**第四組**

**作業說明：**  
 案例設計如何支持學習目標： 說明工具的功能，例如單字查詢、自動生成造句示例，如何幫助使用者提升單字量及應用能力。

 學習者效益： 強調工具能如何透過直觀的操作與即時回饋，增強學習者的自信心，讓學習過程更具吸引力。

 與現實需求的對應： 承接全球化背景，說明工具如何助力學習者應對語言能力在職場和學術領域的挑戰。

 未來效益: 透過本次開發的英文單字字典工具，不僅提升使用者解讀英文造句和詞義的能力，也讓學習更具有互動性和個人化。進一步展望未來，本工具能成為可客製化的單字記憶APP，滿足不同學習者的需求，例如依照個人學習進度與興趣建立專屬字典，並搭配多媒體教學，增強記憶效果。

該工具還可結合語音辨識、即時翻譯等技術，支援學習者在多元環境下應用，打造高效學習系統。不僅適合學生，也能輔助教師設計課程內容，甚至在專業領域提供術語教學。長遠來看，這將促進語言學習工具的進一步數位化與普及化，造福更多學習者。

**背景資料**：

1. Python 用於資料庫操作（CRUD）

這段程式碼實現了對SQLite資料庫的增刪改查（CRUD）操作，用來管理單字資料和文章資料。在這些操作中，包括了：

* 資料庫初始化（init\_db 和 initialize\_database 方法）：用來建立必要的資料表。如果資料表不存在，它們會被自動創建。這是資料庫管理的基礎步驟。
* 新增資料（insert\_word 和 insert\_article 方法）：在資料庫中插入新的單字或文章資料。在插入資料時，使用了SQL語句來確保資料可以正確儲存。
* 查詢資料（search\_data 和 search\_article 方法）：這些方法會根據條件從資料庫中檢索資料。如果符合查詢條件的資料存在，將會返回該資料的詳細資訊。
* 更新資料（update\_word 和 update\_article 方法）：根據給定的條件，更新已存在的資料。例如，修改單字的描述，或更新文章內容和翻譯。
* 刪除資料（delete\_word 和 delete\_article 方法）：刪除特定的單字或文章資料。這些方法會檢查資料是否存在，並執行刪除操作。

2. 網頁爬蟲操作

DataCrawl 類別是一個爬蟲類別，用來從網頁中抓取特定資料。在這段程式碼中，主要是爬取來自Cambridge字典的英漢繁體翻譯資料：

* HTML資料抓取（crawl 方法）：使用 requests 和 BeautifulSoup 庫來爬取網頁內容並解析HTML。請求網頁資料並提取包含單字翻譯、定義、例句等重要區塊的資料。利用網頁上的選擇器解析這些元素，如詞性、翻譯、例句等。
* 氣泡排序（bubble\_sort\_phrases 方法）：將包含「phrase」的單字資料移到所有資料的後面，使其按詞語排序，幫助整理資料。
* 資料輸出為JSON格式：爬取的資料以JSON格式保存，這是便於後續分析、展示或存儲的重要格式。

3. 相關知識

1. SQLite 操作：

* SQLite是一個輕量級的資料庫引擎，嵌入式，適合輕量級應用。它不需要一個獨立的資料庫伺服器，所以在本地儲存非常方便。

1. 爬蟲技術：

* 網頁爬蟲技術常用於從網絡上自動提取資料，這裡使用requests庫進行HTTP請求，BeautifulSoup來解析和提取HTML內容。

1. JSON 格式：

* JSON格式是一種輕量的資料交換格式，通常用於儲存和傳輸結構化資料。在此代碼中，爬取的資料會轉換為JSON格式來儲存。

1. 例外處理：

* 在對資料庫進行操作時，程式使用 try-except 塊來處理可能發生的錯誤，例如在插入重複資料或查詢失敗的情況。

**環境介紹：**  
1. 硬體 (Hardware)

CPU：處理單元，控制整個系統的運行。

記憶體 (RAM)：系統運行的臨時儲存空間。

硬碟或 SSD：數據存儲設備，用於長期存儲操作系統、軟體和數據。

顯示卡 (GPU)：圖形處理單元，負責處理與顯示圖像相關的工作。

輸入設備：如鍵盤、滑鼠，或其他設備，如掃描儀、手寫板等。

網絡設備：例如網卡，實現設備間的通訊與數據傳輸。

2. 作業系統 (Operating System)

