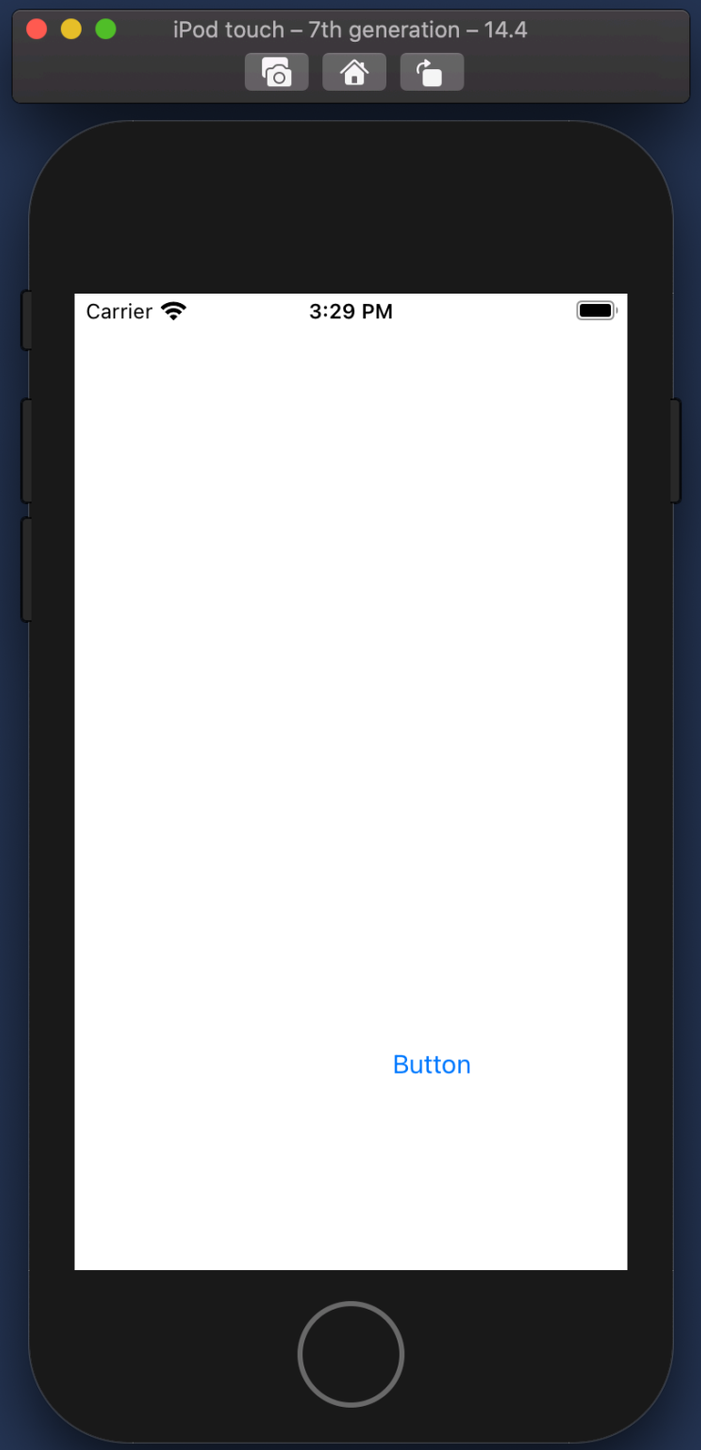

Choose a template for your new project:
Multiplatform
iOS
macOS
watchOS
tvOS
Other
Filter
Application
App
Document App
Game
Augmented Reality App
Sticker Pack App
iMessage App
Framework & Library
Framework
Static Library
Metal Library
Cancel
Previous
Next
No Selection


Choose options for your new project:
Product Name: TestApp
Team: None
Organization Identifier: edu.mobile-course
Bundle Identifier: edu.mobile-course.TestApp
Interface: Storyboard
Life Cycle: UIKit App Delegate
Language: Swift
Use Core Data
Host in CloudKit
Include Tests
Cancel
Previous
Next
No Selection


1
TestApp 〉 iPod touch (7th generation)
TestApp: Ready | Today at 15:09
Main.storyboard
TestApp
TestApp
AppDelegate.swift
SceneDelegate.swift
ViewController.swift
Main.storyboard
2
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
TestApp 〉 TestApp 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
View Controller
Filter
View as: iPhone 11 (wC hR)
56%


TestApp
iPod touch (7th generation)
TestApp: Ready | Today at 15:09
1
Main.storyboard
TestApp
TestApp
AppDelegate.swift
SceneDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcas
LaunchScr
Info.plist
Products
2
Objects
Label
Label
Button
Button
Segmented Control
Text
Text Field
Slider
Switch
Activity Indicator View
Progress View
Page Control
Button
UIButton: Sends an action when tapped
You can set the title, image, and other appearance properties of a button. In addition, you can specify a different appearance for each button state.
Filter
View as: iPhone 11 (wC hR)
56%


