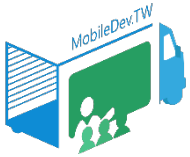
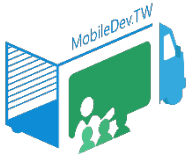


SwiftUI

Ryan Chung

課程大綱

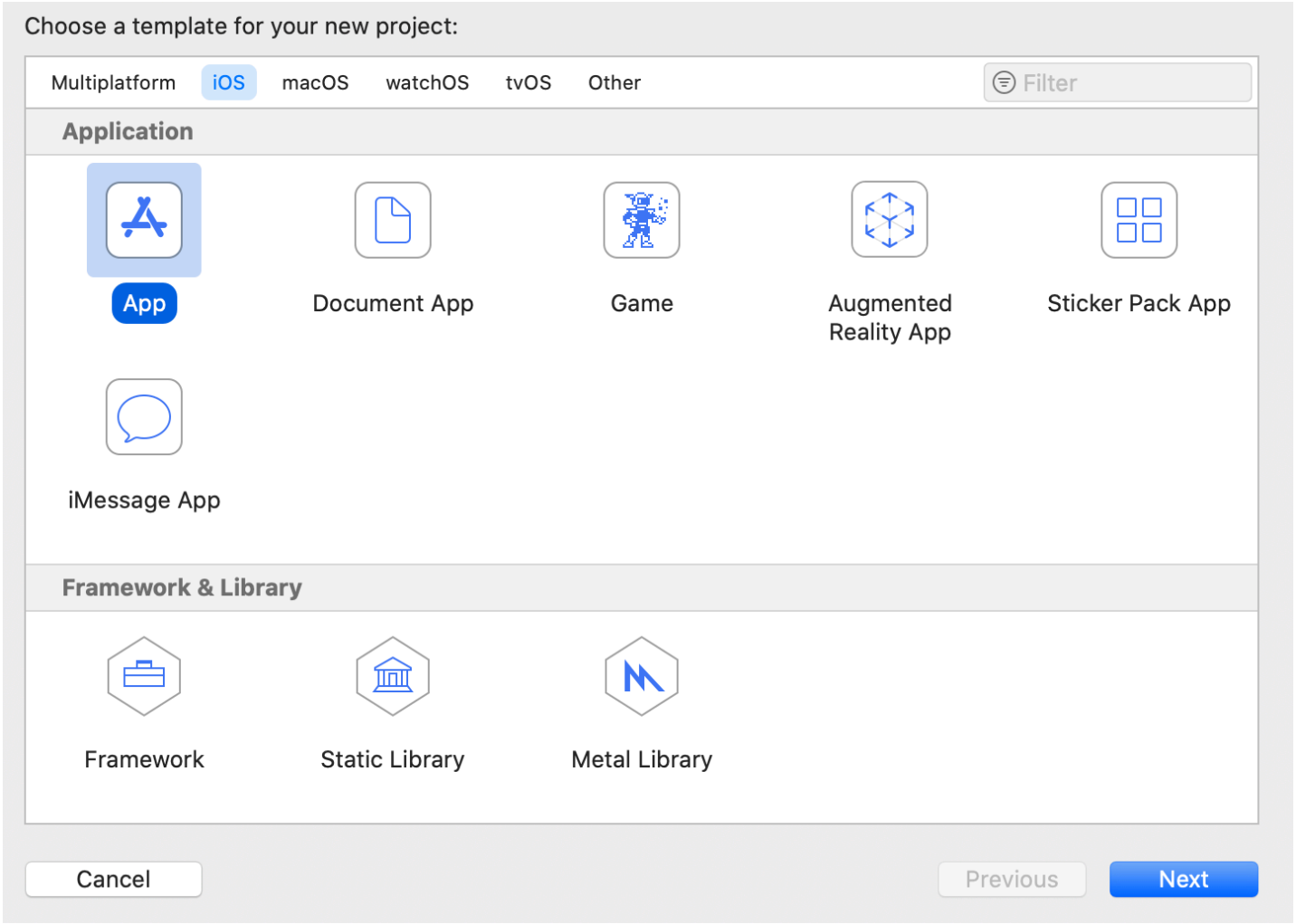
- 文字/圖片呈現
  - 圖文組合 Text, Image
- 按鈕與事件
  - 計數器 Button
  - 剪刀石頭布 Random
- 畫面排版
  - 堆疊 VStack, HStack
  - 冰箱圖文顯示 ZStack
- 清單
  - 餐廳列表 List
  - 個別餐廳介紹 Navigation
  - 跳出頁面 Sheet
- 請求使用者回應
  - 警示視窗 Alert
  - 上拉視窗 ActionSheet
- 設定頁面
  - 條列多擇一 Picker
  - 開關 Toggle
  - 計次 Stepper
  - 滑桿 Slider
- 捲動的世界
  - 課程卡片瀏覽 Scroll
  - 橫向卡片捲動
- 分頁結構



文字處理

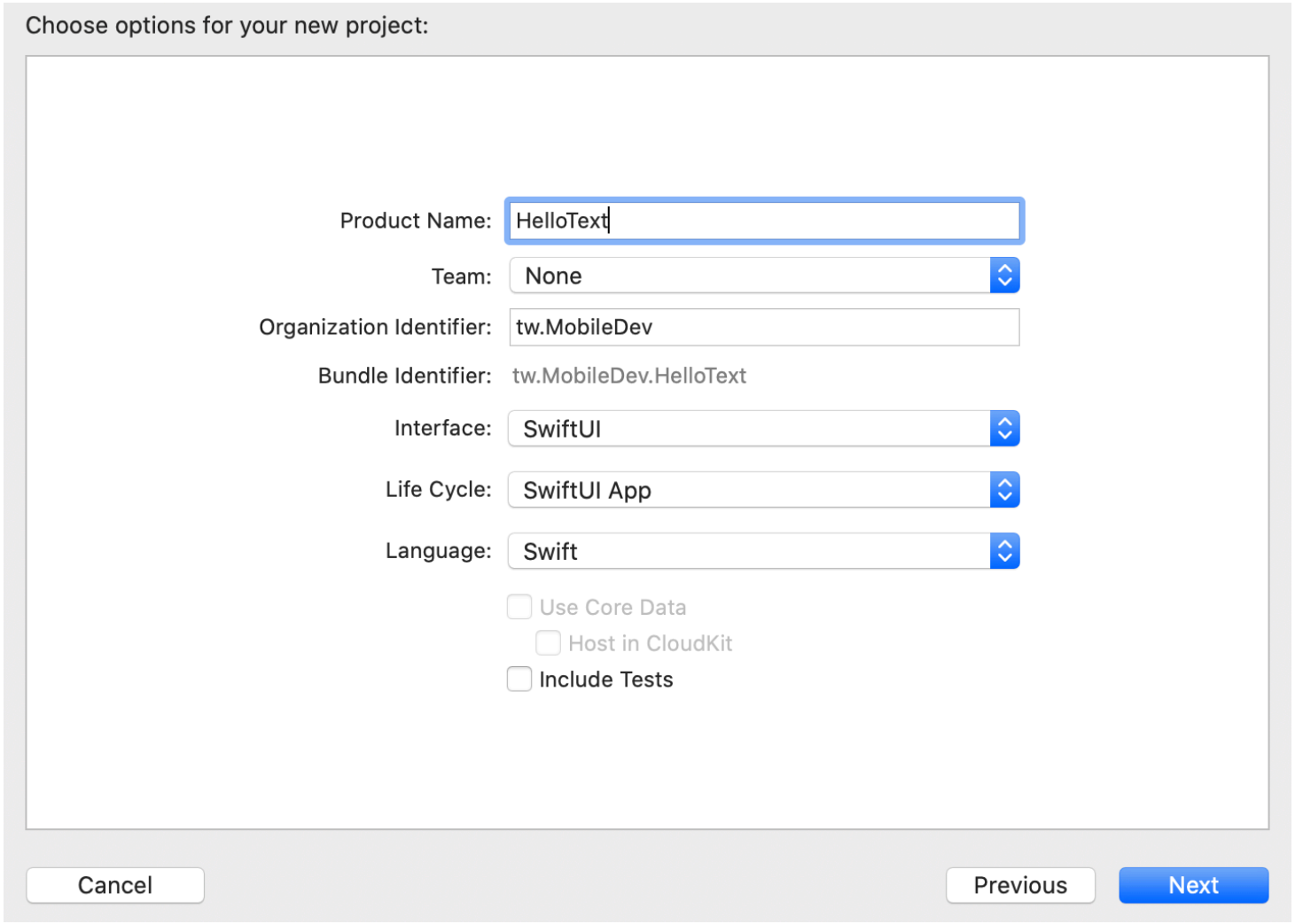
File -> New -> Project

- iOS -> App -> Next



專案名稱 HelloText

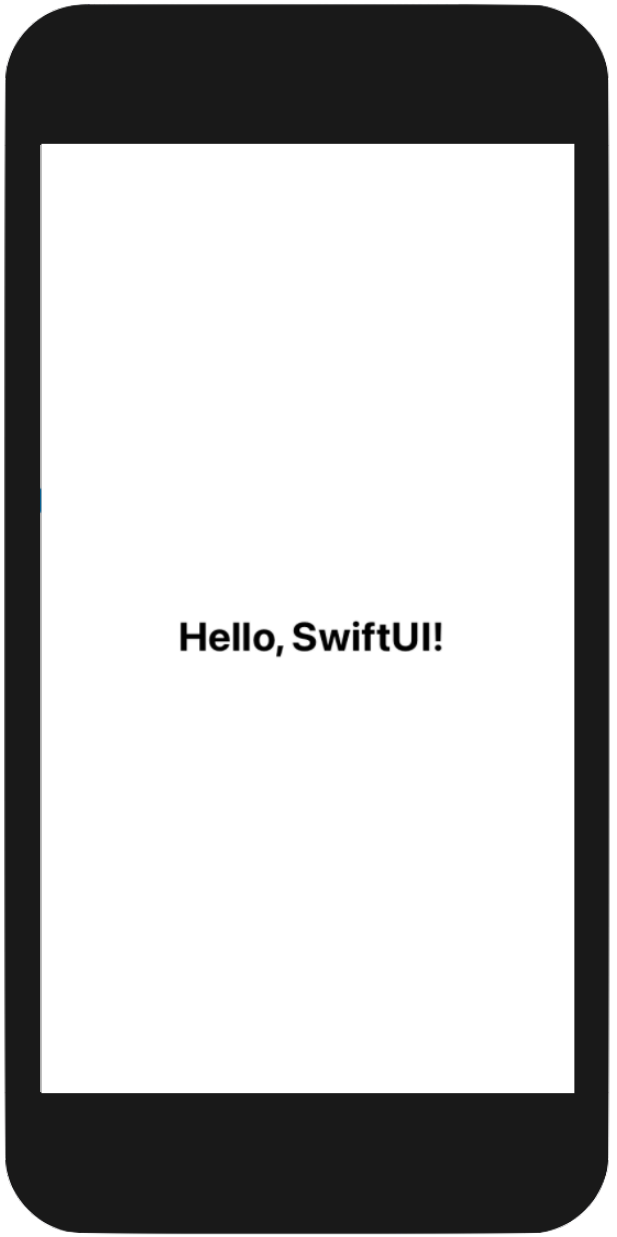
- SwiftUI

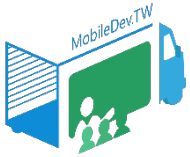


字體變更

- 粗體、標題、邊界

```
import SwiftUI

struct ContentView: View {
    var body: some View {
        Text("Hello, SwiftUI!")
            .fontWeight(.bold)
            .font(.title)
            .padding()
    }
}

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

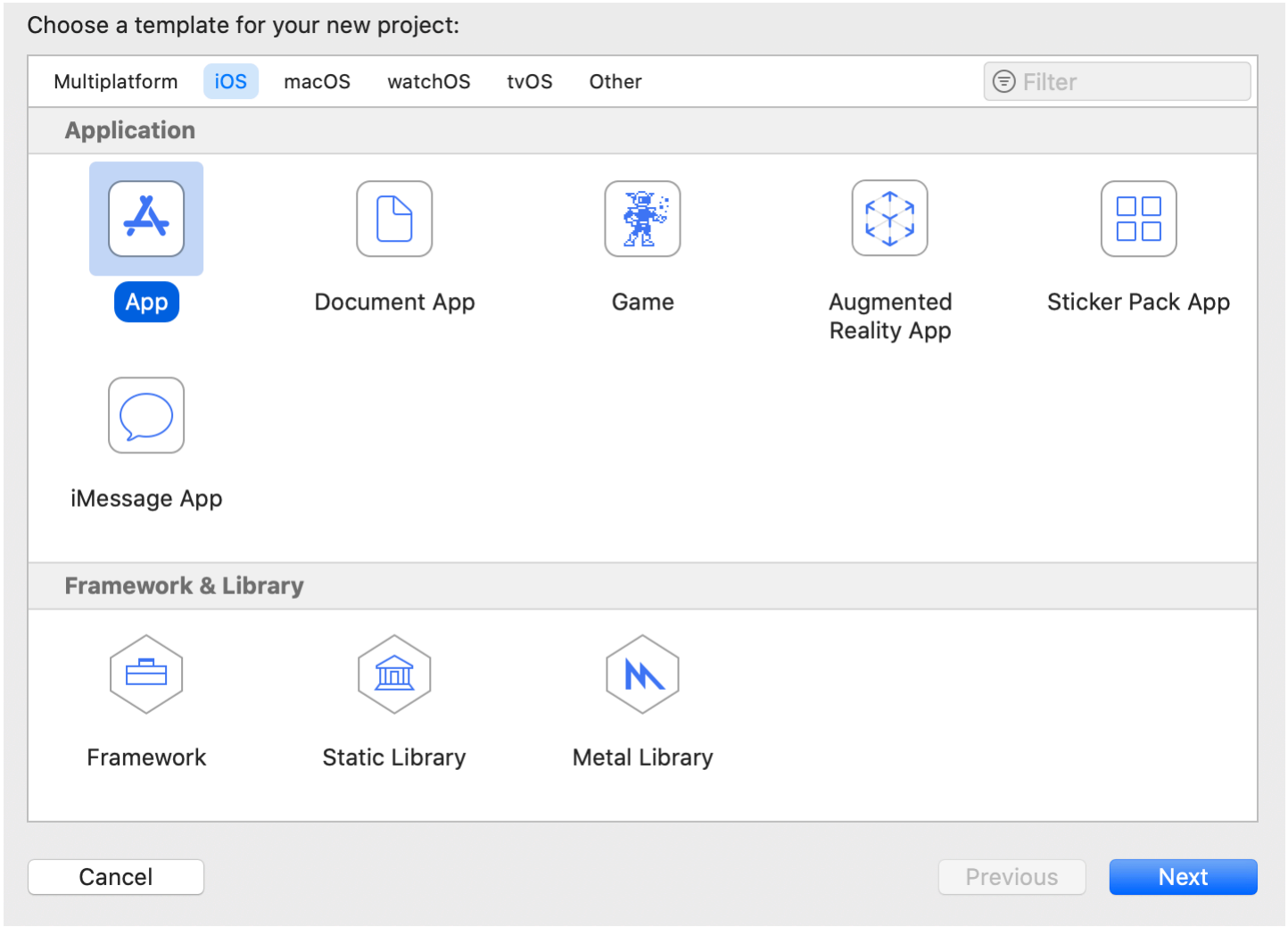




圖片呈現

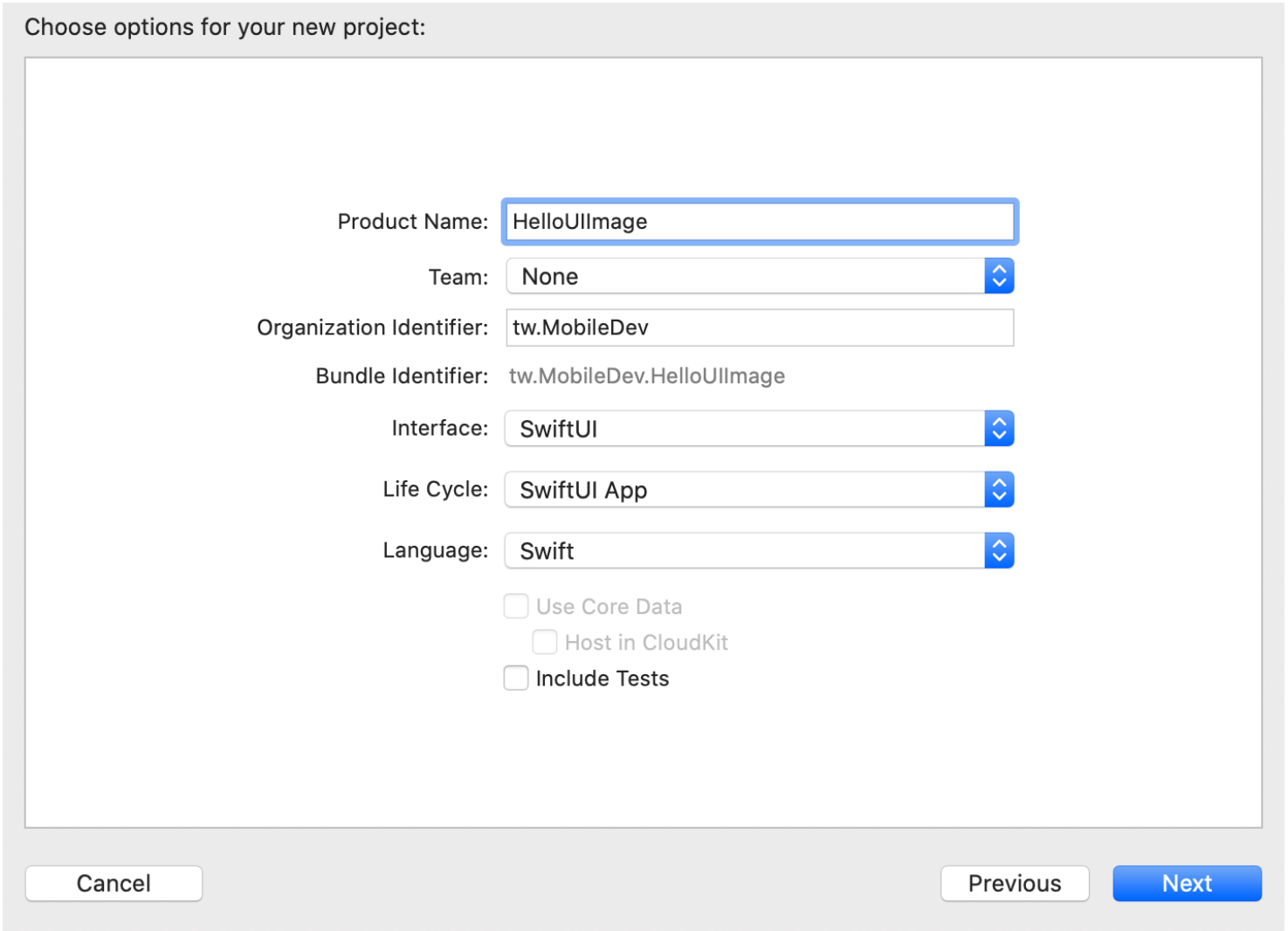
建立新的專案

- iOS -> App



HelloUIImage

- SwiftUI

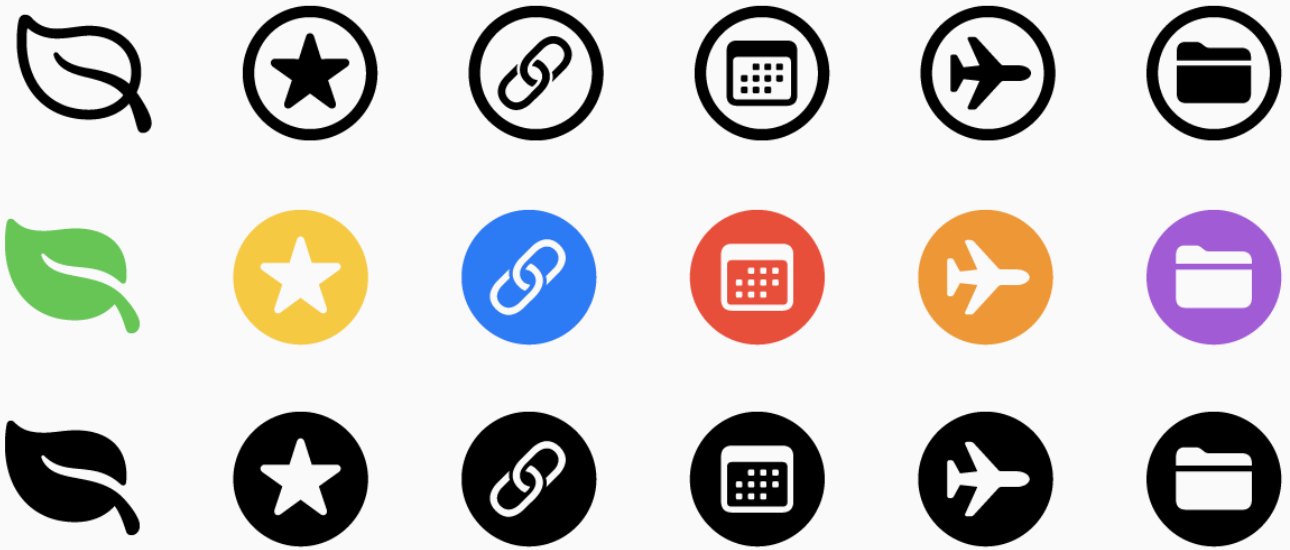


安裝系統ICON

- 下載、安裝

SF Symbols

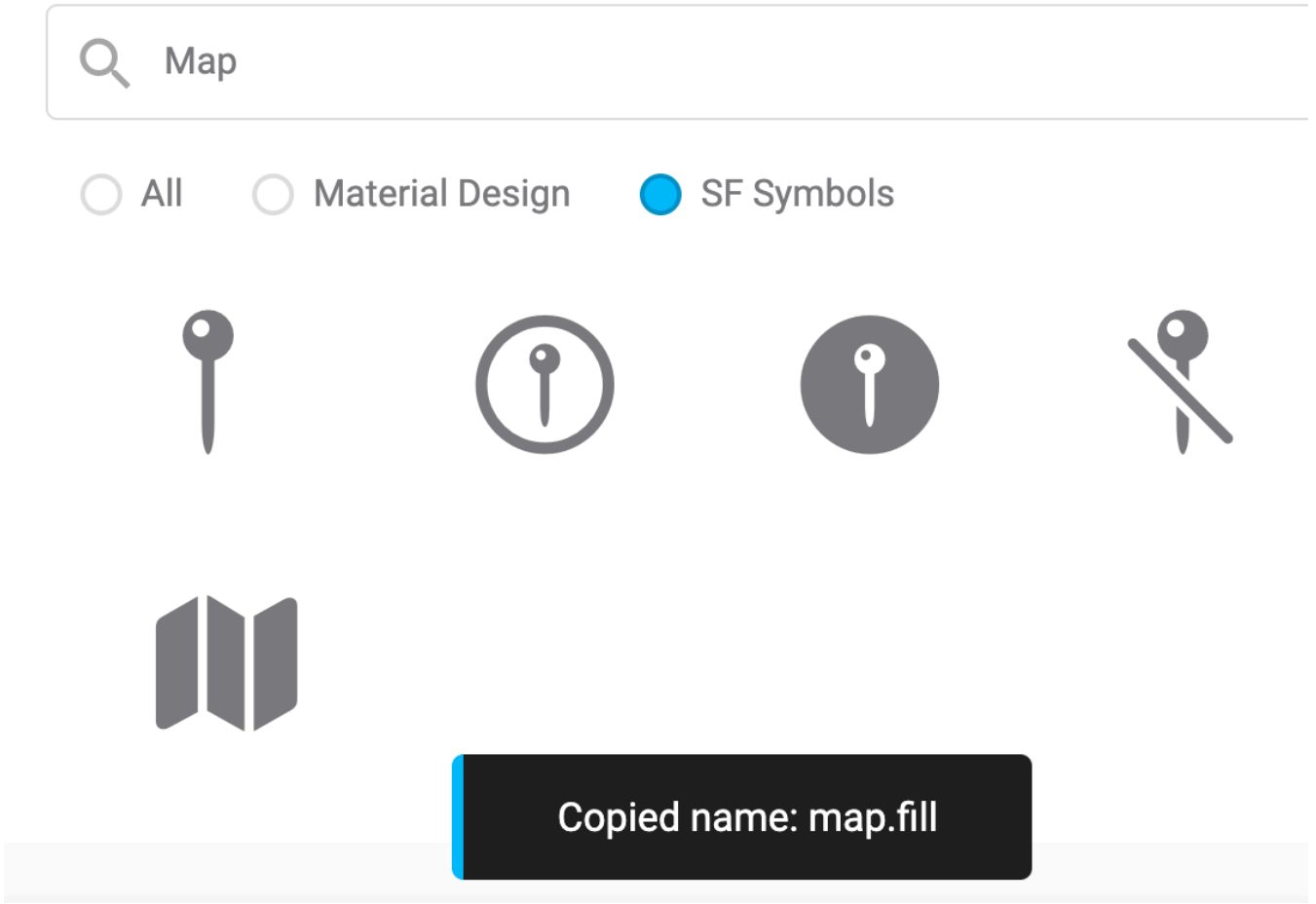
SF Symbols provides a set of over 2,400 consistent, highly configurable symbols you can use in your app. Apple designed SF Symbols to integrate seamlessly with the San Francisco system font, so the symbols automatically ensure optical vertical alignment with text in all weights and sizes.



https://developer.apple.com/design/downloads/SF-Symbols.dmg

SF Symbols查找

- 取得SF Symbol名稱

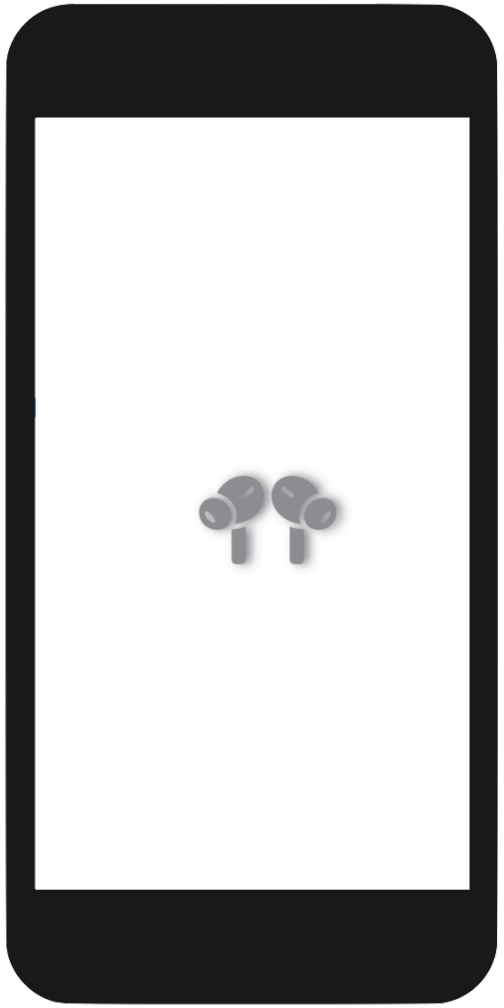


https://hotpot.ai/free_icons?s=sfSymbols

載入圖示

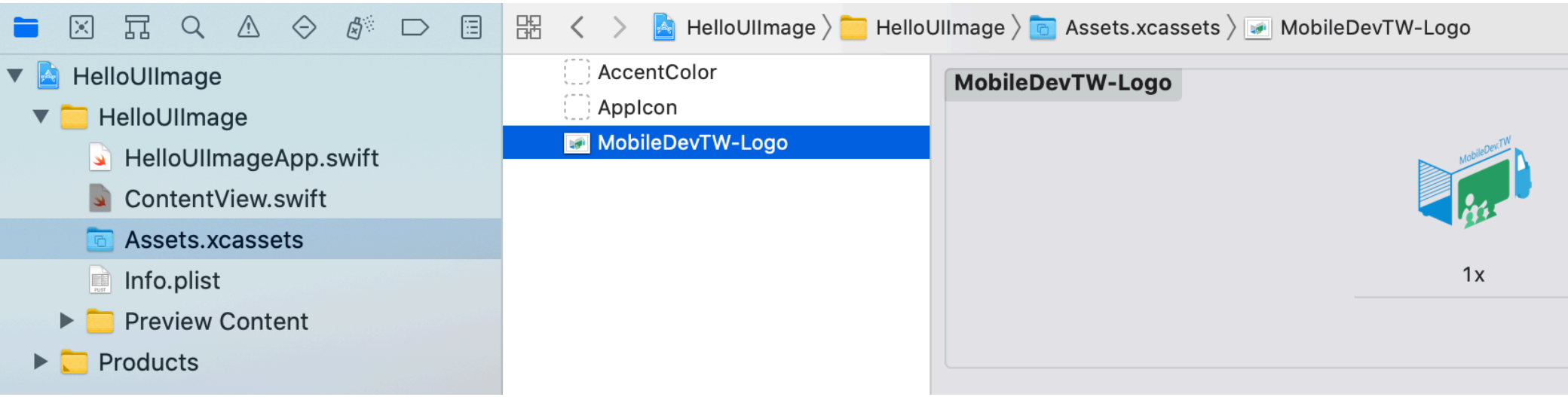
- 圖形選擇、字體大小、顏色、陰影

