

全部作業共同要求：

- 每個小實作的最末行，若有「----- end ----- 實驗者：江俊廷 99 號」請修改為與自己相關的名字與學號（或學號末兩碼）
- 每題都有一個照相檔（可看出已完成），且照相後要加上個人簽名或學號。
簽名內容。主要是「個人學號末兩碼或名字」或 再加上今天日期：x / yy 例如 10 / 24 。
- 傳檔案的檔名：「pX.jpg, 或 X.jpg」其中的 X 代表題號， 例如：「p3.jpg」代表第 3 題
- 每小題存成一個檔案為原則。
如果要集中在一個檔案也可以，只是題號必須要標示清楚，若標示不清楚，將會以沒有作答處理。

(注意 1:請務必要選擇指定題：1,2,3, 5, 8, 10，都有加◎，其他任選 1 題即可)

(注意 2:若有選 P11, 這題是有機會額外加分的特別題)

◎ P1(實作題 1)：P1_ch01_上傳作業 1_「兩種拿到 iris 資料集並且視覺化」的實驗比較

※本題額外要求 1：在顯示兩種方法拿到的資料集時，其中 no_座號=99 要修變為自己相關的座號。

※本題額外要求 2：同時要打開 json 檔與文字檔，查看內容。

參考畫面如下：

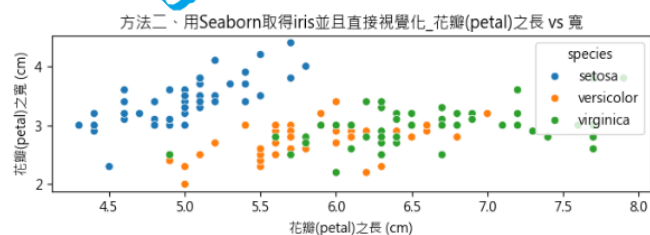
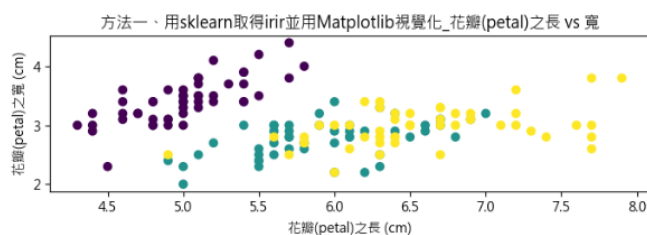
```
48 plt.show()
```

我的座號是99,在兩方法拿到的資料集，在index附近的資料內容是：

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
99	5.7	2.8	4.1	1.3	1.0
100	6.3	3.3	6.0	2.5	2.0

	sepal_length	sepal_width	petal_length	petal_width	species
99	5.7	2.8	4.1	1.3	versicolor
100	6.3	3.3	6.0	2.5	virginica

