

# 教育大數據期末個人專案

**Title:**利用 Python 爬蟲工具(Webdriver)抓取 pchome 資料並分析數據。

**ID:** 411077010

**Name:**蔡岳哲

## Abstraction:

本專案旨在利用 Python 爬蟲技術（使用 Webdriver）來抓取 pchome 網站上特定關鍵字的搜尋結果資料並分析數據。通過這個過程，我們將探討如何撰寫一個程式，並利用工具(Webdriver)進行數據的搜集、整理、分析。本專案將包含文章、程式、資料表等多個面向，並最終完成一個具有實際應用價值的數據分析程式之雛形。

## I: Preface:

在教育領域，大數據的應用已經成為一個引人注目的議題。本專案旨在透過爬蟲技術，將網路上的數據資料收集起來，並通過數據分析提供有價值的見解。

## II: Related documents:

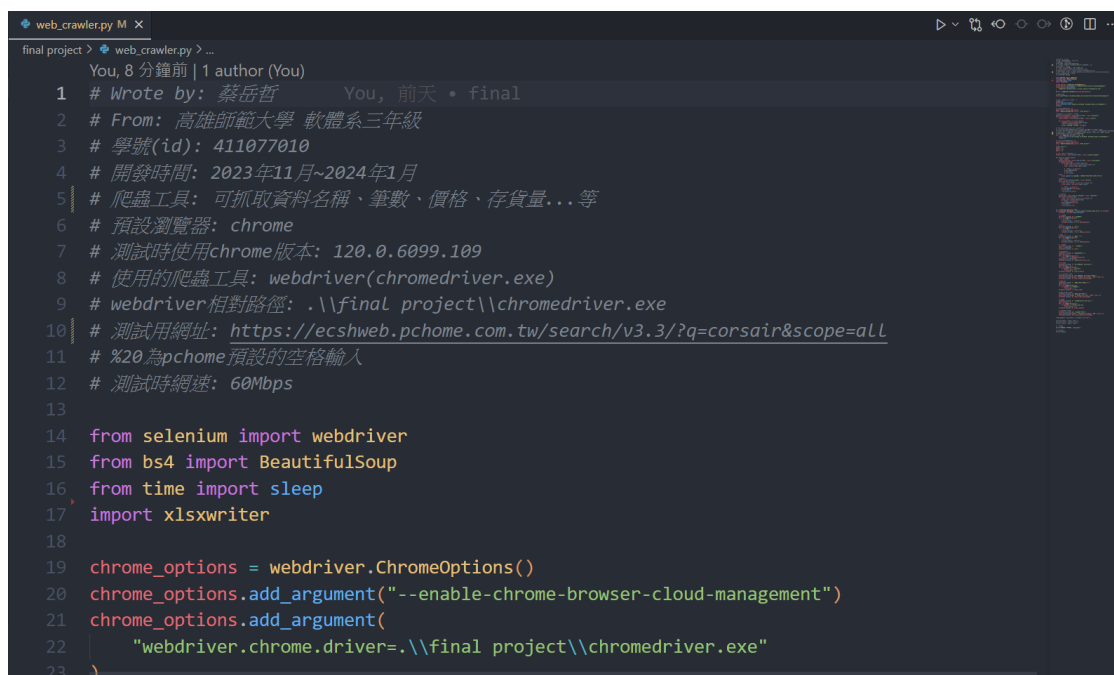
本章節將探討與本專案相關的文件、研究、和先前相關專案的相關性，以建立本專案的理論基礎。

1. 使用的網頁爬蟲工具(webdriver): [chromedriver](#)
2. Selenium 教學: [Selenium 教學](#)
3. 預覽、測試用 pchome 網址: [pchome 搜尋](#)
4. beautiful-soup 教學: [beautiful-soup 教學](#)
5. XlsxWriter 教學: [XlsxWriter 教學](#)