Windows 11

3. 使用工具 (Tools and Software)

編程語言： Python

IDE/編輯器：VS Code

數據庫工具: sqlite3

版本控制工具：Git，配合 GitHub

4. 數據爬取與爬蟲工具

BeautifulSoup、Selenium、Scrapy：處理數據爬取，從網站擷取資料，並進行處理。

API工具：例如 Postman，用於測試 API 請求與響應，進行數據獲取與交流。

**實作方法：**  
1. 環境檢查 (Environment Check)

作業系統版本：Windows 10以上

已安裝依賴庫或軟體：Python 環境中的requests、beautifulsoup4。

網絡連接：檢查網絡連接狀況，以便下載必要的工具或更新。

2. 安裝與設定 (Installation and Setup)

作業系統版本：Windows 10以上

已安裝依賴庫或軟體：Python 環境中的requests、beautifulsoup4。

網絡連接：檢查網絡連接

安裝需要的資料庫系統: SQLite

3. 解決方案步驟

3.1 問題分析與理解

根據題目需求，本作業的目標是設計一個程式，用來抓取網路上的英文字典資料並將其轉換為可視化介面。在這個過程中，我們首先需要分析並明確幾個關鍵步驟：

* 抓取網頁內容：我們需要使用 Python 的爬蟲工具，從網站上抓取英文字典的資料。抓取後的資料需經過解析並抽取所需的字詞、定義和例句等信息。
* 處理字典資料：抓取的資料需要進行過濾和格式化，以便後續使用者查看。
* 製作介面：在抓取並處理好資料後，需要設計一個圖形介面來展示這些資料給用戶。可以使用簡單的 GUI 库如 Tkinter 或 PyQt 來構建。

3. 2 編寫程式

這部分主要涉及以下內容：

* 抓取網頁資料：使用 Python 的爬蟲庫 requests 或 beautifulsoup4 從指定的字典網站抓取英文字典資料。
* 解析 HTML 文檔：抓取回來的資料通常是 HTML 格式，我們需要使用 BeautifulSoup 來提取關鍵資料，例如單字的拼寫、發音、定義等。
* 顯示介面：在前端製作介面並展示抓取的字典資料，可以使用 Tkinter 或類似的 GUI 套件來顯示查詢結果。

3. 3步驟細化

設置環境：

* 安裝爬蟲所需的庫，如 requests, beautifulsoup4, Tkinter 等。
* 測試抓取網站和解析資料的流程。

抓取字典資料：

* 使用 requests 库發送 HTTP 請求來抓取網頁。
* 使用 BeautifulSoup 解析 HTML 並提取字典中的單詞、定義、發音、例句等內容。