兩種拿到iris資料集並且視覺化的兩種方法之實驗比較



[2]: 1 # ●ch01_上傳作業1_兩種拿到iris資料集並且視覺化的實驗比較 ----- end ----- 實驗者：江俊廷99號

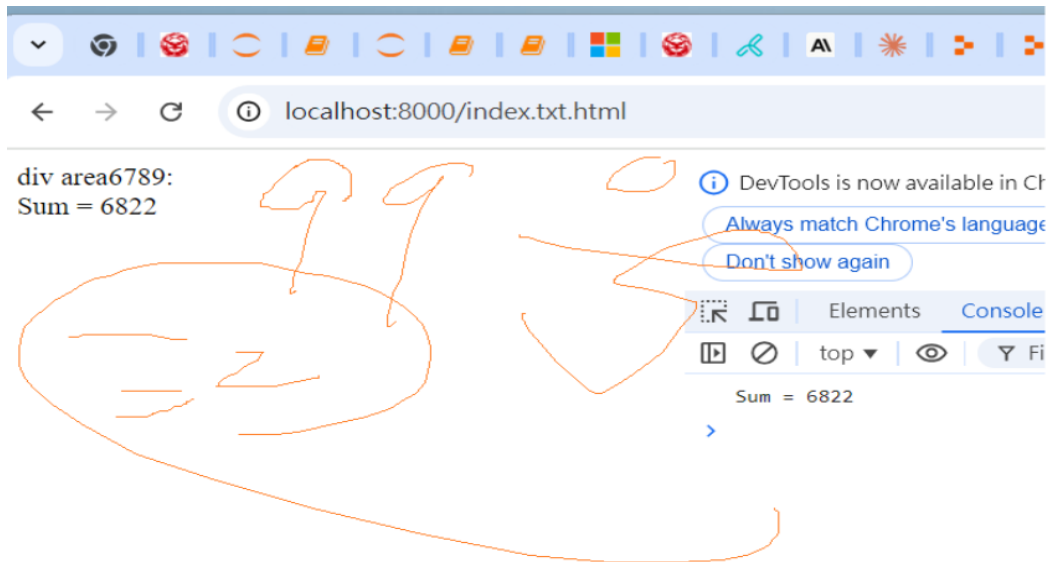
◎ P2(實作題 2)：P2__Lab#ch02-1a_js 入門 localhost(請在 div area01 後面的 01，與 var data 裡 ,0] 的 0，改為自己的學號末四碼)

※本題額外要求 1：基於安全理由。要確定知道資料夾名稱，請把 TTT 改為指定的 TTT1024

※本題額外要求 2：請在 div area01 後面的 01，與 var data 裡 ,0] 的 0，改為自己的學號末四碼)

