

新北市113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	新北市立崇林國民中學		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		<input type="checkbox"/> 資訊素養與倫理組	
領域/科目	科技領域/程式設計單元	實施年級	七年級、八年級	
單元名稱	校園生態紀錄-植物調查局 APP	總節數	共 3 節， 135 分鐘	
設計依據				
學習重點	學習表現	1.運 t-IV-4 能應用運算思維解析問題。 2.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。 3.運 p-IV-2 能利用資訊科技與他人進行有效的互動。 4.運 p-IV-3 能有系統地整理數位資源。 5.運 c-IV-1 能熟悉資訊科技共創工具的使用方法。 6.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	1.科-J-A2 運用科技工具，理解與歸納問題，進而提出簡易的解決之道。 2.科-J-A3 利用科技資源，擬定與執行科技專題活動。 3.科-J-B1 具備運用科技符號與運算思維進行日常生活的表達與溝通。 4.科-J-B2 理解資訊與科技的基本原理，具備媒體識讀的能力，並能了解人與科技、資訊、媒體的互動關係。 5.科-J-B3 了解美感應用於科技的特質，並進行科技創作與分享。
	學習內容	1.資 A-IV-1演算法基本概念。 2.資 P-IV-1程式語言的基本概念、功能及應用 3.資 T-IV-2 資訊科技應用專題。		
議題融入	實質內涵	環境議題、資訊議題		
	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)/植物界			
教材來源	1.翰林版資訊科技教科書、康軒版自然七年級教科書 2.人人可編碼-解謎闖關 Apple 書籍 3.開始編寫程式碼 Apple 書籍 4.學習編寫程式1和2 Apple 書籍 5.iPad Swift App 基礎專案實作作者：Michael Pan 6.Xcode Swift APP 進階專案實作 作者：Michael Pan			
教學設備/資源	iPad、平板鍵盤、觸控大屏			
使用軟體、數位資源或 APP 內容	學習吧平臺、SwiftPlaygrounds APP(開始編寫程式碼、學習程式設計1&學習程式設計2)、書籍 APP			
學習目標				
1.能設計問題的解題流程。 2.能應用非圖像式語言設計程式以解決問題。 3.能透過小組討論分析問題、安排解決問題的次序。 4.能應用非圖像式程式語言與他人合作完成資訊專題。 5.能養成關心自我生活環境，進而熱愛鄉土的情懷。				

教學活動

時間

使用軟體
數位資源
APP 內容

一、教案發想：

(一)、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，走入校園生態系，親近校園植物也是很有趣的一件事情，有關生物單元-植物界相關知識內容的部分，由另一位生物老師於自然領域課程中教授，讓學生在教室習得完整科學知識的脈絡，再結合 5G 新興科技 VR 頭盔沉浸式體驗，了解植物構造與運輸功能，最後，再透過 APP 分組實地踏查校園不同區塊的生態植物，親手繪製校園植物的生態地圖，也可延伸學習至鄰近樂活公園的生態環境踏查，最後，運用學生分組討論與發表檢視學生學習成果，再由老師給予總結指導。

(二)、本教案專注於資訊科技-程式設計單元的設計與實施，採用自主學習四學螺旋式設計，期待學生都能完成 APP 實作。

5mins



Swift
Playgrounds

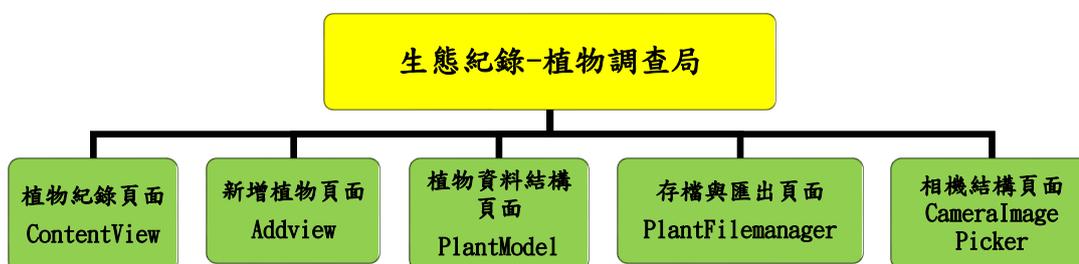
二、生態紀錄-植物調查局 APP 設計思維

(一)、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「生態紀錄-植物調查局」APP。

1. 定義問題：崇林校園各區植物探索拍照記錄。
2. 激發創意：採小組合作激發創意，建構探索 APP。
3. 開發原型：動手實作生態紀錄-植物調查局 APP。
4. 反思修正：完成程式設計，功能測試及除錯。
5. 產出成果：完成生態紀錄-植物調查局 APP。
6. 學生盡量分採用異質性分成若干小組，每組4人為原則。
7. 運用本次專題實作，結合七年級翰林版第3章程式設計，主要讓學生由基礎程式設計實作，並導入資料結構的概念，延伸學習至八年級程式設計陣列單元。

(二)、APP 設計頁面主要架構圖：

10mins



(三)、APP 運用的情境說明：

校園生態植物踏查：全班學生分成若干小組，以自製的 APP 進行校園生態紀錄，範圍以校門口延伸到後校門籃球場。

1. 教師提點 iPad 及 APP 操作流程及注意事項。
2. 教師交代任務，讓學生記錄校園植物，並依照特性進行推測及分類。
3. 學生持 iPad 以 APP 進行校園植物分類、拍攝及壓縮儲存資料，內容包括拍攝的植物相片及推測的植物資訊 CSV 檔。
4. 學生完成踏查任務，各組回到教室進行成果分享，最後由教師進行總結。