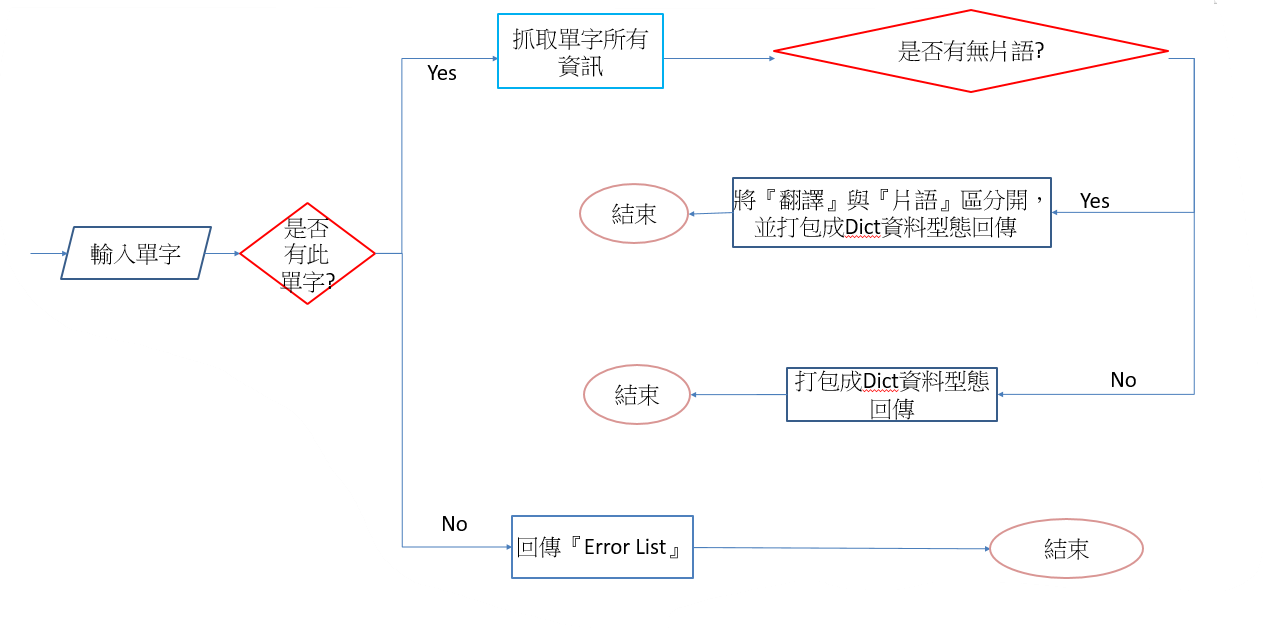
3. 4測試與優化

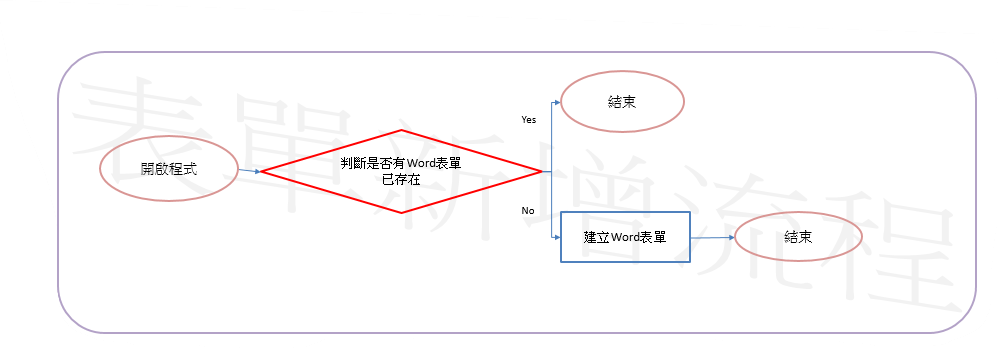
在完成編程後，需進行充分測試，包括：

* 測試使用者輸入不同單字能否正確查詢資料。
* 測試當網站結構發生變化時，程式是否能正確處理這些變化（例如網站上字詞定義的 HTML 結構改變）。
* 優化性能，如果查詢過程較慢，可考慮使用異步請求來加快爬取速度。