```
import SwiftUI


struct ContentView: View {
    var body: some View {
        Image(systemName: "airpodspro")
            .font(.system(size:100))
            .foregroundColor(.gray)
            .shadow(color:.gray, radius: 5, x:5, y:0)
    }
}


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



使用自己的圖片

- 打開左邊導覽列，選擇Assets.xcassets
- 加入自己的圖片，如有需要可重新將圖片命名



載入自己的圖片

- 圖片太大

```
import SwiftUI


struct ContentView: View {
    var body: some View {
        Image("MobileDevTW-Logo")
        /*
        Image(systemName: "airpodspro")
            .font(.system(size:100))
            .foregroundColor(.gray)
            .shadow(color:.gray, radius: 5, x:5, y:0)
        */
    }
}


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

不失真的全貌

- scaledToFit()

```
import SwiftUI


struct ContentView: View {
    var body: some View {
        Image("MobileDevTW-Logo")
            .resizable()
            .scaledToFit()
        /*
        Image(systemName: "airpodspro")
            .font(.system(size:100))
            .foregroundColor(.gray)
            .shadow(color:.gray, radius: 5, x:5, y:0)
        */
    }
}


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

不失真的滿版

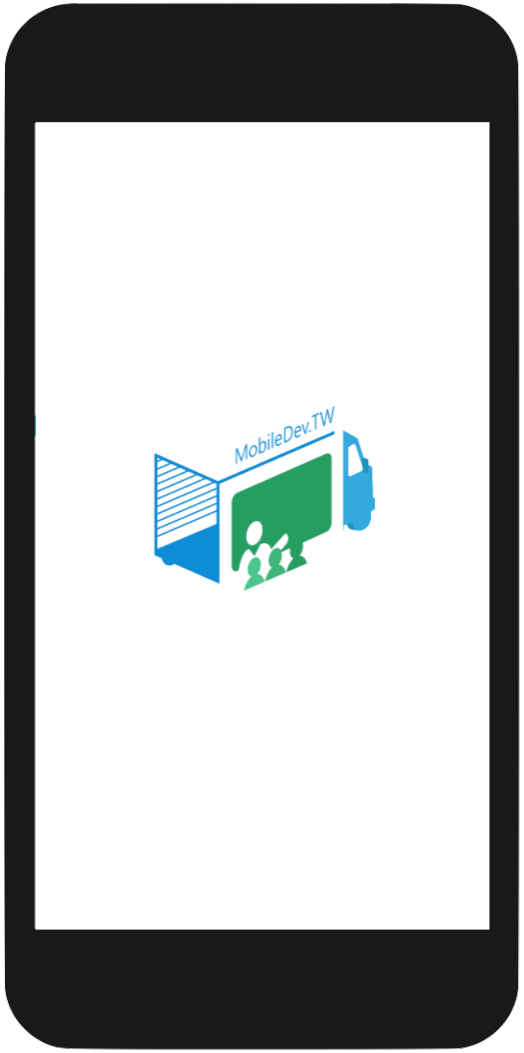
- scaledToFill()

```
import SwiftUI


struct ContentView: View {
    var body: some View {
        Image("MobileDevTW-Logo")
            .resizable()
            .scaledToFill()
        /*
        Image(systemName: "airpodspro")
            .font(.system(size:100))
            .foregroundColor(.gray)
            .shadow(color:.gray, radius: 5, x:5, y:0)
        */
    }
}


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

調整大小

- frame、aspectRatio

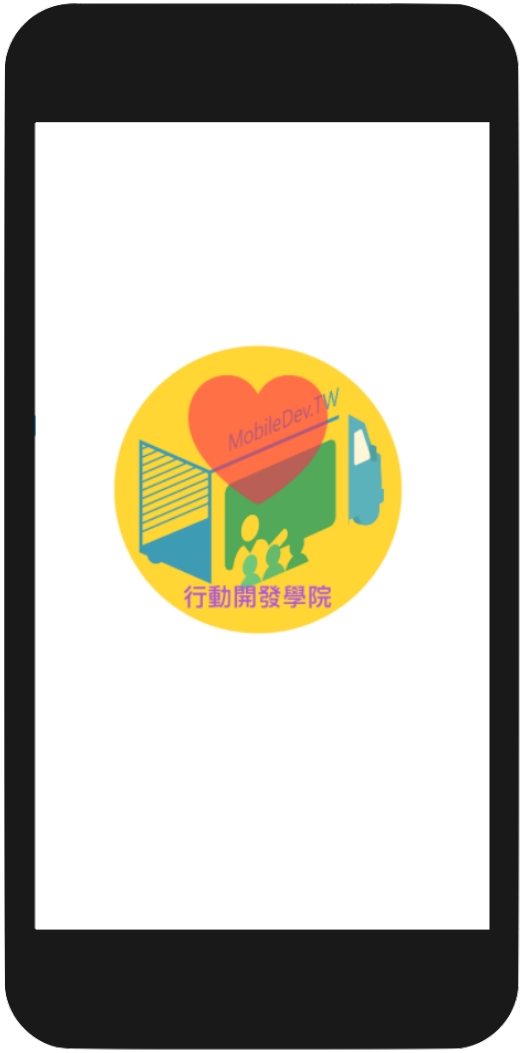
```
import SwiftUI


struct ContentView: View {
    var body: some View {
        Image("MobileDevTW-Logo")
            .resizable()
            .aspectRatio(contentMode: .fit)
            .frame(width:200, height: 200,
                   alignment: .center)
    }
}


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



層疊

- overlay

```
struct ContentView: View {
    var body: some View {
        Image("MobileDevTW-Logo")
            .resizable()
            .aspectRatio(contentMode: .fit)
            .frame(width:200, height:200,
                   alignment: .center)
            .background(Color.yellow)
            .clipShape(Circle())
            .opacity(0.8)
            .overlay(
                Image(systemName: "heart.fill")
                    .font(.system(size: 100))
                    .foregroundColor(.pink)
                    .opacity(0.6)
                    .frame(width: 160, height: 160, alignment: .top)
            )
            .overlay(
                Text("行動開發學院")
                    .fontWeight(.bold)
                    .frame(width: 170, height: 170, alignment: .bottom)
                    .foregroundColor(.purple)
            )
    }
}
```



圖文組合練習

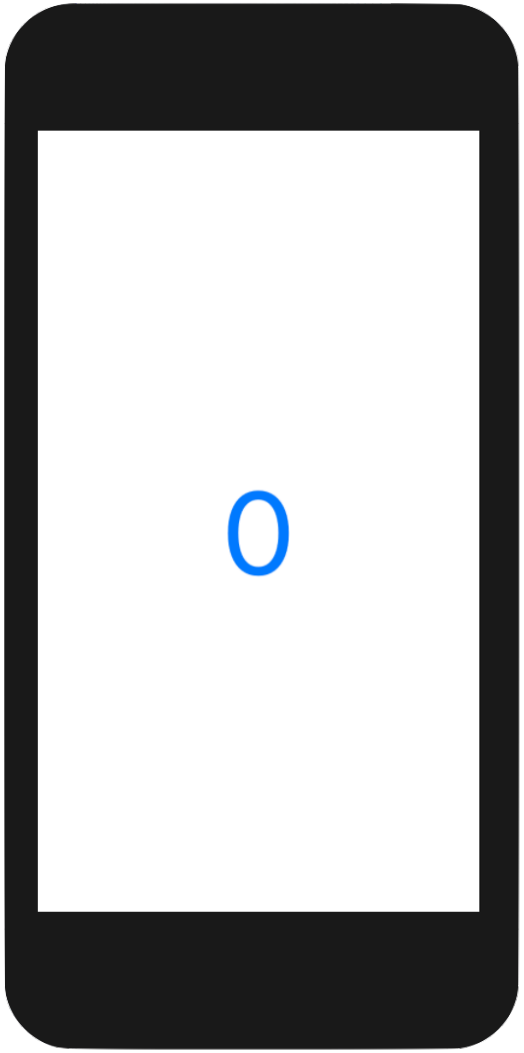
- 圖片滿版
  - 忽略安全區域
  - 填滿
- 下面有文字框
  - 圓角邊框
  - 半透明
  - 白字藍底
  - 靠在圖下方

圖文組合練習

```
struct ContentView: View {
    var body: some View {
        Image("steve-jobs")
            .resizable()
            .aspectRatio(contentMode: .fill)
            .overlay(
                Text("Stay Hungry, \nStay Foolish")
                    .fontWeight(.heavy)
                    .lineSpacing(20)
                    .font(.system(size: 32.0))
                    .foregroundColor(.white)
                    .frame(width: 350, height: 150,
                           alignment: .center)
                    .background(Color.blue)
                    .cornerRadius(30.0)
                    .opacity(0.8)

                ,
                alignment: .bottom
            )
            .edgesIgnoringSafeArea(.all)
    }
}
```