## III: Step

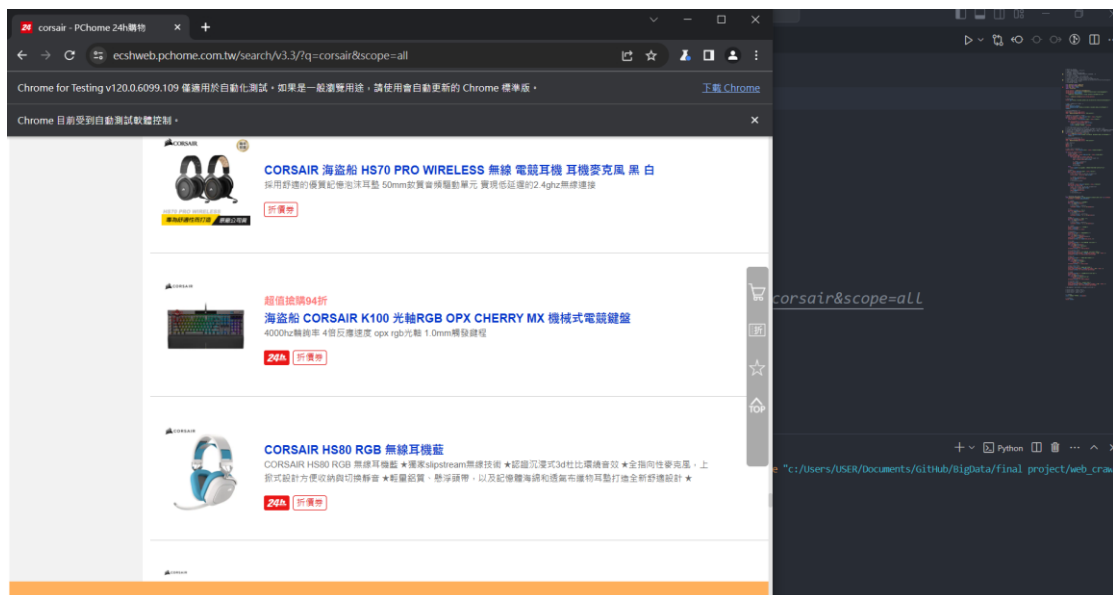
### 1.Article:

(1) run code:



```
web_crawler.py M X
final project > web_crawler.py > ...
You, 8 分鐘前 | 1 author (You)
1 # Wrote by: 蔡岳哲 You, 前天 • final
2 # From: 高雄師範大學 軟體系三年級
3 # 學號(id): 411077010
4 # 開發時間: 2023年11月~2024年1月
5 # 爬蟲工具: 可抓取資料名稱、筆數、價格、存貨量...等
6 # 預設瀏覽器: chrome
7 # 測試時使用chrome版本: 120.0.6099.109
8 # 使用的爬蟲工具: webdriver(chromedriver.exe)
9 # webdriver相對路徑: .\\final project\\chromedriver.exe
10 # 測試用網址: https://ecshweb.pchome.com.tw/search/v3.3/?q=corsair&scope=all
11 # %20為pchome預設的空格輸入
12 # 測試時網速: 60Mbps
13
14 from selenium import webdriver
15 from bs4 import BeautifulSoup
16 from time import sleep
17 import xlsxwriter
18
19 chrome_options = webdriver.ChromeOptions()
20 chrome_options.add_argument("--enable-chrome-browser-cloud-management")
21 chrome_options.add_argument(
22     "webdriver.chrome.driver=\\.\\final project\\chromedriver.exe"
23 )
```

