



1 Modern Programming Language.

2  
3 Midterm Project

4 Carrefour toilet paper price comparison

5  
6  
7 現代程式語言 • 期中報告

8  
9 家樂福衛生紙價格比較

10  
11  
12  
13 ## 第八組: 111210518 陳辰鈺、111210520 吳佳泰、111210524 藍健洲



```
1  import menu
```

```
2  
3  
4  #1. 爬取的目標
```

```
5  
6  #2. 爬取的前置作業
```

```
7  
8  #3. 爬取的步驟
```

```
9  
10 #4. 心得分享
```

```
11  
12 #5. 參考資料
```

```
13  
14
```



```
1  
2 count = 1  
3  
4  
5  
6
```

```
7 爬取的目標  
8  
9
```

```
10  
11  
12 ## Crawling Target  
13  
14
```



```
1 def goal_01():
```

```
2  
3  
4     一開始我們想搜尋全聯小時達的衛生紙，但我們發現此網站  
5     是採用React框架，並且使用了Next.js來進行伺服器端渲  
6     染。這種設計使得網頁載入速度更快，網頁會動態地更新內  
7     容而不改變URL，這給我們的爬蟲之路上造成了很多曲折  
8  
9  
10
```

```
11  
12     ## 於是，我們學會了適時轉彎  
13  
14
```



```
1 def goal_02():
```

```
2  
3  
4     後來我們決定搜尋家樂福線上購的衛生紙，決定要爬  
5     取這個網站的目標的原因是因為我們想要比較各家衛  
6     生紙在此網站上面的價格並將它繪製出來  
7  
8  
9
```

```
10  
11  
12     ## 找標籤好像在考古一樣XD  
13  
14
```



```
1  
2 count = 2  
3  
4  
5  
6
```

# 爬取的前置作業

```
7  
8  
9  
10  
11  
12 ## Preliminary Tasks for Crawling  
13  
14
```



```
1 def process():
```

- 配置VScode環境
- 偽裝成正常的瀏覽器客戶端以利爬取資料

```
10     headers = {"User-Agent" : "Mozilla/5.0 (Windows NT 10.0; Win64; x64)  
11     AppleWebKit/537.36(KHTML, like Gecko) Chrome/80.0.3897.132  
12     Safari/537.6"}  
14
```



```
1 def module_used():
2
3
4     request                ##讀取網頁檔案
5
6     BeautifulSoup          ##解析
7
8     sqlite3                ##資料庫創建與使用
9
10    pandas                  ##讀取資料庫之元素
11
12    matplotlib.pyplot      ##繪圖
13
14
```





```
1  
2 count = 3  
3  
4  
5  
6
```

# 爬取的步驟

```
7  
8  
9  
10  
11  
12 ## Crawling Steps  
13  
14
```



```
1 def step_01():
```

```
4 import requests
5 from bs4 import BeautifulSoup
6 import sqlite3
7
8 list1 = []
9 list2 = []
10
11 url = "https://online.carrefour.com.tw/zh/search/?q=%E6%8A%BD%E5%8F%96%20%E8%A1%9B%E7%94%9F%E7%B4%99"
12 headers = {"User-Agent" : "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36(KHTML, like Gecko) Chrome/80.0.3897.132 Safari/537.6"}
13
14 html = requests.get(url , headers=headers)
15 html.encoding = "UTF-8"
16
17 sp = BeautifulSoup(html.text , "lxml")
18
19 data1 = sp.find_all("div" , class_="commodity-desc") ##name
20 data2 = sp.find_all("em") ## price
```

Python



```
1 def step_01():
```

- 導入所需模組
- 創建兩個空串列備用
- 爬取並提取網頁內所需元素
- 將衛生紙的名字存在data1
- 將衛生紙的價格存在data2



```
def step_02():
```

```
    counter = 1

    for i in data1:
        list1.append((i.find("a" , title=True).text))

    for i in data2:
        temp1 = list(i.text)

        for j in temp1:
            if (j == ","):
                temp1.remove(",")
            elif (j == "$"):
                temp1.remove("$")

        temp2 = "".join(temp1)
        temp2 = int(temp2)

        list2.append(temp2)
```



```
1 def step_02():
2
3
4
5
6
7
8
9
10
11
12
13
14
```

```
db = sqlite3.connect("tissue.db")
cursor = db.cursor()
cursor.execute(
    '''
    CREATE TABLE TISSUE(
        name    CHAR,
        price   INT
    )
    '''
)

for i in range(0, len(list1)):
    cursor.execute("INSERT INTO TISSUE(name , price) VALUES (?, ?)" , (list1[i] , list2[i]))

db.commit()
db.close()
```