TestApp
iPod touch (7th generation)
TestApp: Ready | Today at 15:09
Main.storyboard
TestApp
Main.storyboard
Main.storyboard (Base)
View Controller Scene
View Controller
View
Button
Objects
Label
Label
Button
Button
Segmented Control
Text
Text Field
Slider
Switch
Activity Indicator View
Progress View
Page Control
Button
UIButton: Sends an actio
You can set the title, i
properties of a button
different appearance f
拉到畫面上
Filter
View as: iPhone 11 (wC hR)
56%

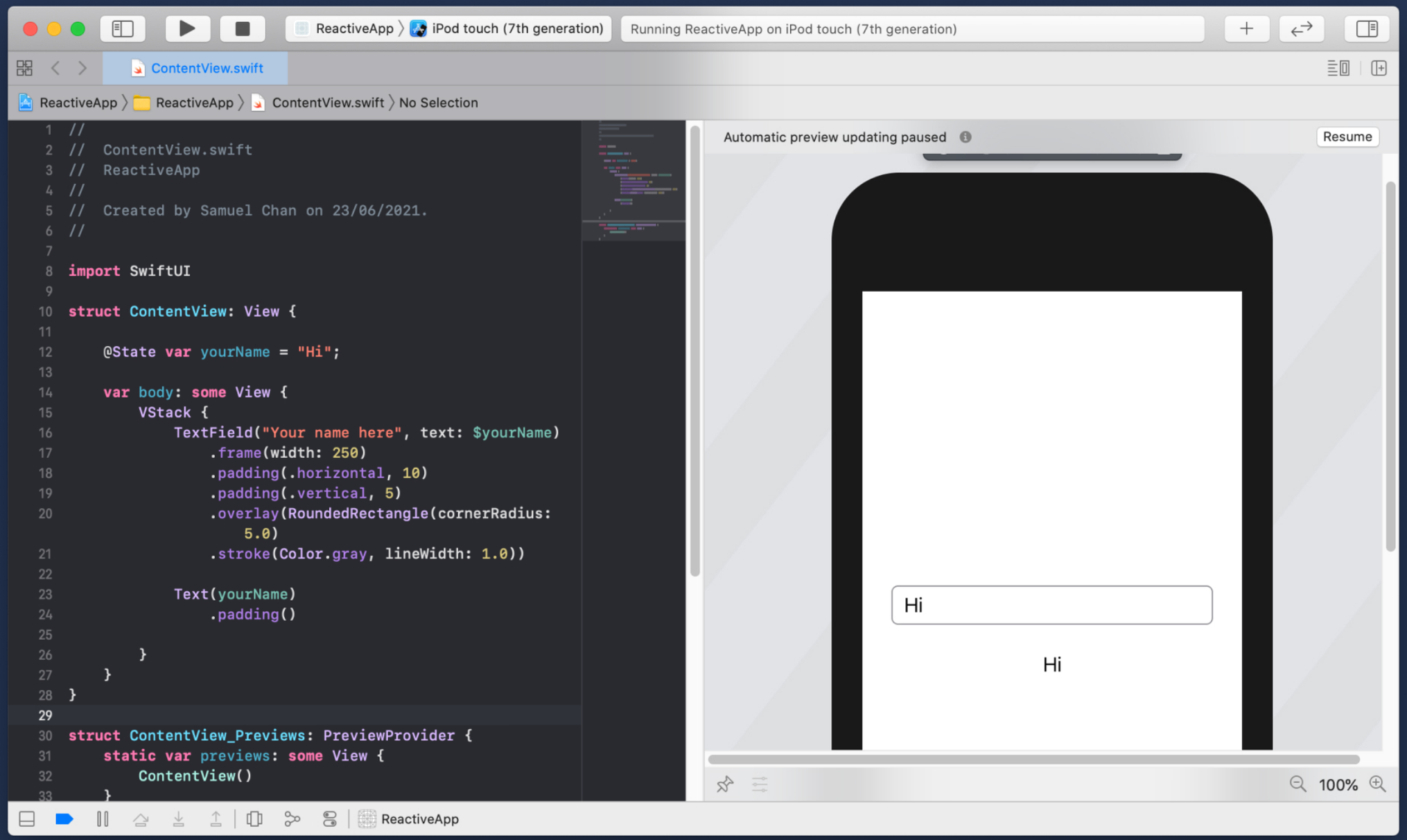

iPod touch – 7th generation – 14.4
Carrier
3:29 PM
Button

加入互動功能
(輸入、計算、輸出)

