


新北市112年度國中小資訊科技優良教案徵選實施計畫

教案設計

服務學校	崇林國中		設計者	孫郁文
參加組別	<input checked="" type="checkbox"/> 程式教育組 <input type="checkbox"/> 人工智慧組			
領域/科目	科技領域/資訊科技		實施年級	八年級
單元名稱	崇林校園尋寶趣		總節數	共 <u> 6 </u> 節 <u> 270 </u> 分鐘
設計依據				
學習重點	學習表現	運 t-IV-3 能設計資訊作品以解決生活問題。 運 t-IV-4 能應用運算思維解析問題。 運 p-IV-1 能選用適當的資訊科技組織思維，並進行有效的表達。 運 p-IV-2 能利用資訊科技與他人進行有效的互動。 運 p-IV-3 能有系統地整理數位資源。 運 c-IV-1 能熟悉資訊科技共創工具的使用方法。 運 c-IV-2 能選用適當的資訊科技與他人合作完成作品。		核心素養
	學習內容	資 A-IV-1 演算法基本概念。 資 P-IV-1 程式語言的基本概念、功能及應用 資 T-IV-2 資訊科技應用專題。		
議題融入	實質內涵	環境議題		
	所融入之學習重點	運 t-IV-3 能設計資訊作品以解決生活問題		
與其他領域/科目的連結		自然領域/生物		
教材來源		翰林版科技(資訊科科技)八上、Swift App 實作入門 for iPad		
教學設備/資源		iPad、聰穎鍵盤		
使用軟體、數位資源或 APP 內容		書籍 APP、the Swift Programming Language、學習吧平臺		
學習目標				
1. 能運用非圖像式程式語言，將文字轉換為語音輸出。				

2. 能發現真實的問題進而拆解問題，並設計問題的解題流程。
3. 能應用非圖像式語言設計程式以解決問題。
4. 能透過小組討論分析問題、安排解決問題的次序。
5. 能應用非圖像式程式語言與他人合作完成資訊科技專題。
6. 能運用網路蒐集資料，以解決學習上遇到的困難。
7. 能養成關心自我生活環境，進而熱愛鄉土的情懷。

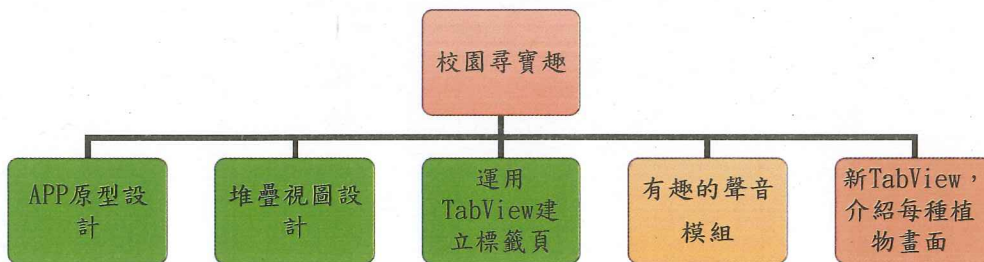
教學活動設計

教學活動內容及實施方式	時間	使用軟體、數位資源或APP內容
<p>一、教案發想：</p> <p>1. 崇林國中位於林口臺地，在歷任校長綠化、美化下，植物多樣群花盛開，全校植物將近 110 種，遍布學校各棟建築物之間的綠地，校園圍牆緊鄰高速公路綠帶 LOHAS PARK，都是師生將常運動、休憩與學習的好地方，為了為讓學生了解學校周邊多樣的植物生態系，因此，本單元課程設計，希望藉由自然領域(生物)植物界知識點出發，結合崇林校園內植物生態環境的探索，運用科技工具及行動載具進行踏查記錄，結合資訊科技程式設計單元，進行跨域學習的校園探索活動，試想，讓學生親自設計個性化認識校園植物 APP，是很有趣的一件事情，有關生物課程植物界知識內容的部分，由生物老師於自然領域課程中教授或跨域協同授課，本教案則專注於資訊科技-程式APP的製作與開發，希望藉由校園植物探索 APP 的製作，讓學生學習運用運算思維解決生活中的問題，呼應崇林國中特色課程「迷霧森林之密室逃脫」，讓程式設計更貼近學生的生活經驗。</p> <p>2. 以 Design Thinking 思維結合科技輔助自主學習模式，設計課程單元，主要以校園植物探究，創意發想設計個性化「崇林校園尋寶趣」APP。</p> <p>(1) 情境分析：崇林校園植物種類繁多，主要以地理位置區分為七大區塊，學生需分組，實地進行植物及生態踏查，了解植物種類及消長情形，以下為七大地理位置區塊：</p> <ul style="list-style-type: none"> ■ 崇真樓、崇善樓 ■ 崇美樓、崇智樓 ■ 楓香大道 ■ 操場 ■ 環校步道 ■ 生態農場 ■ 生態池 <p>(2) 定義問題：崇林校園各區植物探索拍照記錄。</p> <p>(3) 激發創意：採小組合作激發創意，建構探索 APP。</p> <p>(4) 開發原型：動手實作崇林校園尋寶趣 APP。</p> <p>(5) 反思修正：完成程式設計，功能測試及除錯。</p> <p>(6) 產出成果：完成崇林校園尋寶趣 APP。</p> <p>3. 學生盡量分採用異質性分成若干小組，每組 4 人為原則。</p>		<p>無邊記 APP Swift playgrounds APP</p>  <p>學習吧平臺</p>