Python



```
1 def step_02():
```

- 將衛生紙的名稱依序存入list1
- 移除衛生紙價格的"\$"和",", 再存入list2
- 創建一個tissue.db的資料庫
- 在資料庫中創建一個TISSUE的表格
- 將list1和list2的內容依序存入表格



```
1 def step_03():
```

```
2
3
4
5 import sqlite3
6 import pandas
7 import matplotlib.pyplot as plt
8
9 plt.rcParams['font.sans-serif'] = ['Microsoft JhengHei']
10 plt.rcParams['axes.unicode_minus'] = False
11
12 with sqlite3.connect("tissue.db") as con:
13     c = con.cursor()
14     tissue = pandas.read_sql("SELECT * FROM tissue" , con=con)
15
16     list1 = list(tissue.name)
17     list2 = list(tissue.price)
18
19     print(list1)
20     print(list2)
```

Python



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14

```
def step_03():
```

- 在Matplotlib圖表中顯示中文
- 連接到資料庫tissue.db
- 提取資料庫內所需元素
- 將提取出元素(name和price)分別存入list1和list2





```
1 def step_04():
2
3
```

```
4 plt.plot(list1 , list2)
5 plt.title("家樂福衛生紙價格比較")
6 plt.xlabel("品牌")
7 plt.ylabel("價格(元)")
```

Python

```
8 plt.bar(list1 , list2)
9 plt.title("家樂福衛生紙價格比較")
10 plt.xlabel("品牌")
11 plt.ylabel("價格(元)")
```

Python

```
12 plt.barh(list1 , list2)
13 plt.title("家樂福衛生紙價格比較")
14 plt.xlabel("品牌")
15 plt.ylabel("價格(元)")
```

Python



```
1 def step_04():  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14
```

提取list1和list2的內容會製成折線圖、長條圖及橫條圖，並在圖表內加入圖表標題、x軸標籤、y軸標籤，讓圖表更加完整



```
1  
2 count = 4  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14
```

# 心得分享

```
## Experience Sharing
```



```
1 def experience_Chen():
```

```
2  
3  
4     在爬取的過程中我問了AI很多次，我發現它跑出來的結  
5     果不一定是能用的，我們還需要幫它改很多地方才能使用，  
6     他很喜歡在自己的觀點裡面跳針。當我終於爬出結  
7     果時，我很開心，以後比價時不用再開很多分頁了  
8  
9  
10  
11  
12  
13  
14
```



```
1 def experience_Wu():  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14
```

在這次爬蟲作業中，除了鞏固課本的知識外，還學到了新的知識，像是sqlite3和pandas模組的用法、以及如何將在Python的資料存到資料庫，再從資料庫取出資料來繪圖，讓我學習到了很多



```
1 def experience_Blue():  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14
```

我們在過程中有詢問CHATGPT，效果不佳，它給的資訊可能是從自別的網頁的一段擷取上來，除了會有種少了甚麼東西的感覺照著打最後也沒辦法順利執行



```
1  
2 count = 5  
3  
4  
5  
6
```

# 參考資料

```
7  
8  
9  
10  
11  
12 ## References  
13  
14
```



```
1 def references():
2
3
4
5
6
7
8
9
10
11
12
13
14
```

- Kao Jia. (2023, May 28). 【資料分析】如何用Python執行SQL語法. Medium.  
<https://medium.com/@kaojia/%E8%B3%87%E6%96%99%E5%88%86%E6%9E%90-%E5%A6%82%E4%BD%95%E7%94%A8python%E4%B8%8A%E5%9F%B7%E8%A1%8Csql%E8%AA%9E%E6%B3%95-60dfbdd6d3dd>
- 日常tips. (2020, July 30). Python读取sqlite数据并导入DataFrame的两种方法. CSDN.  
<https://blog.csdn.net/veritasalice/article/details/107682782>
- 鄧文淵. (2021). Python初學特訓班 Python Beginner Course (第四版). 碁峯.
- 李天胜. (2022). Python量化交易实战 (第一版). 中國水利水电出版社.





```
1 Thanks; {
```

```
2  
3 第八組:
```

```
4  
5 111210518 陳辰鈺
```

```
6 111210520 吳佳泰
```

```
7 111210524 藍健洲
```



```
10 CREDITS: This presentation template was  
11 created by Slidesgo, including icons by  
12 Flaticon, and infographics & images by Freepik
```

```
13 < Please keep this slide for attribution >
```

```
14 }
```