Swift UI

試試以下例字：

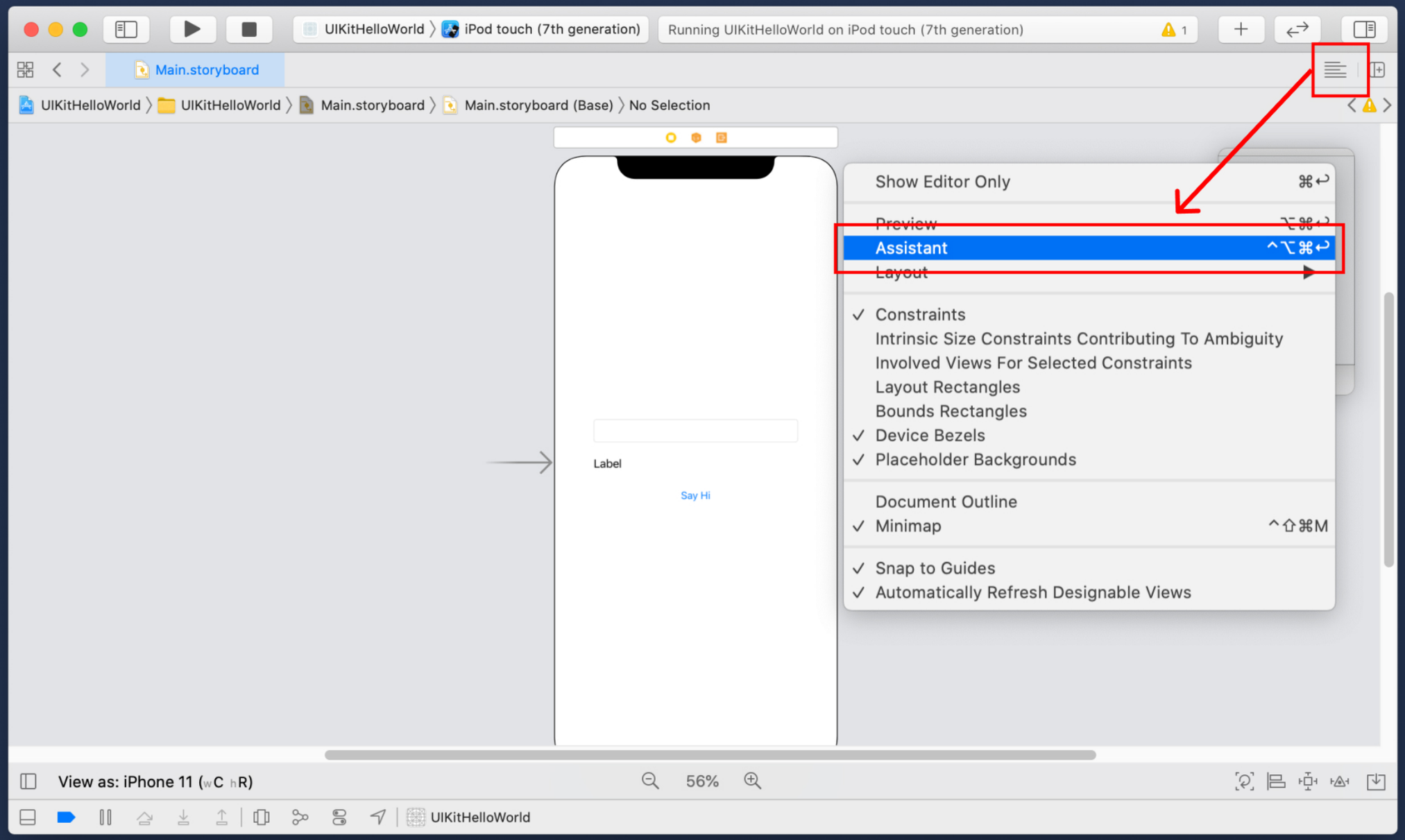
```
...
struct ContentView: View {

    @State var yourName = "Hi";

    var body: some View {
        VStack {
            TextField("Your name here", text: $yourName)
                .frame(width: 250)
                .padding(.horizontal, 10)
                .padding(.vertical, 5)
                .overlay(RoundedRectangle(cornerRadius: 5.0)
                .stroke(Color.gray, lineWidth: 1.0))

            Text(yourName)
                .padding()
        }
    }
}
...
```

