一張含有 文字, 圖形, 字型, 平面設計 的圖片

自動產生的描述

1 1 3 年 Scripting程式設計

*Crypto****資訊系統***

【書面報告】

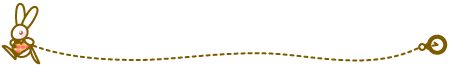
**第一組-組員**

9B017059 余碩峰

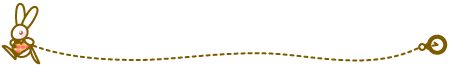
9B017046 沈泰平

9B017039 臧子諒

中華民國113年12月30日

目錄

1. 作業說明......................... 3~6
2. 背景資料....................... 7~11
3. 環境介紹..................... 12~16
4. 實作方法.....................17~33
5. 實作心得.......................34~37
6. 工作分配............................38

作業說明

1. **靈感發想**

　　區塊鏈和加密貨幣市場發展迅速，但相關的數據和資訊往往分散在不同平台，像是幣種價格、市值和相關新聞等資訊需要多次切換平台才能找到，使用起來非常不方便。因此，我希望透過一個爬蟲平台，整合這些資料，讓使用者可以快速查詢幣種的價格變化、最新消息等重要資訊，減少手動查找的麻煩。

1. **目標**

　透過此期末作業，我有設定4個目標，不僅學習爬蟲資料的串接，並更深入學習Python/SQL/HTML/CSS 等語言使用方法。

* 1. **學習網頁爬蟲的基本操作**
* 瞭解如何使用 Selenium 爬取網頁內容，熟悉網頁元素的定位與操作方法。
* 學習如何處理動態網頁數據並實現資料提取。
  1. **強化資料處理與結構化輸出能力**
* 獲取的資料結構化為可讀性高的字典或表格形式，並學習如何輸出為 JSON 或存入資料庫。
  1. **提升程式設計的模組化能力**
* 學習將功能拆分為模組化的函數或類別，提升程式的可讀性與可維護性。
  1. **實踐問題解決與系統優化**
* 解決爬蟲過程中可能遇到的問題（如反爬措施、資料延遲加載等）。
* 使用等待機制和無頭模式提升爬蟲的穩定性與效率。

1. **作業內容**

　　 本作業以 **加密貨幣市場** 作為案例，通過以下功能模組的設計與實現來完成學習目標：

1. **幣價資訊爬取學習**

* 目標網站：「[TradingView價格表](https://tw.tradingview.com/markets/cryptocurrencies/prices-all/)」。
* 需完成實現爬取加密貨幣的基本資訊，如價格、市值、成交量等，並結構化為字典數據。
* **本次收穫**：
  1. 動態表格的處理與數據篩選。
  2. 使用 Python 的字串操作來提取目標幣種資訊。

1. **加密貨幣新聞爬取學習**

* 目標網站：「[Binance 新聞](https://www.binance.com/zh-TC/square/news/all)」。
* 需完成爬取指定加密貨幣的新聞標題、發佈時間與連結，並整理為可讀性高的清單。
* **本次收穫**：
  1. 學習如何動態選擇目標分類。
  2. 處理新聞內容多層結構的定位與提取問題。

1. **無頭模式與資源優化**

* 通過設計無頭模式（headless mode），降低資源消耗並實現後台執行。
* **本次收穫**：
  + 1. 學習如何配置 Selenium 的選項來實現資源優化。

1. **預期效益**
2. **技術能力的提升**

* 熟悉網頁爬蟲框架 Selenium 的操作，能夠應對多種網站結構與場景。
* 掌握模組化程式設計方法，提升軟體設計能力。

1. **實戰經驗的積累**

* 通過解決案例中的實際問題（如反爬措施、動態加載等），為未來處理類似問題打下基礎。
* 模擬實際數據收集場景，提升解決問題的綜合能力。

1. **跨領域知識的應用**

* 學生可藉由此案例拓展對加密貨幣市場的認知，並理解數據分析在金融科技領域的應用。

1. **可擴展應用場景**

* 該作業的設計方法可輕鬆延伸至其他應用場景，如股票市場爬蟲、商品價格追蹤等。

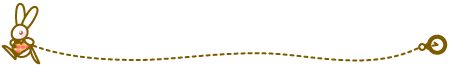
一張含有 黃金 的圖片

描述是以低可信度自動產生背景資料

1. **Python的使用背景**

Python 是一種高階、解釋型且廣泛應用的程式語言，以其簡潔的語法和強大的標準函式庫而聞名。它被廣泛應用於數據分析、人工智慧、網站開發、網頁爬蟲等領域。本次作業選擇 Python 作為主要開發語言，主要原因包括：

