

Python 個人專題

題目：OpenPyXL 模組介紹與應用

組員：方靖

指導老師：蔣寶鑫

1 1 0 年 5 月 1 9 日

目錄：OpenPyXL 模組介紹與應用.....	3
一、前言	3
二、模組介紹與基本操作	3
三、生活實例應用(爬取 Yahoo 股市製作 Excel)....	6
四、使用的優缺點分析	8
五、工作進度(甘特圖)	9
六、遇到的瓶頸、解決方法	9
七、結論	10
八、參考資料	11

題目：OpenPyXL 模組介紹與應用

一、前言

高二多元選修「程式設計」上學期是學習 Python 的基礎操作，下學期會分組製作專題。

這次的專題，我選擇自己一組，在製作過程中自由度也變大，事先規劃的日程表也比較好進行修改，可以往自己有興趣的方向進展。

OpenPyXL 也是個自由度很高的模組，也希望可以使用這個模組讓生活更便利、實用。

二、模組介紹與基本操作

模組介紹

- 建立、讀取、修改 excel 檔案
- 簡易操作就可以滿足日常文書需求
- 利用爬蟲來爬取網站資訊建立數據
- 繪製圖表(視覺化設計)
- Excel 做得到的 OpenPyXL 都做得得到

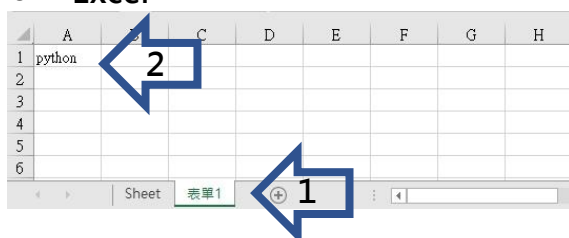
基本操作

步驟 1 建立 Excel 檔案 並輸入資料

● 程式碼

```
1 import openpyxl
2
3 # 建立一個工作簿
4 workbook = openpyxl.Workbook()
5 # 建立一個表單
6 sheet = workbook.create_sheet('表單1') #('表單名稱')
7 # 寫入一個資料
8 sheet.cell(row=1, column=1, value="python") #row(行),column(列)
9 # 儲存
10 workbook.save('檔案名稱.xlsx')
```

● Excel



● 步驟 (上方箭頭表示)

1. 建立 Excel 檔案 和 ('表單 1')。
2. 在行=1 · 列=1 中輸入('python')。
3. 儲存檔案。

步驟 2 讀取檔案 並修改

● 程式碼

```
1 import openpyxl
2 import time
3
4 # 開啟工作簿
5 workbook = openpyxl.load_workbook('檔案名稱.xlsx')
6 # 獲取表單
7 sheet = workbook['表單1']
8
9 # 修改指定的單元格資料 (現在時間)
10 sheet['A1'] = time.strftime('%Y-%m-%d %H:%M:%S', time.localtime())
11
12 # 讀取指定的單元格資料
13 cell = sheet.cell(row=1, column=1).value
14 print(cell)
15
16 workbook.save('檔案名稱.xlsx')
```

- Spyder

```
In [3]: runfile('G:/我的雲端硬碟/程式/py專題/openpyxl 模組/02-讀取檔案資料並修改.py', wdir='G:/我的雲端硬碟/程式/py專題/openpyxl 模組')
2021-05-19 13:02:19
```

- Excel

	A	B	C	D	E	F	G	H
1	2021-05-19 12:54:22							
2								
3								
4								
5								
6								

- 步驟 (上方箭頭表示)

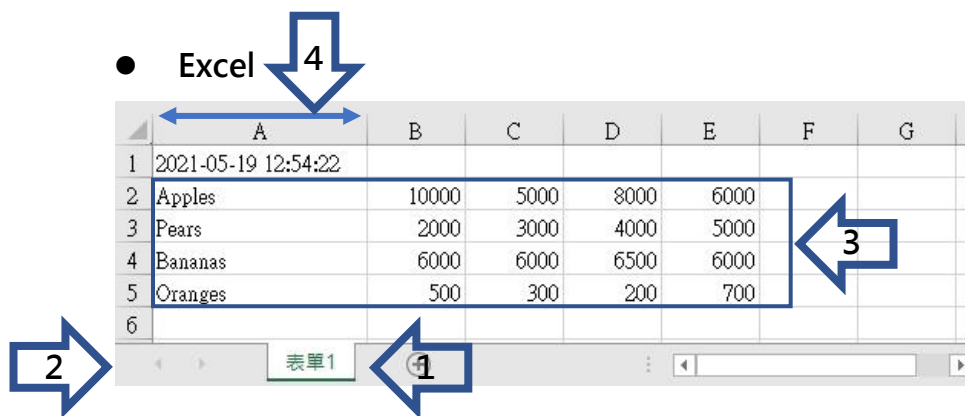
1. 開啟 Excel 檔案 和 ('表單 1')。
2. 輸入當下時間至 Excel。
3. 讀取表單(' A1')方格內資料，輸出至 Spyder 的 Console。
4. 儲存檔案。