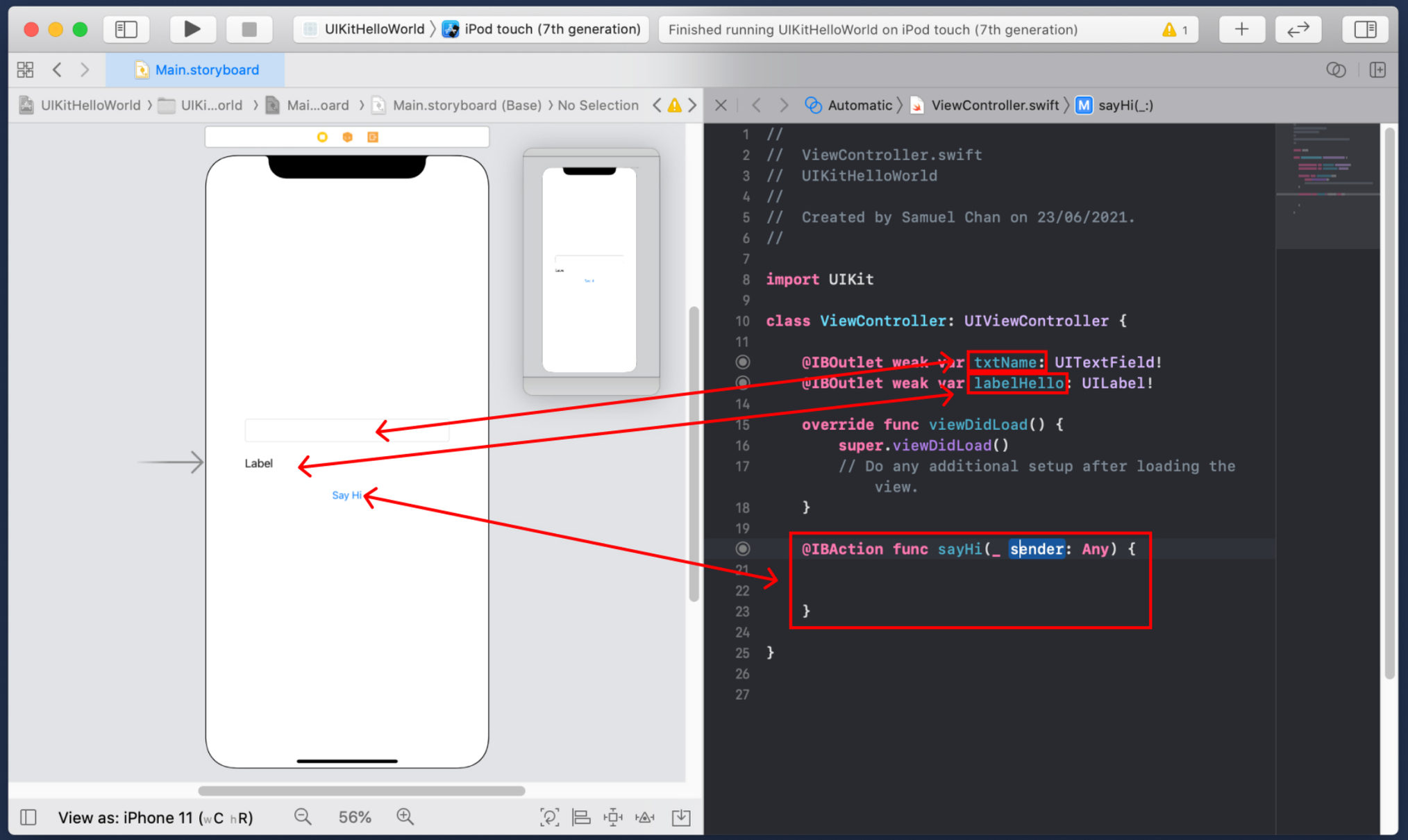

ReactiveApp 〉 iPod touch (7th generation)
Running ReactiveApp on iPod touch (7th generation)
ContentView.swift
ReactiveApp 〉 ReactiveApp 〉 ContentView.swift 〉 No Selection
//
// ContentView.swift
// ReactiveApp
//
// Created by Samuel Chan on 23/06/2021.
//
import SwiftUI
struct ContentView: View {
@State var yourName = "Hi";
var body: some View {
VStack {
TextField("Your name here", text: $yourName)
.frame(width: 250)
.padding(.horizontal, 10)
.padding(.vertical, 5)
.overlay(RoundedRectangle(cornerRadius:
5.0)
.stroke(Color.gray, lineWidth: 1.0))
Text(yourName)
.padding()
}
}
}
struct ContentView_Previews: PreviewProvider {
static var previews: some View {
ContentView()
}
Automatic preview updating paused
Resume
Hi
Hi
100%
ReactiveApp

