

程式語言學習建議指引

411021215 張閔勳

一、前言

W3School 是初學者常用來學習程式語言的網站，但是對於初學者來說有時並不直觀。舉例來說，我想搜尋 loop 相關的主題，但是程式語言中有許多種迴圈，初學者肯定不知道其差別。如果能將所有可能列出來，並顯示相對應主題的介紹，豈不是更加方便嗎？

在網路搜尋時，似乎沒有看到相關的網站，所以決定動手設計一個。雖然 W3School 大致上的功能和本系統相同，但是最大的不同是本系統提供預覽頁面，這樣就不需要進入網頁才知曉其中的內容。預覽的原理是先行將網頁第一段文字爬取下來，再更新到網站上。

二、系統描述

1. 套件、工具使用：

在這個系統中，我使用到 Python Flask 搭配 HTML, CSS 來搭建一個動態網頁，動態網頁的最大特色是網頁內容可以根據使用者需求來改變內容。比較特別的是除了原本的 HTML 語法，Python Flask 讓使用者可以在網頁中安插 Jinja 語法，這樣我們就可以在 HTML 中加入判斷式、迴圈等，使得網也更彈性。另外，我也使用 requests 搭配 beautifulsoup 來爬取指定的網頁內容。

2. 網頁運作流程：

- a、當 app.py 被啟動時，網頁自動進入根目錄，並渲染成 home.html，也就是主頁的呈現。
- b、透過網頁導覽列中的 Learning Suggestions 可以進入系統主要的功能頁。
- c、輸入欲查詢的「程式語言名稱」，如：python, c, c++, javascript 等。若搜尋成功，會被導引至下一個頁面，否則會被刷新成同一頁面。搜尋失敗的最大原因是 w3school 未提供該程式語言的教學。
- d、輸入欲查詢的「主題名稱」，如：loop, list, function 等。（可包含多筆輸入，用空白隔開）若不清楚主題名稱，可以再次輸入「程式語言名稱」。若搜尋成功，會被導引至下一個頁面，此時頁面會呈現所有與關

鍵字相關的結果。

- e、 使用者可以根據該頁面提供的資料篩選出所需的內容。每一筆資料都包含對應網頁的部分內容和網頁連結方便使用者使用。
- f、 當一切結束後，可以透過網頁導覽列中的 About Me, Learning Suggestions 進入想要的頁面。

3. 主要函式概述：

- a、 `@app.route('/'):` 當網頁被開啟，會自動進入根目錄，並呼叫對應的函式。此時，網頁會被渲染為 `home.html`，這個頁面是網頁的主頁。
- b、 `@app.route('/learn', methods=['GET', 'POST']):` 透過導覽列可以進入這個頁面，由於這個頁面包含一個互動輸入框，因此需要 `Post` 來完成，如此後端邏輯才可以進行後續的處理。根據不同的狀態 (`state = 0/1`)，網頁會渲染成不同的頁面，包含 `index.html`, `lang.html`, `results.html`。
- c、 `class Crawler:` 不論是哪個網頁的爬蟲，都包含幾個必要的步驟，因此設計成一個「基底類別」供其他的「衍生類別」使用。
- d、 `class Header_Crawler(Crawler):` 繼承自 `class Crawler`。這個爬蟲設計的目的是要爬取 `w3school` 導覽列上所有的程式語言名稱和對應的連結。如此後續使用者在輸入時，才有對應的資料可以使用。
- e、 `class W3S_Crawler(Crawler):` 繼承自 `class Crawler`。這個類別的主要功能是針對使用者輸入的程式語言進行對應網站的爬蟲，並將正確的結果傳遞給 `Flask`。
- f、 `def getContent(url):` 當使用者輸入欲查詢的主題時，`W3S_Crawler` 會呼叫這個函式，而這個函式會透過對應的連結將網頁爬取下來並回傳。

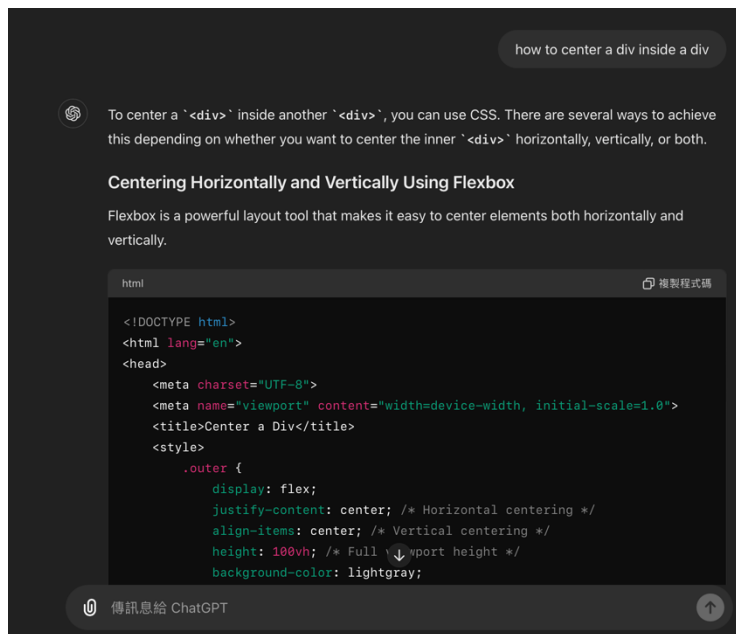
4. 困難與解決：

`Flask` 和一般的網站不太一樣，無法透過相對路徑來存取檔案，特定的檔案必須要存在特定名稱的資料夾。如：`html` 皆須要存在 `templates` 資料夾，`css` 皆需要存入 `static` 資料夾，否則存取會失敗。詳細資料可以參考這份官方說明 <https://flask.palletsprojects.com/en/3.0.x/quickstart/> 包含範例和原理讓人可以輕鬆學會。