步驟 3

多行輸入

- 程式碼

```
1 import openpyxl
2
3 # 開啟工作簿
4 workbook = openpyxl.load_workbook('檔案名稱.xlsx')
5
6 del workbook['Sheet']
7
8 # 獲取表單
9 sheet = workbook['表單1']
10
11 sheet.append(['Apples', 10000, 5000, 8000, 6000])
12 sheet.append(['Pears', 2000, 3000, 4000, 5000])
13 sheet.append(['Bananas', 6000, 6000, 6500, 6000])
14 sheet.append(['Oranges', 500, 300, 200, 700])
15
16 # 調整列的寬度
17 sheet.column_dimensions['A'].width = 20.0 # dimensions(尺寸)
18
19 workbook.save('檔案名稱.xlsx')
```



- 步驟 (上方箭頭表示)

1. 開啟 Excel 檔案 和 ('表單 1') 。
2. 刪除表單('Sheet') 。
3. 輸入串列 。
4. 調整 ('A') 列的寬度至 20 。
5. 儲存檔案 。

三、生活實例應用(爬取當日指定股票(Yahoo 股市)製作 Excel 表格)

- 介紹:

這個範例是參考「Learn Code With Mike」網站上的程式碼，簡化後得到的。

這個程式主要就是將「Yahoo 奇摩股市」的當日資訊，爬取完後匯入至 Excel。裡面有 股票名稱、時間、成交、買進、賣出、漲跌、張數、昨收、開盤、最高、最低。這些網頁上的資訊，會被爬取下來。

● 程式碼

```

1 from bs4 import BeautifulSoup
2 import requests
3 import openpyxl
4 from openpyxl.styles import Font
5
6
7 class Stock:
8     def __init__(self, *stock_numbers):
9         self.stock_numbers = stock_numbers
10
11     def scrape(self):
12
13         result = list()
14
15         for stock_number in self.stock_numbers:
16
17             response = requests.get(
18                 "https://tw.stock.yahoo.com/q/q?s=" + stock_number)
19             soup = BeautifulSoup(response.text.replace("加到投資組合", ""), "lxml")
20
21             stock_date = soup.find(
22                 "font", {"class": "tt"}).getText().strip()[-9:] # 資料日期
23
24             tables = soup.find_all("table")[2] # 取得網頁中第三個表格
25             tds = tables.find_all("td")[0:11] # 取得表格中1到10格
26
27             result.append((stock_date,) +
28                           tuple(td.getText().strip() for td in tds))
29         return result
30
31
32     def export(self, stocks): # openpyxl模組(Module)的save()方法(Method)，傳入Excel檔案的名稱
33         wb = openpyxl.Workbook()
34         sheet = wb.create_sheet("Yahoo股市", 0) # 新增一個新的sheet(工作表)，將它命名為「Yahoo股市」
35         sheet.column_dimensions['B'].width = 15.0 # 調整列寬
36         response = requests.get(
37             "https://tw.stock.yahoo.com/q/q?s=2451") # requests套件取得網頁的回應結果
38         soup = BeautifulSoup(response.text, "lxml") # 使用支援BeautifulSoup套件的lxml解析器來進行解析
39
40         tables = soup.find_all("table")[2] # find_all()方法(Method)，定位所有的表格(table)元素，並且只取得第三個表格(table)
41         ths = tables.find_all("th")[0:11] # 就能夠定位這個表格(table)元素下的前11個標題元素(th)
42         titles = ("資料日期",) + tuple(th.getText() for th in ths)
43         sheet.append(titles)
44
45         for index, stock in enumerate(stocks):
46             sheet.append(stock)
47
48             if "△" in stock[6]:
49                 sheet.cell(row=index+2, column=7).font = Font(color='FF0000') # 紅色
50             elif "▽" in stock[6]:
51                 sheet.cell(row=index+2, column=7).font = Font(color='00A600') # 綠色
52
53         wb.save("yahoostock.xlsx")
54
55 stock = Stock('2451', '2454', '2369', '2330', '0050') # 建立Stock物件
56
57 stock.export(stock.scrape()) # 將爬取的股票當日行情資料匯出成Excel檔案