※本題額外要求 3：請在照相時，要打開 chrome DevTools 的執行畫面

參考畫面如下：

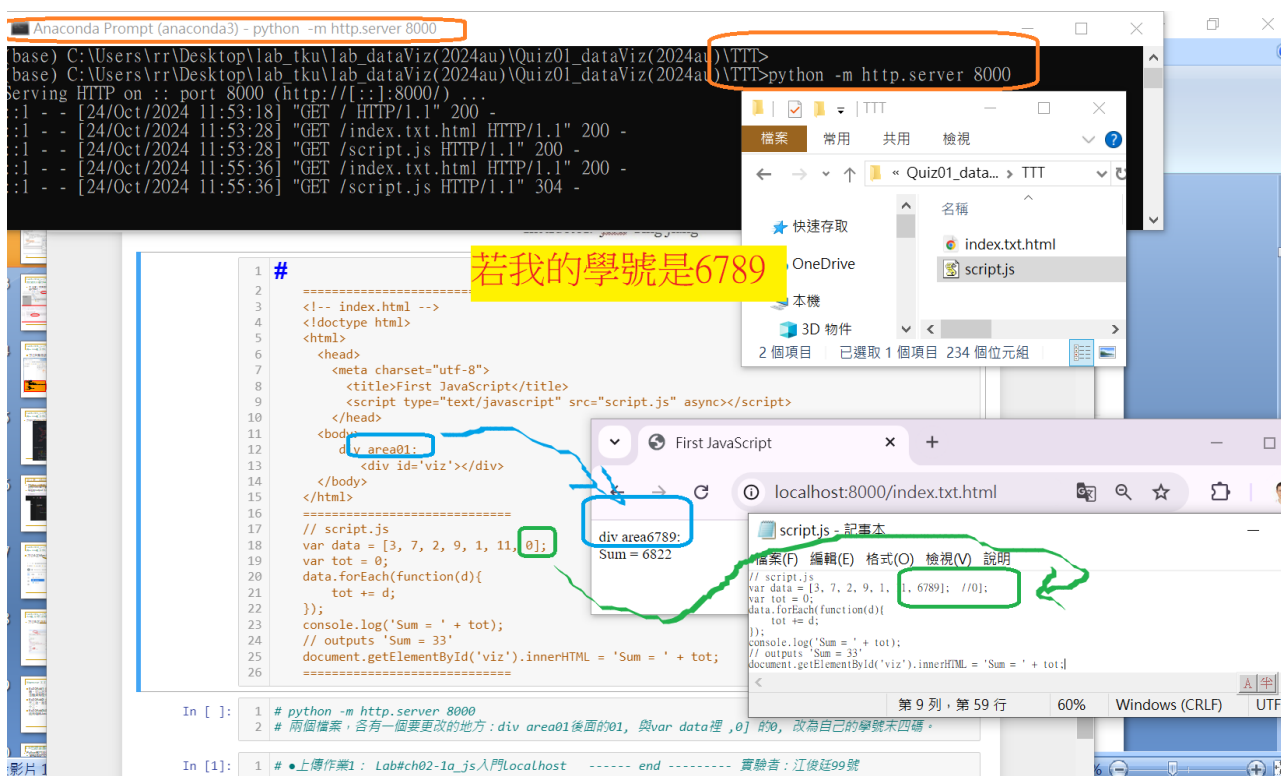


~~~~ 本題的補充教學說明：

(1)畫面左上方，可看到，是執行 `python -m http.server 8000`

(2)這個資料夾裡面有兩個檔案。分別是\*.html，與\*.js

(3)兩個檔案，各有一個要更改的地方：div area01 後面的 01，與 var data 裡 ,0] 的 0，改為自己的學號末四碼。



### ◎ P3(實作題 3)：P3\_ch02\_上傳作業 2\_Lab#2-3\_Example 2-3. Simple data 小數據處理

※本題額外要求 1：## 額外要求：py, js, 都加入 3 筆與自己相關的新資料，例如：

# { 'name': '江俊廷', 'id': 99, 'scores': [80, 81, 82, 83]}, (平均超過 75 才是完美通過！)

# {'name': '江俊廷 2', 'id': 98, 'scores': [75, 75, 75, 75]},

# {'name': '江俊廷 3', 'id': 97, 'scores': [59, 59, 59, 59]},

參考畫面如下：

## ●上傳作業2：簡單小數據處理\_\_py與js

- 要求：py, js, 都加入 3 筆與自己相關的新資料，例如：
- {'name': '江俊廷', 'id': 99, 'scores': [80, 81, 82, 83]}, (平均超過 75 才是完美通過！)
- {'name': '江俊廷 2', 'id': 98, 'scores': [75, 75, 75, 75]},
- {'name': '江俊廷 3', 'id': 97, 'scores': [59, 59, 59, 59]},

要學會jupyter notebook複格到新檔案的方法！

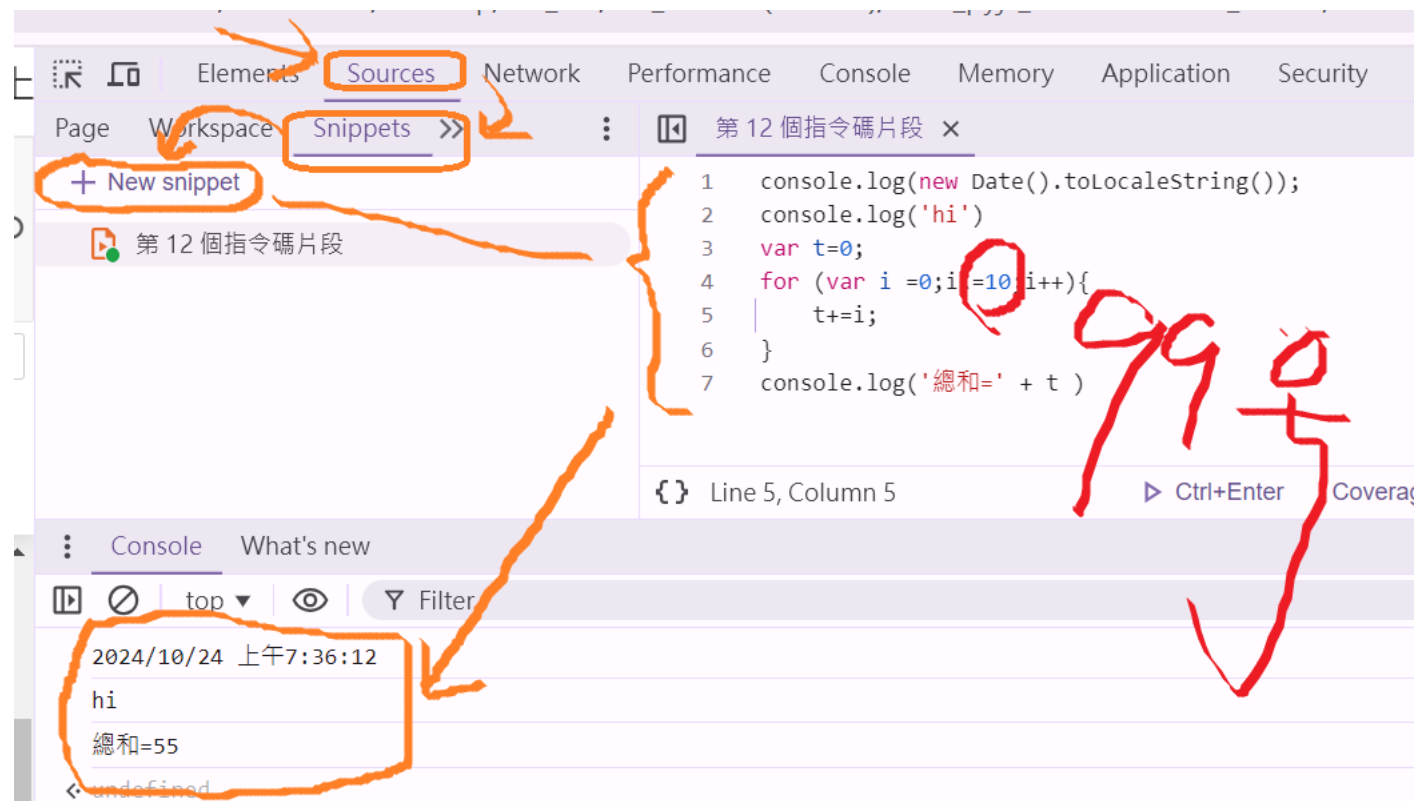
```
jupyter 簡單小數據處理__py與js##### end ##### 製作者：江俊廷99號 Last Checkpoint: 幾秒前 (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help
29 else:
30     sdata['assessment'] = 'failed'
31     # D
32     print("%s's (id: %d) final assessment is: %s"%(sdata['name'], sdata['id'], sdata['assessment'].upper()))
33 #E
34 if __name__ == '__main__':
35     process_student_data(student_data)
Bob's (id: 0) final assessment is: PASSED
Alice's (id: 1) final assessment is: PASSED WITH MERIT
Carol's (id: 2) final assessment is: PASSED
Dan's (id: 3) final assessment is: FAILED
江俊廷's (id: 99) final assessment is: PASSED WITH MERIT