三、生態紀錄-植物調查局 APP 設計

135 mins

(一)、APP 設計環境簡介：

課前自學

1. 學生利用課餘時間進入學習吧了解本單元課程內容，上課前同學須先完成事項以利課程進行，主要內容含括以下三部分：
2. 預習生物課程-植物界單元內容
3. 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

教師導學

APP 設計環境簡介：

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

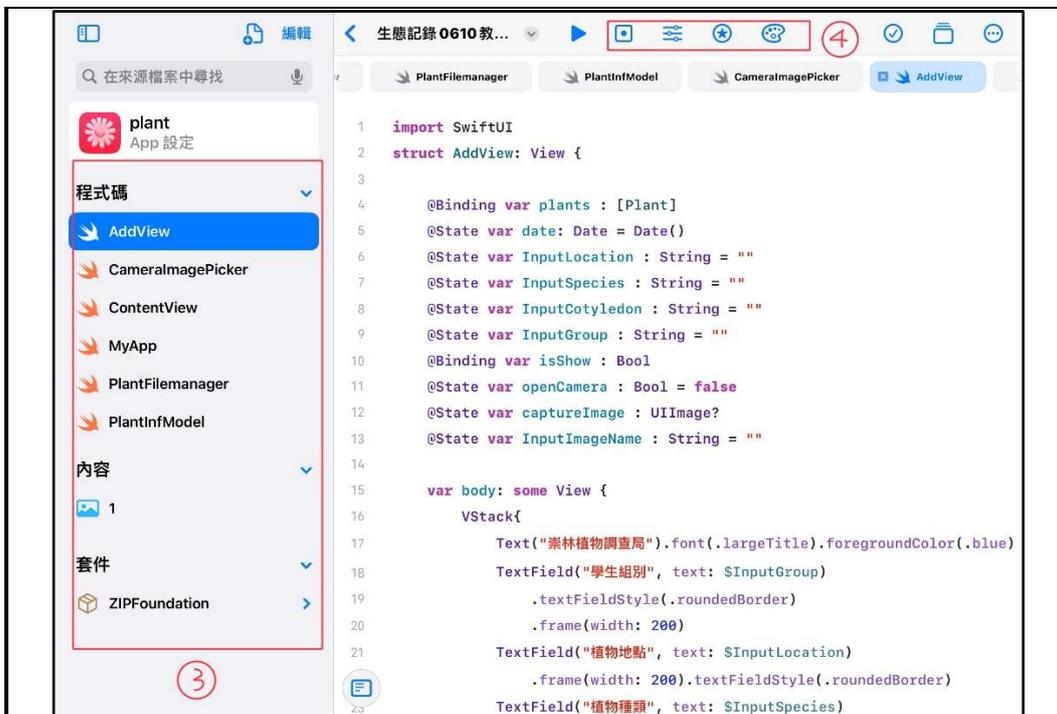


2. 重新命名 APP

1. 程式編輯畫面簡介：基本範例說明

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





(4). 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。



(二)、基本頁面布局設計

課前自學

學生利用課餘時間進入學習吧了解本單元課程內容，上課前同學須先完成事項以利課程進行，主要內容含括以下三部分：

1. 預習生物課程-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，Swiftplaygrounds APP 學習程式設計 1、學習程式設計 2、開始編寫程式碼

教師導學

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，主要顯示頁面有三：

1. 植物紀錄頁面
2. 新增植物頁面
3. 收集列表頁面

組內共學

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，在主畫面中呈現校徽、主要標題-生態紀錄，最具代表性的造型按鈕、植物踏查結果匯出按鈕，以及整個頁面主色及透明度。

1. 植物紀錄頁面：主標題-崇林植物調查局，文字輸入欄位包含:學生組別、植物名稱、植物種類、子葉類型，拍攝植物照片按鈕、儲存、取消兩個按鈕。
2. 可嘗試以各種不同方式布局，也可以使用英文輸入。
3. 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。



