臺北市立建國高級中學 109 學年度第二學期

科技應用科課程學習成果

編者：二年十班 毛柏毅 指導老師：黃敦紀 老師

大數據應用專題：天氣資料獲取與應用

一、 研究動機

天氣與我們的生活息息相關，不論是影響我們穿著的氣溫，或者是牽涉生活作息的降雨，都是每一天每個人不可或缺的資訊。尤其在近幾天臺灣連日乾旱，天氣的資訊便顯得更加重要。因此，我決定先初步研究由中央氣象局提供的氣象預測資料，並且製作圖形化的互動介面，藉以更加瞭解氣象資訊，幫助氣象研究。

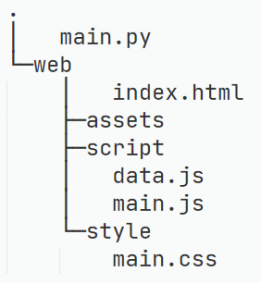
二、 初步設想與製作思路

要利用氣象資料，首先必須找到一個能夠穩定取得的資料來源，在此我決定使用「中央氣象局開放資料平臺」所提供的資料截取 API，使用其中「一般天氣預報——今明 36 小時天氣預報[1]」的資料實作。透過 requests 模組取得 JSON 格式的氣象資料，便可以直接使用 python 操作並利用之。顧名思義，資料中包含了今明 36 小時的天氣預報，其中包括常用的高低溫、降雨機率，還有評估各種氣候因子所得的舒適度和天氣概況，以及最重要的預報對應時段。在此我決定充分利用這些資料，盡可能整齊美觀地將氣溫、降雨機率、舒適度這三個參數呈現出來。

在製作初期，我預設直接將所獲取的資料初步分析並排版後透過終端機輸出，不過這樣缺乏美觀性及互動性，於是我決定透過利用 HTML 呈現畫面的 eel 模組來製作使用者介面。原本我打算沿用使用終端機時的做法，透過把每一個縣市的天氣資料放在各自對應的區塊內，一次在網頁上排開，但這樣會導致可讀性和便利性降低，因此在幾經查找之下，我決定利用 HTML 的 map 元素製作臺灣縣市地圖，當使用者把游標移動到縣市上時，畫面右側便會顯示相對應的天氣資訊，其成果可參見下文「四、功能與操作說明」。

[1] 資料 API 網址：https://opendata.cwb.gov.tw/api/v1/rest/datastore/F-C0032-001

三、 程式架構

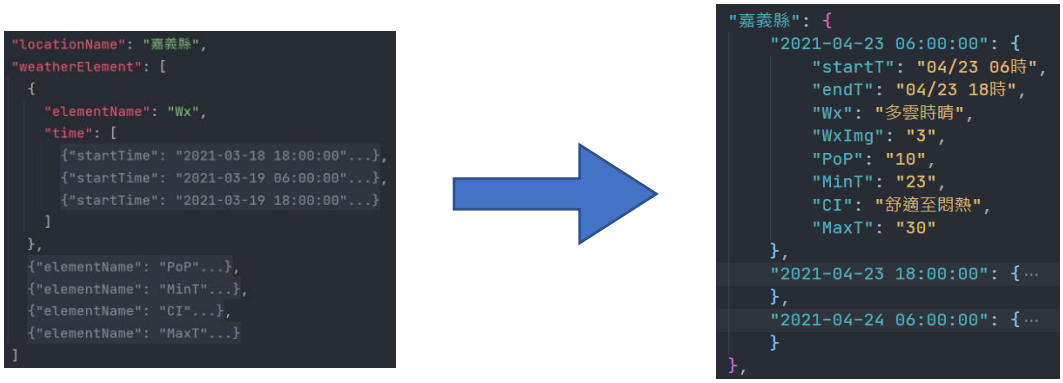


在整個程式裡面，其主要資料結構大致上如右圖，其中包含：使用者介面呈現所需的網頁檔 index.html、main.css 和 assets 資料夾底下的圖片；處理網頁內容的 main.js 和 data.js；以及最重要的主執行檔 main.py。

Python 部分

在 main.py 中主要的兩部分，一是負責取得並預處理天氣資料的 fetch_weather_data 函式，二是使用 eel 開啟網頁程式的主函式。

在簡單的整理之後，原本每個縣市下以項目做分類的天氣資料，就會變成用時段分類，以便使用介面端的呈現。除了整理並重新排序資料之外，我也將原本冗長的時間戳，重新格式化為易讀的樣式，並且額外加上代表天氣概況的圖示編號，詳細的應用可在後文看到。