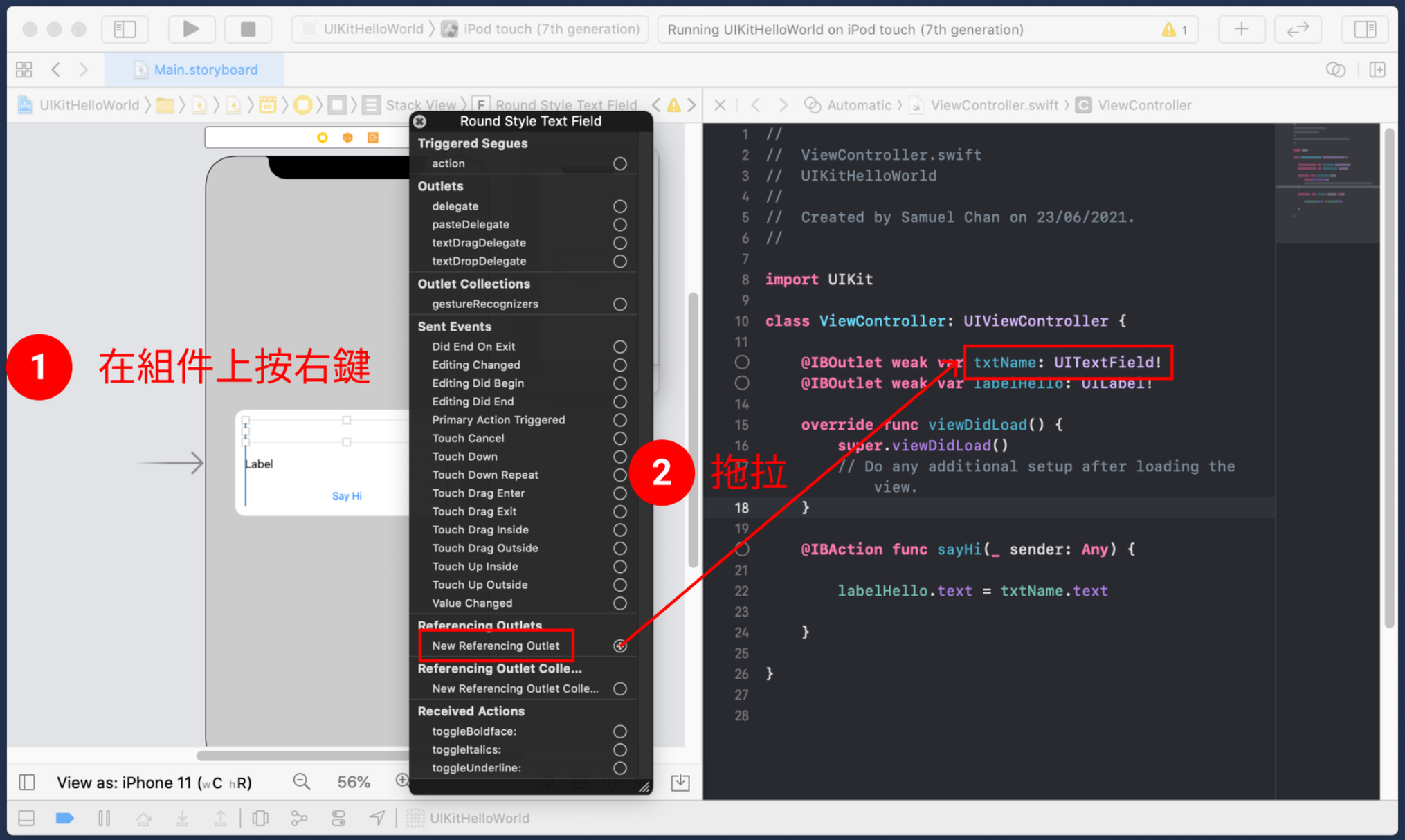

iPod touch – 7th generation – 14.4
Carrier
3:50 PM
Hello, Samuel!
Hello, Samuel!