江俊廷2's (id: 98) final assessment is: PASSED
江俊廷3's (id: 97) final assessment is: FAILED
56
57 // E
58 processStudentData(studentData);
59
Bob's (id: 0) final assessment is: PASSED
Alice's (id: 1) final assessment is: PASSED WITH MERIT
Carol's (id: 2) final assessment is: PASSED
Dan's (id: 3) final assessment is: FAILED
江俊廷's (id: 99) final assessment is: PASSED WITH MERIT
江俊廷2's (id: 98) final assessment is: PASSED
江俊廷3's (id: 97) final assessment is: FAILED
In [2]: 1 # 顯示小數據的內容：
2 print(student_data)
[{'name': 'Bob', 'id': 0, 'scores': [68, 75, 56, 81], 'average': 70.0, 'assessment': 'passed'}, {'name': 'Alice', 'id': 1, 'scores': [75, 90, 64, 88], 'average': 79.25, 'assessment': 'passed with merit'}, {'name': 'Carol', 'id': 2, 'scores': [59, 74, 71, 68], 'average': 68.0, 'assessment': 'passed'}, {'name': 'Dan', 'id': 3, 'scores': [64, 58, 53, 62], 'average': 59.25, 'assessment': 'failed'}, {'name': '江俊廷', 'id': 99, 'scores': [80, 81, 82, 83], 'average': 81.5, 'assessment': 'passed with merit'}, {'name': '江俊廷 2', 'id': 98, 'scores': [75, 75, 75, 75], 'average': 75.0, 'assessment': 'passed'}, {'name': '江俊廷 3', 'id': 97, 'scores': [59, 59, 59, 59], 'average': 59.0, 'assessment': 'failed'}]
```

Lab#2-3\_Example 2-4 Simple data小數據處理with [javascript]

P4(實作題 4)：P4\_ch02\_上傳作業 3\_Lab#2-3Add\_ChromeDevTools 的第一個小程式

※本題額外要求 1：# 額外要求：要求:其中的 10 改為自己的學號末四碼。

參考畫面如下：



◎ P5(實作題 5)：P5\_ch02\_上傳作業 4\_Lab#2-4\_計算「1~10 裡所有奇數平方和」

※本題額外要求 1：其中的 10 改為自己的學號末四碼。

參考畫面如下：

相同問題：「所有奇數平方和」，上格是JavaScript的寫法，下一格換python的寫法