設計示意圖

組間互學

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

教師導學

教師講評，鼓勵表現設計優良的小組作品。

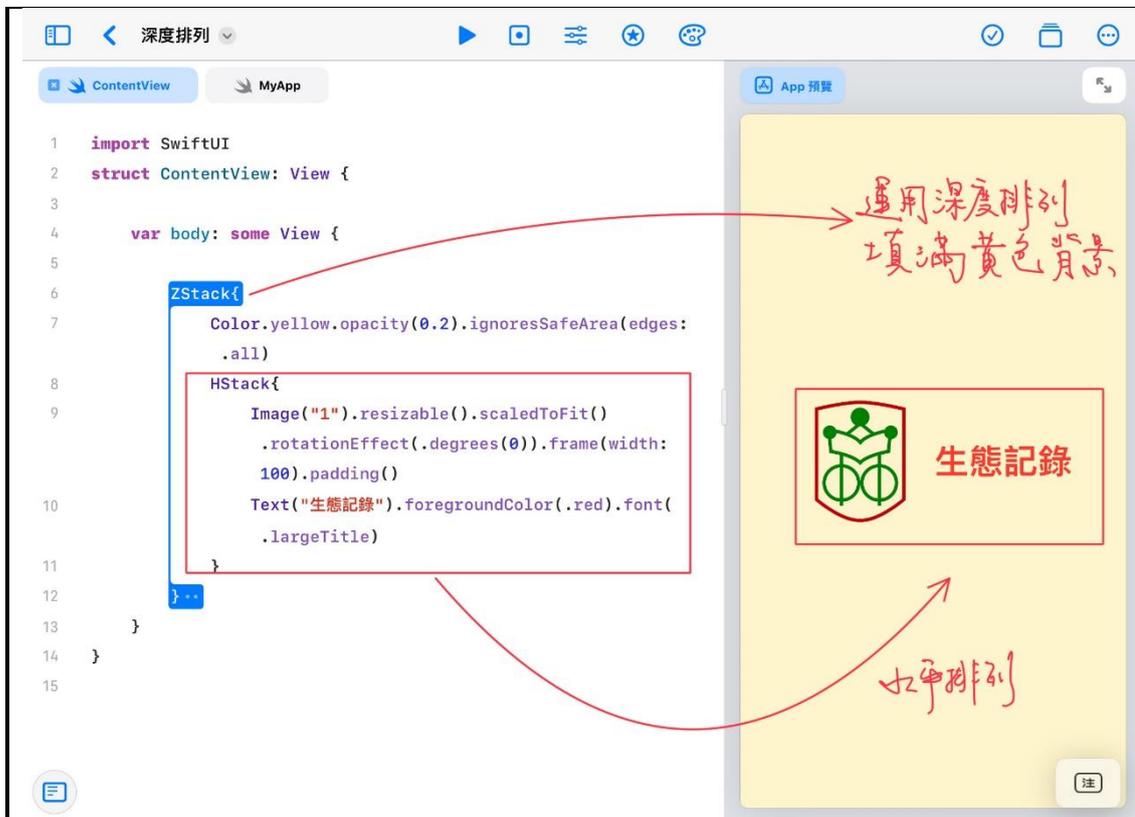
(三)、UI 堆疊視圖設計(使用者介面)

教師導學

1. UI 元件畫面初體驗

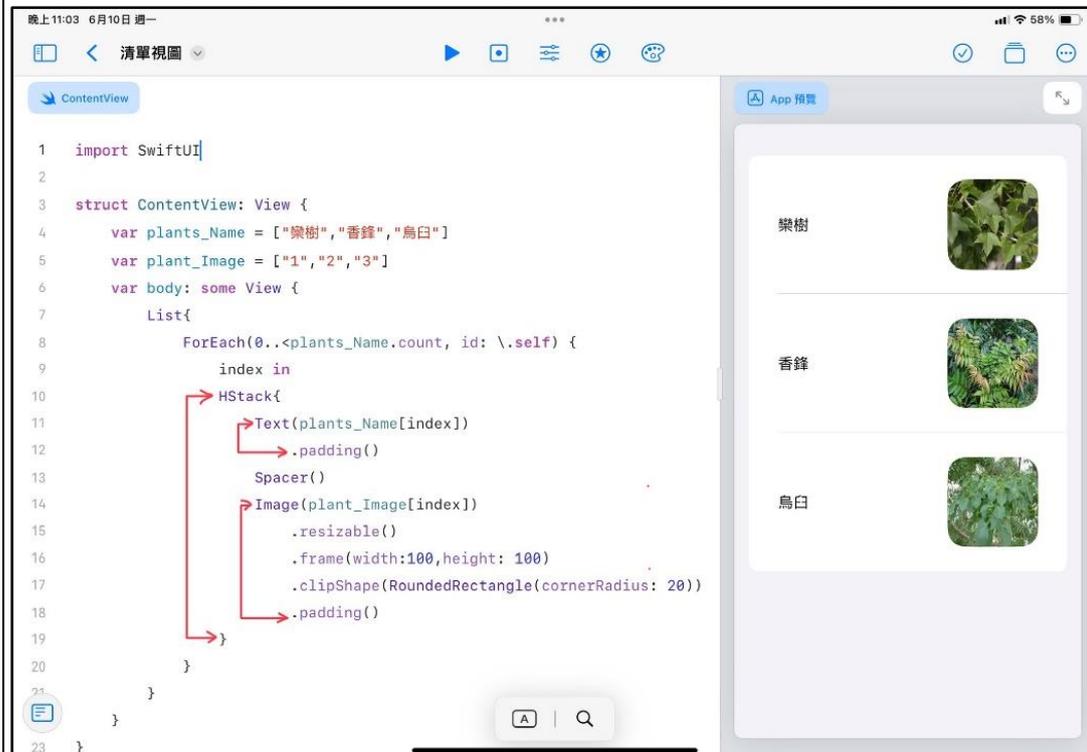
SwiftUI 提供三種不同類型的視圖堆疊，讓學生很直觀就能了解，整個視窗的版面設計。

<p>1. Hstack{}-水平排列視圖</p>	<p>2. VStack{}-垂直排列視圖</p>
<p>3. List{}-清單視圖</p>	<p>4. 運用 ForEach 迴圈設計多樣化的清單視圖</p>
<p>5. Zstack{}-深度排列視圖</p>	



組內共學

1. 教師可先將導學程式範例檔，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，透過小組討論，嘗試動手實作完成以下圖檔：