4. 運用本次專題實作，結合八年級翰林版第2章進階程式設計-陣列篇，主要讓學生學習陣列程式設計實作，以及陣列資料結構的概念與應用。



5. 課程計構圖：



6. 有關校園植物分類及探索知識，在七年級下學期自然領域生物課程已學習完成，本單元設計著重 APP 設計與製作

教學活動一

APP 原型設計

一、課前自學：

學生利用課餘時間進入學習吧，了解本次資訊專題課程上課前，同學須先完成事項，以利課程進行，主要含括以下三部分：

1. 複習七年及下學期生物課程-植物界單元內容
2. 利用午休、早自習、其他空白課程時間，先完成先備知識：
Swiftplaygrounds APP 學習程式設計 1、學習程式設計 2

二、教師導學：APP 視圖設計(10 分鐘)

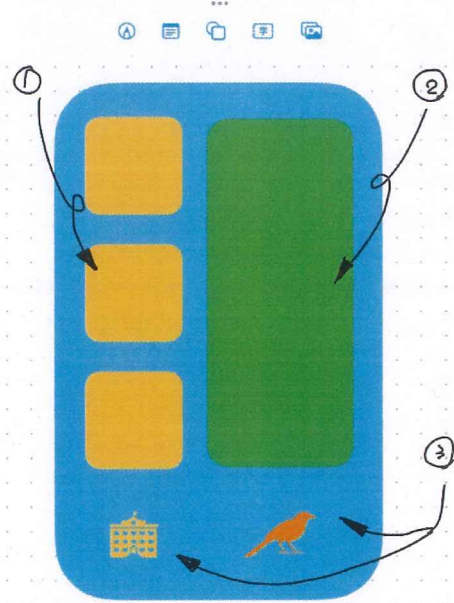
教師運用無邊記 APP，介紹 APP UI 介面設計之技巧及注意事項，並請學生運用巧思，設計 APP 版面配置。

1. 組內共學：

2. 請學生利用 iPad 無邊記 APP，運用工具列各項工具，設計 APP 原型布局，APP 應具有幾項特色，在主畫面中列出七大地理區塊內，最具代表性的植物，包含各種樹木、花草。

20 mins

3. 可嘗試以各種不同方式加以分類，並建立頁籤。
4. 設計分頁標籤介紹代表性的事物，也可以包括中英文解說。
5. 提供學生參考 APP 設計原型圖，讓學生發揮創意，設計自己的 APP 布局。