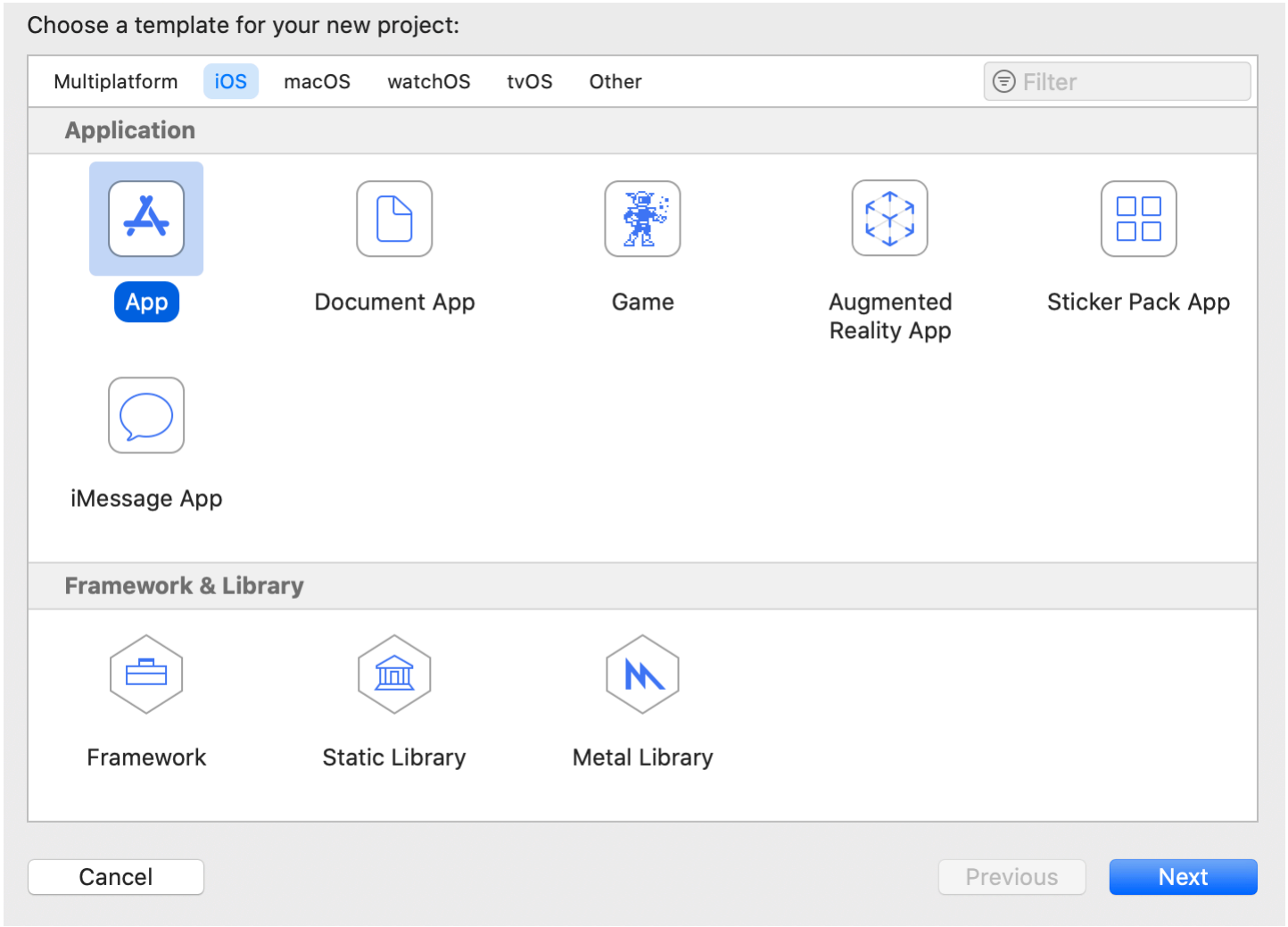
按鈕-計數器

- 顯示一個數字，一開始為0
- 可以點擊，每次增加1



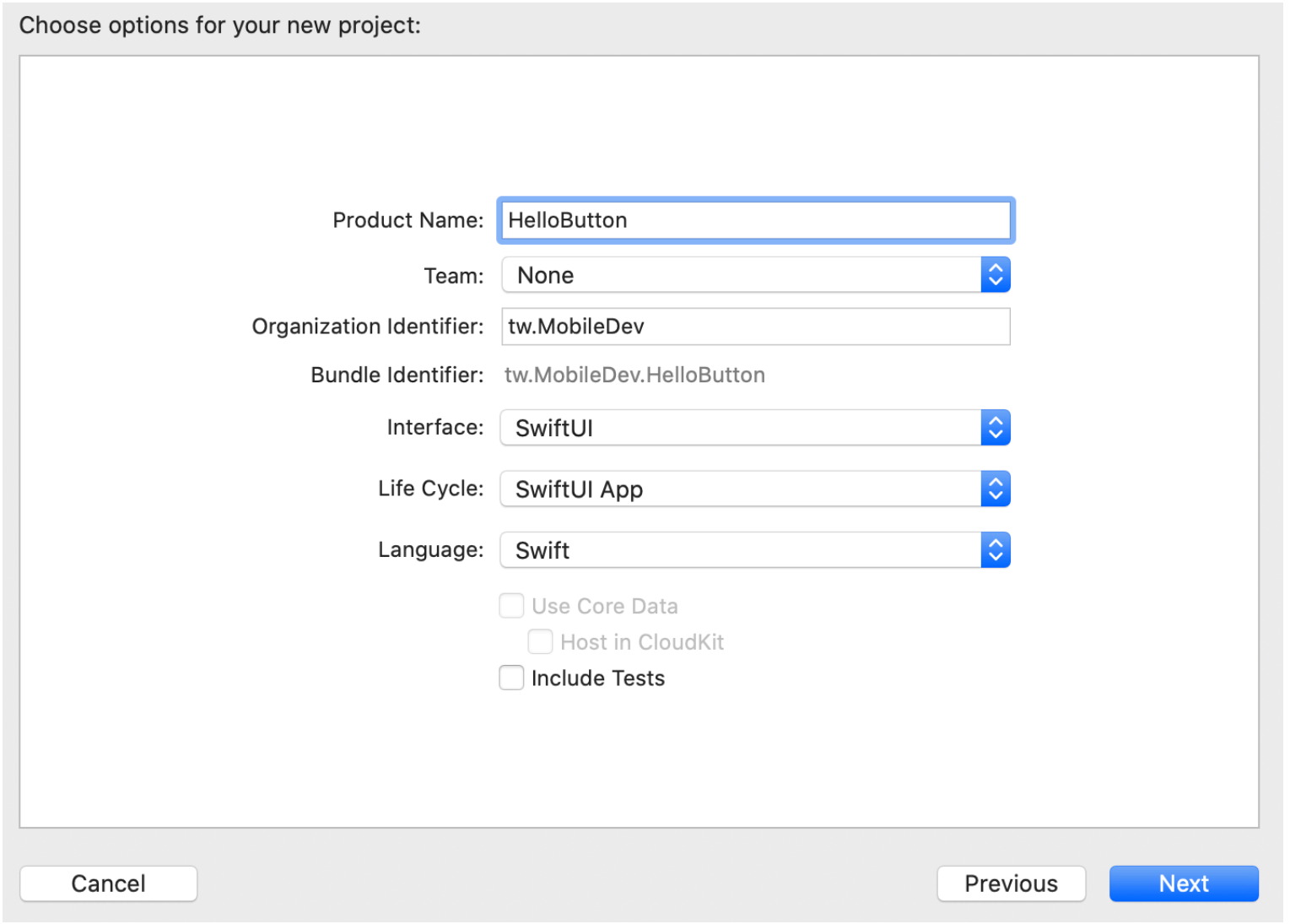
建立新的專案

- iOS -> App



HelloButton

- SwiftUI



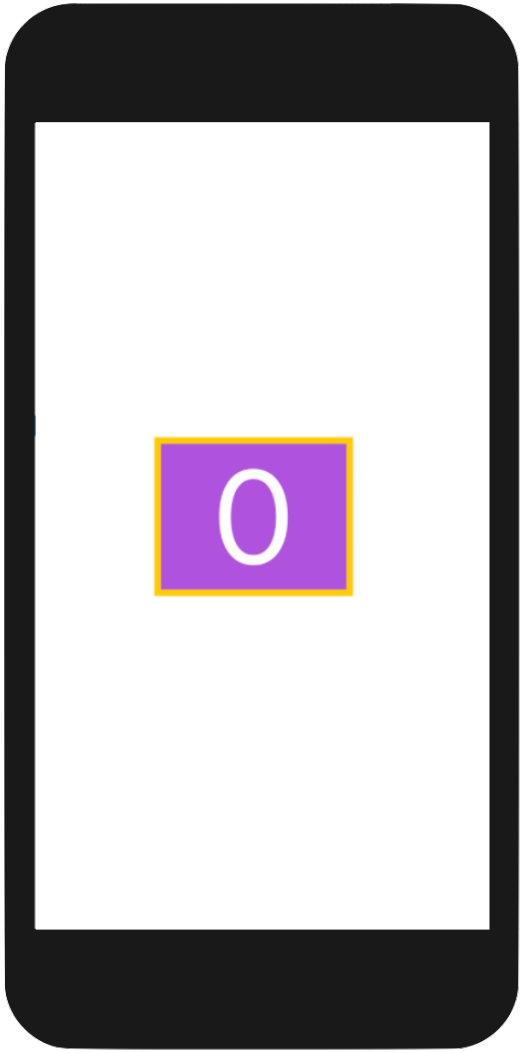
Button

```
import SwiftUI


struct ContentView: View {
    @State var count:Int=0
    var body: some View {
        Button(action:{
            self.count+=1
        }){
            Text(String(count))
                .font(.system(size: 100))
        }
    }
}

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

設定樣式

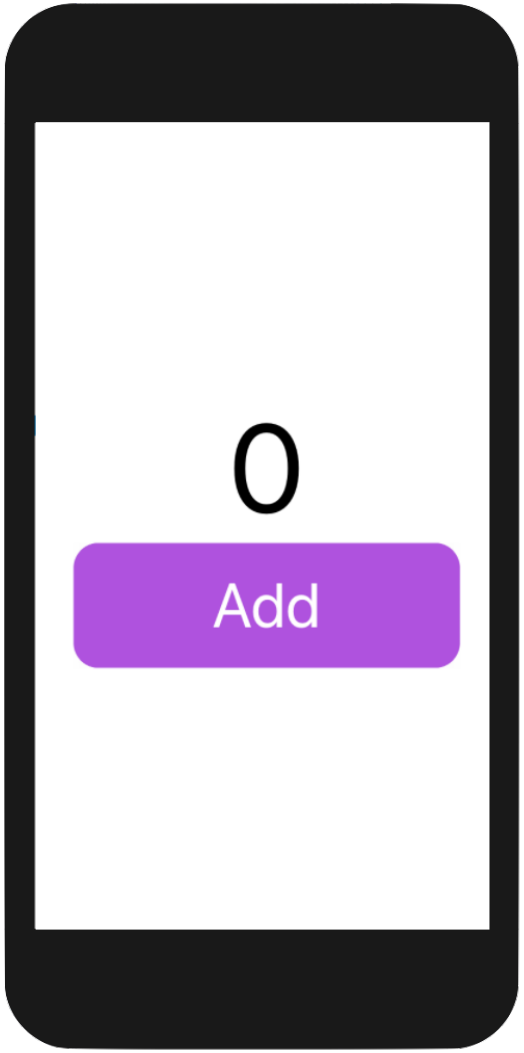
- 邊距、字體大小、顏色、背景、邊框

```
struct ContentView: View {
    @State var count:Int=0
    var body: some View {
        Button(action:{
            self.count+=1
        }){
            Text(String(count))
                .padding(.all, 10)
                .frame(width: 150, height: 120,
                       alignment: .center)
                .font(.system(size: 100))
                .background(Color.purple)
                .foregroundColor(.white)
                .border(Color.yellow, width: 5)
        }
    }
}
```



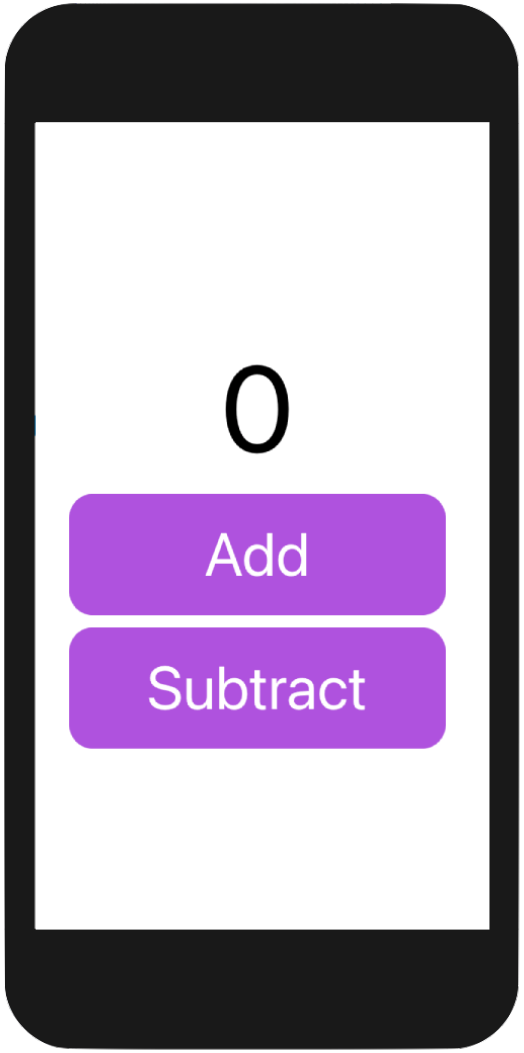
將控制按鈕與顯示文字分開

```
struct ContentView: View {
    @State var count:Int=0
    var body: some View {
        VStack{
            Text(String(count))
                .padding(.all, 10)
                .frame(width: 150, height: 120, alignment: .center)
                .font(.system(size: 100))
            Button(action:{
                self.count+=1
            }){
                Text("Add")
                    .frame(width: 300, height: 100,
                           alignment: .center)
                    .font(.system(size: 50))
                    .background(Color.purple)
                    .padding(.horizontal,5)
                    .foregroundColor(.white)
                    .border(Color.purple, width: 30)
                    .cornerRadius(20)
            }
        }
    }
}
```



再增加一顆按鈕讓數字變小

- 加上padding

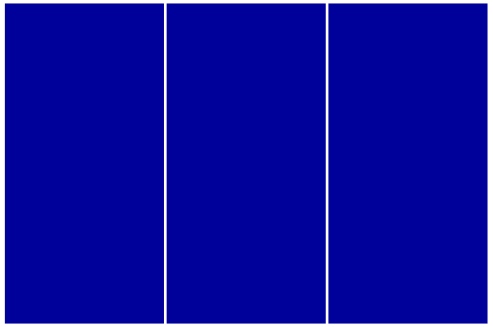


剪刀石頭布

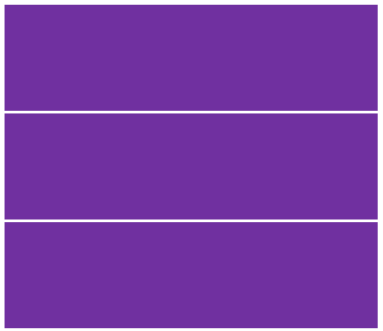
- 亂數、選擇、呈現
  - 產生0, 1, 2的亂數
  - 剪刀、石頭、布
  - 顯示在上方

堆疊

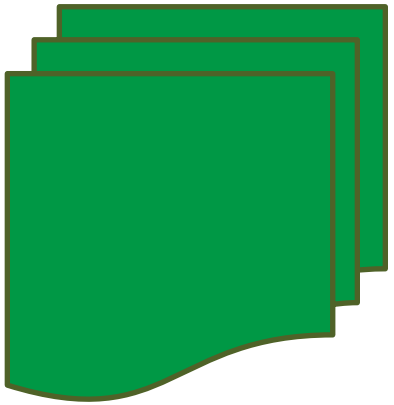
- 水平 H-Stack



- 垂直 V-Stack

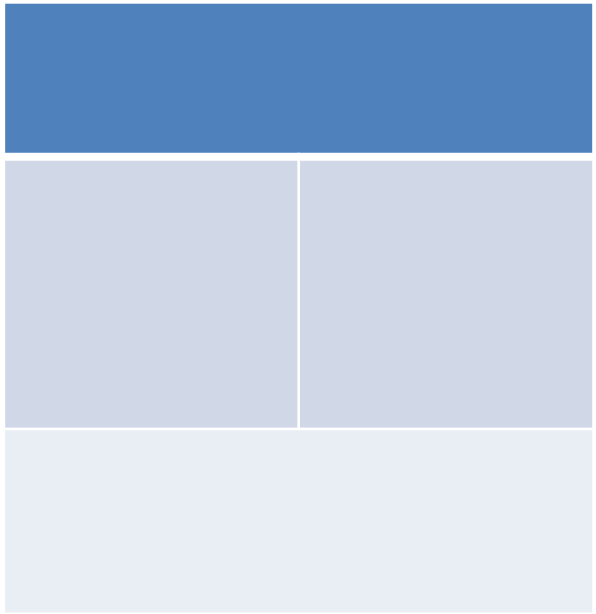


- 層疊 Z-Stack



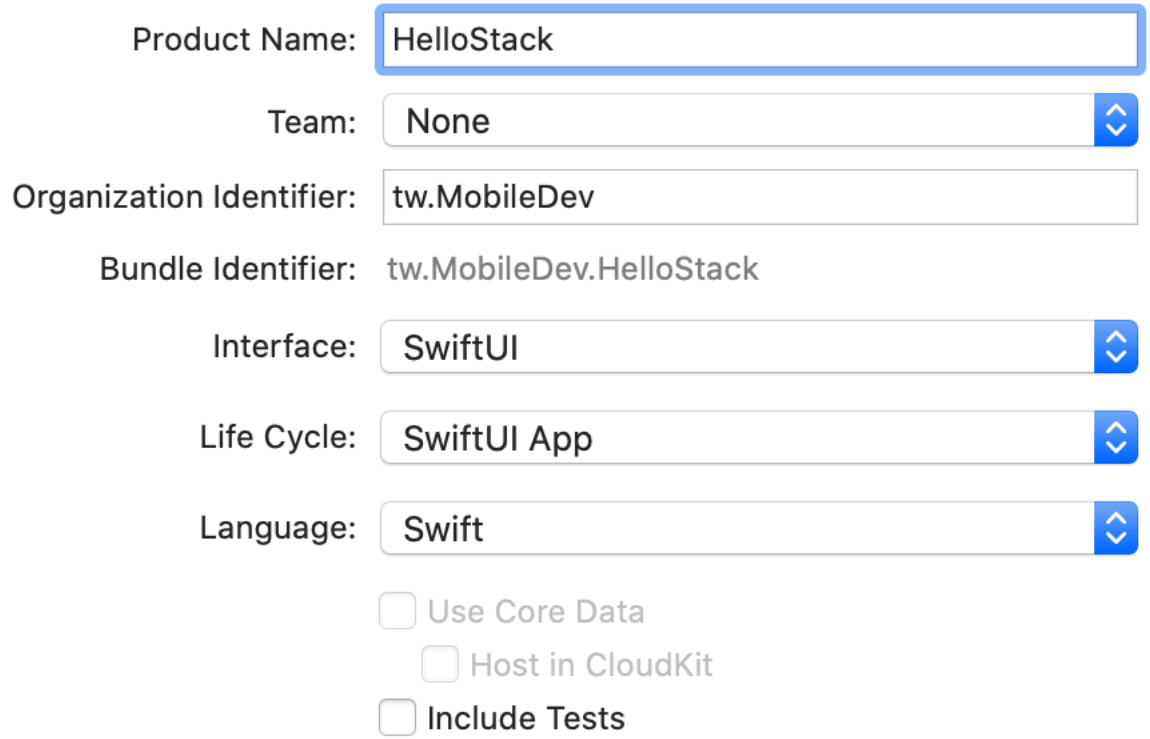
Ryan的冰箱

- 標題：Ryan的冰箱
- 中間兩種水果、下方一條魚



堆疊排版練習

- 建立新專案 -> iOS -> App
- HelloStack



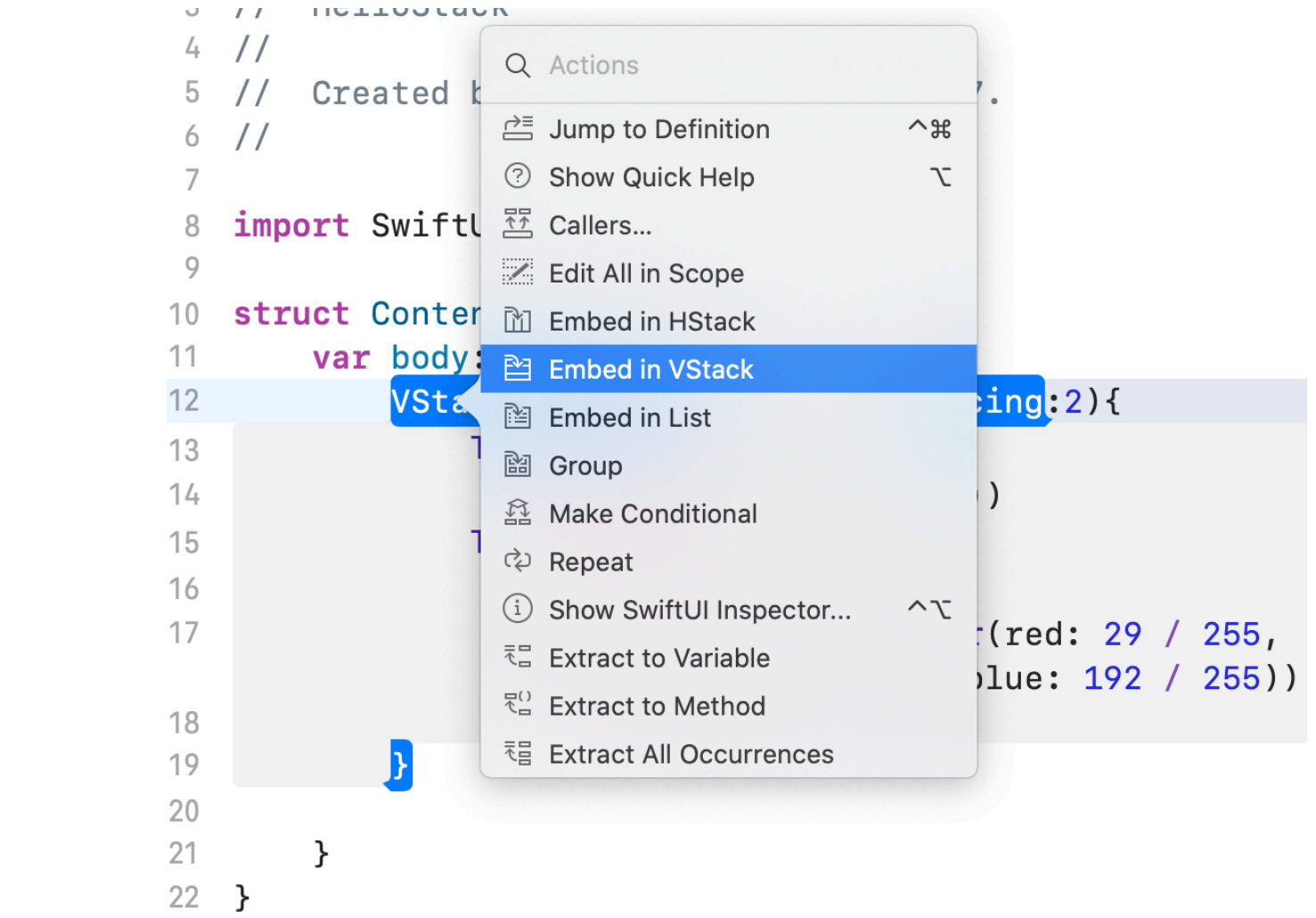
文字堆疊

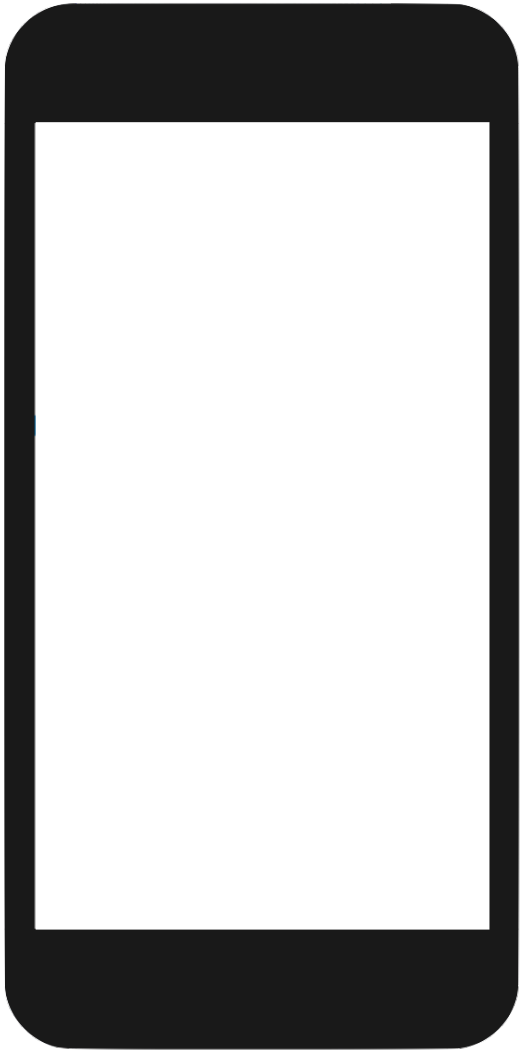
- VStack
  - 文字大小
  - 文字顏色
  - 堆疊對齊、行距

```
struct ContentView: View {
    var body: some View {
        VStack(alignment:.center, spacing:2){
            Text("Ryan的")
                .font(.system(size:30))
            Text("冰箱")
                .font(.title)
                .foregroundColor(Color(red: 29 / 255,
                    green: 40 / 255, blue: 192 / 255))
        }

    }
}
```

增加堆疊層次 Embed in Vstack/HStack

- 選取VStack，按下Command，選擇



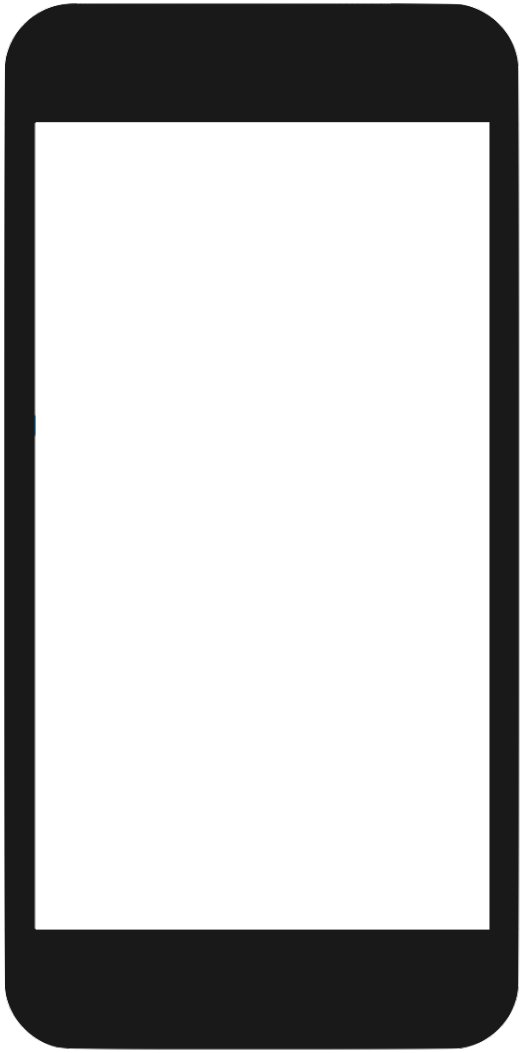


程式碼獨立區塊

- 當層次太多時，可將程式碼獨立出來

