

高雄科技大學

智慧商務系

資料結構
期末專題

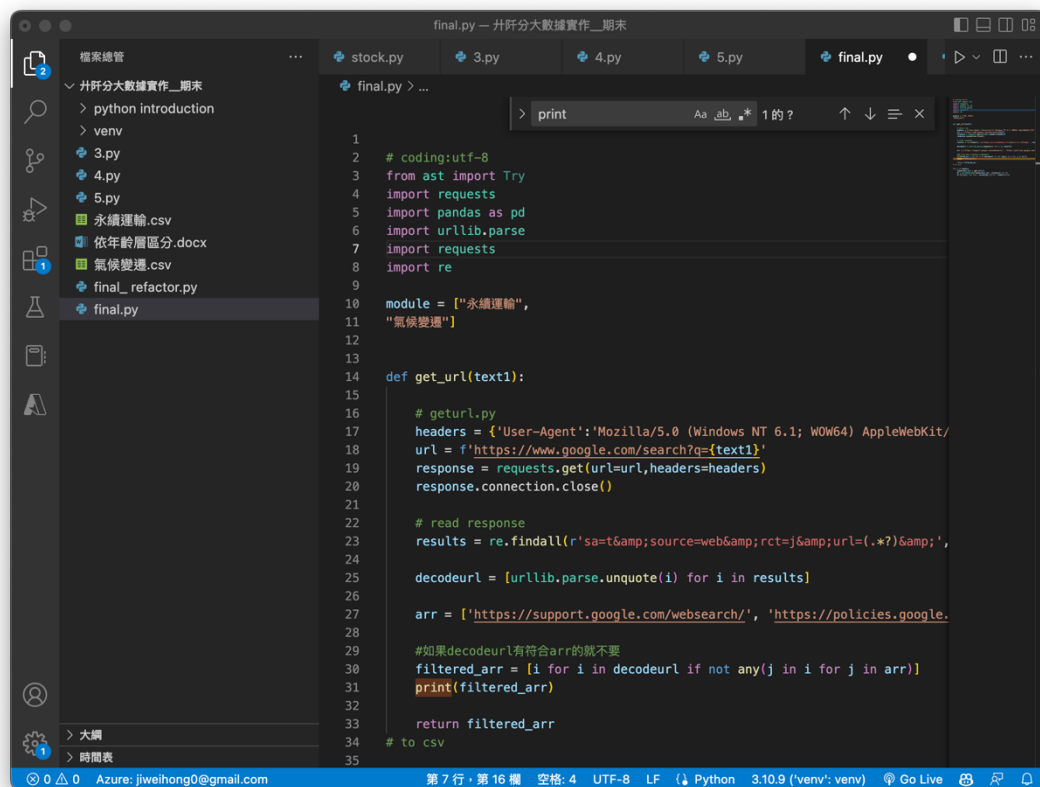
Google 瀏覽器爬蟲

姓名：紀威宏

學號：C109156225

改善說明：

原先的程式是將陣列的詞依序訪問瀏覽器，再將瀏覽器找回來的網址做儲存，現在使用 **hash** 來存儲網址，這樣可以更快地查找和過濾網址。



```
1  # coding:utf-8
2  from ast import Try
3  import requests
4  import pandas as pd
5  import urllib.parse
6  import requests
7  import re
8
9  module = ["永續運輸",
10           "氣候變遷"]
11
12
13
14 def get_url(text1):
15
16     # geturl.py
17     headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/
18 url = f'https://www.google.com/search?q={text1}'
19 response = requests.get(url=url, headers=headers)
20 response.connection.close()
21
22     # read response
23     results = re.findall(r'sa=t&source=web&rct=j&url=(.*?)&',
24
25     decodeurl = [urllib.parse.unquote(i) for i in results]
26
27     arr = ['https://support.google.com/websearch/', 'https://policies.google.
28
29     #如果decodeurl有符合arr的就不要
30     filtered_arr = [i for i in decodeurl if not any(j in i for j in arr)]
31     print(filtered_arr)
32
33     return filtered_arr
34
35 # to csv
```

原程式碼

```
final_refactor.py — 并肝分大数据實作__期末
3.py 4.py 5.py final.py final_refactor.py
final_refactor.py hash_url url
17 # geturl.py > print
18 headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64; AppleWebKit/537.36; Chrome/36.0.2130.11; Safari/537.36)'}
19 url = f'https://www.google.com/search?q={text1}'
20 response = requests.get(url=url, headers=headers)
21 response.connection.close()
22
23 # read response
24 results = re.findall(r'sa=t&source=web&rct=j&url=(.*?)&', response.text)
25
26 decodeurl = [urlib.parse.unquote(i) for i in results]
27
28 arr = ['https://support.google.com/websearch/', 'https://policies.google.com/terms/']
29
30 #如果decodeurl有符合arr的就不要
31 filtered_arr = [i for i in decodeurl if not any(j in i for j in arr)]
32
33 # useheap
34 # topurl = heap_url(text1, filtered_arr)
35 # usehash
36 topurl = hash_url(text1, filtered_arr)
37 print(topurl)
38 return topurl
39 # s用名獨算排名
40
41 # usehash
42 def hash_url(url, filtered_arr):
43     scores = {}
44     for url in filtered_arr:
45         score = calculate_score(url)
46         scores[url] = score
47
48 # 排序網址
49 sorted_urls = sorted(scores.items(), key=lambda x: x[1], reverse=True)
50 return [url for url, _ in sorted_urls]
51
52 # useheap
53 def heap_url(text1, filtered_arr):
54     scores = {}
55     for url in filtered_arr:
56         score = calculate_score(url)
57         scores[url] = score
58     sorted_urls = sorted(scores.items(), key=lambda x: x[1], reverse=True)
59     return [url for url, _ in sorted_urls]
```

重構後