```
In [2]: 1 # 程式目的： 篩選出奇數，並將其平方，然後求和
        2 # # pytho方法1: 用最佳的list compreshension(列表理解)來實現。
        3 nums = range(1, 10 +1 )
        4 odd_squares = [x * x for x in nums if x%2 != 0]
        5 print("所有奇數平方和",sum(odd_squares) )
```

所有奇數平方和 165

```
In [3]: 1 # pytho方法2: 使用 filter(), map(), reduce() 聚合函數來實現
        2 from functools import reduce
        3 nums = list(range(1,10+1 ))
        4 result = reduce(
        5     lambda x, y: x + y,
        6     map(lambda x: x**2,
        7         filter(lambda x: x % 2 != 0, nums)))
        8 print("所有奇數平方和", result)
```

所有奇數平方和 165

```
In [4]: 1 # 額外要求: 其中的10 改為自己的學號末四碼。
        2
        3 # ●上傳作業4_Lab#2-4_計算「1~10裡所有奇數平方和」 ----- end ----- 實驗者：江俊廷99號
```

In [ 1 ]:

1

P6(實作題6): P6\_ch02\_上傳作業5\_Lab#2-5-2\_JS的重覆結構,計算1+2+...+10的總和的實驗(for, while, 與無窮迴圈), 實驗時, 請把10改為「自己的座號加100」。

※本題額外要求1: 要求: 其中的10 改為自己的學號末四碼。

參考畫面如下:

```
35     t=0; i=1;
36     while (true){
37         t+=i;
38         if (i>=10) break;
39         i+=1;
40     }
41     my_log("方法3:"+t)
```

方法1:55

方法2:55

方法3:55

相同問題: 「1+2+...+10的總和」, 上格是JavaScript的寫法, 下一格換python的寫法

```
In [2]: 1 #方法1
2 t=0
3 for i in range(10+1):
4     t+=i
5     print("方法1", t)
6
7 #方法2
8 t=0
9 i=1
10 while i<=10:
11     t+=i
12     i+=1
13     print("方法2", t)
14
15 #方法3
16 t=0
17 i=1
18 while True:
19     if i>10:
20         break
21     t+=i
22     i+=1
23     print("方法3", t)
24
```

方法1 55

方法2 55

方法3 55