```
4  //
5  //  Created by Rya
6  //
7
8  import SwiftUI
9
10 struct ContentView
11     var body: some
12         VStack {
13             VStac                                    ng:2){
14                 Te
15
16                 Te
17
18                                                      red: 29 / 255,
                                                        ue: 192 / 255))
19
20             }
21         }
22
23     }
24 }
```

Actions
Jump to Definition ^⌘
Show Quick Help ⌥
Callers...
Edit All in Scope
Embed in HStack
Embed in VStack
Embed in List
Group
Make Conditional
Repeat
Show SwiftUI Inspector... ^⌥
Extract Subview
Extract to Variable
Extract to Method
Extract All Occurrences

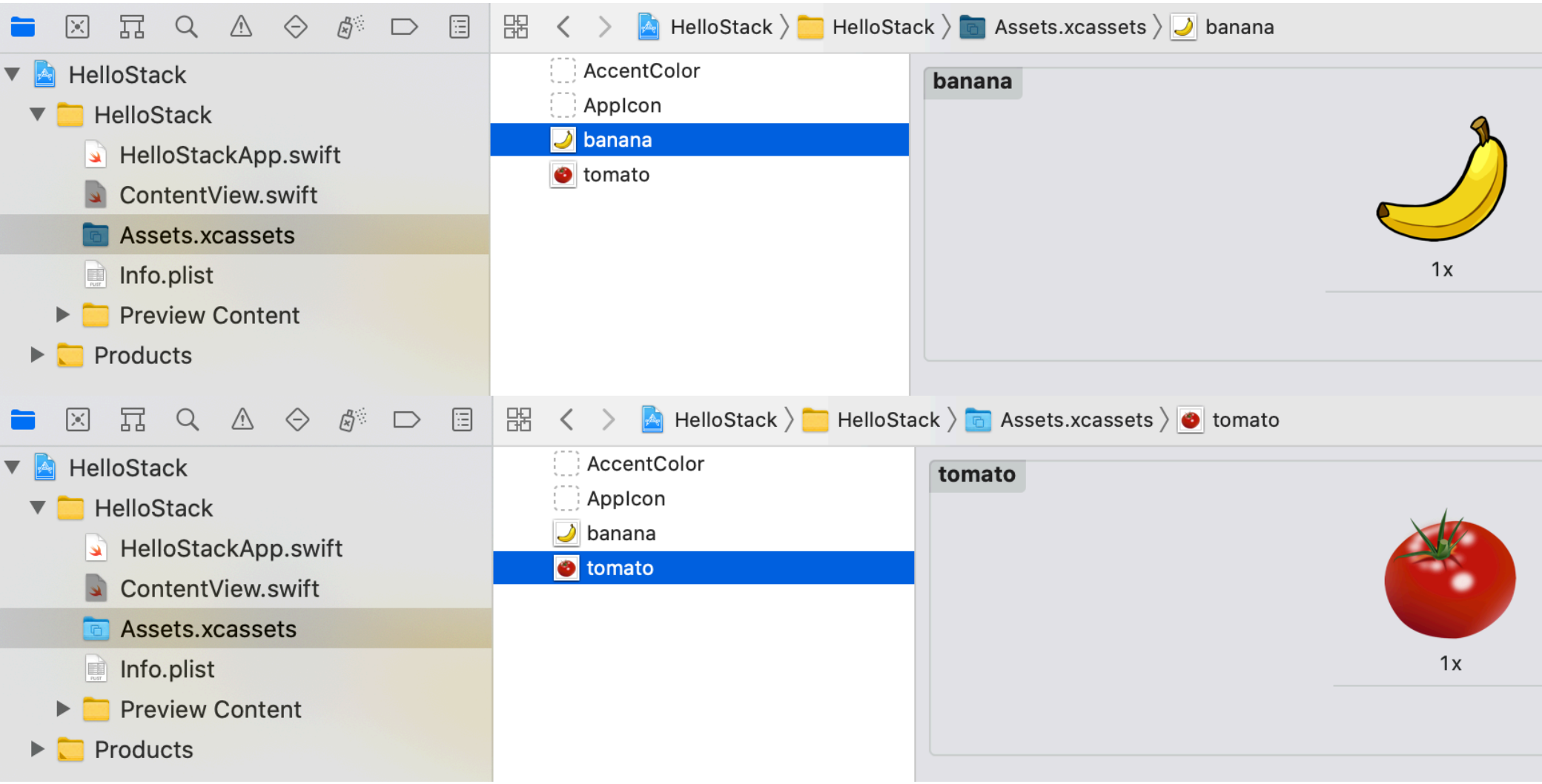


重新命名為TitleView

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

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

struct TitleView: View {
    var body: some View {
        VStack(alignment:.center, spacing:2){
            Text("Ryan的")
                .font(.system(size:30))
            Text("冰箱")
                .font(.title)
                .foregroundColor(Color(red: 29 / 255, green: 40 / 255, blue: 192 / 255))

        }
    }
}
```

加入要練習的圖片

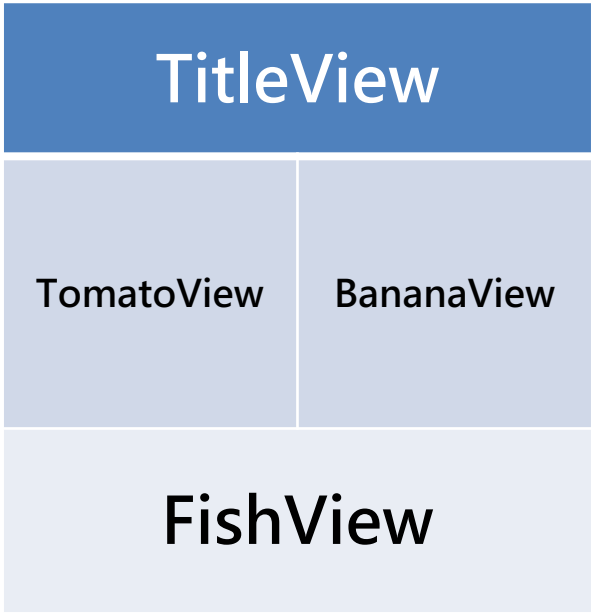
- 左邊導覽列 -> Assets.xcassets

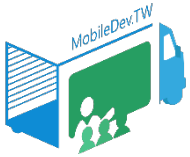


主結構

- 畫面拆解

```
import SwiftUI

struct ContentView: View {
    var body: some View {
        VStack {
            TitleView()
            HStack{
                TomatoView()
                BananaView()
            }
            FishView()
        }
    }
}

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





TitleView

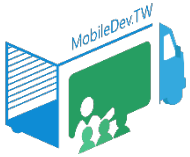
- 最上面

```
struct TitleView: View {
    var body: some View {
        VStack(alignment:.center, spacing:2){
            Text("Ryan的")
                .font(.system(size:30))
            Text("冰箱")
                .font(.title)
                .foregroundColor(Color(red: 29 / 255,
                    green: 40 / 255, blue: 192 / 255))

        }
    }
}
```

TomatoView

- 中間左邊

```
struct TomatoView: View {
    var body: some View {
        VStack{
            Image("tomato")
                .resizable()
                .aspectRatio(contentMode:.fit)
                .frame(width: UIScreen.screenWidth/2-20,
                       alignment: .center)
            Text("Tomato")
                .fontWeight(.bold)
                .font(.system(size: 30))
        }
    }
}
```



BananaView

- 中間右邊

```
struct BananaView: View {
    var body: some View {
        VStack{
            Image("banana")
                .resizable()
                .aspectRatio(contentMode:.fit)
                .frame(width: UIScreen.screenWidth/2-20,
                       alignment: .center)
            Text("Banana")
                .fontWeight(.bold)
                .font(.system(size: 30))
        }.padding(.all, 2)
    }
}
```

FishView

- 最下方

```
struct FishView: View {
    var body: some View {
        VStack{
            Image("fish")
                .resizable()
                .aspectRatio(contentMode:.fit)
                .padding(.all, 15)
            Text("Fish")
                .fontWeight(.bold)
                .font(.system(size: 30))
        }
    }
}

extension UIScreen{
   static let screenWidth = UIScreen.main.bounds.size.width
   static let screenHeight = UIScreen.main.bounds.size.height
   static let screenSize = UIScreen.main.bounds.size
}
```

重構水果區

- 將TomatoView與BananaView結構註解
- 新增FruitView結構

```
struct FruitView: View {
    var imageName:String
    var body: some View {
        VStack{
            Image(imageName)
                .resizable()
                .aspectRatio(contentMode:.fit)
                .frame(width: UIScreen.screenWidth/2-20,
                       alignment: .center)
            Text(imageName.capitalized)
                .fontWeight(.bold)
                .font(.system(size: 30))
        }.padding(.all, 2)
    }
}
```

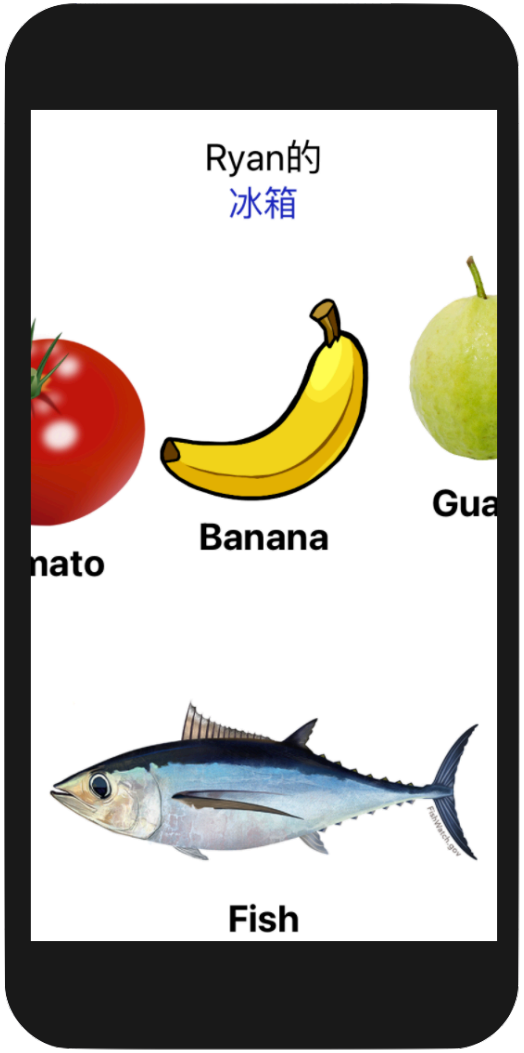
改寫ContentView

- 都換成FruitView

```
import SwiftUI

struct ContentView: View {
    var body: some View {
        VStack {
            TitleView()
            HStack{
                FruitView(imageName: "tomato")
                FruitView(imageName: "banana")
                /*
                TomatoView()
                BananaView()
                */
            }
            FishView()
        }
    }
}

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

可是如果再增加一個...

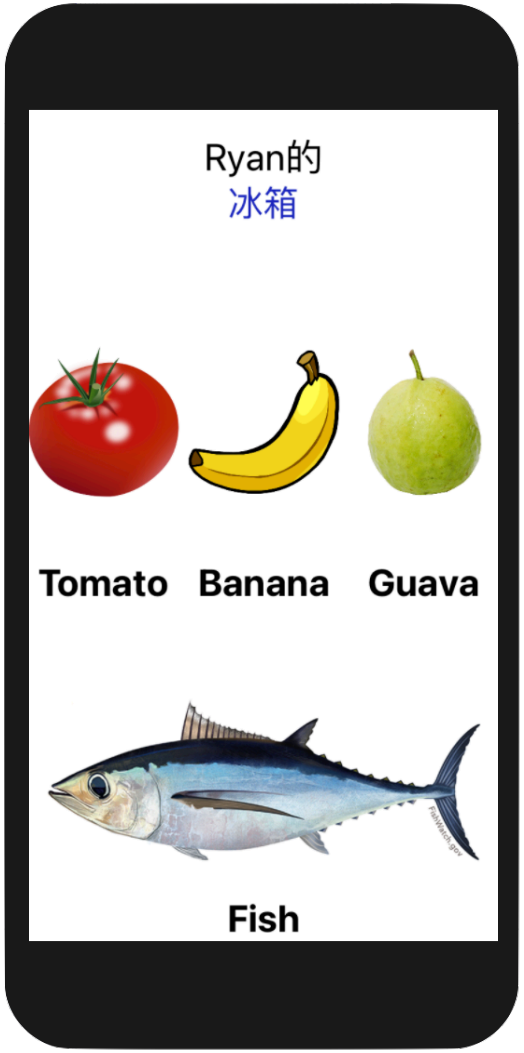
- 每一個水果的寬度問題



改寫FruitView

- infinity

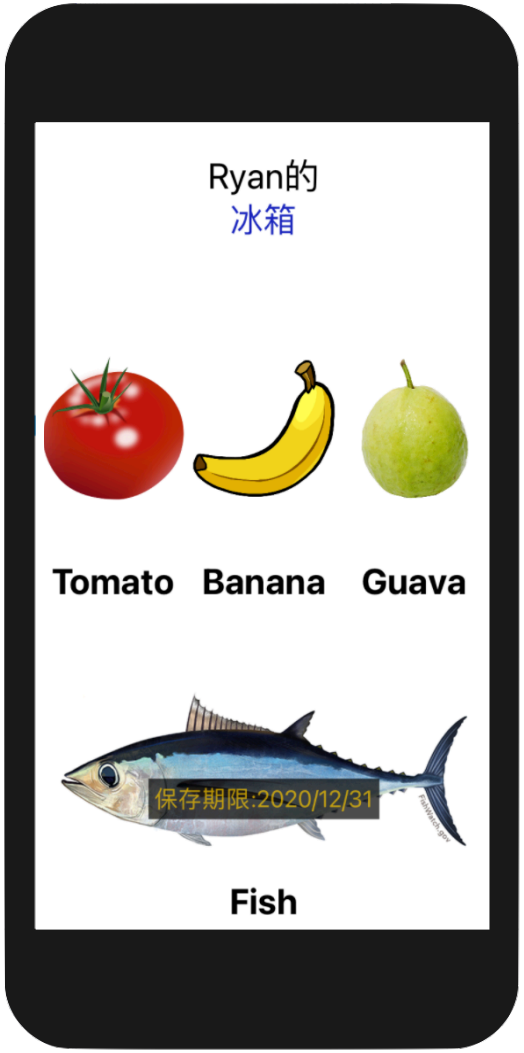
```
struct FruitView: View {
    var imageName:String
    var body: some View {
        VStack{
            Image(imageName)
                .resizable()
                .aspectRatio(contentMode:.fit)
                .frame(height: 200, alignment: .center)
            Text(imageName.capitalized)
                .fontWeight(.bold)
                .font(.system(size: 30))
        }
        .frame(minWidth: 0, idealWidth: 100,
               maxWidth: .infinity, minHeight: 0,
               idealHeight: 100, maxHeight: .infinity,
               alignment: .center)
    }
}
```



增加魚的保存期限

- ZStack

```
struct ContentView: View {
    var body: some View {
        VStack {
            TitleView()
            HStack{
                FruitView(imageName: "tomato")
                FruitView(imageName: "banana")
                FruitView(imageName: "guava")
            }
            ZStack{
                FishView()
                Text("保存期限:2020/12/31")
                    .font(.system(size: 20))
                    .foregroundColor(.yellow)
                    .padding(.all,5)
                    .background(Color.black)
                    .opacity(0.7)
            }
        }
    }
}
```



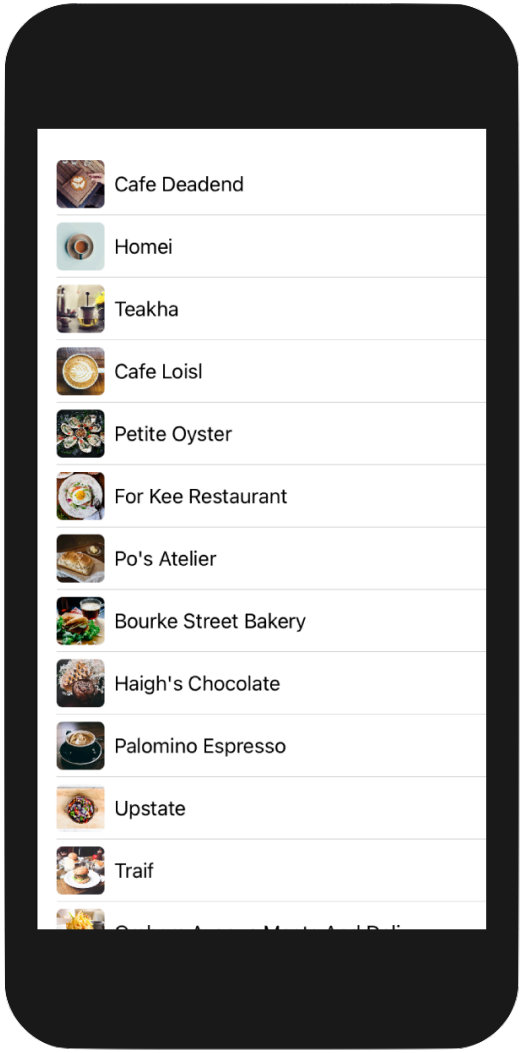
調整保存期限位置

- offset

```
struct ContentView: View {
    var body: some View {
        VStack {
            TitleView()
            HStack{
                FruitView(imageName: "tomato")
                FruitView(imageName: "banana")
                FruitView(imageName: "guava")
            }
            ZStack{
                FishView()
                Text("保存期限:2020/12/31")
                    .font(.system(size: 20))
                    .foregroundColor(.yellow)
                    .padding(.all,5)
                    .background(Color.black)
                    .opacity(0.7)
                    .offset(x: 40, y: 20)
            }
        }
    }
}
```