- (1). 特色照片區
- (2). 特色說明區
- (3). 標籤頁項目標示區

三、組間互學：

展示與分享，學生完成作品繳交至學習吧平臺，教師展示作品，並請小組發表分享設計心得。

四、教師導學：

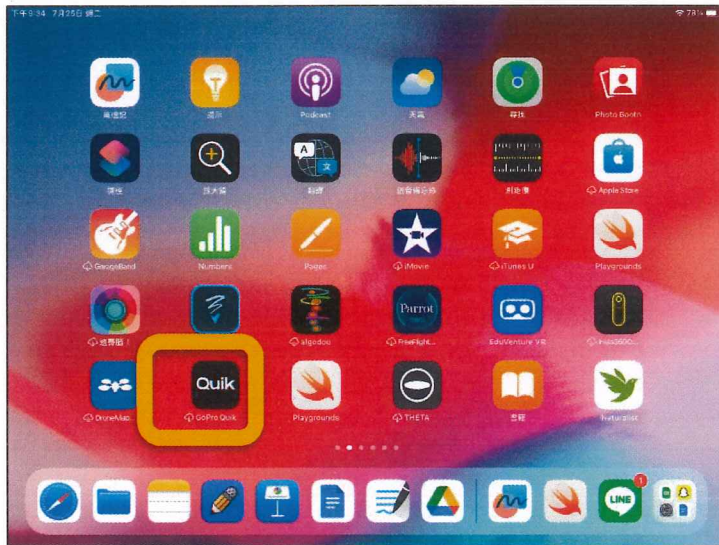
教師講評，鼓勵表現設計優良的小組作品。

教學活動二

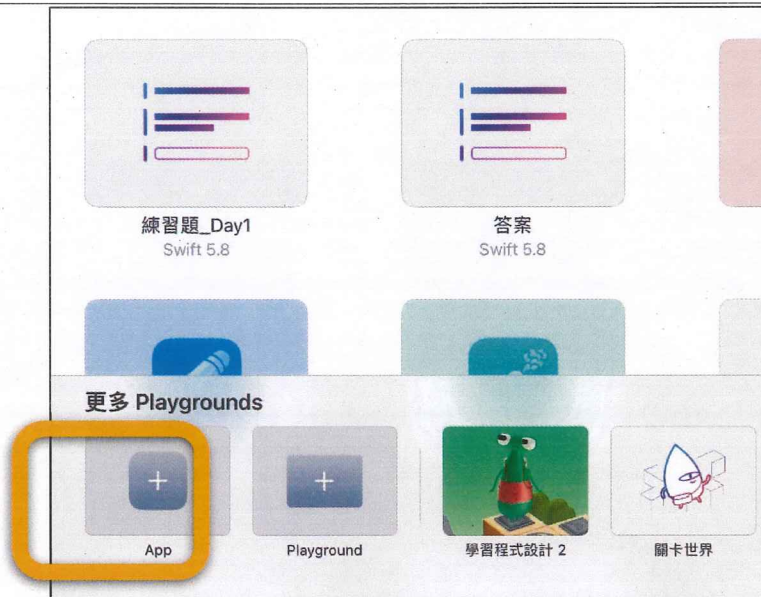
使用堆疊視圖設計 APP 的 UI(使用者介面)