1. **語法簡單、易於學習**
   * Python 的語法設計簡單直觀，非常適合用於教學和快速開發。
2. **豐富的第三方支援**
   * 例如 : 這次作業中使用的爬蟲框架 **Selenium**，是一個專業的網頁操作工具，可以方便地處理各類靜態與動態網頁。
3. **跨平台支援**
   * Python 能在 Windows、macOS 和 Linux 等多個平台上運行，方便開發者部署程式。
4. **本次作業問題的介紹**
5. **主題背景**
   * 加密貨幣（Cryptocurrency）作為一種數位資產，在近年來引起了廣泛的關注。其價格波動劇烈，並伴隨著大量的市場資訊與新聞。投資者需要有效的工具來追蹤價格、分析市場並快速獲取相關資訊。
   * 本作業設計了一個基於 Python 的工具，實現了 **幣價爬取** 與 **新聞爬取** 的功能，旨在幫助用戶快速獲取相關資料。
6. **問題陳述**
   * **問題 1：如何爬取動態加載的網頁內容？**  
      現代網站通常採用 JavaScript 動態渲染技術，使得靜態爬蟲工具難以抓取所需內容。本作業引入了 Selenium 作為動態爬蟲工具，模擬使用者操作以獲取網頁內容。
   * **問題 2：如何提升爬蟲的穩定性與效率？**  
      爬取過程中可能面臨反爬機制、網頁加載延遲等問題。本作業透過設置等待機制與無頭模式來解決這些挑戰，並實現高效、穩定的爬取過程。
   * **問題 3：如何組織與展示爬取的資料？**  
      從不同網站提取的數據需要進行處理與結構化，才能提高可讀性與後續分析的方便性。本作業將爬取的數據結構化為 Python 字典，並輸出為易於查看的格式（例如 : 表格或 JSON）。
7. **相關技術知識**
8. **Selenium 簡介**
   * Selenium 是一款自動化測試工具，常用於網頁測試與爬蟲。它能夠模擬瀏覽器操作，如點擊、滾動、輸入等，並支援 JavaScript 動態渲染的網站。
   * **功能特點**：
     + 支援多瀏覽器操作（如 Chrome、Firefox）。
     + 能夠處理複雜的網頁結構與 AJAX 加載內容。
9. **無頭模式（Headless Mode）**
   * 無頭模式是指在無需顯示圖形介面的情況下執行瀏覽器操作，適合後台運行爬蟲，能有效降低系統資源消耗。
10. **網頁結構與定位技術**
    * **HTML 結構**：網頁是由 HTML 標籤組成的結構化文件。爬蟲需要通過特定的標籤或屬性來定位目標內容。
    * **CSS Selector** 和 **XPath** 是常用的元素定位技術。
11. **反爬機制與解決方案**
    * **常見反爬措施**：
      + 檢測快速多次請求。
      + JavaScript 驗證或 CAPTCHA。
    * **解決方案**：
      + 設置請求延遲（time.sleep）。
      + 模擬人類操作（如滾動與滑動）。
12. **Python 資料結構與模組化設計**
    * **字典與清單**：便於結構化存儲資料，如爬取的幣種名稱、價格、市值等。
    * **模組化編程**：將功能拆分為獨立函數或類別，提升程式的可讀性與維護性。
13. **加密貨幣市場概述**
    * **市場特點**：加密貨幣價格波動性大，且受政策、技術、新聞等多方面因素影響。
    * **應用場景**：爬取數據後可用於價格趨勢分析、交易策略制定、投資決策支持等。

環境介紹

1. **作業系統**
2. **主要系統**
   * **Windows 10 或以上**：

相容性高且支援廣泛的開發工具。

* + **其他可用系統**
    - **macOS：**

提供穩定性與優化效能的替代選項。

* + - **Linux：**

適合熟悉 Unix 系統的開發者，可選擇 Ubuntu 22.04 LTS 等版本。

1. **開發與執行工具**
2. **程式語言**
   * **Python 3.7 或以上版本**：
     + 確保支援最新的函式庫與框架功能，並提供強大的第三方庫支援。
3. **開發工具**
   * **Visual Studio Code**
     + 輕量化且強大的編輯器，支援 Python 開發並提供豐富的擴張功能。
     + **擴充功能**：Python、Pylance、Selenium Helper 等。