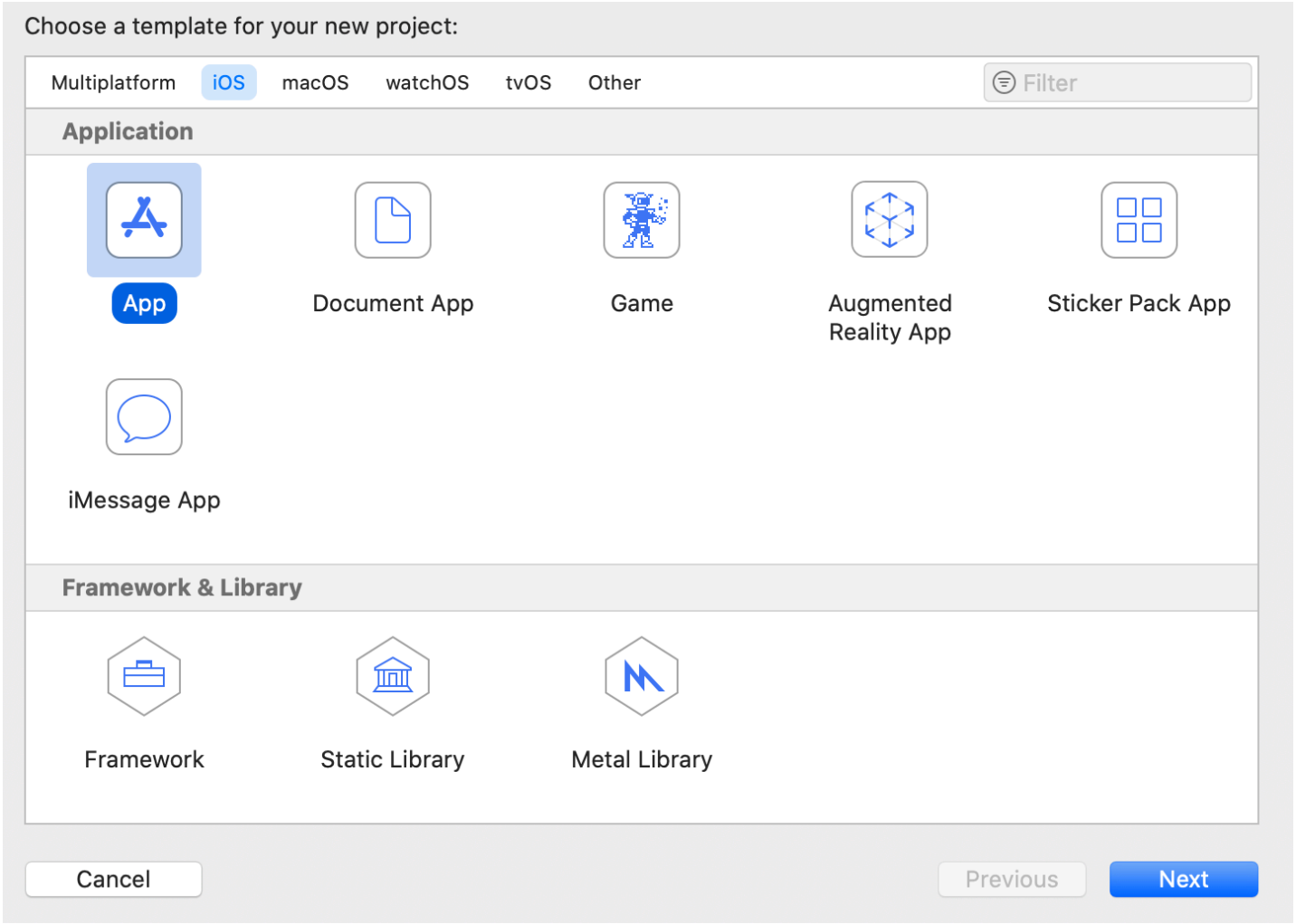
條列式

- 餐廳清單
  - 小圖 + 餐廳名稱
- 點擊之後可以顯示該餐廳細節



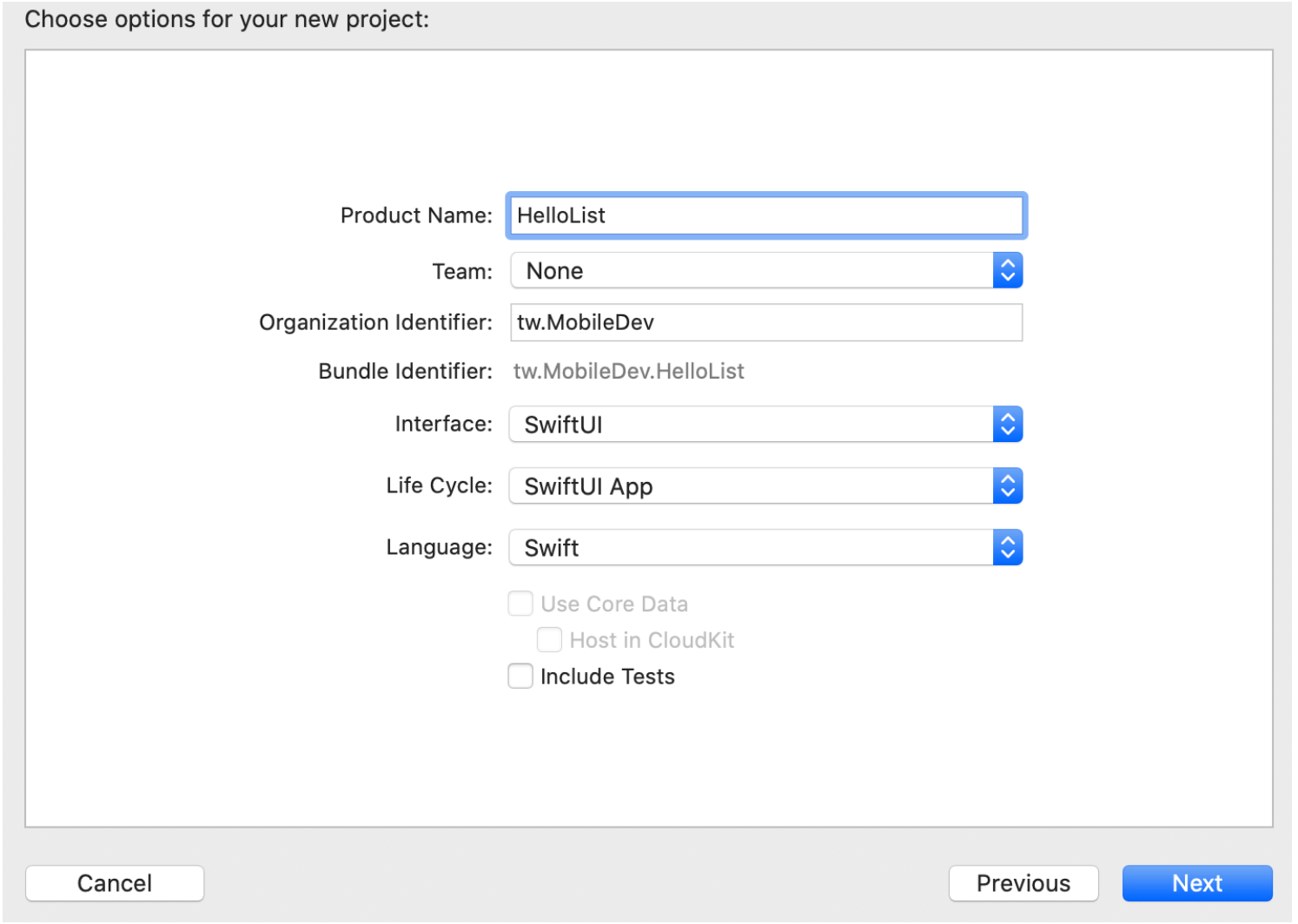
File -> New -> Project

- iOS -> App -> Next



專案名稱 HelloList

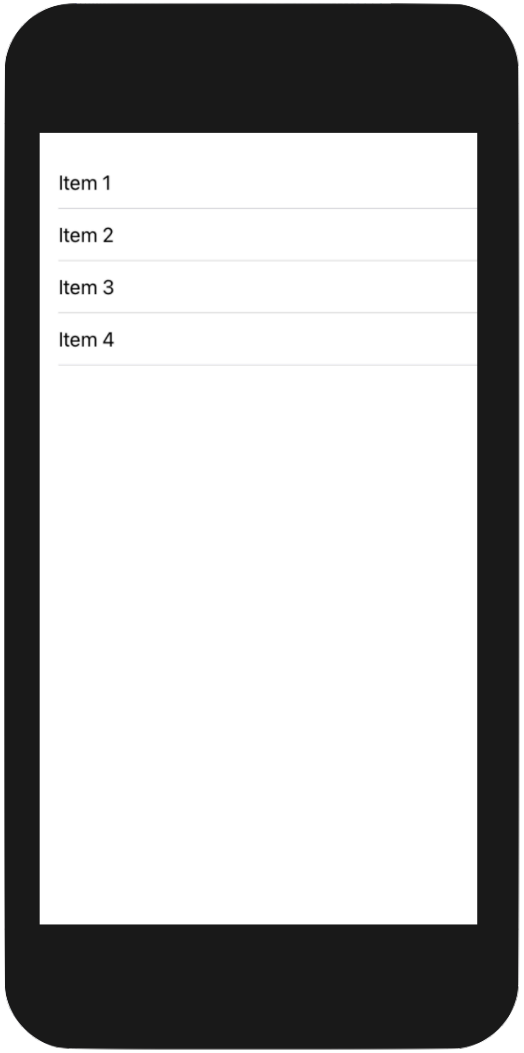
- SwiftUI



ContentView.swift

- 加入清單 List

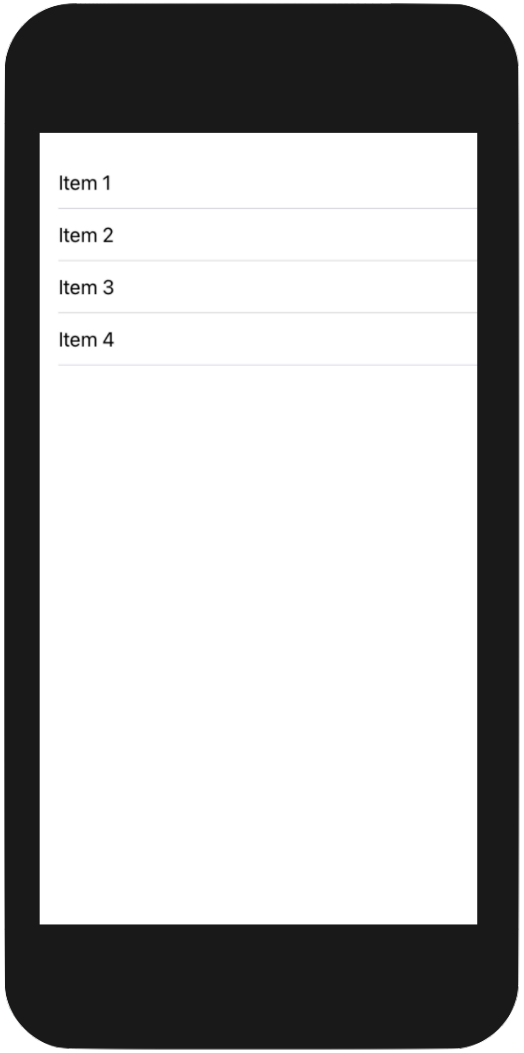
```
import SwiftUI

struct ContentView: View {
    var body: some View {
        List{
            Text("Item 1")
            Text("Item 2")
            Text("Item 3")
            Text("Item 4")
        }
    }
}

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



ContentView.swift

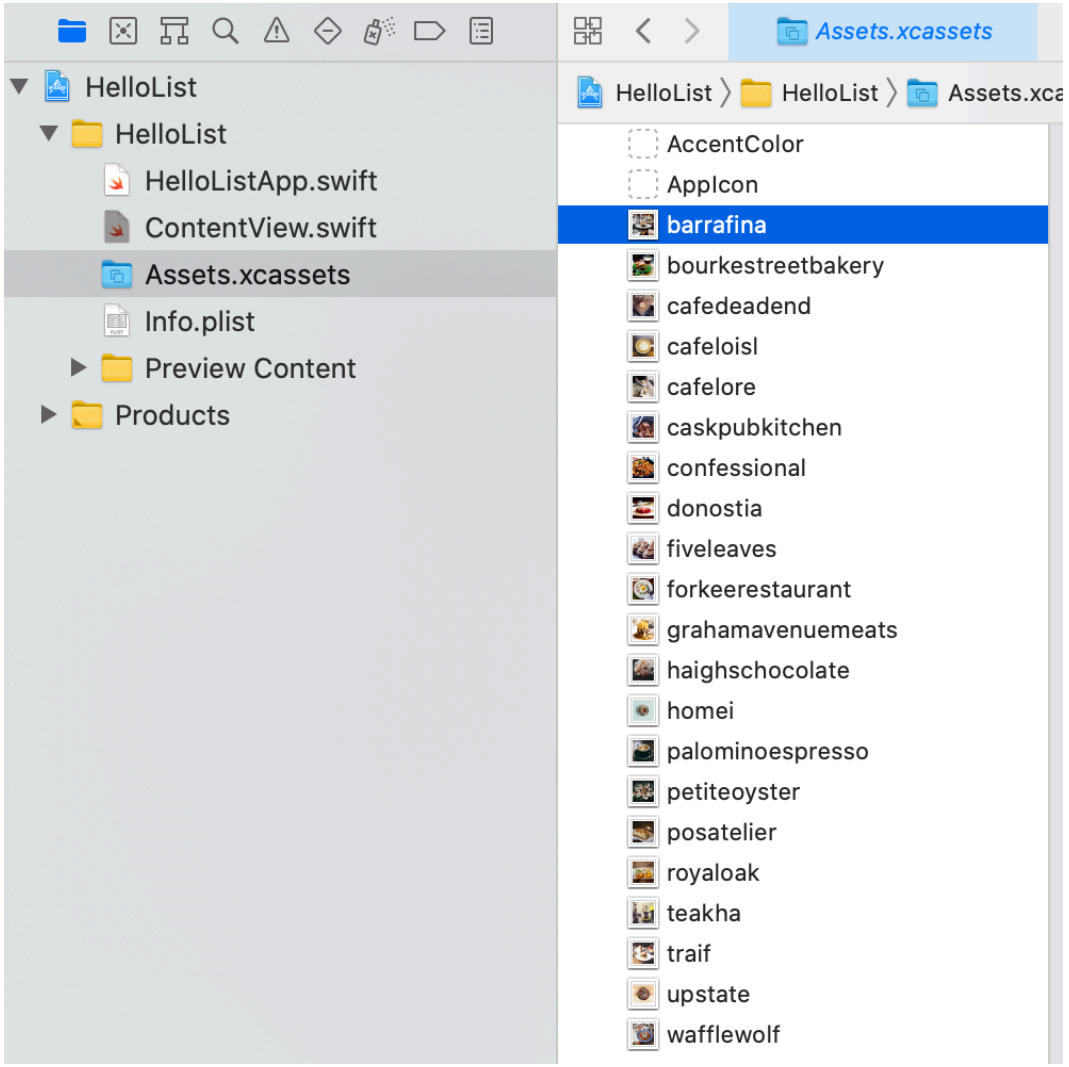
- 用迴圈加入清單項目

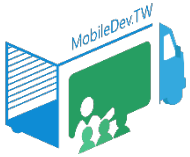
```
import SwiftUI

struct ContentView: View {
    var body: some View {
        List{
                ForEach(1...4, id:\.self){ index in
                 Text("Item \(index)")
                }
        }
    }
}

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



將預先準備好的小圖放入專案中

- Assets.xcassets





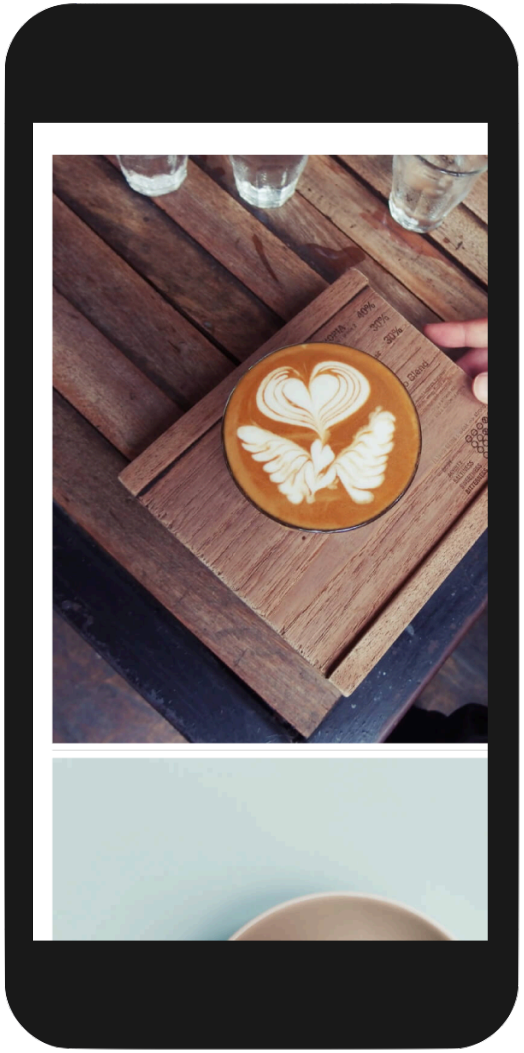
加入項目名稱與圖片名稱

- 宣告在ContentView中

```
import SwiftUI

struct ContentView: View {
    var restaurantNames = ["Cafe Deadend", "Homei", "Teakha",
"Cafe Loisl", "Petite Oyster", "For Kee Restaurant", "Po's
Atelier", "Bourke Street Bakery", "Haigh's Chocolate",
"Palomino Espresso", "Upstate", "Traif", "Graham Avenue
Meats And Deli", "Waffle & Wolf", "Five Leaves", "Cafe Lore",
"Confessional", "Barrafina", "Donostia", "Royal Oak", "CASK
Pub and Kitchen"]


    var restaurantImages = ["cafedeadend", "homei", "teakha",
"cafeloisl", "petiteoyster", "forkeerestaurant",
"posatelier", "bourkestreetbakery", "haighschocolate",
"palominoespresso", "upstate", "traif", "grahamavenuemeats",
"wafflewolf", "fiveleaves", "cafelore", "confessional",
"barrafina", "donostia", "royaloak", "caskpubkitchen"]
```

改寫List來套用資料

- 直接作用在List上
  - 圖片
  - 文字
  - Hstack
- 結果可以滑，但圖片太大

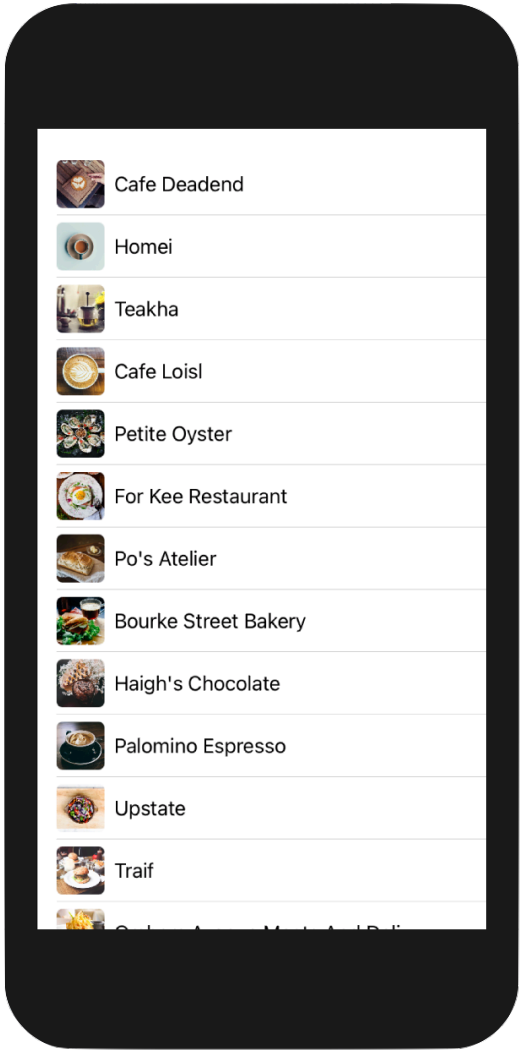


```
var body: some View {
        List(restaurantNames.indices, id:\.self){
            index in
            HStack{
                Image(self.restaurantImages[index])
                Text(self.restaurantNames[index])
            }
        }
}
```

調整圖片呈現大小

- 40x40、圓角邊框

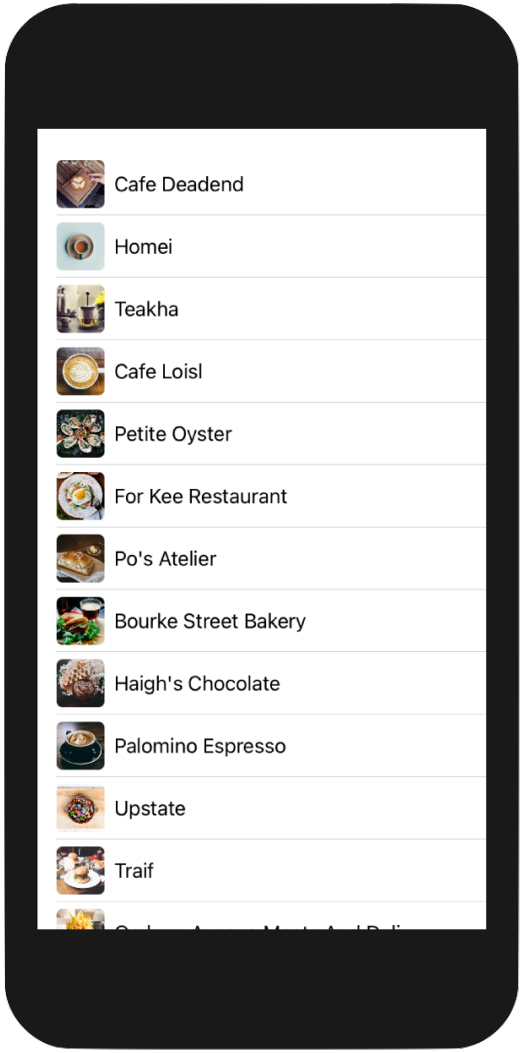
```
var body: some View {
        List(restaurantNames.indices, id:\.self){
            index in
            HStack{
                Image(self.restaurantImages[index])
                    .resizable()
                    .frame(width: 40, height: 40)
                    .cornerRadius(5)
                Text(self.restaurantNames[index])
            }
        }
}
```



重構資料

- 合併餐廳名稱與餐廳圖片

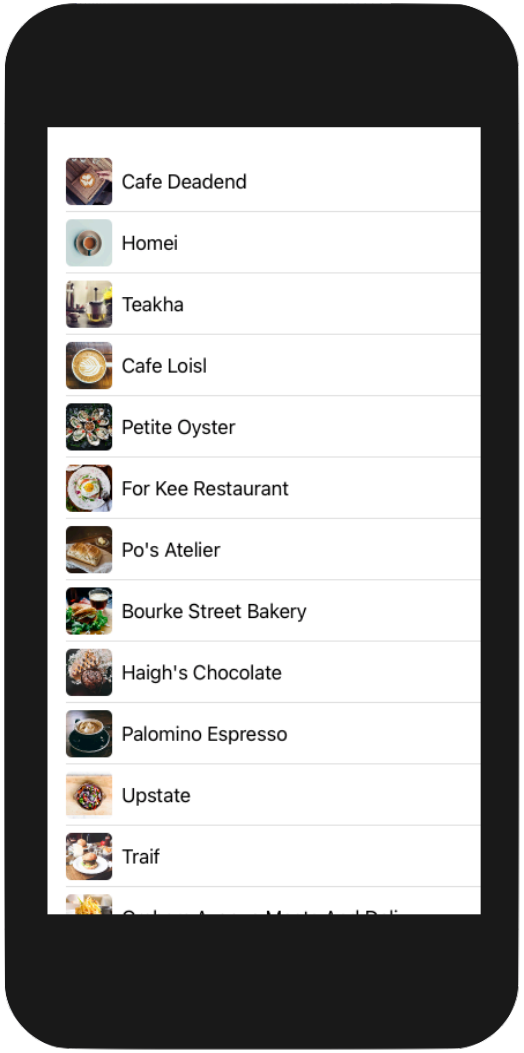
```
struct ContentView: View {

    struct Restaurant {
        var name:String
        var image:String
    }

    var restaurants = [
        Restaurant(name: "Cafe Deadend", image: "cafedeadend"),
        Restaurant(name: "Homei", image: "homei"),
        Restaurant(name: "Teakha", image: "teakha"),
        Restaurant(name: "Cafe Loisl", image: "cafeloisl"),
        Restaurant(name: "Petite Oyster", image: "petiteoyster"),
        Restaurant(name: "For Kee Restaurant", image: "forkeerestaurant"),
        Restaurant(name: "Po's Atelier", image: "posatelier"),
        Restaurant(name: "Bourke Street Bakery", image: "bourkestreetbakery"),
        Restaurant(name: "Haigh's Chocolate", image: "haighschocolate"),
        Restaurant(name: "Palomino Espresso", image: "palominoespresso"),
        Restaurant(name: "Upstate", image: "upstate"),
        Restaurant(name: "Traif", image: "traif"),
        Restaurant(name: "Graham Avenue Meats And Deli", image: "grahamavenuemeats"),
        Restaurant(name: "Waffle & Wolf", image: "wafflewolf"),
        Restaurant(name: "Five Leaves", image: "fiveleaves"),
        Restaurant(name: "Cafe Lore", image: "cafelore"),
        Restaurant(name: "Confessional", image: "confessional"),
        Restaurant(name: "Barrafina", image: "barrafina"),
        Restaurant(name: "Donostia", image: "donostia"),
        Restaurant(name: "Royal Oak", image: "royaloak"),
        Restaurant(name: "CASK Pub and Kitchen", image: "caskpubkitchen")
    ]
```



重構資料

- 修正List寫法

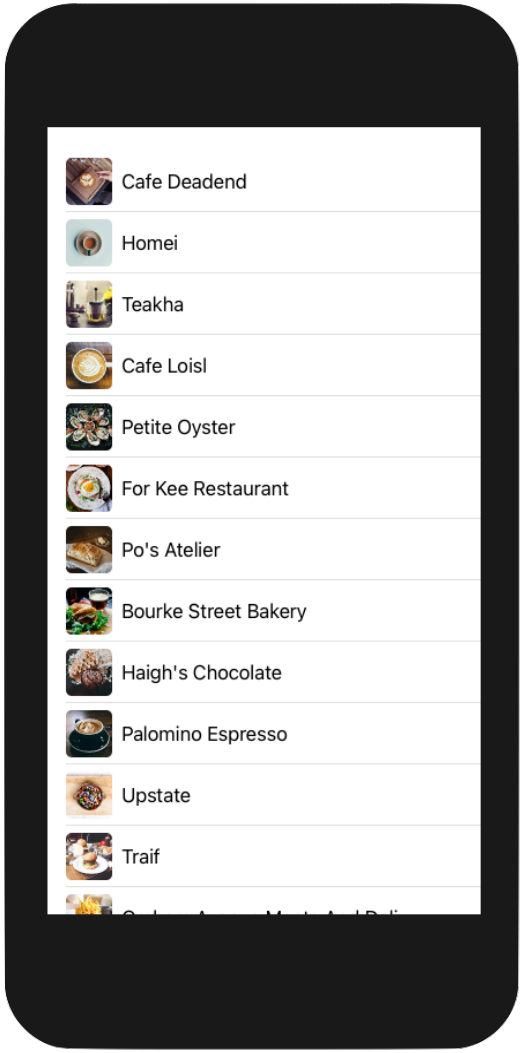
```
var body: some View {
        List(restaurants, id:\.name){
            thisRestaurant in
            HStack{
                Image(thisRestaurant.image)
                    .resizable()
                    .frame(width: 40, height: 40)
                    .cornerRadius(5)
                Text(thisRestaurant.name)
            }
        }
}
```

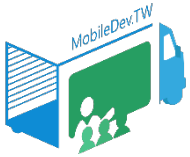


確認結構具單一識別符

- 簡化

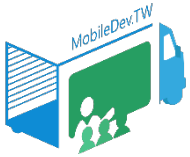
```
struct Restaurant: Identifiable{
        var id = UUID()
        var name:String
        var image:String
    }


var body: some View {
        List(restaurants){ thisRestaurant in
            HStack{
                Image(thisRestaurant.image)
                    .resizable()
                    .frame(width: 40, height: 40)
                    .cornerRadius(5)
                Text(thisRestaurant.name)
            }
        }
    }
```





清單與細節

- 延續前一個範例
- 點擊查看餐廳細節



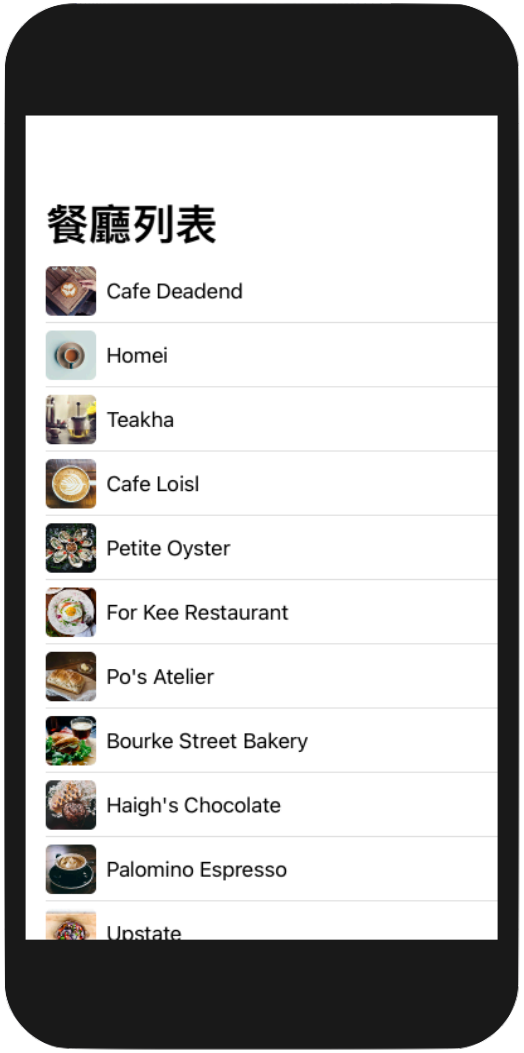
複製專案

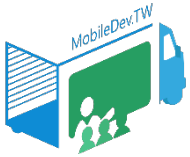
- 複製HelloList專案
- 修改資料夾名稱作為區隔

在List外包上一層導覽View

- NavigationView
- 設定導覽列標題文字

```
var body: some View {
        NavigationView{
            List(restaurants){ restaurantItem in
                BasicImageRow(thisRestaurant: restaurantItem)
              }
            .navigationBarTitle("餐廳列表")
        }
    }
```