教師導學：

1. 打開 Swiftplaygrounds APP，更多 Playgrounds，新增一個空白 APP 範本



70
mins

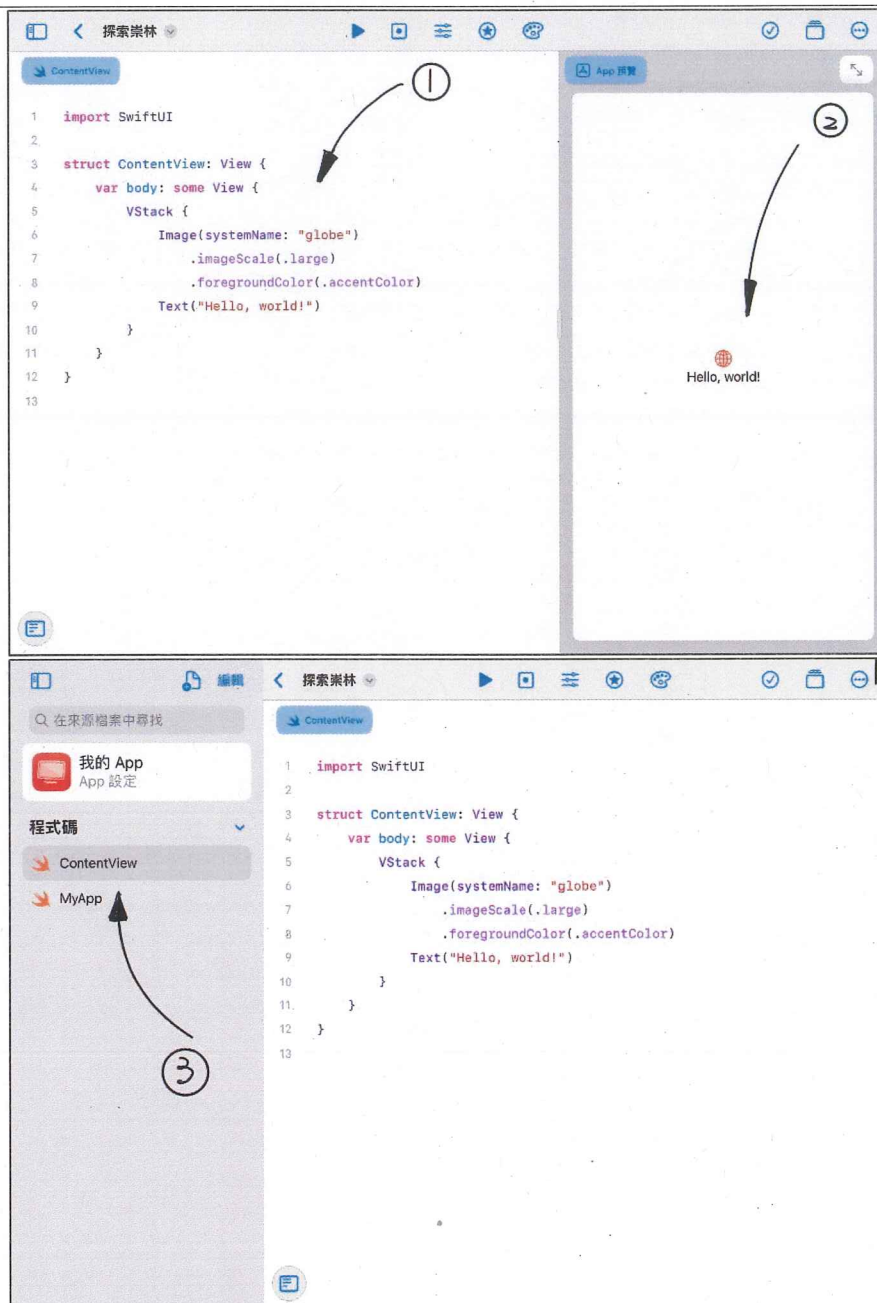


2. 重新命名 APP



3. 程式編輯畫面簡介：基本範例說明

- (1). 主要編輯區
- (2). APP 預覽區
- (3). 程式碼檔案與資料區

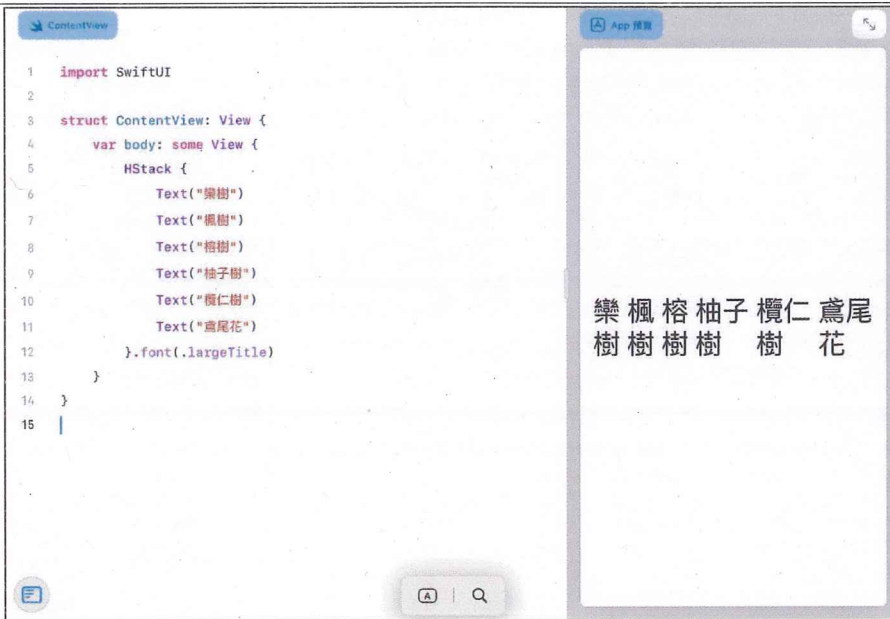


(4). 瀏覽資料庫控制項目：所見及所得，控制項目、顏色和圖像，並快速加到你的程式碼中

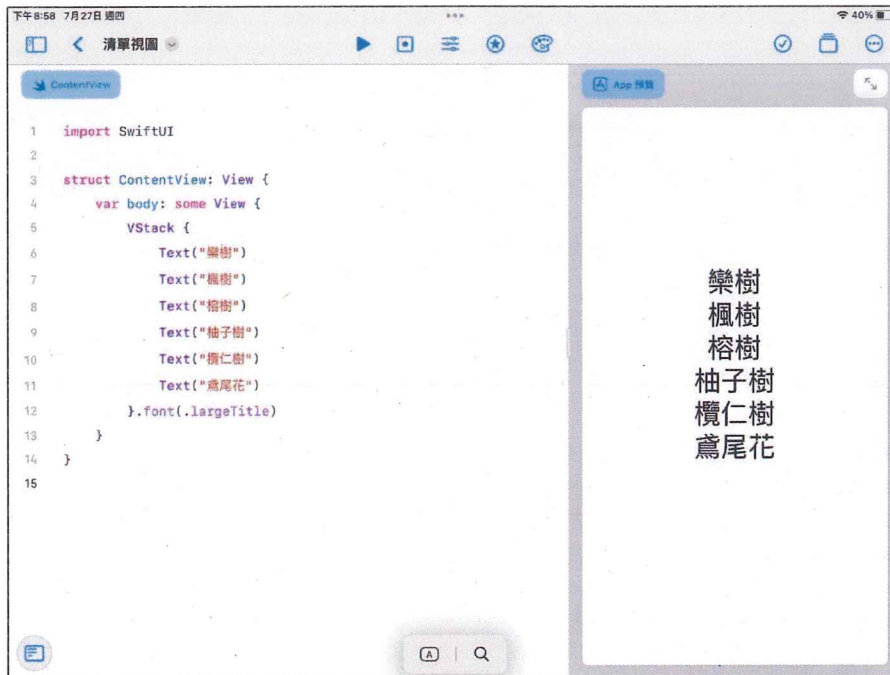
4. UI 元件畫面初體驗：(35 分鐘)