1. **虛擬環境管理工具**
   * **virtualenv**
     + 用於創建獨立的虛擬環境，確保第三方庫之間無相依性衝突。
2. **第三方套件**
   * **Selenium**
     + 用於網頁操作及爬取動態內容。
   * **Ttkbootstrap**
     + 採用 Metro UI 風格的主題。
   * **tkinter**
     + 內建於 Python 中，用於建立簡單的圖形使用者介面。
   * **black**
     + Python 格式化工具，確保程式碼符合 PEP 8 規範。
3. **網頁瀏覽器**
   * **Google Chrome**
     + 作為爬蟲目標的主要瀏覽器，需安裝與之匹配的 ChromeDriver。
   * **ChromeDriver**
     + Selenium 與瀏覽器的橋接工具，需根據 Chrome 版本進行對應下載。
4. **目標網站與數據來源**
5. **TradingView 加密貨幣市場頁面**
   * URL: <https://tw.tradingview.com/markets/cryptocurrencies/prices-all/>
   * 用於獲取幣種價格、市值、交易量等市場數據。
6. **Binance 新聞區**
   * URL: <https://www.binance.com/zh-TC/square/news/all>
   * 用於獲取加密貨幣相關的新聞資訊及分析。
7. **其它支援工具**
8. **版本管控工具**
   * **Git**
     + 用於管理程式版本，方便記錄開發過程與協同作業。
9. **命令列工具**
   * **Windows CMD 或 PowerShell**
     + 用於安裝套件及啟動程式。
10. **備份與協作工具**
    * **Google Drive 或 GitHub**
      + 用於程式碼備份與跨設備同步共享。

一張含有 黃金 的圖片

描述是以低可信度自動產生實作方法

1. **環境檢查**
2. **建立虛擬環境**

* 使用虛擬環境來隔離專案環境

一張含有 文字, 收據, 字型, 白色 的圖片

自動產生的描述

1. **啟動虛擬環境**

* Windows:

一張含有 字型, 文字, 白色, 設計 的圖片

自動產生的描述

成功啟動後，終端提示應顯示 “**(.venv)**”

**3. 安裝必要的套件**

* 1. **tkinter  
     提供應用的圖形化介面。**
  2. **ttkbootstrap  
     使網頁介面質感昇華。**
  3. **Selenium  
     用於爬取目標網站數據（如幣價與新聞）。**
  4. **black  
     程式碼格式化工具，用於保持程式符合 PEP 8 規範。**
* 將以下套件安裝至虛擬環境中：

一張含有 文字, 字型, 收據, 白色 的圖片

自動產生的描述  


**4. 檢查專案結構**

* 確保專案的基本檔案結構如下：

一張含有 文字, 字型, 螢幕擷取畫面, 白色 的圖片

自動產生的描述

**5. 執行專案**

* 切換到專案目錄後，執行以下指令：

一張含有 文字, 字型, 白色, 設計 的圖片

自動產生的描述

1. **解決方案**
   1. **系統架構功能介紹**

此系統主要分成兩部分，分別是幣種新聞和幣種資訊，新聞會以最新時間進行排序，並可透過連結點開，幣種資訊在爬蟲後可進行 CRUD，用以紀錄和備註資料。

一張含有 文字, 螢幕擷取畫面, 字型, Rectangle 的圖片

自動產生的描述

* 1. **資料庫建置**

使用了Python的sqlite3模組來建立和操作SQLite資料庫，資料庫建置及資料表屬性如下：

1. **幣種資訊表 (crypto\_data)：**
   * + - * **id (主鍵)**：自動遞增的唯一識別碼。
         * **crypto**：幣種名稱，例如 BTC、ETH。
         * **price**：幣種價格，以浮點數儲存。
         * **change**：幣種漲跌百分比，以字串形式儲存。
         * **volume**：24小時成交量，以字串形式儲存。
         * **market\_cap**：市值，以字串形式儲存。
         * **timestamp**：記錄資料建立時間。
         * **note**：備註欄位，用於記錄額外資訊，可留空。
2. **insert\_data(data)**