先準備細節頁的內容

- 放在ContentView中

```
struct RestaurantDetailView:View{
        var restaurant:Restaurant
        var body: some View{
            VStack{
                Image(restaurant.image)
                    .resizable()
                    .aspectRatio(contentMode: .fill)
                    .clipped()
                Text(restaurant.name)
                    .font(.system(.title, design:.rounded))
                    .fontWeight(.black)
                Spacer()
            }
        }
    }
```

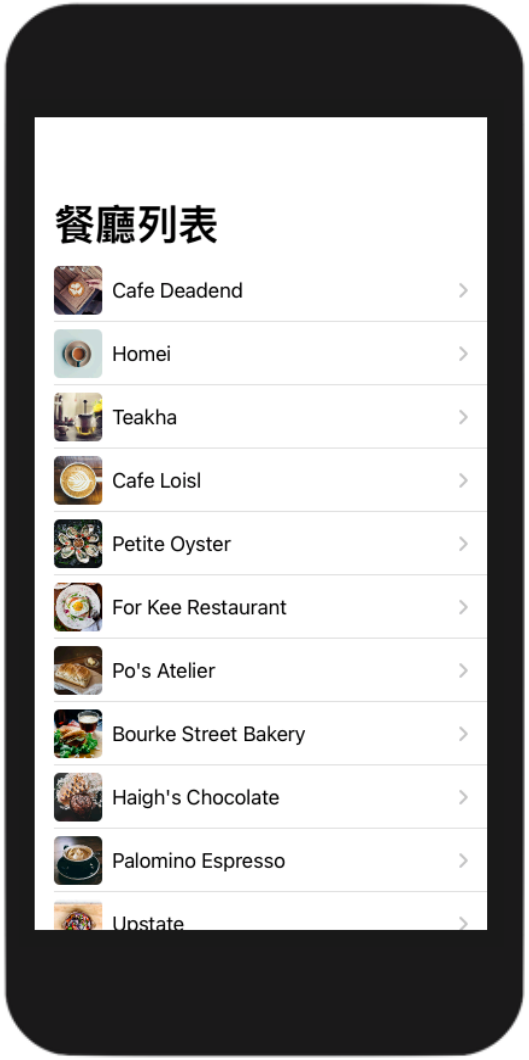
修改List增加導覽連結

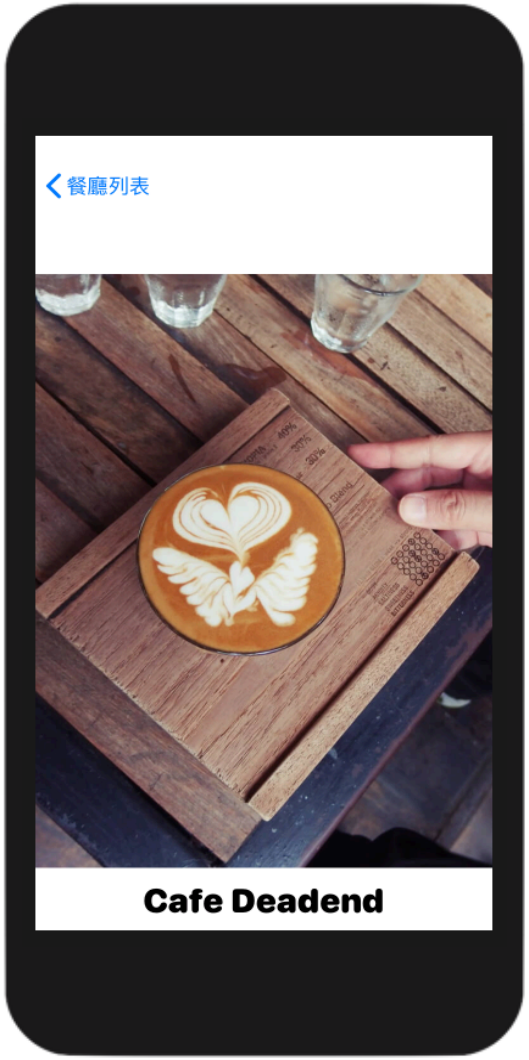
- NavigationLink

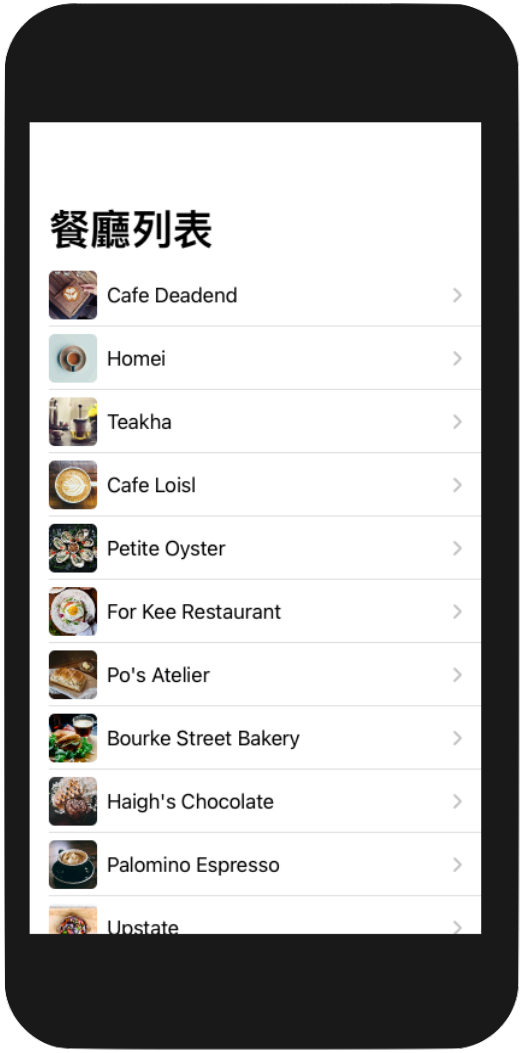
```
var body: some View {
        NavigationView{
            List(restaurants){ restaurantItem in
                NavigationLink(destination:
                  RestaurantDetailView(restaurant: restaurantItem)){
                    BasicImageRow(thisRestaurant: restaurantItem)
                }
              }
            .navigationBarTitle("餐廳列表")
        }
    }
```

測試

- 點擊項目 -> 出現細節 -> 返回清單

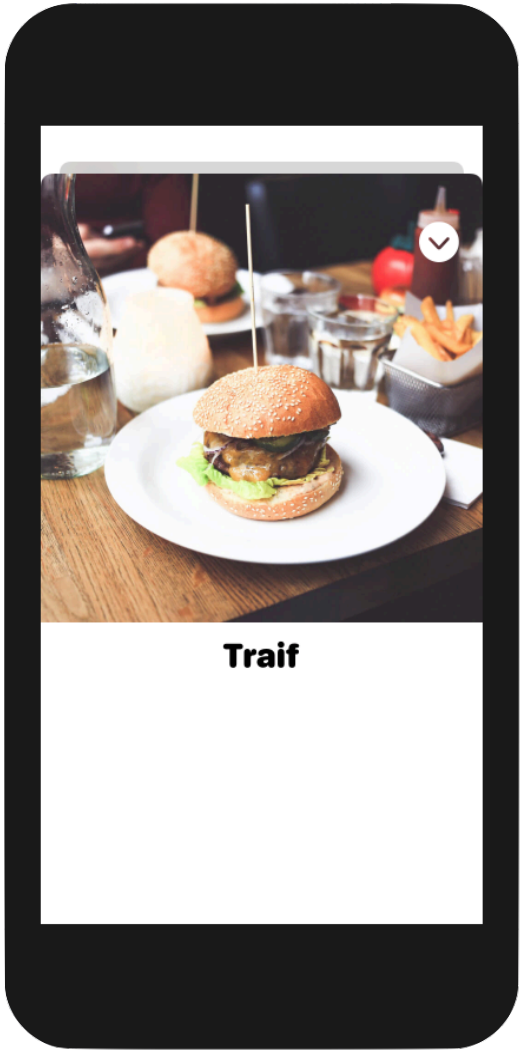


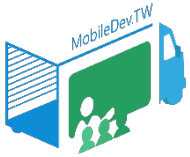




延伸：跳出頁面呈現餐廳細節

- 在餐廳列表點擊餐廳項目
  - 跳出頁面
  - 可下滑關閉或點擊右上方按鈕關閉





宣告變數

- 作為跳出頁面判斷使用

```
import SwiftUI



struct ContentView: View {

    @State var showDetailView = false
    @State var selectedRestaurant:Restaurant?
```

修改呈現細節頁面方式

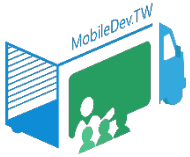
```
var body: some View {
        NavigationView{
            List(restaurants){ restaurantItem in
                    BasicImageRow(thisRestaurant: restaurantItem)
                        .onTapGesture {
                            self.showDetailView = true
                            self.selectedRestaurant = restaurantItem
                        }
            }
            .sheet(item:self.$selectedRestaurant){ thisRestaurant in
                     RestaurantDetailView(restaurant: thisRestaurant)
             }
            /*
            List(restaurants){ restaurantItem in
                NavigationLink(destination:
                  RestaurantDetailView(restaurant: restaurantItem)){
                    BasicImageRow(thisRestaurant: restaurantItem)
                }
              }
            */
            .navigationBarTitle("餐廳列表")
        }
    }
```

增加跳出頁面右上角關閉按鈕

```
struct RestaurantDetailView:View{
        @Environment(\.presentationMode) var presentationMode
        var restaurant:Restaurant
        var body: some View{
            ScrollView{
                VStack{
                    Image(restaurant.image)
                        .resizable()
                        .aspectRatio(contentMode: .fill)
                        .clipped()
                    Text(restaurant.name)
                        .font(.system(.title, design:.rounded))
                        .fontWeight(.black)
                    Spacer()
                }
            }
            .overlay(
                HStack{
                    Spacer()
                    VStack{
                        Button(action:{
                            self.presentationMode.wrappedValue.dismiss()
                        },label:{
                            Image(systemName:"chevron.down.circle.fill")
                                .font(.largeTitle)
                                .foregroundColor(.white)
                        })
                        .padding(.trailing, 20)
                        .padding(.top, 40)
                        Spacer()
                    }
                }
            )
        }
    }
```

請求使用者回應的頁面

- 背景顏色變更



HelloAlert

- 建立專案HelloAlert (iOS -> App)
- 在ContentView中增加兩個變數

```
import SwiftUI


struct ContentView: View {

    @State var showAlert = false
    @State var colorArray:Array = [255.0,255.0,255.0]
```

層疊一個矩形作為背景

- 底下一層矩形
- 上面有一個按鈕，按鈕有設定alert顯示

```
var body: some View {
        ZStack{
            Rectangle()
            Button(){
            }
             .alert(){
            }
        }
```

矩形設定

- 顏色用變數

```
Rectangle()
     .fill(Color(red: colorArray[0]/255, green: colorArray[1]/255, blue: colorArray[2]/255))
     .frame(minWidth: 0, idealWidth: 100, maxWidth: .infinity,
            minHeight: 0, idealHeight: 100, maxHeight: .infinity, alignment: .center)
     .edgesIgnoringSafeArea(.all)
```

按鈕設定

- 顏色用變數

```
Button(action:{ self.showAlert = true }){
                Text("把背景換成紅色")
                .frame(width: 300, height: 100, alignment: .center)
                .font(.system(size: 30))
                .background(Color.purple)
                .padding(.horizontal,5)
                .foregroundColor(.white)
                .border(Color.purple, width: 30)
                .cornerRadius(20)
        }
        .alert(isPresented:$showAlert){
                Alert(title: Text("背景顏色變更"),
                      message: Text("你是否確定要把背景顏色變成紅色?"),
                      primaryButton: .default(Text("確定"),
                        action: {colorArray = [255.0, 0.0, 0.0]}),
                      secondaryButton:.default(Text("取消"),
                        action: {colorArray = [255.0, 255.0, 255.0]}))
        }
```

提供使用者更多選擇

- ActionSheet

複製上一個專案

- Button顯示文字修改

```
Button(action:{
                self.showAlert = true
            }){
                Text("設定背景顏色")
                .frame(width: 300, height: 100, alignment: .center)
                .font(.system(size: 30))
                .background(Color.purple)
                .padding(.horizontal,5)
                .foregroundColor(.white)
                .border(Color.purple, width: 30)
                .cornerRadius(20)
            }
```

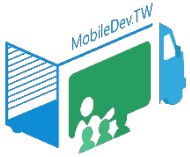
Alert -> ActionSheet

- 將Alert換成ActionSheet
- 增加選項

```
.actionSheet(isPresented:$showAlert){
                ActionSheet(title: Text("背景顏色變更"),
                      message: Text("請選擇"),
                      buttons:[
                        .default(Text("紅色")){colorArray = [255.0, 0.0, 0.0]},
                        .default(Text("綠色")){colorArray = [0.0, 255.0, 0.0]},
                        .default(Text("藍色")){colorArray = [0.0, 0.0, 255.0]},
                        .default(Text("白色")){colorArray = [255.0, 255.0, 255.0]},
                        .cancel()
                      ])
            }
```

設定頁面

- 更多控制項目
  - 條列選擇
  - 開關
  - 計次
  - 滑桿



ContentView.swift

- 新建專案HelloSettings，進入ContentView
- 宣告變數

```
import SwiftUI


struct ContentView: View {

    private var displayFontType =
                    [".default", ".rounded",".monospaced",".serif"]
    @State var displayFontSelected = 0
    @State var IsDeepScheme = false
    @State var colorArray:Array = [255.0,255.0,255.0]
    @State var stepperValue = 0
    @State var sliderValue = 0.0
```

NavigationView -> Form

```
import SwiftUI


struct ContentView: View {

    private var displayFontType = [".default", ".rounded",".monospaced",".serif"]
    @State var displayFontSelected = 0
    @State var IsDeepScheme = false
    @State var colorArray:Array = [255.0,255.0,255.0]
    @State var stepperValue = 0
    @State var sliderValue = 0.0
    var body: some View {
            NavigationView{
                Form{

                }
                .navigationBarTitle("Settings 設定")
            }
     }
}


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

Form -> Section

- Picker

```
Form{
      Section(header: Text("字型設定")){
          Picker(selection:$displayFontSelected,
                 label:Text("字型選擇(\(displayFontSelected))")){
              ForEach(0..<displayFontType.count,id:\.self){
                  Text(self.displayFontType[$0])
              }
          }
      }
 }
```

Form -> Section

- Toggle

```
Form{
      Section(header: Text("字型設定")){
          Picker(selection:$displayFontSelected,
                 label:Text("字型選擇(\(displayFontSelected))")){
              ForEach(0..<displayFontType.count,id:\.self){
                  Text(self.displayFontType[$0])
              }
          }
      }
     Section(header: Text("背景風格")){
           Toggle(isOn:$IsDeepScheme){
               Text("深色(\(String(IsDeepScheme)))")
           }
       }

 }
```

Form -> Section

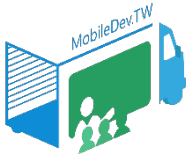
- Stepper

```
Form{
      Section(header: Text("字型設定")){
          Picker(selection:$displayFontSelected,
                 label:Text("字型選擇(\(displayFontSelected))")){
              ForEach(0..<displayFontType.count,id:\.self){
                  Text(self.displayFontType[$0])
              }
          }
      }
     Section(header: Text("背景風格")){
           Toggle(isOn:$IsDeepScheme){
               Text("深色(\(String(IsDeepScheme)))")
           }
       }
     Section(header: Text("計數器")){
           Stepper("Stepper(\(stepperValue))",
                   onIncrement: {stepperValue+=1},
                   onDecrement: {stepperValue-=1})
       }
 }
```

Form -> Section

- Slider