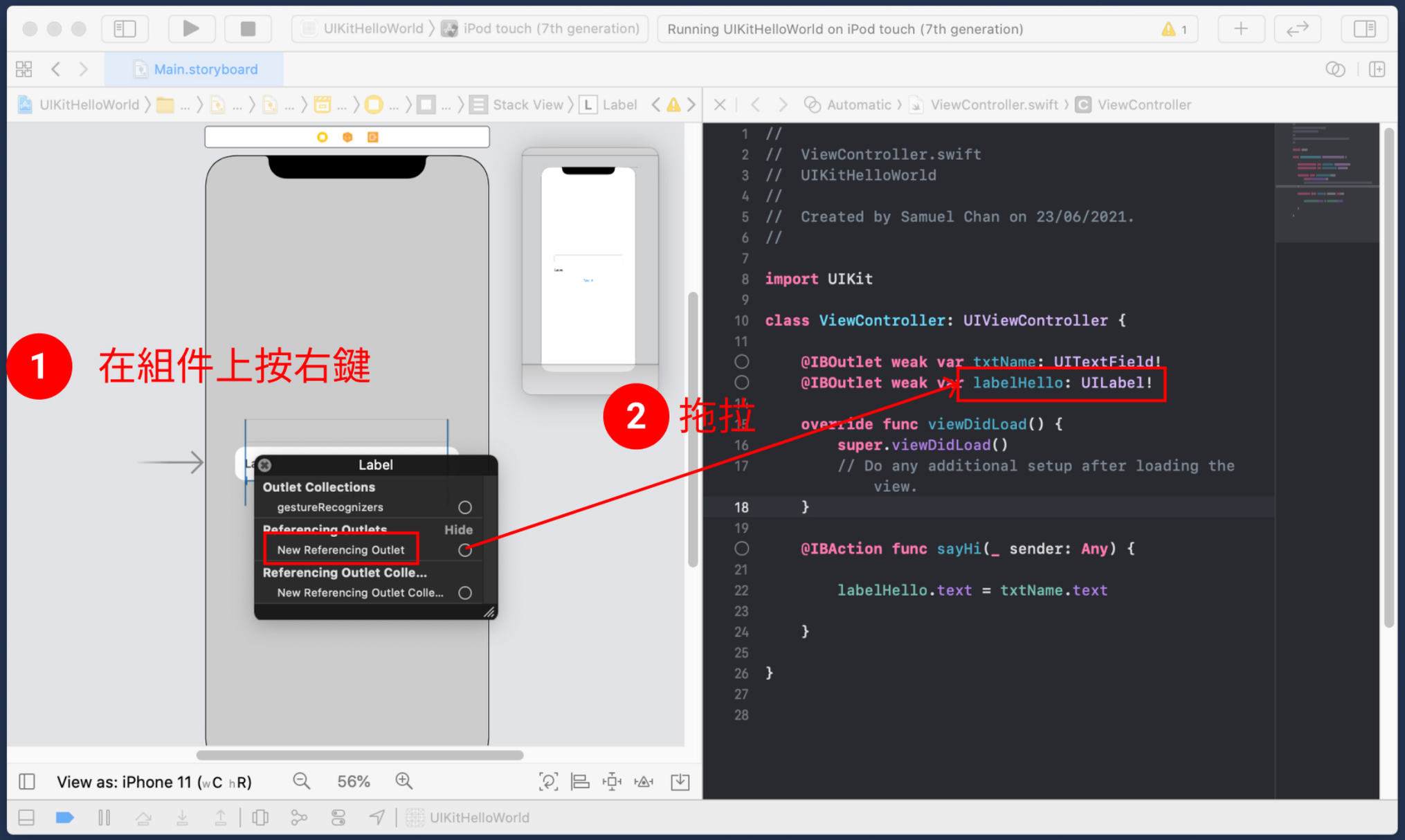
UIKit

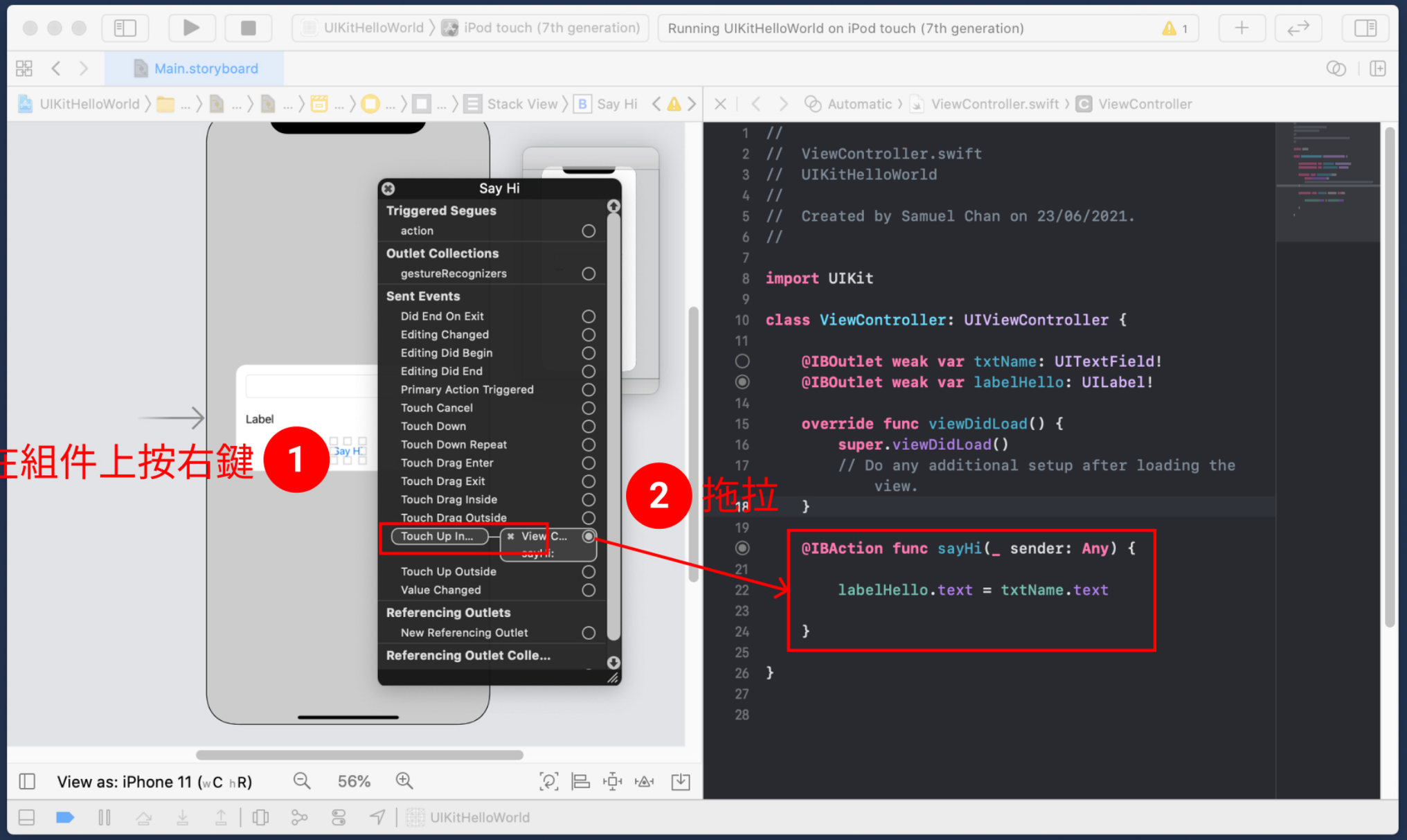
1. 建立介面
2. 將介面與程式連結
3. 編寫程式控制介面


UIKitHelloWorld
iPod touch (7th generation)
Running UIKitHelloWorld on iPod touch (7th generation)
Main.storyboard
UIKitHelloWorld
UIKitHelloWorld
Main.storyboard
Main.storyboard (Base)
No Selection
Label
Say Hi
Show Editor Only
Preview
Assistant
Layout
Constraints
Intrinsic Size Constraints Contributing To Ambiguity
Involved Views For Selected Constraints
Layout Rectangles
Bounds Rectangles
Device Bezels
Placeholder Backgrounds
Document Outline
Minimap
Snap to Guides
Automatically Refresh Designable Views
View as: iPhone 11 (wC hR)
56%
UIKitHelloWorld


UIKitHelloWorld
iPod touch (7th generation)
Finished running UIKitHelloWorld on iPod touch (7th generation)
Main.storyboard
UIKitHelloWorld
UIKi...orld
Mai...oard
Main.storyboard (Base)
No Selection
Automatic
ViewController.swift
sayHi(_:)
Label
Say Hi
View as: iPhone 11 (wC hR)
56%
//
// ViewController.swift
// UIKitHelloWorld
//
// Created by Samuel Chan on 23/06/2021.
//
import UIKit
class ViewController: UIViewController {
@IBOutlet weak var txtName: UITextField!
@IBOutlet weak var labelHello: UILabel!
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view.
}