* 接受字典形式的輸入數據，並將其插入到資料表中。
* 支持可選的 note 欄位，默認值為空字符串。

1. **get\_data(crypto)**

* 根據幣種名稱檢索其歷史數據，並按 timestamp 排序。
* 返回所有相關記錄的列表。

1. **update\_data(crypto, timestamp, note)**

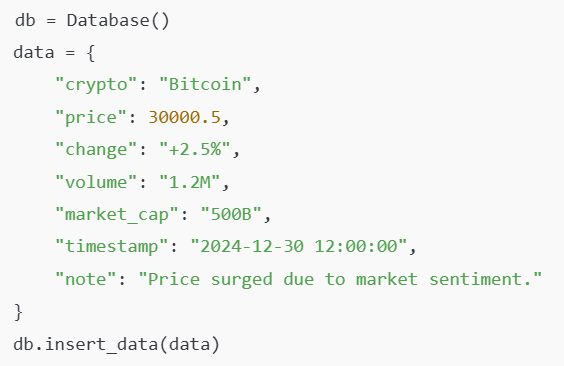
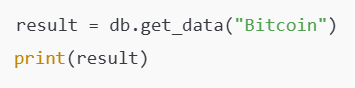
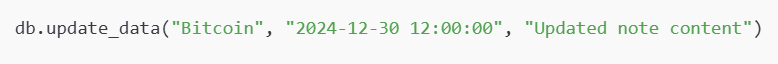
* 更新特定幣種和時間戳記錄中的 note 欄位。

1. **delete\_data(crypto, timestamp)**

* 刪除指定幣種和時間戳的數據。

1. **close()**

* 關閉與資料庫的連接，釋放資源。
* **程式運作方式**

1. **新增資料**  
   
2. **搜索資料  
   **
3. **更新資料  
   **
4. **刪除資料  
   **
5. **關閉資料庫  
   **
   1. **CryptoApp GUI 系統開發** 我們透過 Tkinter 框架的圖形化應用程式 (GUI)，用於顯示加密貨幣的即時市場數據及新聞資訊，並提供與 SQLite 資料庫的互動功能。功能包括查詢、新增、編輯和刪除加密貨幣相關數據，以及顯示最新新聞和快速連結的功能。

* **程式功能分析**

1. **主要功能模組**
2. **CryptoApp 類別**

包含整個 GUI 的設計與功能實現，包括元件初始化、用戶交互處理邏輯，以及與數據庫的通信。

1. **資料庫操作**

使用 Database 類，實現幣種資訊的新增、檢索、更新和刪除功能，確保數據可以長期保存。

1. **即時數據獲取**

通過 get\_crypto\_data() 與 get\_news() 獲取加密貨幣的市場數據與新聞資訊。

1. **主要功能描述**
2. **幣種數據查詢與顯示**

提供一個下拉選單允許用戶選擇幣種並檢索即時數據。

1. **使用 Treeview 顯示結果表格。**
2. **數據庫操作**
3. 支持用戶將即時查詢結果新增至 SQLite 資料庫。
4. 支持資料庫內數據的搜索、備註更新和刪除。
5. **最新新聞展示**
6. 按頁面顯示新聞資訊，提供上下頁翻頁功能。
7. 點擊新聞中的連結可直接在瀏覽器中打開。
8. **UI 元件**
9. **主介面**

* 幣種下拉選單、搜尋按鈕、結果顯示表格、新聞表格。
* 分頁按鈕控制新聞展示。

1. **資料庫搜尋視窗**
2. 包括搜尋功能、結果表格，以及編輯和刪除數據的按鈕。
3. **第三方模組**
4. tkinter：用於構建圖形化介面。
5. ttkbootstrap : 用於提升頁面質感。
6. sqlite3：與 SQLite 資料庫交流。
7. webbrowser：用於打開新聞連結。

* **程式運作邏輯**

1. **初始化**
2. 創建主窗口，設置字體、佈局以及初始化數據庫連接。
3. **幣種數據查詢**
4. 用戶選擇幣種後，通過 fetch\_data() 調用 get\_crypto\_data() 並更新 Treeview 表格。
5. **數據庫交流**
6. 點擊 "新增" 按鈕後，將當前查詢結果連同備註存儲至資料庫。
7. 點擊 "資料庫搜尋" 打開新窗口進行資料庫查詢。
8. 用戶可編輯或刪除特定記錄。
9. **新聞展示**
10. 查詢數據後自動調用 get\_news()，並按頁顯示結果。