```
Form{
      Section(header: Text("字型設定")){
          Picker(selection:$displayFontSelected,
                 label:Text("字型選擇(\(displayFontSelected))")){
              ForEach(0..<displayFontType.count,id:\.self){
                  Text(self.displayFontType[$0])
              }
          }
      }
    Section(header: Text("背景風格")){
          Toggle(isOn:$IsDeepScheme){
              Text("深色(\(String(IsDeepScheme)))")
          }
      }
    Section(header: Text("計數器")){
          Stepper("Stepper(\(stepperValue))",
                  onIncrement: {stepperValue+=1},
                  onDecrement: {stepperValue-=1})
      }
    Section(header: Text("滑桿(\(sliderValue,
                    specifier: "%.2f"))")){
          Slider(value: $sliderValue, in: 0...1)
      }
 }
```

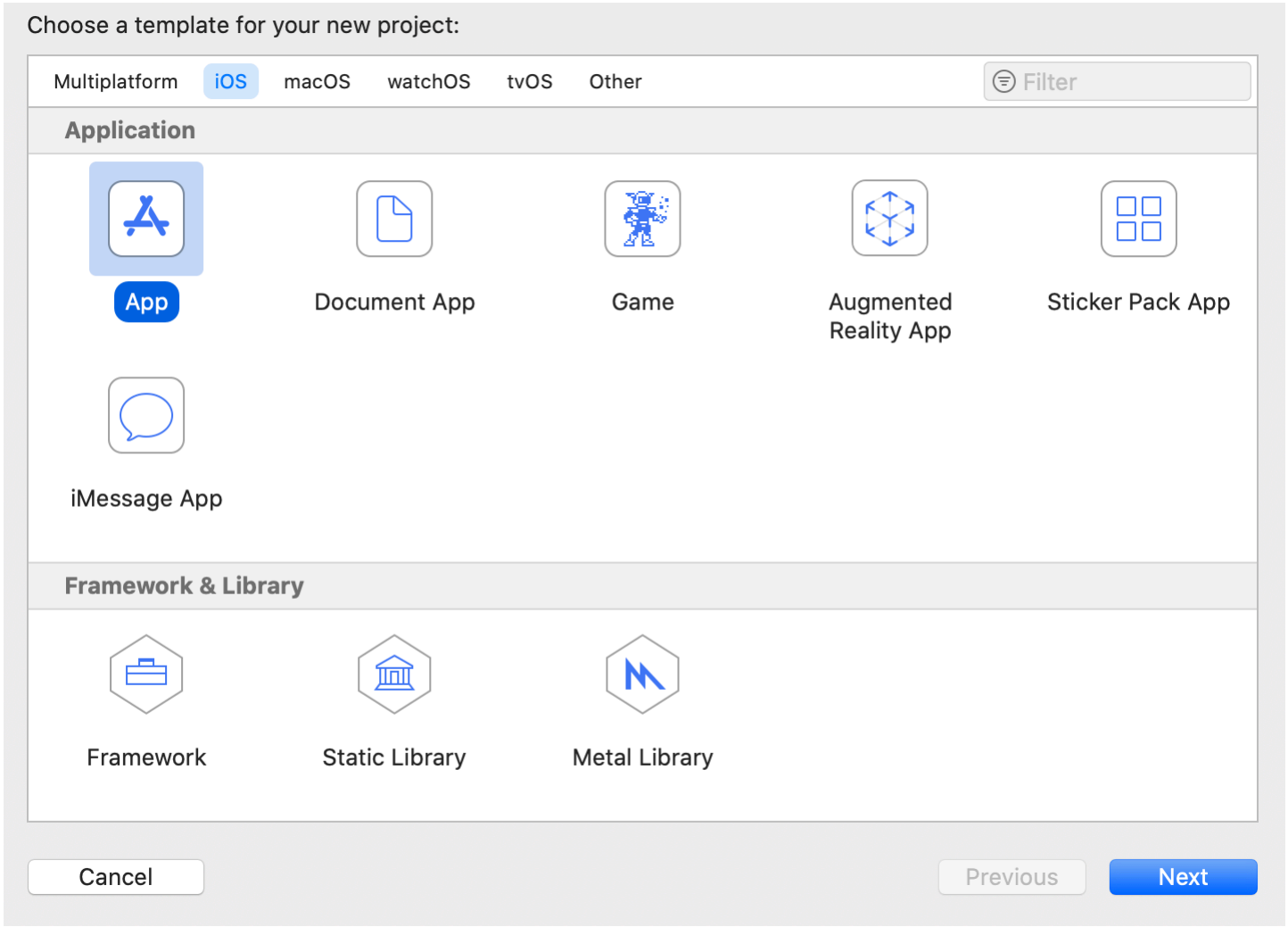


捲動的世界

- ScrollView
- Carousel UI

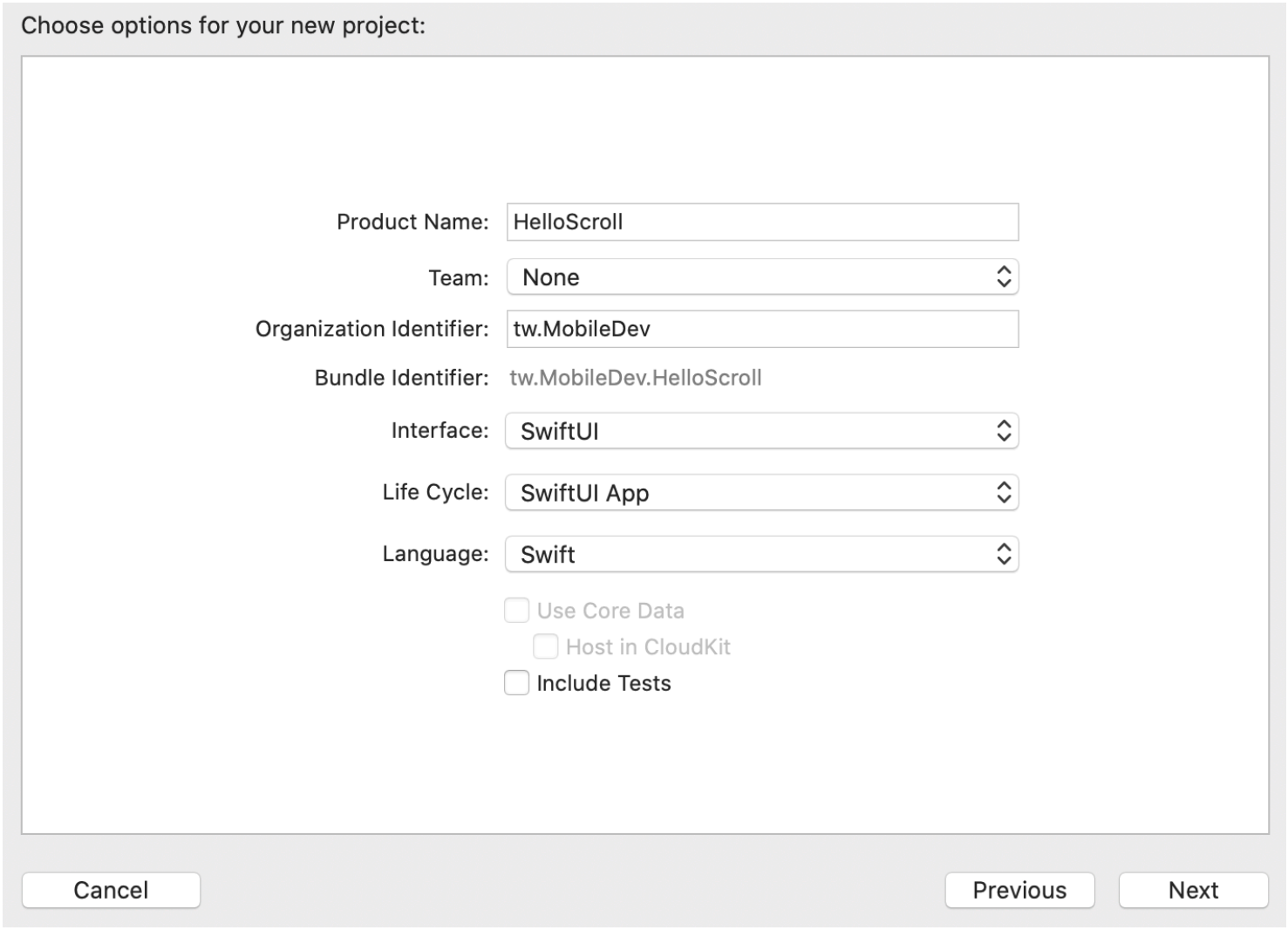
建立新的專案

- iOS -> App



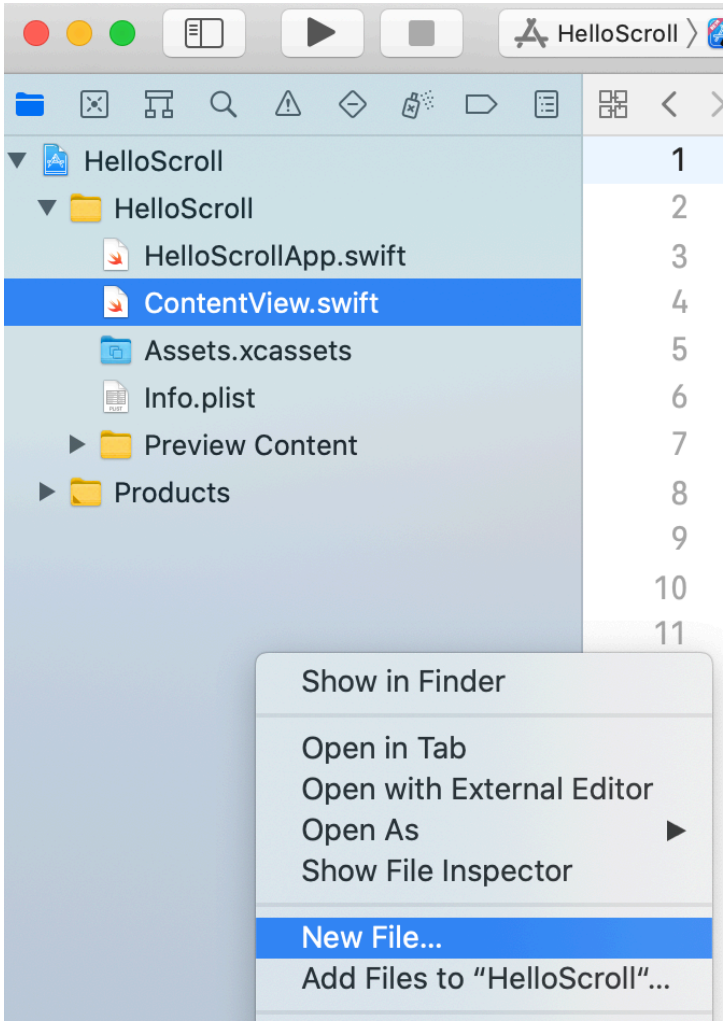
HelloScroll

- SwiftUI



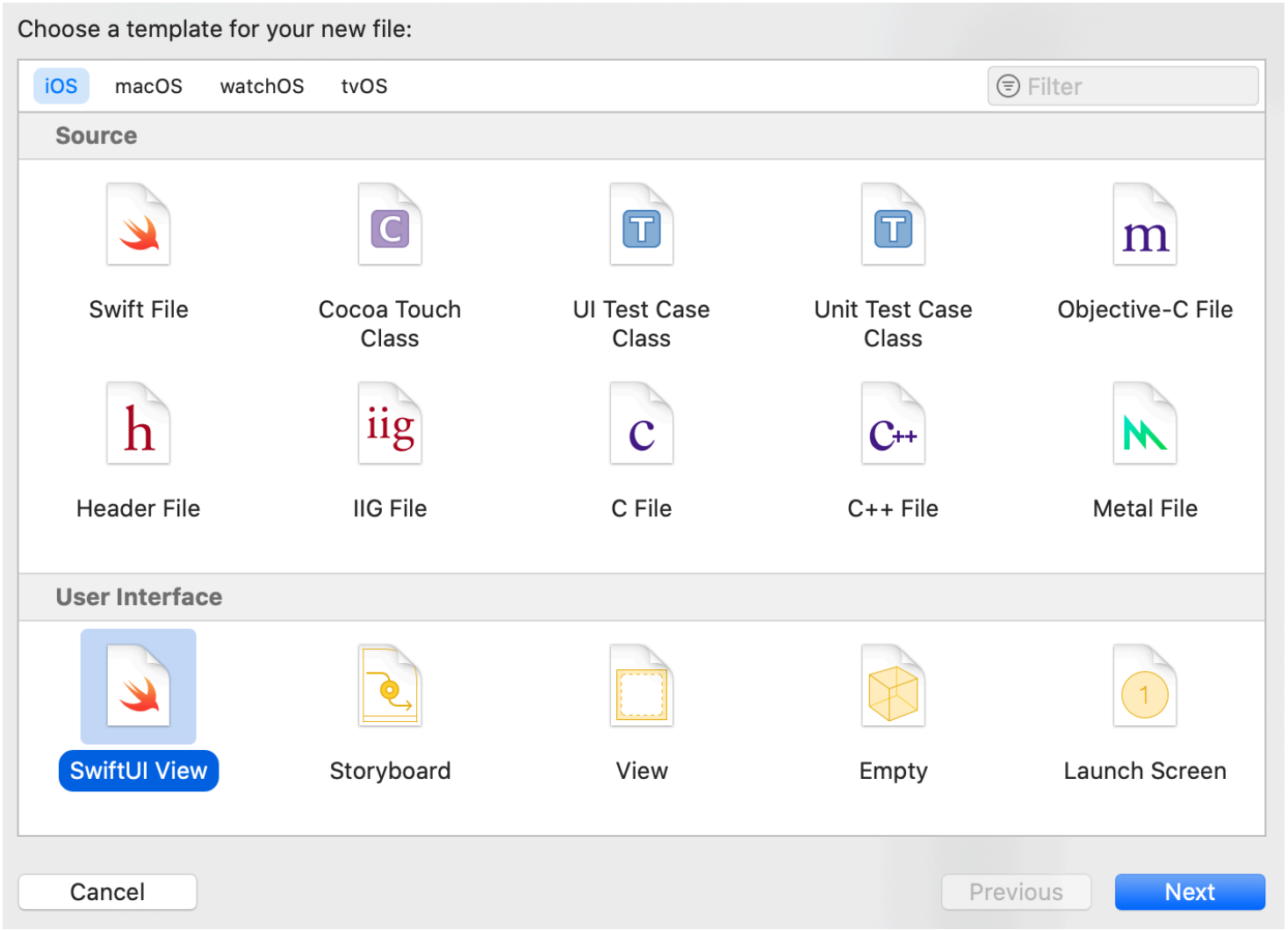
新增檔案

- 導覽區 -> 右鍵 -> New File...



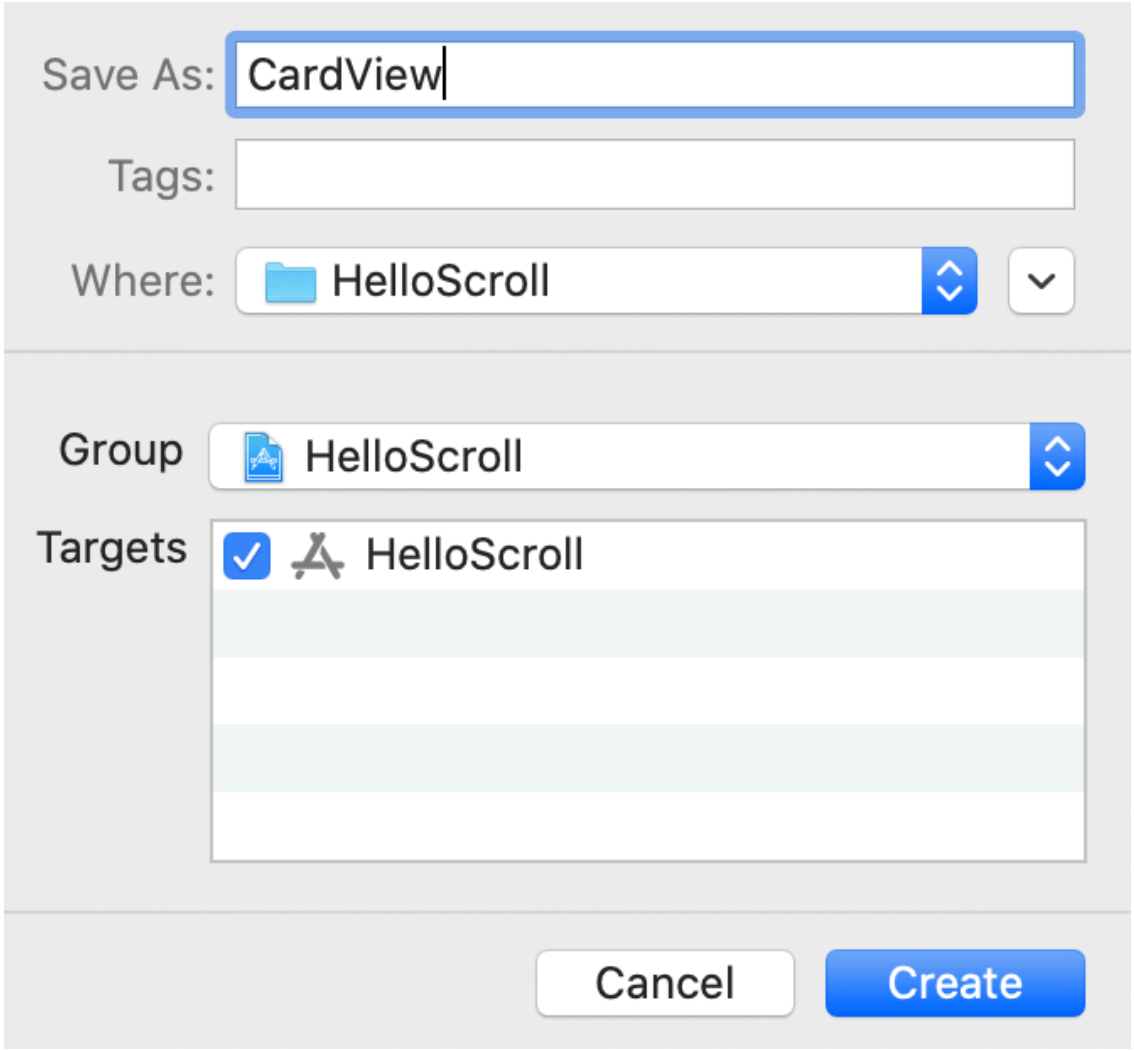
選擇樣板

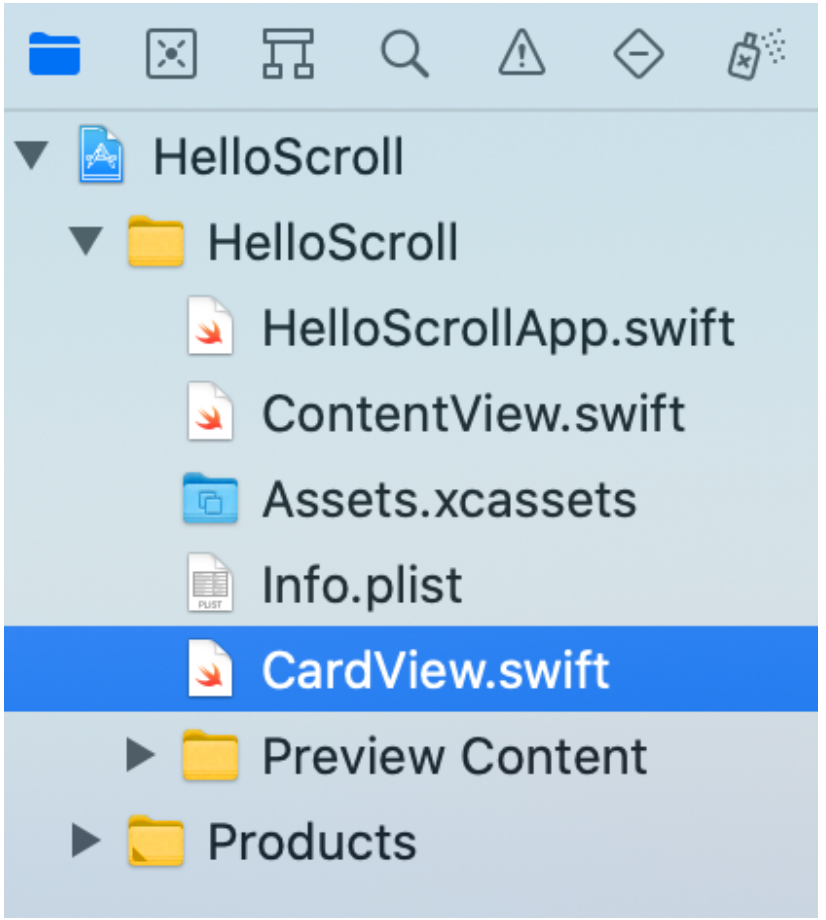
- User Interface -> SwiftUI Views



命名

- CardView -> Create 確定放置層次



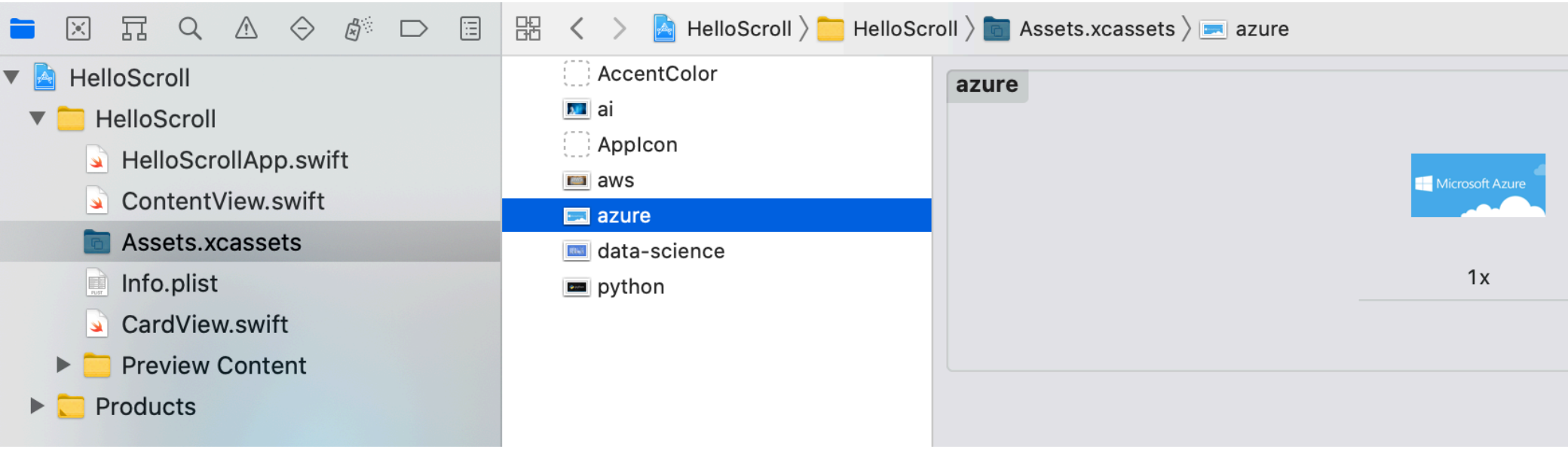


素材準備

- 至少5份
  - 圖片
  - 主題
  - 描述
  - 副標題
- 例如
  - 5個喜歡的藝人
  - 5部喜歡的電影
  - 5個喜歡的遊戲

將素材加入至專案中

- 左邊導覽列 -> Assets.xcassets



先做出第一張卡片 CardView.swift

- 圖片+文字(主標題/次標題/描述)

```
struct CardView: View {
    var body: some View {
        VStack{
            Image("ai")
                .resizable()
                .aspectRatio(contentMode:.fit)
            VStack(alignment:.leading){
                Text("Intro to AI")
                    .font(.headline)
                    .foregroundColor(.secondary)
                Text("人工智慧導論")
                    .font(.title)
                    .fontWeight(.black)
                    .foregroundColor(.primary)
                    .lineLimit(3)
                Text("12hr")
                    .font(.caption)
                    .foregroundColor(.purple)
            }
        }
    }
}
```

文字靠左對齊

- infinity, leading, padding

```
struct CardView: View {
    var body: some View {
        VStack{
            Image("ai")
                .resizable()
                .aspectRatio(contentMode:.fit)

            VStack(alignment:.leading){
                Text("Intro to AI")
                    .font(.headline)
                    .foregroundColor(.secondary)
                Text("人工智慧導論")
                    .font(.title)
                    .fontWeight(.black)
                    .foregroundColor(.primary)
                    .lineLimit(3)
                Text("12hr")
                    .font(.caption)
                    .foregroundColor(.purple)
            }
            .frame(minWidth: 0, idealWidth: 100,
                   maxWidth: .infinity, alignment: .leading)
            .padding(.horizontal,10)
        }
    }
}
```

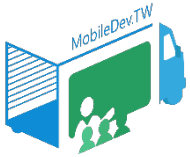
調整版面，變得更像卡片

```
struct CardView: View {
    var body: some View {
        VStack{
            Image("ai")
                .resizable()
                .aspectRatio(contentMode:.fit)
            VStack(alignment:.leading){
                Text("Intro to AI")
                    .font(.headline)
                    .foregroundColor(.secondary)
                Text("人工智慧導論")
                    .font(.title)
                    .fontWeight(.black)
                    .foregroundColor(.primary)
                    .lineLimit(3)
                Text("12hr")
                    .font(.caption)
                    .fontWeight(.bold)
                    .foregroundColor(.purple)
            }
            .frame(minWidth: 0, idealWidth: 100, maxWidth: .infinity,
                   alignment: .leading)
            .padding(.leading,15)
            .padding(.bottom, 10)
        }
        .background(Color(red: 255/255, green: 204/255, blue: 153/255))
        .cornerRadius(20)
        .overlay(
            RoundedRectangle(cornerRadius: 20)
                .stroke(Color.gray,lineWidth: 2)
        )
        .padding(.all, 10)
    }
}
```

重構CardView

- 宣告變數：圖片、課程中英文名稱、時數

```
import SwiftUI


struct CardView: View {
    var image:String
    var courseNameUS:String
    var courseNameTW:String
    var courseHours:String

    var body: some View {
        VStack{
            Image(image)
                .resizable()
                .aspectRatio(contentMode:.fit)
            VStack(alignment:.leading){
                Text(courseNameUS)
                    .font(.headline)
                    .foregroundColor(.secondary)
                Text(courseNameTW)
                    .font(.title)
                    .fontWeight(.black)
                    .foregroundColor(.primary)
```



修改呼叫

- 將內容變成傳入參數

```
struct CardView_Previews: PreviewProvider {
    static var previews: some View {
        CardView(image: "ai", courseNameUS: "Intro to AI",
                 courseNameTW: "人工智慧導論", courseHours: "12hr")
    }
}
```

輸入5門課程資料

```
struct CardView_Previews: PreviewProvider {
    static var previews: some View {
        VStack{
            CardView(image: "ai", courseNameUS: "Intro to AI",
                     courseNameTW: "人工智慧導論", courseHours: "12hr")
            CardView(image: "python", courseNameUS: "Intro to Python",
                     courseNameTW: "Python程式語言", courseHours: "24hr")
            CardView(image: "azure", courseNameUS: "Intro to Azure",
                     courseNameTW: "Azure操作入門", courseHours: "18hr")
            CardView(image: "data-science", courseNameUS: "Intro to Data Science",
                     courseNameTW: "資料科學導論", courseHours: "12hr")
            CardView(image: "aws", courseNameUS: "Intro to AWS",
                     courseNameTW: "AWS操作入門", courseHours: "18hr")
        }
    }
}
```

但卻變成了...

- 配合畫面壓縮
- 無法滾動

加上ScrollView

```
struct CardView_Previews: PreviewProvider {
    static var previews: some View {
        ScrollView{
            VStack{
                CardView(image: "ai", courseNameUS: "Intro to
                    AI", courseNameTW: "人工智慧導論",
                    courseHours: "12hr")
                CardView(image: "python", courseNameUS:
                    "Intro to Python", courseNameTW: "Python
                    程式語言", courseHours: "24hr")
                CardView(image: "azure", courseNameUS: "Intro
                    to Azure", courseNameTW: "Azure操作入門",
                    courseHours: "18hr")
                CardView(image: "data-science", courseNameUS:
                    "Intro to Data Science", courseNameTW: "
                    資料科學導論", courseHours: "12hr")
                CardView(image: "aws", courseNameUS: "Intro
                    to AWS", courseNameTW: "AWS操作入門",
                    courseHours: "18hr")
            }
        }
    }
}
```

增加上方說明文字

- 副標題、主標題