```

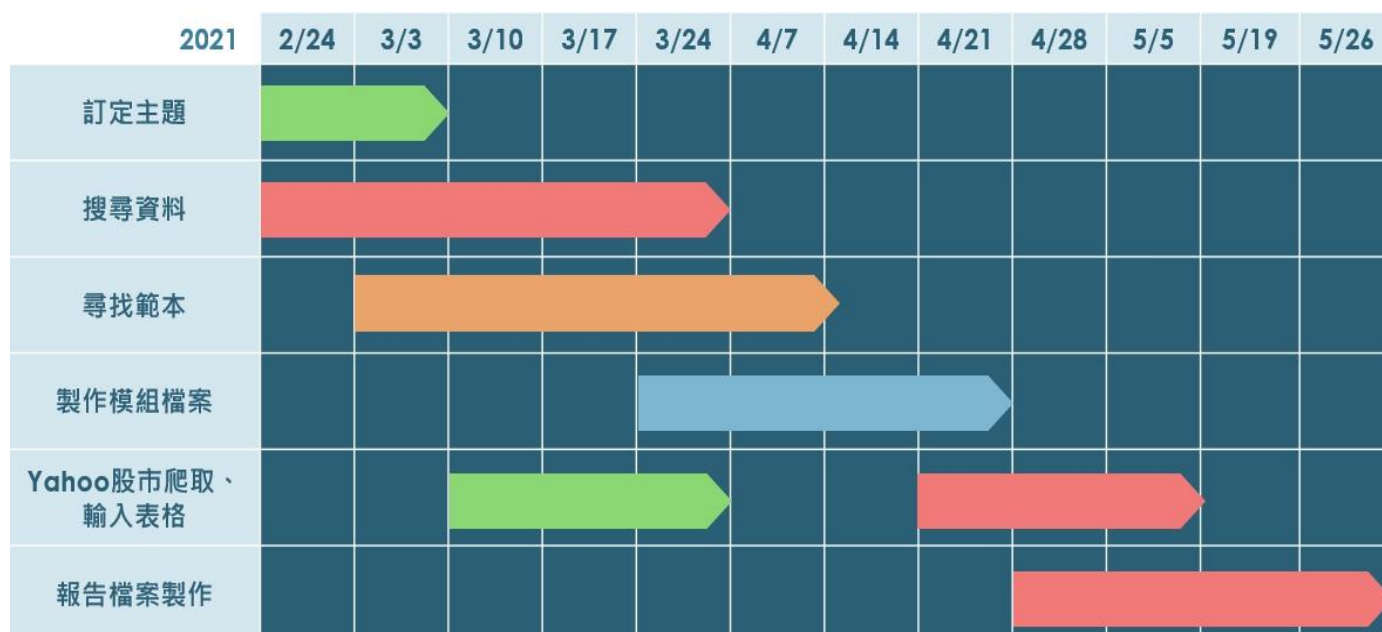
● Excel

	A	B	C	D	E	F	G	H	I	J	K	L
1	資料日期	股票代號	時間	成交	買進	賣出	漲跌	張數	昨收	開盤	最高	最低
2	110/05/19	2451創見	13:30	69.4	69.0	69.4	△0.1	1,418	69.3	68.1	70.0	67.6
3	110/05/19	2454聯發科	13:30	943	943	944	▽51	12,443	994	983	992	943
4	110/05/19	2369菱生	13:30	16.05	16.00	16.05	△1.15	10,869	14.90	15.00	16.10	14.80
5	110/05/19	2330台積電	13:30	567	567	568	▽5	28,802	572	571	572	565
6	110/05/19	0050元大台灣50	13:30	131.40	131.40	131.55	▽0.90	16,900	132.30	132.20	132.65	130.50

四、使用的優缺點分析

項目	優、缺點分析
優點	<ol style="list-style-type: none">1. 簡易的操作就可以滿足日常文書需求。2. 利用同樣的程式碼，可簡化重複性高的資料處理。3. 利用爬取網站資料，方便建立大量資訊數據。
缺點	<ol style="list-style-type: none">1. 寫程式並非視覺化操作，沒有辦法快速上手。2. 如果修改的東西不會重複使用，或是沒有大量的資料處理，就沒有必要特別使用 OpenPyXL 模組。若是直接操作 Excel 比較方便。3. 像這次爬取網路上的資料，但如果網頁有更動會改版，就要重新撰寫程式。