SwiftUI 提供三種不同類型的視圖堆疊，讓學生很直觀就能了解，整個視窗的版面設計。

(1). Hstack-水平排列視圖



(2). VStack-垂直排列視圖



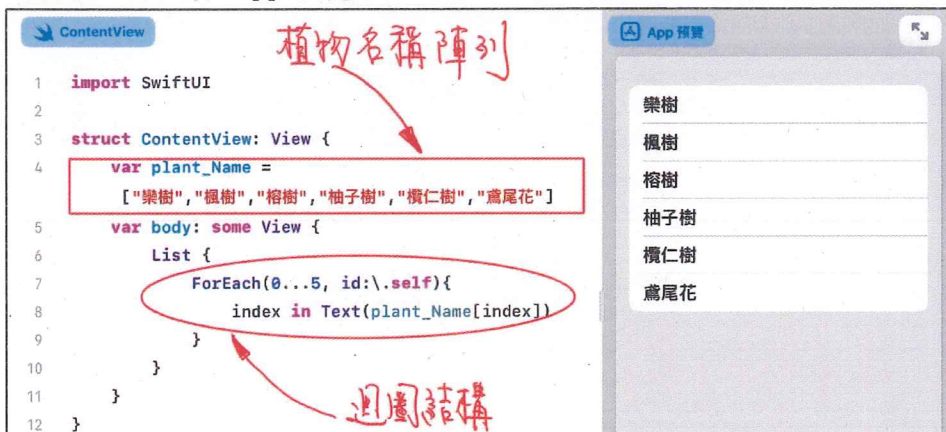
(3). List-清單視圖

List 也是 SwiftUI 的 View，畫面的呈現會以清單滾動式列表呈現。



5. 建立更多樣化的清單視圖

我們使用一維陣列來儲存清單資料，每個陣列元素都有一個索引值，用來存取資料，我們宣告一個變數 plant_Name 賦予清單資料的陣列，運用 ForEach 迴圈依序取出陣列資料，顯示在 App 預覽區。




組內共學：

1. 教師可先將程式建立範例檔，並利用 iPad Airdrop 功能，迅速提供給學生，這樣可以避免學生打字錯誤，先學會看懂，知道如何修改就很厲害了。（要請學生特別注意程式中的大括弧）
2. 請學生依據老師課堂教授內容，嘗試將校園植物踏查分類收集到的植物相片，上網搜尋植物相關屬性，原產地，分布校園位置，分類建立植物名稱陣列、植物圖片名稱陣列、植物產地陣列等。
3. 參考老師範例程式，試著使用 ForEach 迴圈結構，將每個陣列元素顯示在 List 清單中。
4. Image() 是圖片容器，後面有黑點加圖片修飾符，可以修飾圖片造型喔！修飾符的種類很多，可查閱 Swiftplaygrounds APP 右上角的「文件」功能。


```

1 import SwiftUI
2
3 struct ContentView: View {
4     var plant_image = ["Luan tree", "grapefruit tree", "Bird Banyan"]
5     var body: some View {
6         List{
7             ForEach(0...2, id: \.self){
8                 index in
9                 HStack{
10                    Image(plant_image[index])
11                    .resizable()
12                    .frame(width: 100, height: 100)
13                }
14            }
15        }
16    }
17 }

```



圖片修飾符

提示：

- 修飾符 `resizable()`：縮放圖片到目前可用空間大小。
- 修飾符 `frame(width:height:alignment:)`：將此視圖放置在具有指定大小框架內。
- 修飾符 `clipShape(RoundedRectangle(cornerRadius:30))`：修飾圖片為圓角矩形。

```

1 import SwiftUI
2
3 struct ContentView: View {
4     var plant_Name = ["臺灣樂樹", "柚子樹", "雀榕"]
5     var plant_image = ["Luan tree", "grapefruit tree", "Bird Banyan"]
6     var body: some View {
7         List{
8             ForEach(0...2, id: \.self){
9                 index in
10                HStack{
11                    Image(plant_image[index])
12                    .resizable()
13                    .frame(width: 100, height: 100, alignment: .center)
14                    .clipShape(RoundedRectangle(cornerRadius: 30))
15                    VStack{
16                        Text(plant_Name[index])
17                    }
18                }
19            }
20        }
21    }
22 }