3. 參考教師導學範例程式，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。

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。

5. UI 堆疊視圖設計的活動，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，帶領學生一起完成基礎學習演練。

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。

(四)、崇林植物調查局 APP 頁面架構：

教師導學

教師說明本次 APP 設計主要 swift 檔案如下：

	<p>①新增植物頁面程式 AddView,swift</p> <p>②啟動相機程式(由教師提供範本，學生會呼叫此程式，就成功囉！ CameraImagePicker.swift</p> <p>③APP 主要頁面程式 ContentView.swift</p> <p>④打開 APP 最先執行的程式 MyApp.swift</p> <p>⑤準備存取資料的函數，包含資料的儲存、載入、壓縮、匯出成 CSV 檔。(教師提供學生不需撰寫) PlantFileManager.swift</p> <p>⑥新增植物的資料結構 PlantInfModel.swift(教師提供學生不需撰寫)</p>
---	--

各頁面互動關係如下：



■生態紀錄-植物調查局製作 ContentView 程式碼及分段說明連結：

https://docs.google.com/document/d/1tnhXpmX7FU5MzWhHo_Jz43CxaR4IKwlyYKNb07dz2eM/edit?usp=sharing

■生態紀錄-植物調查局製作 AddView 程式碼及分段說明連結：

<https://docs.google.com/document/d/1b5TW32D0DqPogdnzXKk-Yf-AIUGJdo8QncjfihPaiqE/edit?usp=sharing>

■生態紀錄-植物調查局製作 CameraImagePicker 程式碼連結：

<https://docs.google.com/document/d/10PzPNKYCEkS81BiQLmp26boyZMvrKs7IBanfOSbeC8k/edit?usp=sharing>

■生態紀錄-植物調查局製作 PlantFileManager 程式碼連結：

<https://docs.google.com/document/d/10PzPNKYCEkS81BiQLmp26boyZMvrKs7IBanfOSbeC8k/edit?usp=sharing>

■生態紀錄-植物調查局製作 PlantinfModel 程式碼連結：

<https://docs.google.com/document/d/1vBfazYIO682WCp7mf2Des5V7tTBek3kN9DTq6W9ocqw/edit?usp=sharing>

■壓縮套件：

<https://github.com/weichsel/ZIPFoundation?tab=readme-ov-file>

■生態紀錄-植物調查局完整 APP 壓所檔連結：

https://drive.google.com/drive/folders/1-UI500BP-DU_6GikgMBmJQYWZiy3GbTT?usp=sharing

考量是七年級學生，大部分的頁面程式會由練習範本中提供，主要讓學生學習撰寫的頁面會集中在(1)新增植物頁面程式 AddView.swift 及(2)APP 主要頁面程式 ContentView.swift 兩個頁面，其他 swift 頁面程式碼，可以直接提供給學生，只須把要學生撰寫程式碼的部分挖空，讓學生嘗試撰寫並預覽結果，學習效果會更好，並可減少學生學習程式的挫折感。

(五)、新增植物頁面(新增 swift 檔案：AddView)：

教師導學

1. 文字欄位、按鈕初體驗：

2. SwiftUI 提供文字欄位，(就像是一個空白的文字方塊，讓你可以由鍵盤輸入內容，讓學生很直觀在視窗中輸入資料)，或是透過瀏覽資料庫控制項目(編輯畫面正上方-控制項目-文字欄位)至接帶出，也可避免學生打錯字，基本語法如下：

(1).文字欄位：

```
TextField("Placeholder", text: Value)
```

Placeholder：輸入欄位提示文字

Value：綁定的變數，會儲存輸入的內容。

提示：

修飾符.textFieldStyle(roundedBorder)圓角邊框

修飾符.frame(width: ,height)框架尺寸

(2).按鈕：

常用<型式一>

```
Button("Button"){
```

```
    Action
```

```
}
```

常用<型式二>

```
Button("Button"){
```

```
    Action
```

```
}
```

Button：按鈕名稱

Action：按鈕按下執行動作

提示：

修飾符.frame(width: 整數,height: 整數)按鈕尺寸

修飾符.background(.顏色)按鈕背景顏色

修飾符.cornRadius()按鈕圓角

組內共學

1.請學生在檔案與資料區新增 Swift 檔案，檔名為：AddView。

2.新增一個 view 的結構，程式碼先以中文區塊表示，就能讓學生很清楚整個程式的架構。

```
import SwiftUI
struct AddView: View {
```

```
// 程式碼 A.宣告變數與定義函
```

數

```
var body: some View {
```

```
// 程式碼 B.新增標題及文字欄位
```

```
// 程式碼 C.拍攝照片按鈕
```

```
// 程式碼 D.儲存和取消按鈕
```

```
}
}
```



3.教師 Airdrop 作業範例程式檔-附件 1 程式檔，請小組互相討論，根據教師導學所學，如何新增以下欄位，並加以美化。

序號	欄位名稱	相關變數	
①	崇林植物調查局	x	
②	學生組別	InputGroup	
③	植物地點	InputLocation	
④	植物種類	InputSpecies	
⑤	子葉類型	InputCotyledon	
⑥	拍攝照片	openCamera	
⑦	儲存	newplant	
⑧	取消	isShow	

4.作業範例程式檔，如附件 1，請教師刪除註解//程式碼 A.宣告變數與定義函、//程式碼 B.新增標題及文字欄位灰色文字，即可將檔案提供給學生練習使用。

■檔案連結：<https://drive.google.com/drive/folders/1owYtbz3F1jyCf8m-IF8Wm8bjz5yn2CqT?usp=sharing>

組間互學

- 1.展示與分享，學生完成作品繳交至學習吧平臺或 AirDrop 給老師，教師展示作品，並請小組發表分享設計心得。
- 2.教師教學參考範例程式檔，如附件 2。