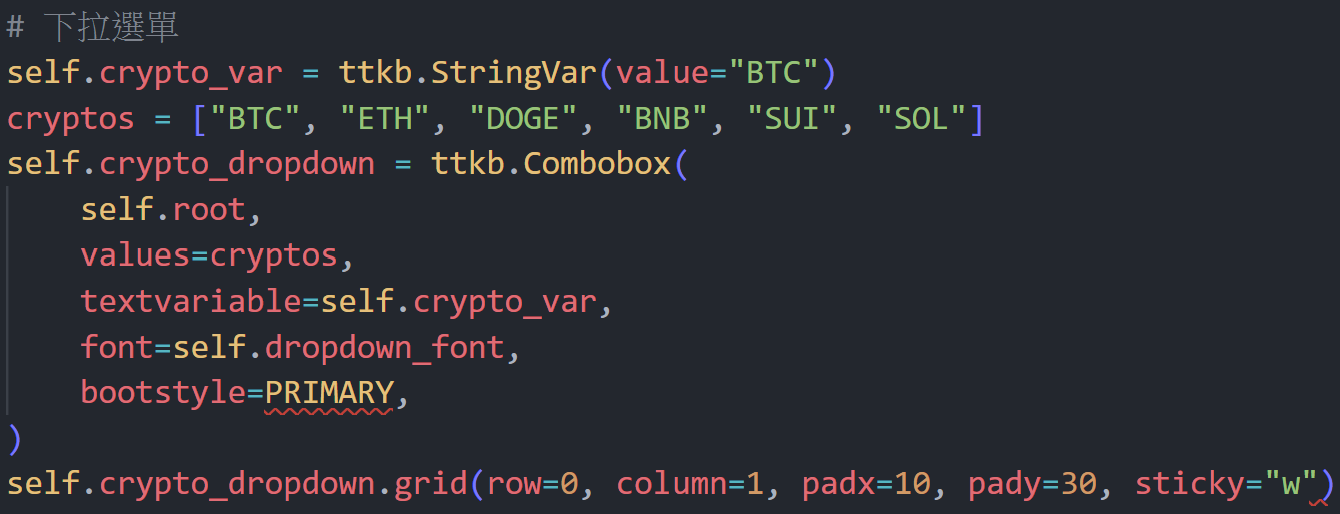
1. **Selenium程式功能介紹**
2. **get\_driver()**
3. 初始化一個無頭模式的 Selenium WebDriver，以便在沒有圖形介面的環境中執行瀏覽器操作。
4. 配置選項包括無頭模式、禁用 GPU、以及避免共享內存問題。
5. **get\_crypto\_data(crypto)**
6. 爬取 [TradingView](https://tw.tradingview.com/markets/cryptocurrencies/prices-all/)  的幣種資料。
7. 搜尋目標加密貨幣，提取其價格、漲跌幅、成交量、市值等資訊。
8. 回傳資料以字典形式表示。

1. **get\_news(crypto)**
2. 爬取 [Binance 新聞頁面](https://www.binance.com/zh-TC/square/news/all) 的新聞資料。
3. 透過分類按鈕定位目標幣種新聞，提取新聞標題、連結和發佈時間。
4. 以清單形式回傳新聞資料，每則新聞為一個字典。
   1. **使用者介面介紹**
5. **主頁面：**