```



5. 教學活動一，主要的目的是讓學生嘗試將自己拍攝的圖片及植物名稱，放入清單列表中，並且很整齊地排列出來。

6. 如果學生對堆疊視圖設計還是不清楚，老師可放慢速度，帶領學生一起完成附錄一、基礎學習單演練。

組間互學：

1. 展示與分享，學生完成作品繳交至學習吧平臺，教師展示作品，並請小組發表分享設計心得。

教師導學：

2. 教師講評，鼓勵表現設計優良的小組作品。

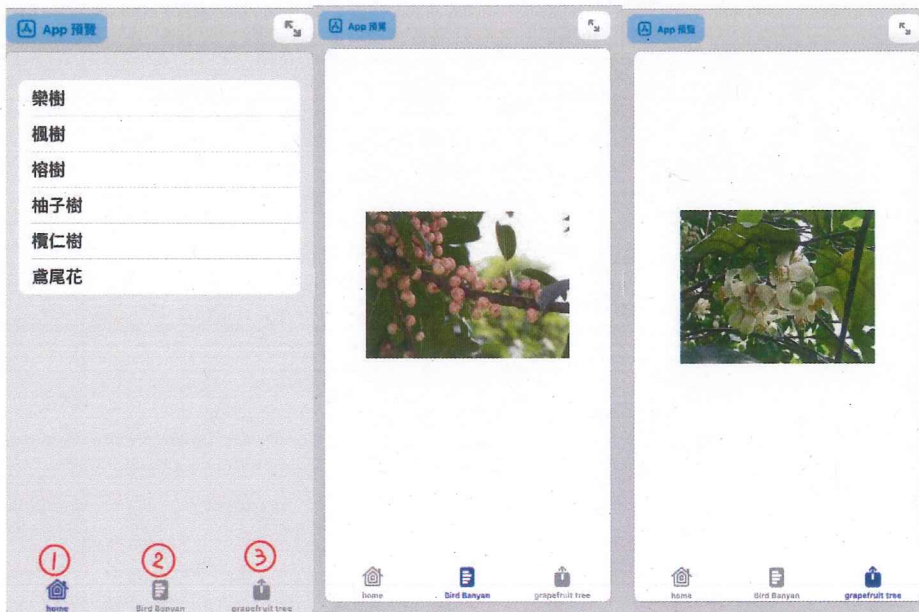
教學活動三

運用 TabView 建立標籤頁

教師導學：

1. TabView 也是 SwiftUI 中的一個 View 容器，可用於不同的標籤頁之間切換，會讓 APP 顯示的內容更為豐富，操作步驟簡單明瞭。
2. 透過 TabView，使用者可以從畫面底部標籤中選擇不同的選項後，切換到相應的標籤頁面。
3. TabView{ }後接修飾符.tabItem{ Label(“名稱“，systemImage:”內見圖檔名稱“)}，這樣就能在 APP 預覽畫面底部建立不同的標籤項目。

45
mins



```
ContentView
1 import SwiftUI
2 struct ContentView: View {
3     var plant_Name = ["樂樹", "楓樹", "榕樹", "柚子樹", "欖仁樹", "鳶尾花"]
4     var body: some View {
5         TabView{
6             List {
7                 ForEach(0...5, id:\.self){
8                     index in Text(plant_Name[index])
9                 }
10            }.tabItem { Label("home",systemImage: "homekit") }
11            TabView{
12                Image("Bird Banyan")
13            }.tabItem{ Label("Bird Banyan",systemImage:
14                "chart.bar.doc.horizontal.fill") }
15            TabView{
16                Image("grapefruit tree")
17            }.tabItem{ Label("grapefruit tree",systemImage:
18                "square.and.arrow.up.fill") }
19        }..
20    }
21 }
```


- (1). 標籤頁項目 1 :一維陣列內含 6 個元素，["欒樹", "楓樹", "榕樹", "柚子樹", "欖仁樹", "鳶尾花"]
- (2). 標籤頁項目 2 : 切換至圖片至 Bird Banyan
- (3). 標籤頁項目 3 : 切換至圖片至 grapefruit tree

組內共學：

1. 教師可先將程式建立範例檔，並利用 iPad Airdrop 功能，迅速提供給學生，這樣可以避免學生打字錯誤，先學會看懂，知道如何修改就很厲害了，要請學生特別注意程式中修飾符的位置。
2. 不用太複雜的程式語法，強調 APP 視覺畫面安排，這讓學生才會有成就感。

組間互學：

1. 展示與分享，學生完成作品繳交至學習吧平臺，教師展示作品，並請小組發表分享設計心得。

教師導學：

1. 教師講評，鼓勵表現設計優良的小組作品。

教學活動四

有趣的聲音模組

教師導學：

1. 在 import SwiftUI 之下，匯入語音模組 import AVFounctio
2. 建立函式 saySomething，加入以下四行程式碼：
第一行是指定要說出的文字
第二行是決定發音的語系
其餘行是建立語音合成器，並使用選定的聲音來說出文字
3. 當按鈕被按下時，會使用選定的聲音來念出文字。
4. 變換各國語言(語系)代碼，練習有趣的發音。
5. 範例畫面布局如下圖

45 mins

```

1  import SwiftUI
2  import AVFoundation
3
4  struct ContentView: View {
5      let synthesizer = AVSpeechSynthesizer()
6      @State var words = "崇林團隊好棒棒!"
7      func saySomething(words:String){
8          let utterance = AVSpeechUtterance(string: words)
9          let voice = AVSpeechSynthesisVoice(language: "Zh-TW")
10         utterance.voice = voice
11         synthesizer.speak(utterance)
12     }
13     var body: some View {
14         VStack {
15             Image(systemName: "teddybear.fill")
16                 .imageScale(.large)
17                 .foregroundColor(.red)
18             Text("\(words)").font(.title)
19             Button("請按我") {
20                 saySomething(words:words)
21             }
22         }
23     }
24 }

```

① 匯入語音模組

② 指定說出文字

決定語系 說出文字

呼叫函數



崇林團隊好棒棒!
請按我

組內共學：

1. 教師可先將程式建立範例檔，並利用 iPad Airdrop 功能，迅速提供給學生，這樣可以避免學生打字錯誤。
2. 將本次教學活動所學讓 iPad 讀出文字，同時運用本單元所學，修改專題製作教學活動二，讓放入 List 中的文字 Text() 皆能發出聲音。