(六)、新增 ContentView 頁面：

教師導學

- 1.先列出頁面主要程式架構：
struct ContentView: View

// A.宣告變數與定義函數

```
var body: some View {  
  VStack {
```

// B.新增校徽圖片、新增標題"生態紀錄"、新增按鈕

// C.列出植物踏查記錄

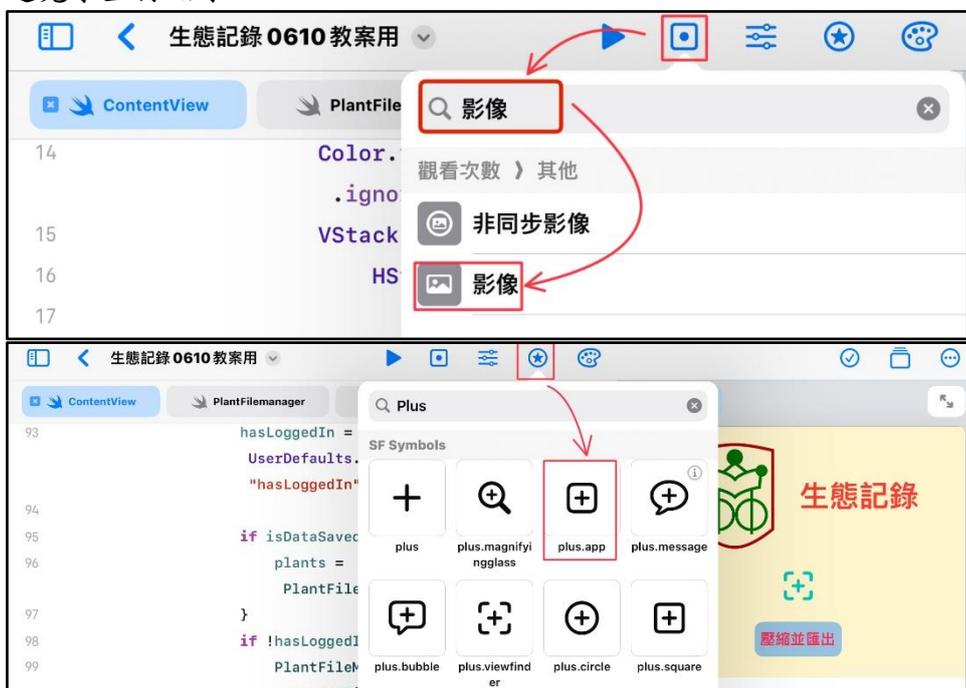
```
}
```

```
.onAppear{
```

// D.畫面載入前要執行的內容

```
}
```

2. SwiftUI 提供影像指令，讓你可以由外部匯入圖檔，或是取用內建圖檔，或是透過瀏覽資料庫控制項目(編輯畫面正上方-控制項目-影像)直接帶出，也可避免學生打錯字。



(1).影像指令初體驗，基本語法如下：

```
Image ("Image Name",)
```

Image Name：圖檔檔名

提示：

修飾符.foregroundColor()前景顏色

修飾符.font(.largeTitle)字體大小

3.教師 Airdrop 作業範例程式檔，如附件 2 程式檔，請小組互相討論，根據教師導學所學，如何新增以下欄位，並加以美化。

序號	欄位名稱	相關變數	
①	校徽與標題	x	
②	加號按鈕	isShowAddView	

③	壓縮並匯出	x	
④	我的推測	x	植物踏查列表
⑤	植物地點	plants.[index].location	
⑥	植物種類	plants.[index].species	
⑦	子葉類型	plants.[index].cotyledon	

```

struct ContentView: View
// A.宣告變數與定義函數
var body: some View {
Vstack {
// B.新增校徽圖片、新增標題"生態紀錄"、新增按鈕

// C.列出植物踏查記錄
}

.onAppear{

// D.畫面載入前要執行的內容

}

```



4.教師提供範例程式檔，引導學生完成檔案。

5.作業範例程式檔，如附件2，請教師刪除註解//程式碼 B.新增校徽圖片、新增標題"生態紀錄"、新增按鈕，即可將檔案提供給學生練習使用。

■檔案連結：

https://drive.google.com/drive/folders/1doLf2CsZg7iSjgLuztcQMpu4Gow2ZA_T?usp=sharing