@IBAction func sayHi(_ sender: Any) {
}
}


UIKitHelloWorld
iPod touch (7th generation)
Running UIKitHelloWorld on iPod touch (7th generation)
Main.storyboard
Stack View
Round Style Text Field
Automatic
ViewController.swift
ViewController
Round Style Text Field
Triggered Segues
action
Outlets
delegate
pasteDelegate
textDragDelegate
textDropDelegate
Outlet Collections
gestureRecognizers
Sent Events
Did End On Exit
Editing Changed
Editing Did Begin
Editing Did End
Primary Action Triggered
Touch Cancel
Touch Down
Touch Down Repeat
Touch Drag Enter
Touch Drag Exit
Touch Drag Inside
Touch Drag Outside
Touch Up Inside
Touch Up Outside
Value Changed
Referencing Outlets
New Referencing Outlet
Referencing Outlet Colle...
New Referencing Outlet Colle...
Received Actions
toggleBoldface:
toggleItalics:
toggleUnderline:
1
在組件上按右鍵
2
拖拉
Label
Say Hi
View as: iPhone 11 (wC hR)
56%
UIKitHelloWorld
//
// ViewController.swift
// UIKitHelloWorld
//
// Created by Samuel Chan on 23/06/2021.
//
import UIKit
class ViewController: UIViewController {
@IBOutlet weak var txtName: UITextField!
@IBOutlet weak var labelHello: UILabel!
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view.
}
@IBAction func sayHi(_ sender: Any) {
labelHello.text = txtName.text
}
}