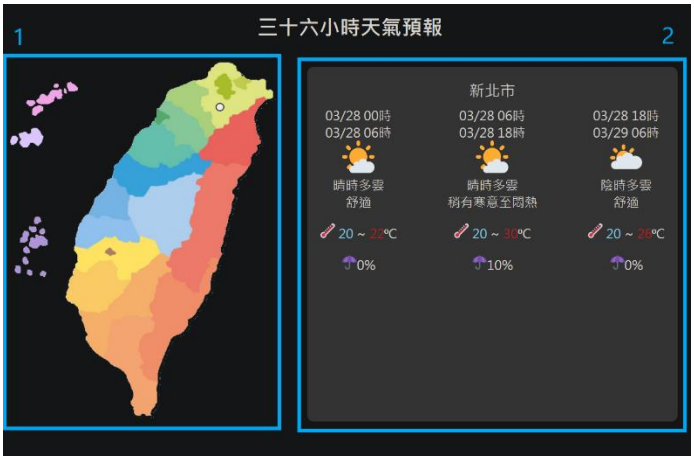
在 main.py 中也包括 `eel.init("web")`和 `eel.start("index.html")`這兩行程式，其意義在於將 web 資料夾作為使用介面所需網頁的根資料夾，並且將 web/index.html 作為該介面的首頁，於程式處理完畢後在瀏覽器中開啟，再接下來就是 HTML 和 JS 派上用場的時候了。

HTML & JavaScript 部分

首先為了能夠讓網頁伺服器和 python 主程式互相溝通，在此需要引入 /eel.js，且需在 python 端需要使用的函式前方標上`@eel.expose`的標籤，方可在 JavaScript 端使用（在 eel 物件底下，作為一個「會回傳 promise 的函式」使用）。在 JS 順利接收到預先處理好的天氣資料時，便可以生成對應的內容，並利用 jQuery 加進最終呈現的網頁中。

此外，透過前文所提到的 map 元素，對於每一個縣市生成對應的觸控範圍，在指到一縣市時將其對應的天氣資料顯示出來，並且隱藏其他縣市的內容，便可以達到觀看單一縣市天氣資料的效果。為了能夠凸顯目前所選中的縣市，我也額外加上了一個指標元件，在每一次更換選中縣市時移動到相對應的位置。包括多邊形碰撞箱以及指標的位置（還有天氣圖示的編號與檔名之映射），其他所需的資料都儲存在 data.js，在引入的同時便會被載入。在這些程式的互相配合之下，便能夠做出一個簡易的縣市天氣查詢系統。

四、 功能與操作說明



實際運行程式所得的操作界面如右圖，可分為縣市選擇（地圖）和天氣資訊兩個部分。在畫面左側，滑鼠移至一縣市的位置時，首先畫面右側會出現該縣市接下來 36 小時的天氣概況，以 12 小時為一時段橫向排列。此外地圖上會有一表示選中縣市的小白點，顯示在地圖上對應縣市的位置。

天氣概況部分，對於每一個時段的資訊，由上至下分別為：該時段的起始時間和結束時間、天氣概述以及對應的圖示、用顏色區分的最低溫和最高溫、降雨機率等。此外若降雨機率超過 30%，代表降雨機率的數值左方之雨傘圖示會被替換為下雨的圖示，提醒使用者要記得帶傘。以上所呈現的資料，基本上就已經足夠一般人日常生活使用。

要閱覽此專題的原始碼，可於專案的 GitHub 程式庫查看。而利用 pyinstaller 預先打包好的執行檔，則可以在這裡下載。

五、 遭遇困難與解決辦法

使用者介面

誠如前文所述，為了能夠製作易讀且簡潔的使用者介面，我選擇搭配容易處理的 HTML 來製作。不過這樣一來，為了動態生成網頁內容，勢必得接觸到我還不太熟悉的 JavaScript。尤其是 eel 處理跨語言函式的方式，為了避免兩側不同步運算造成的時間差，其使用的 promise 概念更是我從來沒有接觸過的。不過也趁著這次機會，我也學到了許多 JavaScript 的特殊概念，對於未來製作其他也相當有幫助。

此外，由於要呈現全臺灣數十個縣市的天氣資料，我必須透過某種方式讓使用者自行選擇要呈現的縣市，最終所採用的方式，便是前文數次提及的 map 元素。透過背景圖片搭配預先繪製好的碰撞箱，方才得以達到選擇縣市顯示對應資料的效果。

授權碼與環境變數

由於中央氣象局開放資料平臺所提供的資料，需要提供事前申請好的授權碼（Authorization Key）才能夠提取，我勢必要將這段資料儲存在應用程式的某處。不過為了資安方面的考量，我也不可能將其直接放在 python 的主執行檔內。在多方參考之下，我決定利用 dotenv 模組來隱藏授權碼。透過將其放在獨立的.env 文件內，並且搭配簡單的一行 load_dotenv()，便能夠讀取並使用裡面所存的授權碼，進而順利提取天氣資料供畫面呈現使用。

六、 省思與未來展望

本次所製作的程式，由於時間有限僅有製作提取資料和畫面呈現的部分，畫面的美化也做的不甚理想。此外由於一開始仍未構想使用者介面的時候，沒有想到要使用 JavaScript，以至於如今需要在 python 和 JS 兩側下工夫的窘境。有鑑於 JS 在線上資料處理和畫面呈現上的優勢，若未來有機會，我希望能夠使用全 JS 來製作使用者介面，亦即使用 Node.js 來作為開發環境。此外，本次沒能夠利用其他氣象局提供的資料，也沒能夠做到分析數據的功能，對於本次的專題也是一大遺憾。若有機會，我也希望可以深入研究氣候資料，透過自己撰寫的資料分析程式來真正分析這些資訊，進而製作出真正的「大數據」應用程式。