(七)、生態紀錄-校園踏查：

將全班學生分為五組，以自製的 APP 進行校園生態紀錄，範圍以校門口延伸到後校門籃球場，學生手持 ipad 進行校園踏查。

課前自學

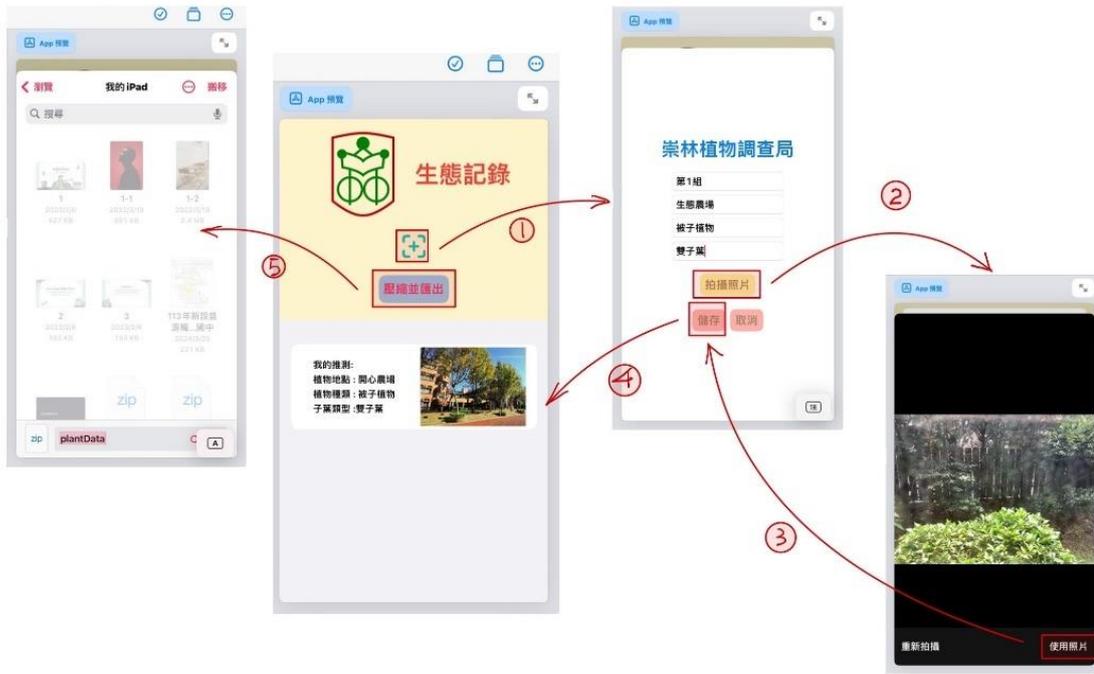
學生利用課餘時間登入學習吧熟悉 ipad 操作流程，並確認小組成員負責校園範圍，以不重覆為原則。

教師導學

教師提供小組任務，每組須在 25 分鐘之內找到蘚苔植物、蕨類植物、裸子植物和被子植物的其中三種，並以 APP 進行分類、拍攝及上傳資料。並提醒同學不可單獨行動，更要注意安全。