## (2) crawling web:



## (3) data analysis by code(demo part of code):

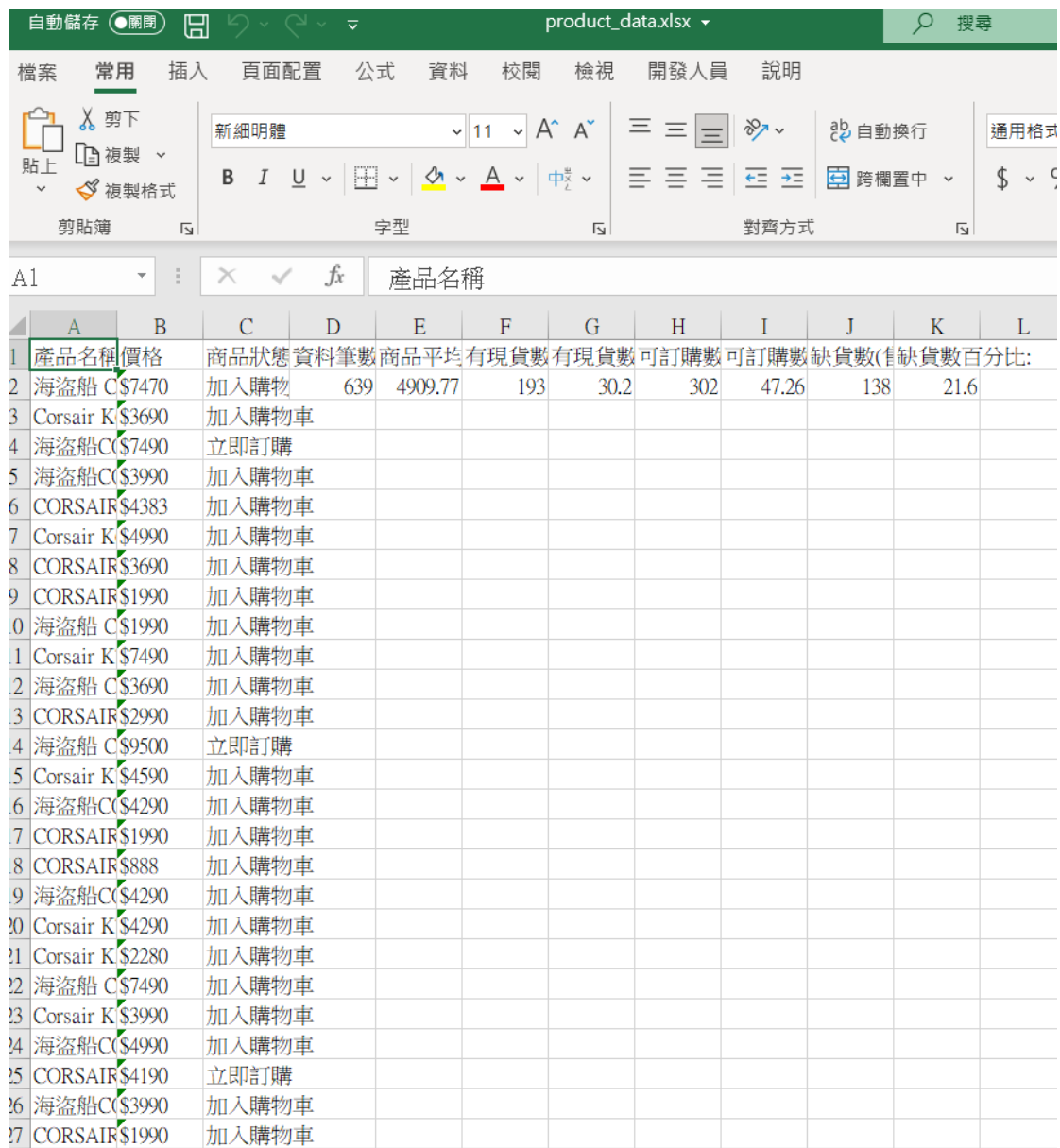
```
for i in range(len(pprice)):
    total_price += int(pprice[i][1:])
total_price /= len(pprice)
worksheet.write(1, 4, round(total_price, 2))

# 有現貨數
worksheet.write(0, 5, "有現貨數(可加入購物車):")
total_product = 1
for i in range(len(phave)):
    if phave[i] == "加入購物車":
        total_product += 1
worksheet.write(1, 5, total_product)

# 有現貨百分比
worksheet.write(0, 6, "有現貨數(可加入購物車)百分比:")
total_product_percentage = round(total_product * 100 / total, 2)
worksheet.write(1, 6, total_product_percentage)

# 可訂購數
worksheet.write(0, 7, "可訂購數(立即訂購):")
total_order = 1
for i in range(len(phave)):
    if phave[i] == "立即訂購":
        total_order += 1
```