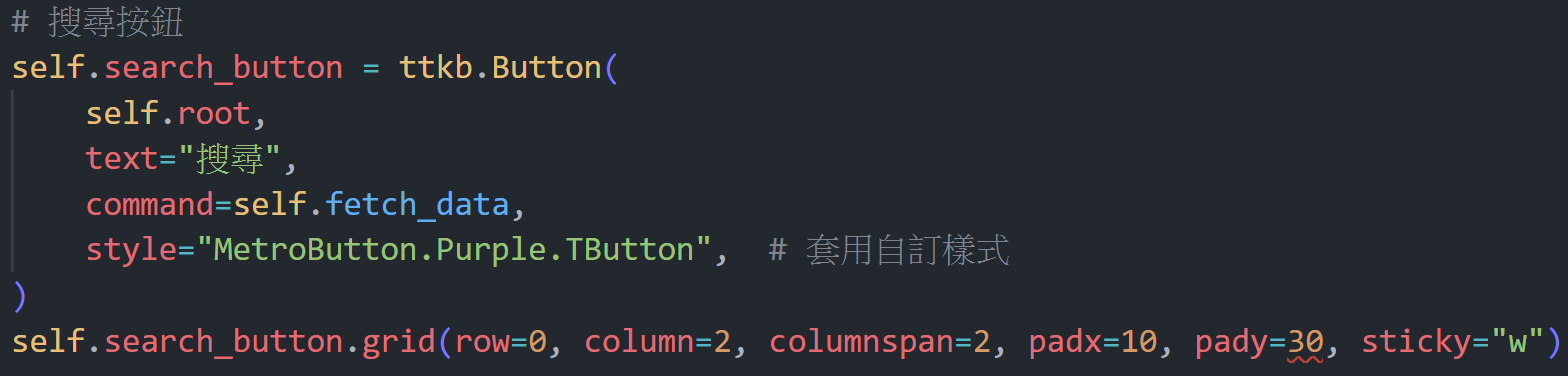


* **主頁面爬蟲 :** 在頁面中，可以透過左上的下拉式選單選取幣種，並透過右側的搜尋爬蟲資料，爬蟲結束並可看到幣種資訊和新聞，且點擊新聞連結，可直接打開瀏覽器觀看完整新聞。