組間共學：

1. 展示與分享。
2. 學生完成作品繳交，教師展示作品，並請小組發表分享設計心得。
3. 教師課堂觀察、學生作品發表、學生互評。

教師導學：

1. 教師講評，鼓勵表現設計優良的小組作品。

```

1 import SwiftUI
2 import AVFoundation
3 struct ContentView: View {
4     var plant_Name = ["棠樹", "楓樹", "榕樹", "柚子樹", "欖仁樹", "鳶尾花"]
5     let synthesizer = AVSpeechSynthesizer()
6     func speaker(sayString: String){
7         let utterance = AVSpeechUtterance(string: sayString)
8         utterance.voice = AVSpeechSynthesisVoice(language: "zh-TW")
9         synthesizer.speak(utterance)
10    }
11    var body: some View {
12        TabView{
13            List {
14                ForEach(0...5, id:\.self){
15                    index in Text(plant_Name[index])
16                        .onTapGesture {
17                            speaker(sayString: plant_Name[index])
18                        }
19                }
20            }.tabItem { Label("home",systemImage: "homekit") }
21            TabView{
22                VStack{
23                    Image("Bird Banyan")
24                    Text("Bird Banyan")
25                        .onTapGesture {
26                            speaker(sayString:"Bird Banyan" )
27                        }
28                }
29            }.tabItem{ Label("Bird Banyan",systemImage:
30                "chart.bar.doc.horizontal.fill") }
31            TabView{
32                VStack{
33                    Image("grapefruit tree")

```

- ①匯入語音模組
- ②語音函式
- ③onTapGesture{ }開啟點一下手勢，同時，呼叫語音函式說出植物名稱
- ④onTapGesture{ }開啟點一下手勢，同時，呼叫語音函式說出圖片植物名稱

教學活動五

建立一個新 TabView，主要展示詳細介紹每種植物畫面

視學生學習情形，教師再提供學生延伸學習

1. 本次教學活動有一定困難度，教師宜先提供程式範例檔，並利用 iPad Airdrop 功能，迅速提供給學生，再詳細解說程式各部分功能。
2. 提供學生的範例檔，一方面解說程式架構，一方面也能複習先前所學習過的課程內容。
3. 本次教學活動，主要建立一個 struct BasicTextImageROW，放在原程式 struct ContentView :View { } 之後。

90
mins

```

42
43 struct BasicTextImageRow: View {
44     var imageName: String
45     var name:String
46     var type: String
47     var location: String
48     @State var isShow = false
49     var body: some View {
50         TabView{
51             VStack(alignment:.leading, spacing: 50) {
52                 ZStack{
53                     Image(isShow ? "Green":imageName)
54                     .resizable()
55                     ② .frame(height: 300, alignment: .leading)
56                     .cornerRadius(20) ③
57                     .rotation3DEffect(.degrees(isShow ? 0:180) , axis:
58                         (x: 0.0, y:1.0, z: 0.0) ④
59                     )
60                     .onTapGesture {
61                         withAnimation{isShow.toggle()} ⑤
62                     }
63                     Text(isShow ? name : "")
64                     .font(.system(.largeTitle, design: .rounded))
65                     .foregroundColor(.white) ⑥
66                 }
67             }
68             .VStack( alignment: .center, spacing: 10){
69                 Text(type)
70                 .font(.system(.body, design: .rounded))
71                 Text(location)
72                 .font(.system(.body, design: .rounded))
73                 .foregroundColor(.gray)
74             }
75         }
76     }
77 }

```

①
三元運算子
決定圖片是否呈現
"綠色"

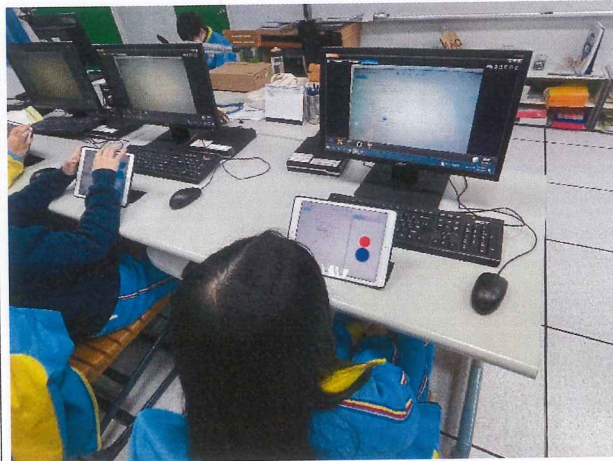
- ①三元運算子：簡化版的 if else 條件判斷語法，語法中使用「？」，利用「:」將條件句 true 的值和 false 的值分開，
如果 isShow 的值是 true: 圖片會呈現 Image("Green")
如果 isShow 的值是 false: 圖片會呈現 Image(imageName)
- ②修飾符：frame(height:300) 限定圖片高度 300
- ③修飾符：cornerRadius(20) 圖片呈現 20 度圓角
- ④修飾符：rotation3DEffect(isShow ? 0:180 ,axis:(x:0.0, y:1.0, z:0.0))
如果 isShow 的值是 true:圖片以 y 軸為中心旋轉 180 度
如果 isShow 的值是 false:圖片以 y 軸為中心旋轉 0 度
- ⑤修飾符：onTapGesture{is }開啟點一下手勢點按一下，改變 isShow 的布林值
- ⑥修飾符：Text()的文字字體及顏色

