

Replit (repl):

<https://replit.com/@tryso26/Trysofieldlcclasspythoncrawlerex2?v=1>

Replit (Dev URL):

<https://805d8201-467c-4d06-8e12-a476ee53cf7d-00-neduz4dreqoa.sisko.replit.dev/>

(Dev URLs are temporary and will sleep after you leave the workspace.)

Github (repository):

https://github.com/tryso26/Tryso_field/tree/lcclass_pythoncrawler_ex2

Github (HTML):

https://github.com/tryso26/Tryso_field/blob/lcclass_pythoncrawler_ex2/ex2.html

GitHub & BitBucket HTML Preview (將 Github (HTML)複製貼上可預覽):

<https://htmlpreview.github.io/?>

備註:

不好意思拖了點時間，10/01 花了一些時間從頭學習 ssh key pair 建立和透過 github repository 儲存與分享，html 檔案透過 GitHub & BitBucket HTML Preview 工具 (<https://github.com/htmlpreview/htmlpreview.github.com>) 確認可以正確顯示網頁素材 (只排版未用其他顯示器確認過，不排除轉換裝置可能有跑版問題)。

直接將 html preview 轉換出來的網址放到瀏覽器呼叫，會被 github 擋，所以可能要勞煩審閱者將 Github (HTML)複製到預覽工具網頁執行。

產生 Replit deploy 可正確預覽的連結方法這邊還要花時間摸索一下 (Github 程式碼有做微小的修正更新，Replit 還是舊版)。但因確認 Github HTML preview 可以正確預覽，這邊就將先上傳的資料夾撤下，將雲端空間空出來。

2024.10.03 更新

因為 replit 提供與 github 的協作工具，於是將原先的 repl 刪除，希望改以 import Github 的 repl 尋求資料同步。

1. 然而花了不少時間處理載入問題，有大量問題需要利用 Shell 工具使用 git bash 語法修改環境 (這邊只能完全請求 AI 工具協助)，較大的問題略舉如：
載入 repository 時止 import main 分支，但當中的其他 branches 是完全消失的，只能藉由 `git checkout <branch_name>` 手動調閱分支名稱 (若使用 `git clone -b <branch_name> https://github.com/<username>/<repository>.git`，只會把分支內容的檔案載入，但無法將整個分支帶進來，分支清單仍然只會有 main，當然也違論 push 回去)；
2. 若新增 repl 時會自動建好網頁開發環境，但 import github 則需要手動建立開發環境和轉發端口，這邊嘗試新建一個全新的 html, css, js repl 後將 .replit 檔案內容複製，抄到目標 repl 後更改 `entrypoint="index.html"` 的目標路徑到正確檔名，安裝 Replit's Python

tools 並在 Shell 執行 `python -m http.server` 成功使 Networking 自動建立 internal port , 成功在 Webview 預覽檔案結構與 html 畫面。

然後成功建立環境後仍然有不能解決的問題:

- 1) 現在進入 repl 要使 Webview 連結到內部端口，需要每次都在 Shell 執行 `python -m http.server ;`
- 2) 如同同學遇到的問題，現在 repl 要在作者離開工作空間時能夠外部執行，須要使用 replit 的付費計畫來部署(deploy) repl，才能得到永久 Dev URL。

所以綜合上述現階段要利用 replit 與 github 協作至少有兩個大門檻，一者熟悉 git bash 而且 有耐心處理 replit 預設資料載入時的優化問題，第二是須要透過付費計畫部署 repl 才能使外部訪客得以直接透過 Networking 轉接內外端口瀏覽 repl 的執行內容。

因此現階段免費或初階使用者資料儲存在 github，透過 VScode 與 github 協作，內部開發時使用 Live Server 套件預覽結果，分享網址時藉由 GitHub & BitBucket HTML Preview 進行結果預覽，還是親民而且簡易非常多。