1. 全班同學分為五組，並討論組內同學負責的校園區域。

2. 教師計時 25 分鐘，讓同學手持 ipad 到校園觀察並紀錄、分類植物。



組內共學

小組在 25 分鐘之內進行校園植物踏查，彼此約時間集合討論紀錄的結果，確認植物的分類是否正確，再接著尋找植物的分布。於預定時間完成植物紀錄並回到教室。

1. 同學分組進行校園植物觀察，利用自製 APP 進行紀錄及分類。
2. 教師巡視同學踏查狀況，即時提供 APP 使用的協助，並注意同學行蹤。

組間互學

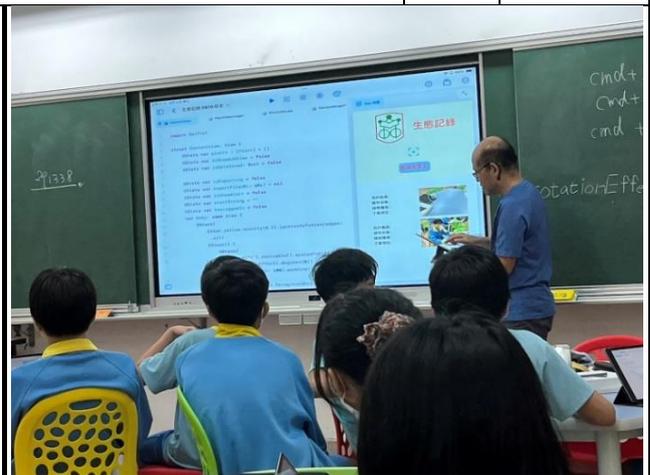
進行小組競賽，讓同學分享紀錄結果並由教師進行結算分數並總結課程。

1. 小組上傳 APP 紀錄的結果至雲端。
2. 小組輪流上台分享紀錄的植物結果及分類情形。
3. 教師提供回饋並提點其他的同學補充內容，最後總結課程活動。

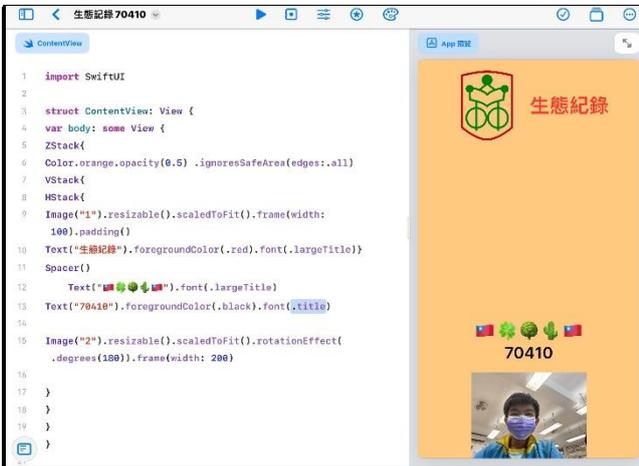
教學
成果



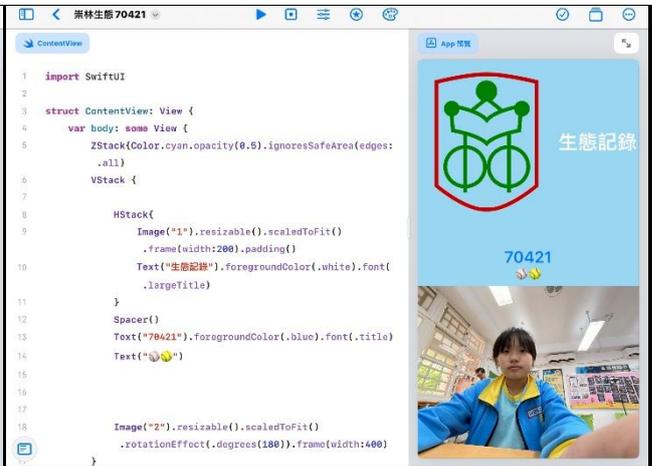
團隊夥伴教學共備



教師展示 APP



SwiftUI 視覺化程式設計



SwiftUI 視覺化程式設計



程式設計課堂一隅



程式設計課堂一隅



學生運用植物調查局 APP 踏查一隅



學生運用植物調查局 APP 踏查成果分享

教學心得與省思

一、今年第一次與生物老師合作進行跨域 PBL 教學課程單元，採取以下多元評量方式檢核學生學習成效：

1. 教師課間小組觀察：

教師課堂間，不斷於各小組間巡視觀察學生課堂表現，適時引導各組學生完成任務，同時記錄各組學生表現與互動情形，如發現大部分組別有卡關現象，可暫停所有組別學習任務，進行教師導學，協助學生解決問題。

	<p>2.運用學習平臺即時預覽，即時投影(Airplay)功能抽驗學生作品： 每個小單元的組內課間實作，主要在於觀念的澄清，學生都必須動手實作，適時的抽驗可以強化學生學習成效</p> <p>3.善用學習平臺加分機制： 在 Coding 的學習上，班級學生表現差異較大，採用 S 形異質性分組，有濟弱扶傾的功效，並適時以小組獎勵方式，鼓勵組內同學互相合作，一起解決問題。</p> <p>4.小組總結報告： 分組學生代表進行報告，將學習紀錄歷程 Airplay 投影至布幕、大屏或電子白板，提供各組學生參考並深化學生本單元學習內涵。</p> <p>5.總結性評量： 各組學習任務結束後，搭配學習吧平臺線上評量，檢視學生在本單元學習成就，題目不需要多。</p> <p>6.教師講評與建議 最後，再由授課教師進行講評，並展示學生個別自主學習成果、教師觀察小組學習歷程，並總結本單元學習內容及學生學習表現，作為未來執行本單元活動的重要參考。</p> <p>7.課程單元反思： 運用 google 表單進行學生自評與互評，觀察學生學習心得，包括學習內容熟悉度及對教材的體驗與心得並提供教師回饋，作為修正課程設計的參考資料。</p> <p>二、本課程設計是繼112年科技領域-掃落葉機器人單元後，跨域結合自然領域-生物單元，所發展出結合學校在地特色的課程，經過半年時間的醞釀與實施，從學校數位社群會議開始教學主題的發想，到社群跨域共備，再回到領域課程單元教學活動的設計，最後實際產出落實於學生的學習，一路走來著實不易，特別感謝課程設計核心夥伴盈欣老師及鈺淋老師。</p> <p>三、科技領域教學設計主要由孫郁文老師、沈鈺淋老師負責，緊扣七年級程式設計、演算法單元為主，希望提供學生不一樣的學習體驗，將小學所學圖像式程式語言轉換至文字式語言，我們提供了大量的模板(包含了數個 Swift 檔案)，也嘗試運用 Xcode 開發 Playgrounds Guide Tutorial APP，以適合學生課前自主學習或補救學習，減少學生學習程式的障礙，而能得到學習程式的樂趣，利用所學及所得很直觀的方式產出 iPad APP，最後，再利用自己設計的 APP，運用於生物課程進行校園植物的踏查，就算功能測試無法完全正常，最後也能使用教師派發的範例程式檔，未來也能擴充延伸至校園或林口地區鳥類及其他生態的踏查。</p> <p>四、本次 PBL 專題式資訊科技課程讓學生學習到變數、串列、迴圈以及 UI 介面的設計，對七年級學生稍有難度，適合在七年級下學期實施，建議可由7年級上學期以遊戲式學習方式，逐步帶出語法的觀念，例如使用 iPad SwiftPlaygrounds APP-開始編寫程式碼 Swift5.8版，配合各版本學習內容如程式設計單元及演算法單元一併實施，不但能提升學生的學習動機，學生的學習成就也超乎師長預期，如果學生學習的不錯，也可以讓學生自學-學習程式設計1、學習程式設計2、APP 使用入門。</p>
<p>參考資料</p>	<p>1.學習編寫程式1和2(教師指南) iPad 書籍 APP 2.人人可編碼解謎闖關(教師指南) iPad 書籍 APP 3.iPad Swift App 基礎專案實作作者：Michael Pan 4.Xcode Swift APP 進階專案實作 作者：Michael Pan 5.教育部教育大市集-自然科學之植物體與細胞 VR 教材 6.教師示範影片連結：https://youtu.be/ZRxxC8_tUQk</p>
<p>附錄</p>	<p>附件1：作業範例程式檔-附件1(簡易版) 附件2：作業範例程式檔-附件2(簡易版) 附件3：生態紀錄-植物調查局完整 APP 壓縮檔連結： https://drive.google.com/drive/folders/1-UI500BP-</p>

[DU_6GikgMBmJQYWZiy3GbTT?usp=sharing](#)