五、工作進度(甘特圖)



六、遇到的瓶頸、解決方法

我一開始還不太清楚自己要做什麼，因為沒有隊友可以幫忙一起提出意見，但在許多思考製作 OpenPyXL 模組，我一開始在網路上尋找範例，偶然發現了 Yahoo 股市的利用，而且因為我近期有研究一些股市的資料，所以當我看到有辦法利用簡單的爬取網頁資料、結合 OpenPyXL 後我就決定要做這個主題了。

雖然我對這個主題很有興趣，但製作難度對我來說還是有點太高了，還有很多不了解的東西。像是 類別(Class)和 建構式(Constructor)，在課堂上都沒有學習到。

需要花一些時間去研究，當我再有不懂的地方，會認真地上網找資料或是詢問厲害的同學，雖然有些地方還是沒有很了解，但大部分的程式內容都有包含註解了。

七、結論

雖然這次專題的分組採兩人一組，但也可以一人一組。這次我的是選自己一組。我會做這個決定原因是，自己研究的自由度比較高，而一年級(去年)上 Visual Basic 專題的時候是三人一組，雖然當時也做得蠻艱辛的，但兩三個人一起做，效率還是會高上不少，在學期中的時候就差不多完成了，後面的上課時間感覺有點浪費。這次因為沒有隊友 只有我一個人，所以這次研究的過程中，我有先大概規劃日程表。讓自己的進度不要落後，因為沒有隊友會互相提醒。但同時也訓練自己的自主能力，要時常檢視自己的進度，且不要想著隨時會有人來幫助。

這學年學校上的多元選修教授的是 Python，也是我第一次使用 Python，在使用的過程中發現與 C++ 語法的邏輯上有許多不同，在 C++ 中要學習的基礎非常多像要宣告變數、指派記憶體位址等等，光理解這些就要花費很多時間。而 Python 相較之下簡化了很多，可以減少許多不必要的文字，減少輸入的時間，就像是在跟電腦直接進行溝通的「直譯式語言」。另一方面 Python 會比較講求縮排，要特別空 4 格。初期使用時常會忘記，導致程式敘寫的時間變長，但透過課堂上的練習，逐漸熟練，使錯誤率降低。

透過這學期的專題製作，發現在 Python 中還有很多自己不知道的地方，也朝著目標努力，很順利的完成了這學期的研究。之後應該還有非常多的機會可以做自己有興趣的程式及研究。

八、參考資料

import、import as、from import-

<https://openhome.cc/Gossip/Python/ImportImportAsFrom.html>

[Python 爬蟲教學]活用 openpyxl 套件將爬取的資料寫入 Excel 檔案-

<https://www.learncodewithmike.com/2020/08/python-write-to-an-excel-file-using-openpyxl-module.html>

def 陳述句-<https://openhome.cc/Gossip/Python/DefStatement.html>

Python 第三方庫之 openpyxl(11) - IT 閱讀-

<https://www.itread01.com/content/1531679902.html>

python 中操作 mysql 的 pymysql 模組詳解-

<https://codertw.com/%E8%B3%87%E6%96%99%E5%BA%AB/16063/>

Python 3.1 快速導覽 - 例外處理 try-except 陳述-

<https://pydoing.blogspot.com/2011/01/python-try.html>

[Python 爬蟲教學]輕鬆學會 Python 網頁爬蟲與 MySQL 資料庫的整合方式-

<https://www.learncodewithmike.com/2020/08/python-scraper-integrate-with-mysql.html?m=1>

教學筆記本: python class 教學及範例-[http://python-](http://python-learnnotebook.blogspot.com/2018/11/python-class.html)

[learnnotebook.blogspot.com/2018/11/python-class.html](http://python-learnnotebook.blogspot.com/2018/11/python-class.html)

—

Python 使用 openpyxl 模組讀取、寫入 Excel 檔案教學與範例-

<https://officeguide.cc/python-openpyxl-read-write-excel-file-tutorial-examples/>

—

[Python 爬蟲教學]活用 openpyxl 套件將爬取的資料寫入 Excel 檔案-

<https://www.learncodewithmike.com/2020/08/python-write-to-an-excel-file-using-openpyxl-module.html>