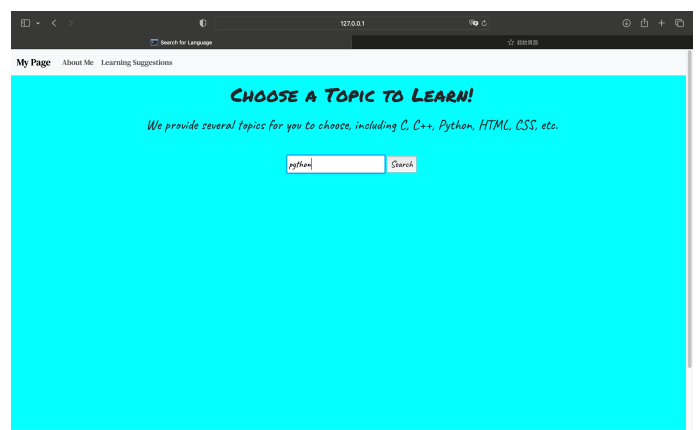
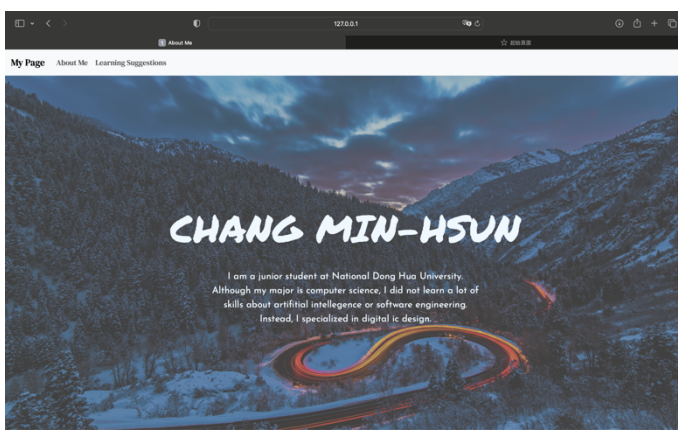
由於網站包含多個頁面，但是所有頁面都共享一個導覽列，若設計網頁後，發現導覽列需要更新內文，那麼我們需要多次更改相同的內容。此

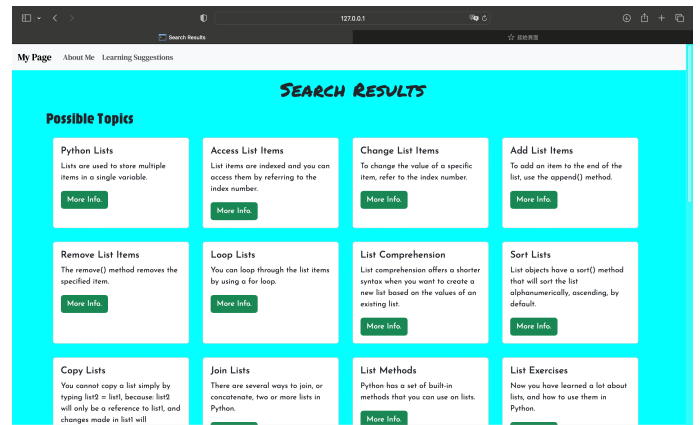
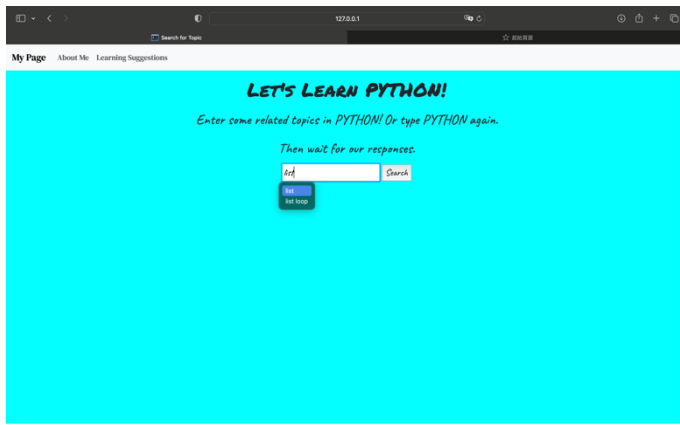
時，我們可以將相同的部分獨立出來成為新的 `base.html`，再透過 `{% extends base.html %}` 將這部分加入其他 `html`。如此，當導覽列需要更動，只需要修改 `base.html` 的部分即可。

網頁設計中，排版其實是很需要經驗的，舉例來說，要將一個物件置中的方式有很多種，像是透過計算 `margin` 或是透過 `transform` 等都可以達到。不熟悉時，ChatGPT 其實幫了大的忙，如圖。雖然不是每次都可以達到理想目標，但是了解 ChatGPT 的錯誤也是一種學習。



5. 系統截圖：以查詢 Python 中的 list 為例





三、結果與反思：

本系統最大的問題是並不支援所有程式語言，因為本系統爬取的對象是W3School。接下來是關鍵字的識別，如果使用者輸入的是程式語言的縮寫，或是以中文輸入都會出現搜尋失敗。

未來這個系統可以爬取更多的教學網站，讓使用者達到更好的體驗。也許可以加入簡單的人工智慧來加強解析關鍵字的能力。