```
struct CardView_Previews: PreviewProvider {
    static var previews: some View {
        ScrollView{
            VStack{
                Text("Mobile Development Academy")
                    .frame(minWidth: 0, idealWidth: 100,
                           maxWidth: .infinity, alignment: .leading)
                    .padding(.all, 10)
                    .foregroundColor(.secondary)
                Text("行動開發學院培訓課程")
                    .font(.title)
                    .frame(minWidth: 0, idealWidth: 100,
                            maxWidth: .infinity, alignment: .leading)
                    .padding(.leading, 10)
                CardView(image: "ai", courseNameUS: "Intro to AI",
                         courseNameTW: "人工智慧導論", courseHours: "12hr")
                CardView(image: "python", courseNameUS: "Intro to Python",
                         courseNameTW: "Python程式語言", courseHours: "24hr")
```

延伸練習

- 橫向捲動版本

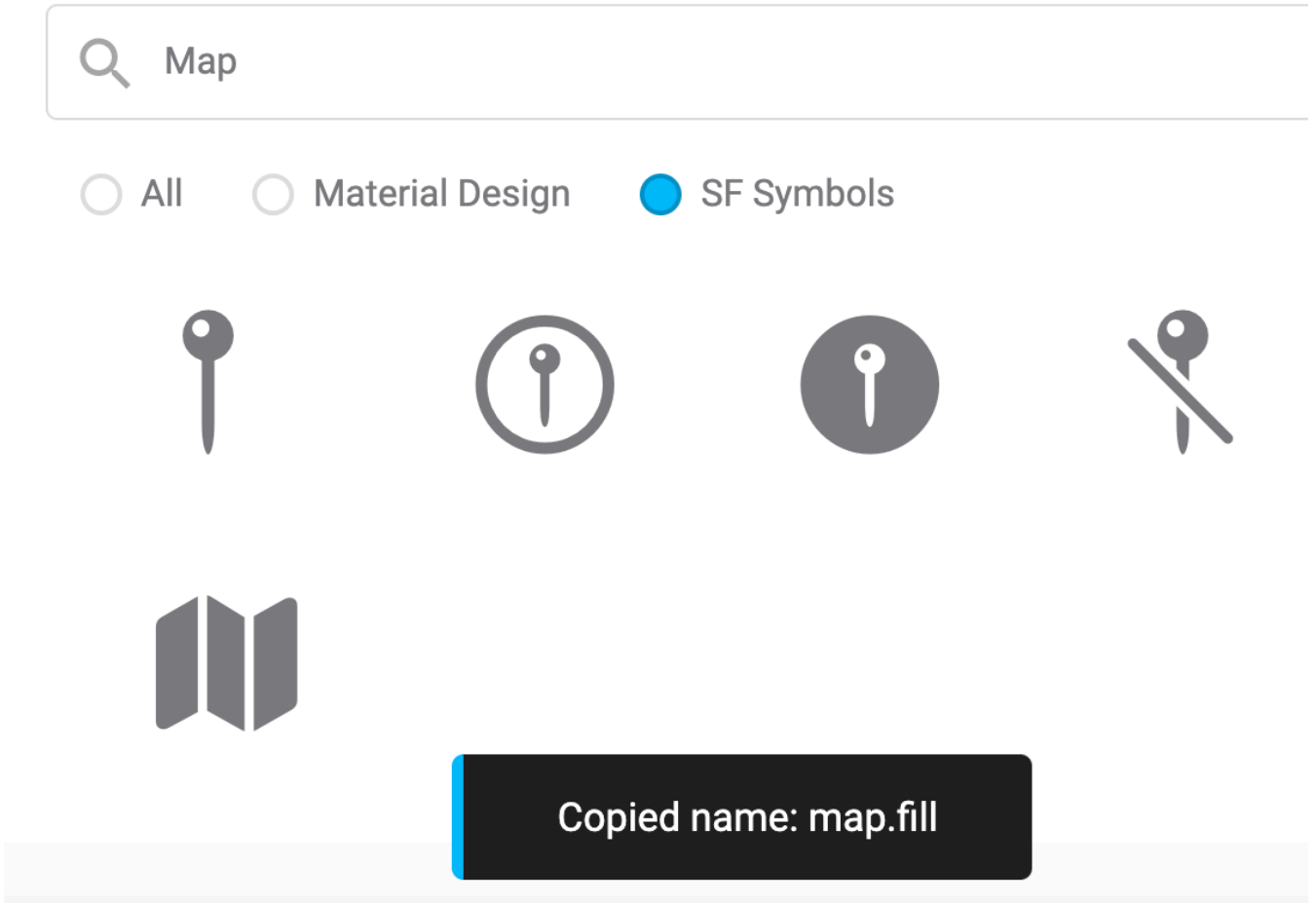
```
struct CardView_Previews: PreviewProvider {
    static var previews: some View {
            VStack{
                Text("行動開發學院培訓課程")
                    .font(.title)
                Text("Mobile Development Academy")
                    .foregroundColor(.secondary)
                ScrollView(.horizontal){
                    HStack{
                        CardView(image: "ai", courseNameUS:
                                 "Intro to AI", courseNameTW:
                                 "人工智慧導論", courseHours: "12hr")
                        .frame(width: 300, height: 300)
                        CardView(image: "python", courseNameUS:
                        "Intro to Python", courseNameTW:
                        "Python程式語言", courseHours: "24hr")
                        .frame(width: 300, height: 300)
```

分頁架構

- 建構一個行動開發學院 App
  - 歡迎頁面
    - 單頁服務圖文介紹
  - 課程簡介
    - 導覽清單
  - 科技小字典
    - 新知識卡片式瀏覽

挑選小圖 - SF Symbols查找

- 取得SF Symbol名稱



https://hotpot.ai/free_icons?s=sfSymbols

挑選小圖

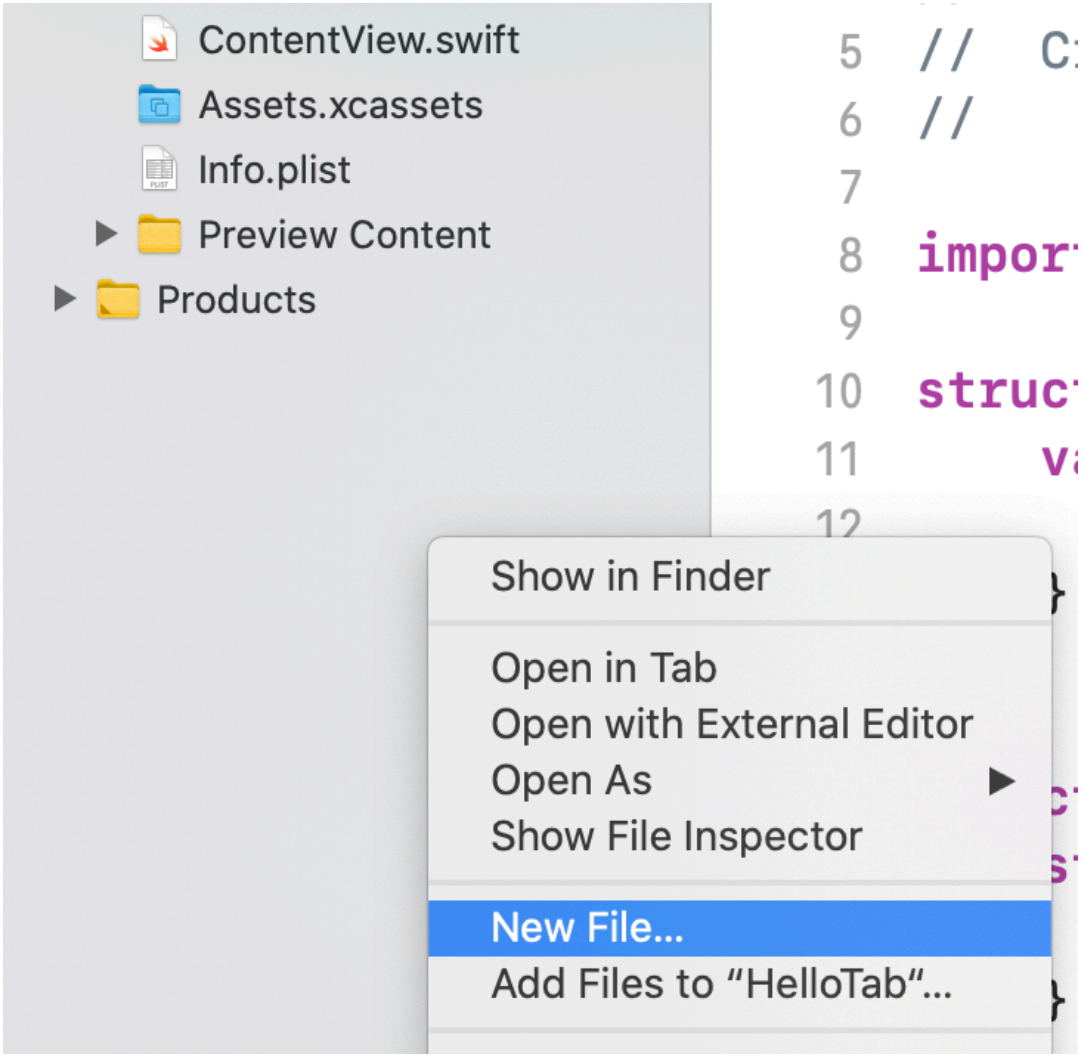
- 歡迎
  - rosette
- 課程清單
  - list.dash
- 知識卡片
  - book

分頁架構 - ContentView

```
import SwiftUI

struct ContentView: View {
    var body: some View {
        VStack{
            Text("行動開發學院")
                .font(.largeTitle)
                .fontWeight(.heavy)
                .foregroundColor(.primary)
            TabView{
                Text("第一頁")
                    .tabItem {
                        Image(systemName: "rosette")
                        Text("Welcome")
                    }
                Text("第二頁")
                    .tabItem {
                        Image(systemName: "list.dash")
                        Text("Courses")
                    }
                Text("第三頁")
                    .tabItem {
                        Image(systemName: "book")
                        Text("Learn")
                    }
            }
        }
    }
}
```

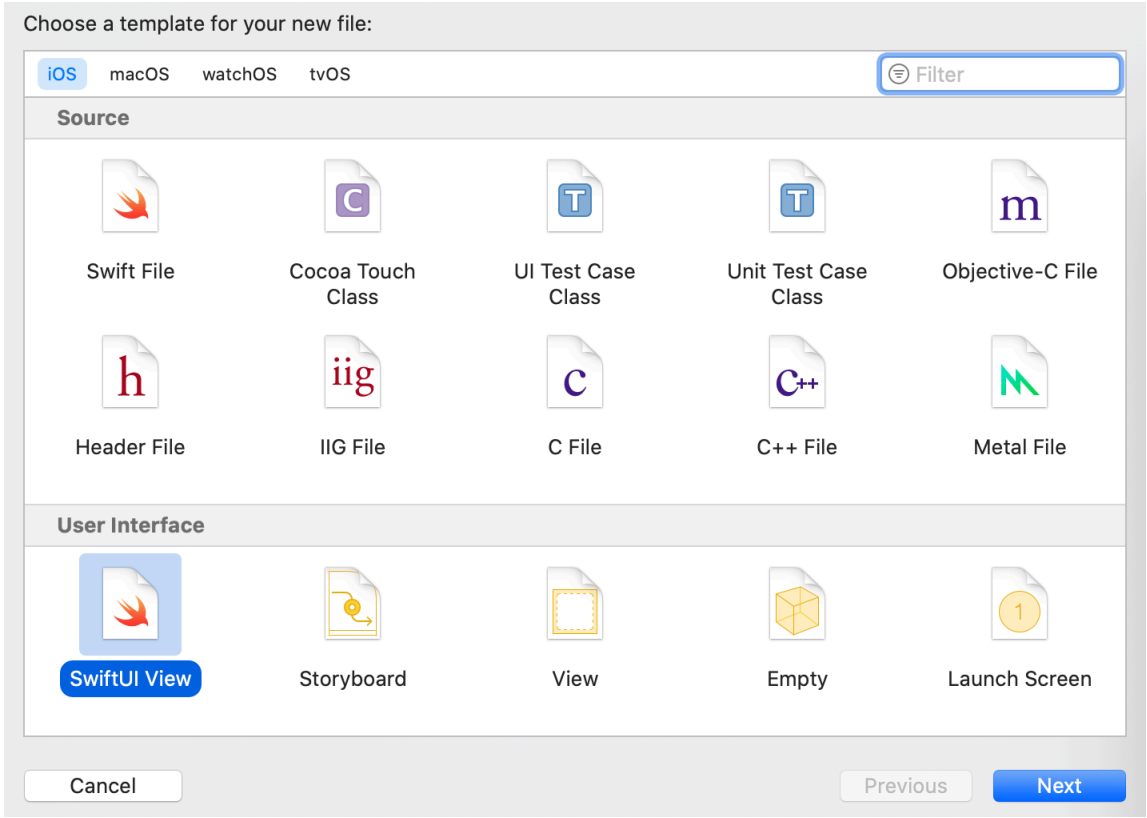
從第一頁開始

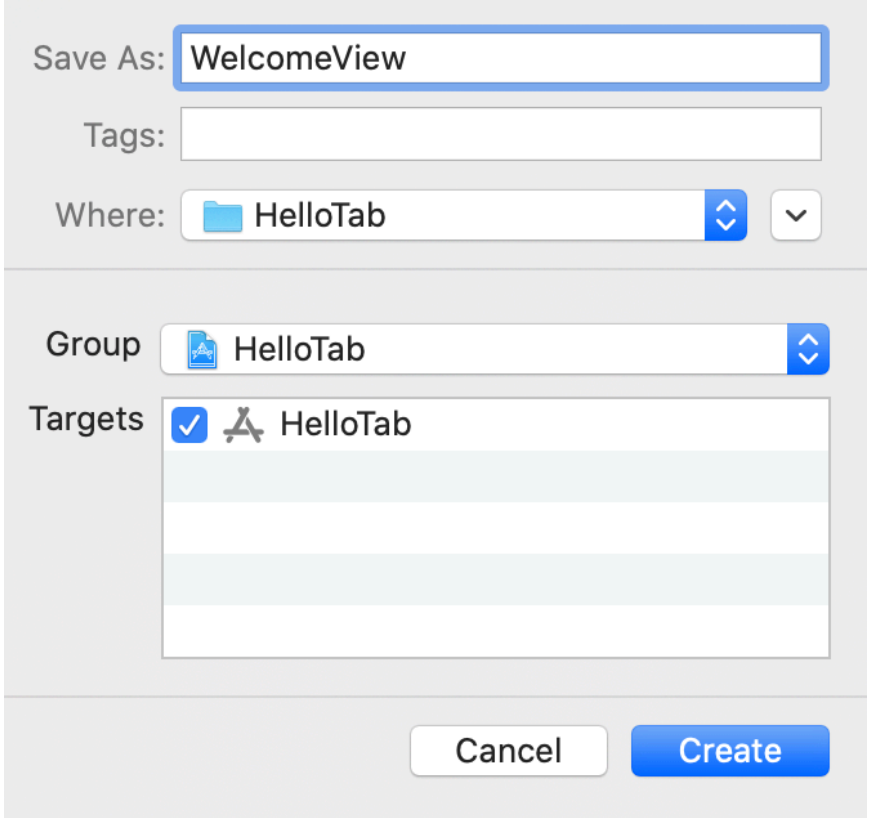
- 左邊專案導覽區->右鍵新增檔案



選擇樣板

- iOS -> User Interface -> SwiftUI View
- 命名為WelcomeView

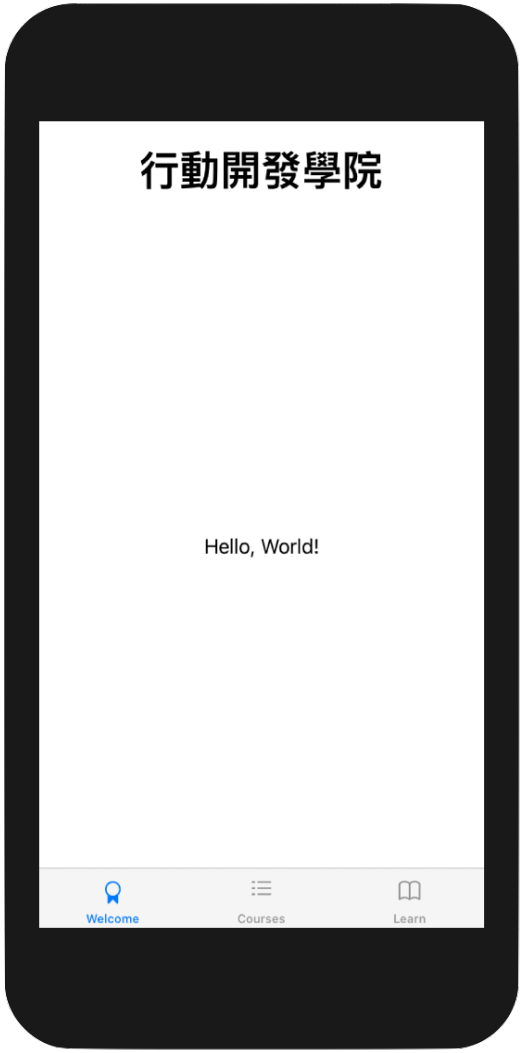




設定第一個分頁為WelcomeView

- ContentView.swift

```
struct ContentView: View {
    var body: some View {
        VStack{
            Text("行動開發學院")
                .font(.largeTitle)
                .fontWeight(.heavy)
                .foregroundColor(.primary)
            TabView{
                WelcomeView()
                    .tabItem {
                        Image(systemName: "rosette")
                        Text("Welcome")
                    }
                Text("第二頁")
                    .tabItem {
                        Image(systemName: "list.dash")
                        Text("Courses")
                    }
                Text("第三頁")
                    .tabItem {
                        Image(systemName: "book")
                        Text("Learn")
                    }
            }
        }
    }
}
```



設定WelcomeView圖文顯示

- 完成後再回到ContentView

```
struct WelcomeView: View {
    var body: some View {
        VStack{
            Image("mobiledevtw")
                .resizable()
                .aspectRatio(contentMode: .fit)
            Text("資訊技術培訓\nWeb/APP/AI應用開發")
                .fontWeight(.heavy)
                .lineSpacing(20)
                .font(.system(size: 32.0))
                .foregroundColor(.white)
                .frame(width: 350, height: 150, alignment: .center)
                .background(Color.blue)
                .cornerRadius(20.0)
                .multilineTextAlignment(.center)
        }

    }
}
```

CourseListView

- 新增檔案CourseListView.swift
- 以餐廳清單為基底，修改成課程

CourseListView.swift

- 程式架構

```
import SwiftUI

struct CourseListView: View {

    @State var showDetailView = false
    @State var selectedCourse:Course?

    struct Course: Identifiable{ ••• }

    var courses = [ ••• ]

    struct BasicImageRow: View { ••• }

    var body: some View { ••• }

    struct CourseDetailView:View{ ••• }
}

struct CourseListView_Previews: PreviewProvider {
    static var previews: some View {
        CourseListView()
    }
}
```

變數宣告

- 切換頁面用的、課程資料結構

```
import SwiftUI


struct CourseListView: View {

    @State var showDetailView = false
    @State var selectedCourse:Course?

    struct Course: Identifiable{
        var id = UUID()
        var name:String
        var image:String
        var description:String
    }
```

課程資料

- 記得放圖片至Assets.xcassets

```
var courses = [
        Course(name: "Intro to AI", image: "ai", description: "人工智慧介紹"),
        Course(name: "Intro to Python", image: "python", description: "Python介紹"),
        Course(name: "Intro to Data Science", image: "data-science", description: "資料科學介紹"),
        Course(name: "Intro to Azure", image: "azure", description: "微軟Azure介紹"),
        Course(name: "Intro to AWS", image: "aws", description: "亞馬遜AWS介紹"),
    ]
```

單一課程清單項目

- 小圖、課程名稱

```
struct BasicImageRow: View {
        var thisCourse:Course
        var body: some View {
            HStack{
                Image(thisCourse.image)
                    .resizable()
                    .frame(width: 40, height: 40)
                    .cornerRadius(5)
                Text(thisCourse.name)
            }
        }
    }
```

課程列表

- 並設定調出細節頁面

```
var body: some View {
        NavigationView{
            List(courses){ courseItem in
                    BasicImageRow(thisCourse: courseItem)
                        .onTapGesture {
                            self.showDetailView = true
                            self.selectedCourse = courseItem
                        }
            }
            .sheet(item: self.$selectedCourse){ thisCourse in
                CourseDetailView(course: thisCourse)
            }
            .navigationBarTitle("課程列表")
        }
    }
```

課程細節頁面

```
struct CourseDetailView:View{
        @Environment(\.presentationMode) var presentationMode
        var course:Course
        var body: some View{
            ScrollView{
                VStack{
                    Image(course.image)
                        .resizable()
                        .aspectRatio(contentMode: .fill)
                        .clipped()
                    Text(course.name)
                        .font(.system(.title, design:.rounded))
                        .fontWeight(.black)
                    Spacer()
                    Text(course.description)
                        .font(.system(.subheadline, design:.rounded))
                        .fontWeight(.light)
                    Spacer()
                }
            }
            .overlay(
                HStack{
                    Spacer()
                    VStack{
                        Button(action:{
                            self.presentationMode.wrappedValue.dismiss()
                        },label:{
                            Image(systemName:"chevron.down.circle.fill")
                                .font(.largeTitle)
                                .foregroundColor(.white)
                        })
                        .padding(.trailing, 20)
                        .padding(.top, 40)
                        Spacer()
                    }
                }
            )
        }
    }
```

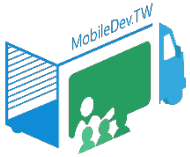
CardView

- 製作一簡單知識卡片頁，點擊可切換

CardView.swift

- 新增檔案(View)

```
import SwiftUI

struct TermAndDescription: Identifiable{ ••• }

var myDictionary = [ ••• ]

struct CardView: View { ••• }

struct CardView_Previews: PreviewProvider {
    static var previews: some View {
        CardView()
    }
}
```



資料結構、內容

- 新增要使用的資料結構
- 新增幾筆內容

```
import SwiftUI


struct TermAndDescription: Identifiable{
    var id = UUID()
    var term:String
    var description:String
}


var myDictionary = [
    TermAndDescription(term: "R2 Score", description: "迴歸分析數值型常用評量指標"),
    TermAndDescription(term: "Confusion Matrix", description: "分類常用評估方法"),
    TermAndDescription(term: "Precision", description: "精準度，從模型的角度出發，評估機器學習的成效"),
    TermAndDescription(term: "Recall", description: "召回率，從真實世界的角度出發，評估機器學習的成效")
]
```

卡片結構

```
struct CardView: View {
    @State var currentCard = 0
    var body: some View {
        VStack{
            VStack{
                Text(myDictionary[currentCard].term)
                    .font(.title)
                    .padding(.all, 10)
                Text(myDictionary[currentCard].description)
                    .font(.body)
                    .foregroundColor(.blue)
                    .padding(.all, 10)
            }
            .frame(minWidth: 0, idealWidth: 100, maxWidth: 300,
                   minHeight: 0, idealHeight: 100, maxHeight: 300, alignment: .center)
            .background(Color.yellow)
            .onTapGesture {
                if currentCard<myDictionary.count-1{
                    currentCard+=1
                }else{
                    currentCard=0
                }
            }
            Text("點擊查看下一張")
                .padding(.all, 10)
        }
    }
}
```

測試

- 確認三個分頁是否都正常