UIKitHelloWorld
iPod touch (7th generation)
Running UIKitHelloWorld on iPod touch (7th generation)
Main.storyboard
Stack View
Label
Automatic
ViewController.swift
ViewController
1 在組件上按右鍵
2 拖拉
Label
Outlet Collections
gestureRecognizers
Referencing Outlets
Hide
New Referencing Outlet
Referencing Outlet Colle...
New Referencing Outlet Colle...
View as: iPhone 11 (wC hR)
56%
UIKitHelloWorld
//
// ViewController.swift
// UIKitHelloWorld
//
// Created by Samuel Chan on 23/06/2021.
//
import UIKit
class ViewController: UIViewController {
@IBOutlet weak var txtName: UITextField!
@IBOutlet weak var labelHello: UILabel!
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view.
}
@IBAction func sayHi(_ sender: Any) {
labelHello.text = txtName.text
}
}

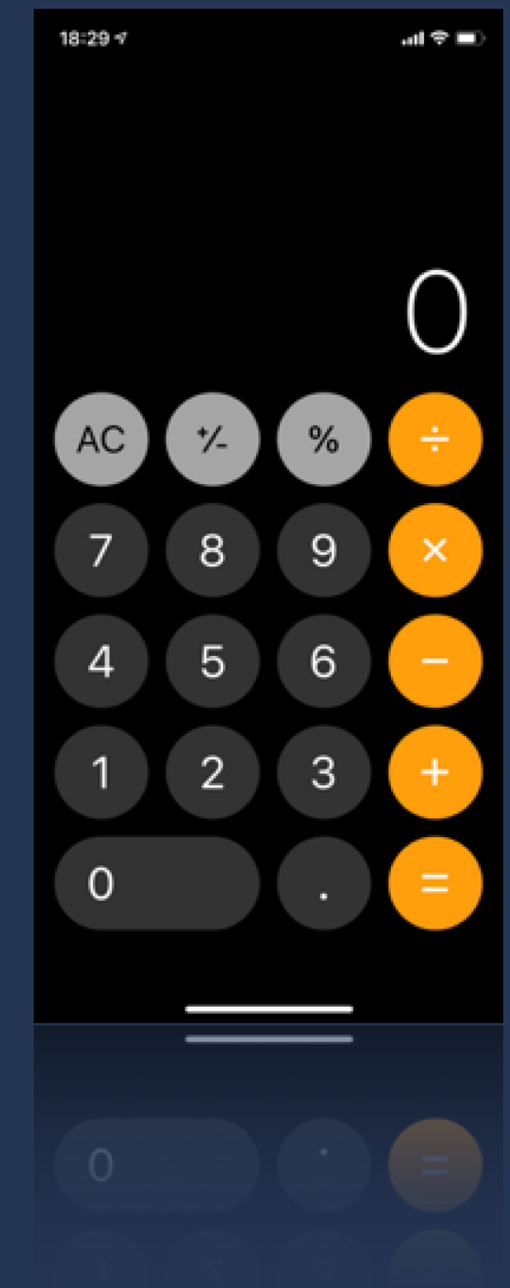

UIKitHelloWorld
iPod touch (7th generation)
Running UIKitHelloWorld on iPod touch (7th generation)
Main.storyboard
Stack View
Say Hi
Automatic
ViewController.swift
ViewController
Label
Say Hi
Triggered Segues
action
Outlet Collections
gestureRecognizers
Sent Events
Did End On Exit
Editing Changed
Editing Did Begin
Editing Did End
Primary Action Triggered
Touch Cancel
Touch Down
Touch Down Repeat
Touch Drag Enter
Touch Drag Exit
Touch Drag Inside
Touch Drag Outside
Touch Up In...
Touch Up Outside
Value Changed
Referencing Outlets
New Referencing Outlet
Referencing Outlet Colle...
在組件上按右鍵
1
2
拖拉
//
// ViewController.swift
// UIKitHelloWorld
//
// Created by Samuel Chan on 23/06/2021.
//
import UIKit
class ViewController: UIViewController {
@IBOutlet weak var txtName: UITextField!
@IBOutlet weak var labelHello: UILabel!
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view.
}
@IBAction func sayHi(_ sender: Any) {
labelHello.text = txtName.text
}
}
View as: iPhone 11 (wC hR)
56%
UIKitHelloWorld

當按下按鈕後將 Label 文字改成 TextField 的內容

```
import UIKit

class ViewController: UIViewController {
...
    @IBAction func sayHi(_ sender: Any) {
        labelHello.text = txtName.text
    }
}
```

課堂作業 Week 1 (計算分數!)

- 嘗試按照 iOS 的內置計算機程式，製作原型 (Mockup) 介面
- 請仔細閱讀下頁要求，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].week1.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/EI4QMXQHWD7sM8xfdPYC
4. 到期日：2021年6月30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 或欠交沒有分數。