**選單程式**



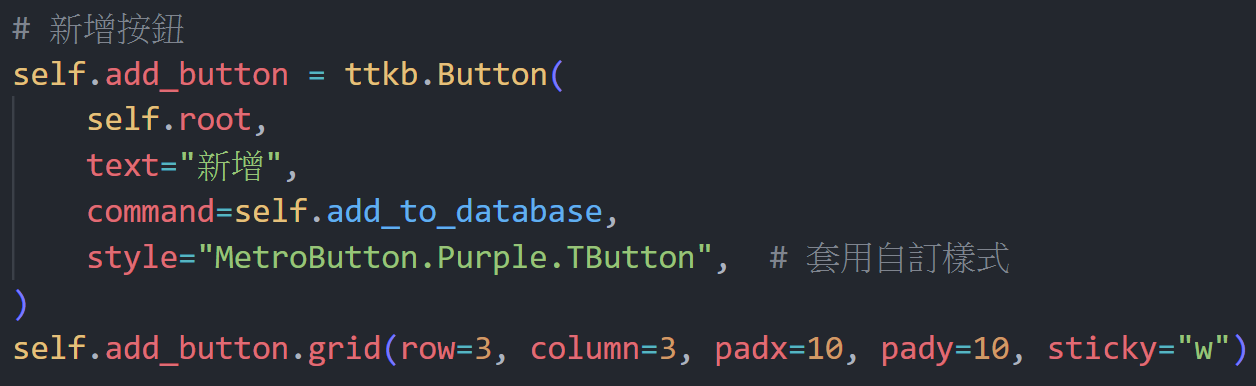
**搜尋按鈕樣式**



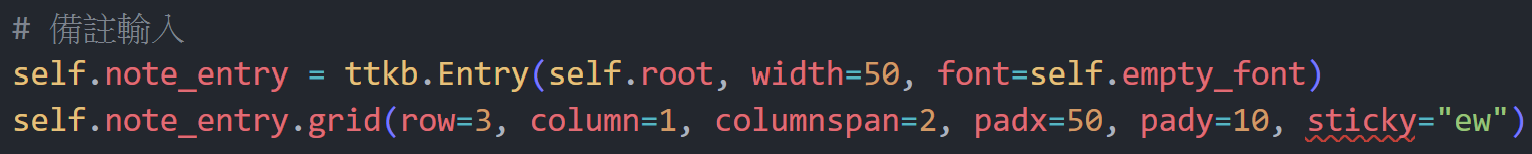


* **主頁面新增 :** 再搜尋到幣種資訊後，可以選取資料，並點擊新增，便會將幣種資訊新增至資料庫，也可在備註欄位加上想要的資訊，在此輸入幣價漲幅備註。

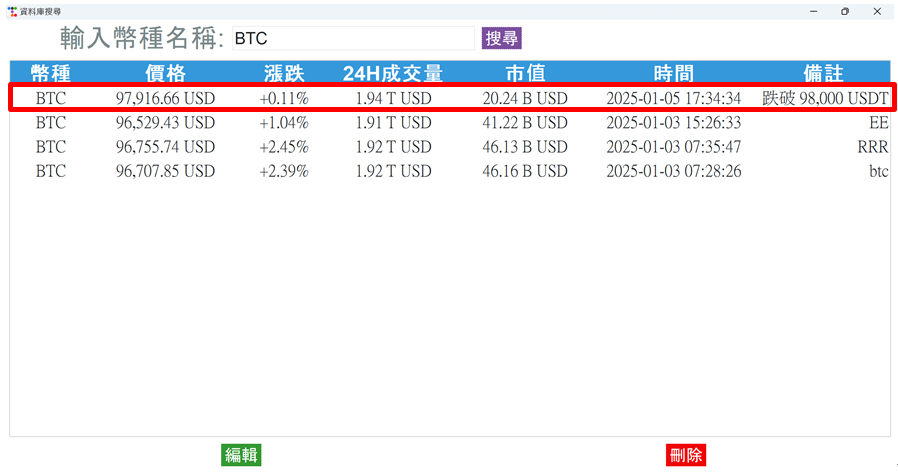
**新增按鈕樣式**



**備註輸入框樣式**







* **主頁面搜尋 :** 可透過右上紅框的筆記本按鈕，搜尋已儲存的幣種資料，並點擊搜尋，便會彈出該資料的頁面。

**筆記本按鈕樣式**



**資料庫搜尋樣式**







* **筆記本編輯和刪除 :** 在搜尋到幣種資訊後，可以選取資料，並點擊編輯，即可修改備註，也可隨意刪除已儲存的資料。

一張含有 黃金 的圖片

描述是以低可信度自動產生實作心得

這次的爬蟲網站的建立是個充滿挑戰性的任務，透過組長碩峰的教學以及帶領下，從無到有完成了整個專案。在開始這個作業之前，我們先花了一些時間研究和計劃。我們想要確定網站的目標和功能，並設計一個符合使用者需求的介面。這一步驟是非常重要的，因為一個好的計劃可以幫助我們在後續的開發過程中更加順利地進行。  
 在設計階段，我們遇到了一些挑戰。其中之一是如何選擇合適的顏色和框架，以創建一個令人愉悅且易於閱讀的網站。我們花了一些時間研究色彩心理學的原則，以確保我們的設計能夠傳達出所需的情感和信息。另外，還需要考慮到不同的設備，以確保網站在各種環境下都能正常運行和呈現。

主要困難是正確爬取目標網站的資料，以及整理和儲存爬取後的資訊。這需要我投入更多的時間和精力來學習和理解這些新技術，目標網站結構複雜，我們需要分析HTML結構，調整爬取邏輯來獲取正確的數據。同時，爬取後的資料格式常需要清理和重新整理，才能顯示於 tkinter 和存入 SQLite 資料庫。

除了技術上的挑戰，還需要跟組員進行有效的溝通和協作。在分組作業中，我們需要合作來完成整個網站的開發過程。這要求我們在討論和解決問題時保持開放的態度，並學會彼此聆聽和尊重。通過有效的溝通和協作，我們能夠充分發揮每個成員的專業能力，並最終建立一個出色的網站。

分組作業讓我們有機會從他人那裡學習和獲得不同的觀點。每個分組成員都帶來了自己獨特的技能和想法，這豐富了我們對爬蟲網站建立的理解和知識。在這個過程中，我們學到了何謂團隊合作、如何有效地溝通和管理時間。這些是在未來工作中非常寶貴的技能。

透過這次實作，我們學會了如何解決爬取資訊失敗的問題，並加深了對資料庫設計和操作的理解。並且善用ttkbootstrap將介面樣式調整為更加友善的質感風格，這次專案雖然困難，不過加強了我們的團隊合作能力，並對未來的工作有所裨益。但也讓我們對程式設計和問題解決更有信心，希望未來可以將這些技能應用到更多專案中。

一張含有 文字, 螢幕擷取畫面, 軟體, 網頁 的圖片

自動產生的描述**頁面修正前**

**頁面修正後**



一張含有 黃金 的圖片

描述是以低可信度自動產生工作分配

* **程式撰寫：**余碩峰70%、沈泰平20%、臧子諒10%
* **文件編輯：**余碩峰70%、沈泰平15%、臧子諒15%
* **簡報製作：**余碩峰70%、沈泰平15%、臧子諒15%

**感謝賞閱**