附件1：作業範例程式檔-附件1(簡易版)

```
import SwiftUI
```

```
struct AddView: View {
```

```
    // A.宣告變數與定義函數
```

```
    @Binding var plants : [Plant]
```

```
    @State var date: Date = Date()
```

```
    @State var InputLocation : String = ""
```

```
    @State var InputSpecies : String = ""
```

```
    @State var InputCotyledon : String = ""
```

```
    @State var InputGroup : String = ""
```

```
    @Binding var isShow : Bool
```

```
    @State var openCamera : Bool = false
```

```
    @State var captureImage : UIImage?
```

```
    @State var InputImageName : String = ""
```

```
    var body: some View {
```

```
        VStack{
```

```
            // B.新增標題及文字欄位
```

```
                Text("崇林植物調查局").font(.largeTitle).foregroundColor(.blue)
```

```
                TextField("學生組別", text: $InputGroup)
```

```
                    .textFieldStyle(.roundedBorder)
```

```
                    .frame(width: 200)
```

```
                TextField("植物地點", text: $InputLocation)
```

```
                    .frame(width: 200).textFieldStyle(.roundedBorder)
```

```
                TextField("植物種類", text: $InputSpecies)
```

```
                    .frame(width: 200).textFieldStyle(.roundedBorder)
```

```
                TextField("子葉類型", text: $InputCotyledon)
```

```
                    .frame(width: 200).textFieldStyle(.roundedBorder)
```

```
            //C.拍攝照片按鈕
```

```
                Button("拍攝照片"){
```

```
                    openCamera = true
```

```
                }.foregroundColor(.brown).font(.system(size:20)).frame(width: 100,height:
```

```
                40).background(Color.yellow.opacity(0.4)).cornerRadius(10)
```

```

        CameraImagePicker(image: $captureImage)
    }
    if let image = captureImage {
        Image(uiImage:image)
            .resizable()
            .scaledToFit()
            .frame(height: 200)
            .padding()
    }
}

```

HStack{

```

//D.儲存和取消按鈕
Button("儲存"){
    let formatter = DateFormatter()
    formatter.dateFormat = "yyyyMMddHHmm"
    var newplant = Plant( date: date,group:InputGroup,scientificName:
    InputLocation, family: InputSpecies, feature: InputCotyledon)
    if let selectimage = captureImage{
        let imageName = formatter.string(from: Date())
        newplant.imageName = imageName
        PlantFileManager.shared.saveImage(image: selectimage,
        imageName:imageName)
    }
    plants.append(newplant)
    PlantFileManager.shared.saveData(data: plants)
    UserDefaults.standard.set(true, forKey: "isDataSaved")
    isShow = false
}.foregroundColor(.brown).font(.system(size:20)).frame(width: 60,height:
40).background(Color.red.opacity(0.4)).cornerRadius(10)
Button("取消"){
    isShow = false
}.foregroundColor(.brown).font(.system(size:20)).frame(width: 60,height:

```

```
40).background(Color.red.opacity(0.4)).cornerRadius(10)
```

```
}  
}  
}  
}
```

```
import SwiftUI
```

```
struct ContentView: View {
```

```
    @State var plants : [Plant] = []
```

```
    @State var isShowAddView = false
```

```
    @State var isDataSaved: Bool = false
```

```
    @State var isExporting = false
```

```
    @State var exportFileURL: URL? = nil
```

```
    @State var isShowAlert = false
```

```
    @State var alertString = ""
```

```
    @State var hasLoggedIn = false
```

```
    var body: some View {
```

```
        ZStack{
```

```
            Color.yellow.opacity(0.2).ignoresSafeArea(edges:.all)
```

```
            VStack() {
```

```
                HStack{
```

```
                    Image("1").resizable().scaledToFit().rotationEffect(.degrees(0)).frame(width: 100).padding()
```

```
                    Text("生態記錄").foregroundColor(.red).font(.largeTitle)
```

```
                }
```

```
                Button{
```

```
                    isShowAddView = true
```

```
                }label: {
```

```
                    Image(systemName:
```

```
                        "plus.viewfinder").font(.largeTitle).foregroundColor(.mint)
```

```
                }.sheet(isPresented: $isShowAddView){
```

```
                    AddView(plants: $plants, isShow: $isShowAddView)
```

```
                }
```

```
            }Button("壓縮並匯出") {
```

```
                PlantFileManager.shared.saveCSVData(data: plants)
```

①

②

3

```
if let zipURL = PlantFileManager.shared.zipAllFile() {
    exportFileURL = zipURL
    isExporting = true
} else {
    alertString = "壓縮檔案出現錯誤"
    isShowAlert = true
}
}.frame(width: 100,height: 40)
.background(.blue.opacity(0.4)).cornerRadius(10)
.padding()
.fileMover(isPresented: $isExporting, file: exportFileURL) { result
in
    switch result {
    case .success(let file):
        print(file.absoluteString)
    case .failure(let error):
        print(error.localizedDescription)
    }
}
.alert(isPresented: $isShowAlert) {
    Alert(title: Text(alertString), dismissButton: .default(Text("確定"))) {
        alertString = ""
    })
}
}
}
List{
    ForEach(0..
```

4

```
Text("植物種類 : \${plants[index].species}")
Text("子葉類型 : \${ plants[index].cotyledon}")

}
.font(.footnote)
.padding()
Spacer()
if let imageName = plants[index].imageName{
    if let image =
        PlantFileManager.shared.loadImage(imageName:
            imageName){
        Image(uiImage: image)
            .resizable()
            .scaledToFill()
            .frame(width: 150,height: 80)
        }
    }
}.frame(height: 100)

} .onDelete{ indexSet in
    if let index = indexSet.first{
        if let imageName = plants[index].imageName{
            PlantFileManager.shared.deleteImage(imageName:
                imageName)
        }
    }
    plants.remove(atOffsets: indexSet)
    PlantFileManager.shared.saveData(data: plants)
}
}
```