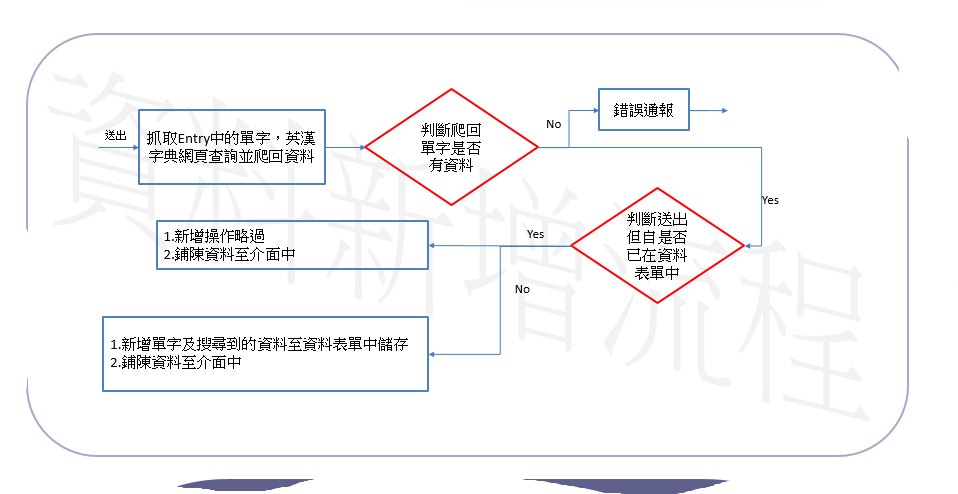
4.流程圖

爬蟲流程

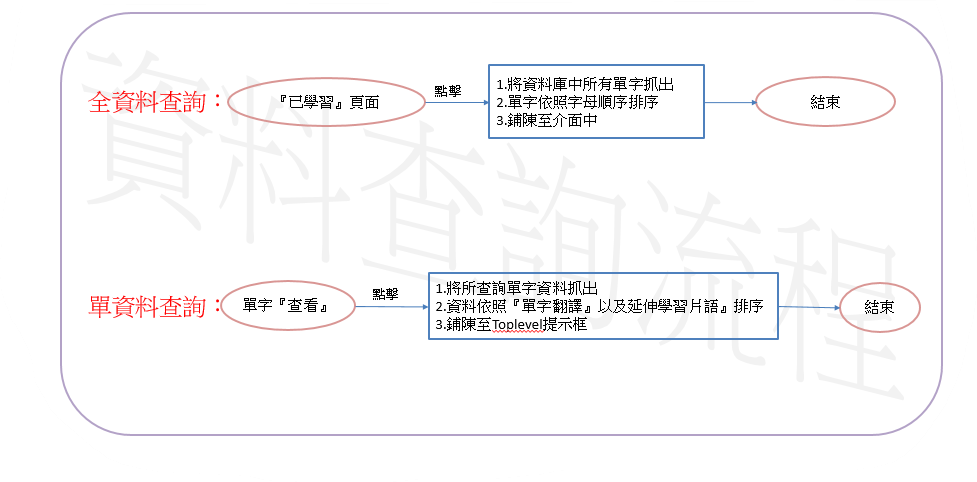


資料表單

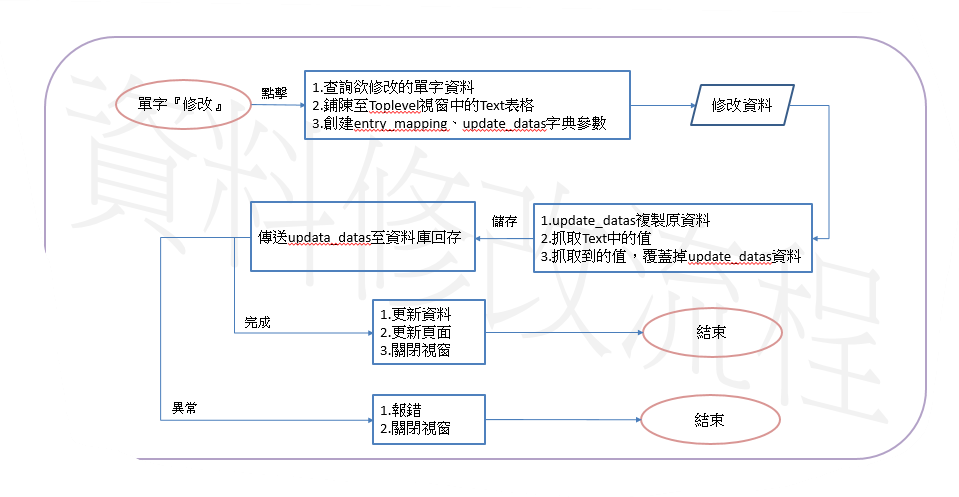
資料新增



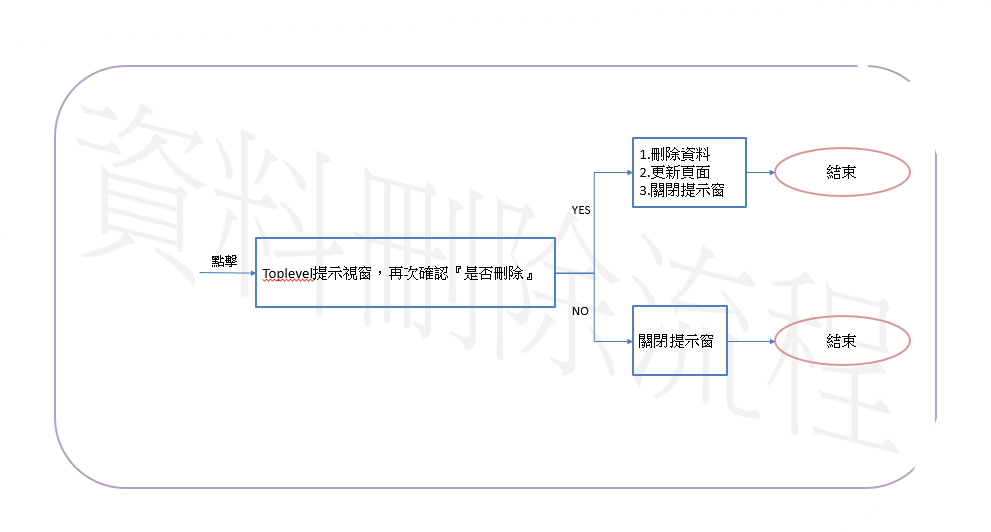
資料查詢



資料修改



資料刪除



**實作心得：**

孫惟琪: 在這次的學習過程中，我對 Tkinter 介面的理解更加深入。透過建立簡單的視覺介面，我能更清楚地理解 Python 在 GUI 程式設計中的運用。從設定視窗、布局，到處理事件，每個步驟的細節都需要仔細規劃與實現。此外，這次也讓我深刻體會到作為程式設計師，除了技術上的學習，也必須考量與團隊其他成員的合作。實際上，開發過程中我們不僅面對著代碼本身，還需要協調各方需求與意見。這讓我明白，在開發大型專案或是工作環境中，跨部門的合作、有效的溝通和共同達成目標是不可或缺的。因此，我學會了更注重如何與他人協同工作，這是我在學習過程中最深刻的體會。

王思焰:

使用技術與工具

1. 圖形界面設計 (GUI)：
   * 使用 tkinter 套件構建 GUI，並輔助以 ttk 提供更現代化的視覺樣式。
   * 包括 Canvas 與滾動條的組合用來呈現動態內容。
2. 資料處理：
   * 整合資料庫操作 (透過 WordDatas 類別)。
   * 包括單字的新增、查詢、修改與刪除。
3. 功能性頁籤 (Tabs)：
   * Notebook 用於分頁設計，實現查詢頁面與已學習單字頁面的功能切換。
4. 事件處理：
   * 使用 bind 綁定滾輪滾動事件及清單點擊事件。
   * 透過按鈕回調實現資料查詢與互動操作。
5. 視窗管理：
   * 使用 Toplevel 創建模態窗口，處理更詳細的資料檢視與編輯功

張云昇:

**附註：**  
工作分配

|  |  |  |  |
| --- | --- | --- | --- |
| 工作內容 | 孫惟琪 | 王思焰 | 張云昇 |
| 程式設計資料庫 | 0% | 100% | 0% |
| 介面 | 20% | 80% | 0% |
| ppt | 50% | 50% | 0% |
| 文件 | 100% | 0% | 0% |

參考文獻

------------------github操作文章----------------

https://code.yidas.com/git-commands/

-----------------TTK套件操作參考文章----------------

https://hackmd.io/@peterju/SJoPn4\_HT

https://steam.oxxostudio.tw/category/python/tkinter

https://chatgpt.com/

------------------爬蟲參訪網站--------------------------

https://dictionary.cambridge.org/zht/

--------GUI套件比較參考文章(可考慮是否有要放的必要)---------

https://www.pythonguis.com/faq/which-python-gui-library/