#### (4) convert into xlsx:



	A	B	C	D	E	F	G	H	I	J	K	L
1	產品名稱	價格	商品狀態	資料筆數	商品平均	有現貨數	有現貨數	可訂購數	可訂購數	缺貨數	缺貨數	百分比
2	海盜船 C	\$7470	加入購物車	639	4909.77	193	30.2	302	47.26	138	21.6	
3	Corsair K	\$3690	加入購物車									
4	海盜船C	\$7490	立即訂購									
5	海盜船C	\$3990	加入購物車									
6	CORSAIR	\$4383	加入購物車									
7	Corsair K	\$4990	加入購物車									
8	CORSAIR	\$3690	加入購物車									
9	CORSAIR	\$1990	加入購物車									
10	海盜船 C	\$1990	加入購物車									
11	Corsair K	\$7490	加入購物車									
12	海盜船 C	\$3690	加入購物車									
13	CORSAIR	\$2990	加入購物車									
14	海盜船 C	\$9500	立即訂購									
15	Corsair K	\$4590	加入購物車									
16	海盜船C	\$4290	加入購物車									
17	CORSAIR	\$1990	加入購物車									
18	CORSAIR	\$888	加入購物車									
19	海盜船C	\$4290	加入購物車									
20	Corsair K	\$4290	加入購物車									
21	Corsair K	\$2280	加入購物車									
22	海盜船 C	\$7490	加入購物車									
23	Corsair K	\$3990	加入購物車									
24	海盜船C	\$4990	加入購物車									
25	CORSAIR	\$4190	立即訂購									
26	海盜船C	\$3990	加入購物車									
27	CORSAIR	\$1990	加入購物車									

## 2.Program:

使用 Python 爬蟲技術，利用 Webdriver，從定義爬取的目標網站抓取所需文字與資料，透過程式編寫，實現自動化的數據收集與分析。

## 3.Hardware:

考慮到爬蟲可能需要長時間運行，以下將討論適用的配置，確保順利執行整個數據搜集過程。

# 預設瀏覽器: *chrome*

# 測試時使用 *chrome* 版本: *120.0.6099.109*

# 使用的爬蟲工具: *webdriver(chromedriver.exe)*

# *webdriver* 相對路徑: *.\final project\chromedriver.exe*

# 測試用網址:

<https://ecshweb.pchome.com.tw/search/v3.3/?q=corsair&scope=all>

# 測試時網速: *60Mbps*

#### IV: System establishment(Data analyse):

使用簡單的加法將資料總數得出來，利用除法將資料百分比分析出來。

以下為其中一個範例:

```
# 缺貨數(百分比)
worksheet.write(0, 10, "缺貨數百分比:")
total_soldout_percentage = round(total_soldout * 100 / total, 2)
worksheet.write(1, 10, total_soldout_percentage)
```

#### V: Conclusion:

最終，我們將總結整個專案，回顧所取得的成果、遇到

的挑戰，可能的擴展方向和改進方法，以提升系統的效能和應用價值。

### (1)取得的成果:

可以將前端文件除去 `css`、`js` 等 `html` 渲染，並獲取指定的 `name`、`class` 中的資料。將獲得的資料分析並自動寫入到 `excel` 中，以便使用者直接了解數據。

### (2)遇到的挑戰:

若網速不夠快，爬蟲的速度便會大大降低，所需時間會變多很多。若網站搜尋到的筆數太多，也會造成爬蟲時間過長，並且抓取到的資料到最後也可能與使用者所搜尋的無關。

### (3)可能的擴展方向和改進方法:

可以利用其他工具，或是再次進行資料分類，將過於不符的資料去除，以得到更有用的總資料。

利用其他資料結構、其他工具來爬取資料，或許可以提升效能。

**##報告:**[報告影片](#)