4. 讓學生要先學會看懂程式架構，知道如何修改，呈現它們想要的畫面就對了。
5. 提供以下完整範例程式碼提供老師教學參考。


```

import SwiftUI
import AVFoundation
struct ContentView: View {
    var plant_Name = ["台灣欒樹", "柚子樹", "雀榕"]
    var plant_Image = ["Luan tree", "grapefruit tree", "Bird Banyan"]
    var plant_location = ["Taiwan", "Chong lin", "Lodon"]
    var plant_type = ["無患子科", "芸香科", "桑科"]
    let synthesizer = AVSpeechSynthesizer()
    func speaker(sayString: String){
        let utterance = AVSpeechUtterance(string: sayString)
        utterance.voice = AVSpeechSynthesisVoice(language: "zh-TW")
        synthesizer.speak(utterance)
    }
    var body: some View {
        TabView{
            List {
                ForEach(0...2, id:\.self){
                    index in HStack{
                        Image(plant_Image[index])
                            .resizable()
                            .scaledToFit()
                            .frame(width: 100, alignment: .center)
                            .cornerRadius(30)
                        Text(plant_Name[index])
                            .onTapGesture {
                                speaker(sayString: plant_Name[index])
                            }
                    }
                }
            }.tabItem { Label("崇林植物", systemImage: "homekit") }
            TabView{
                ForEach(0..

```

說明:運用 iPad 進行程式設計



說明:進行堆疊視圖設計

教學成果



說明:學生互助合作學程式



說明:學生測試程式功能

教學心得與省思

1. 本次教學單元所設計的教案，上學年實施的部分僅限於堆疊視圖設計，因為使用 iPad 內建的浮動鍵盤，尤其在編寫程式碼的過程中，操作上不是很方便，比較複雜的程式撰寫難度更高，因此，學校購買一個班級數量的藍牙鍵盤，相信112學年度將 iPad 融入程式設計，將會讓資訊科技課程更有趣。
2. 在「教學活動五」有些難度，再編寫教案的過程中有些掙扎，最後還是放進去，教師在教學過程中，盡量提供學生範例程式檔，讓學生先看懂程式架構，再思考如何修改，會是比較容易成功的，學生也比較會有成就感。
3. iPad Swift Playgrounds APP 內有許多書籍可提供教學使用，包含內建感測器的運用、機器學習、APP 的設計、圖形繪製等，未來可以多加利用。

參考資料

1. Swift App 實作入門 for iPad 教師專班 Day3 講師: Michael Pan
2. 學習編寫程式1和2(教師指南) iPad 書籍 APP
3. 人人可編碼解謎闖關(教師指南) iPad 書籍 APP
4. The Swift Programming Language Swift5.7 Edition

附錄

附錄一、學習單
 附錄二、教師編寫之範例程式連結 <https://drive.google.com/file/d/14oax-pv-5mPcVaGRqNjc8Ku0fxKuwFMD/view?usp=sharing>

附錄一、SwiftUI 使用堆疊視圖基礎設計

一、提示

1.關鍵字：堆疊類型

(1).VStack{ }:垂直堆疊、HStack{ }:水平堆疊、ZStack{ }:深度堆疊

2.關鍵字：文字類型

(1).Text()文字

3.關鍵字：影像類型

(1).Image()影像



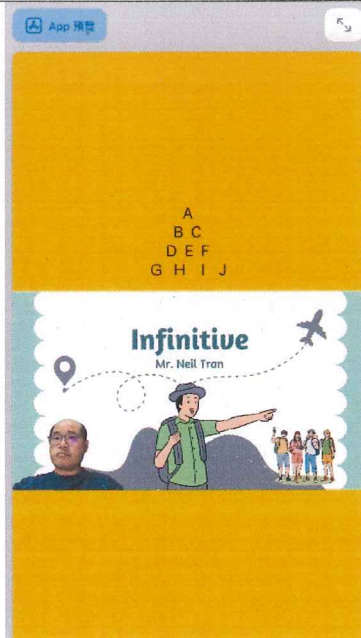
二、請利用堆疊(佈局)類型，完成以下版面設計：

1.填入單一背景色。

2.英文字母 A 至 J，分成四列，置中依序排列。

3.插入圖片，並調整大小。

提示：影像可調整大小 `resizable()`、縮放至適當大小 `scaledToFit()`



1.畫面程式碼：