```
In [3]: 1 # 上傳作業5_Lab#2-5-2_JS的重覆結構, 計算1+2+...+10的總和的實驗 ----- end ----- 實驗者: 江俊廷99號
```

P7(實作題 7)： # P7\_ch03\_上傳作業 1\_Lab#3-0a-FileOpen 逐行讀資料的方法\_\_文字檔與 JSON 檔  
※本題額外要求 1：# 請把出現「name 江俊廷」與「987654321」的地方改為「name + 自己的姓名」與「自己的學號」

參考畫面如下：

```
3 dat="123","456" ##error 測試
4 dat="abcccccccccccccccccccc" ##error 測試
5 dat='{ "name江俊廷": "987654321", "year": 20 }'
6
7 with open('in.json', 'w', encoding='utf-8') as f:
8     f.write(dat)
9
10
11 print('\ndat準備好了---用文字檔讀檔檢查檔案的內容')
12 ##用方法三印出內容、●● 這是配合 with 的高效率寫法！
13 with open('in.json', "r", encoding='utf-8') as fp:
14     for line in fp:
15         print(line, end='')
16
17 print('\ndat準備好了---用json_直接載入load')
18 ### JSON讀入#####
19 dat_load=''
20 import json
21 with open('in.json', "r", encoding='utf-8') as fp:
22     dat_load=json.load(fp)
23 print(dat_load) ##直接是正確的 json的物件格式！
24
```

dat準備資料

```
dat準備好了---用文字檔讀檔檢查檔案的內容
{ "name江俊廷": "987654321", "year": 20 }
dat準備好了---用json_直接載入load
{'name江俊廷': '987654321', 'year': 20}
```

1]: 1 # 請把出現「name 江俊廷」與「987654321」的地方改為「name + 自己的姓名」與「自己的學號」  
2 # ●ch03\_上傳作業1：逐行讀資料的方法\_\_文字檔與JSON檔 ----- end ----- 實驗者：江俊廷99號



◎ P8(實作題 8)： P8\_ch03\_上傳作業2\_\_爬蟲相關 --- (1)爬 JSON 數據 (2)爬 CSV 數據

※本題額外要求 1：要自己找到 json 的網址，並且爬蟲成功。

參考作法的步驟：

1. 政府資料開放平臺: <https://data.gov.tw/> (為了避免資料發生不同步的情況，請由此開始連結！！)
2. 搜尋關鍵字：「不動產實價登錄 淡水」「歷年不動產實價登錄資訊-買賣案件-111 年-淡水區」
3. 正常情況，第 1 個會是「[歷年不動產實價登錄資訊-買賣案件-101 年-淡水區](#)」請點進去 / 「CSV 旁邊的 檢視資料」/ 再複製 JSON 的鏈結網址：
4. 只要爬蟲 JSON 成功即完成。(CSV 省略)
5. 請視情況(因為每年都有可能發生局部修改資料格式) 修改程式，可以印出前 14 筆資料，地區，改為 "[歷年不動產實價登錄資訊-買賣案件-111 年-淡水區](#)"

※注意，為了避免換頁問題，請設為每頁顯示 100 項，就不會發生找不到資料的情況！



參考畫面如下：  
(只以中和區為例)

P8\_●ch03\_上傳作業2\_\_爬蟲相關 --- (1)爬JS... Last Checkpoint: 3 分鐘前 (unsaved changes)

View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykern)

```
1 ##網路上opendata的 JSON 用爬蟲方法，爬政府資料開放平台的 opendata https://data.gov.tw/
2 # 不動產實價登錄資訊-買賣案件-中和區 https://data.gov.tw/dataset/146474
3 import requests
4 # json 2024年版
5 http='https://quality.data.gov.tw/dq_download_json.php?nid=146474&md5_url=cca3539e4734542313e6406
6 response = requests.get(http) ##直接get會自動生成一個session()
7 #client = requests.session()
8 #response = client.get(url=http) # 執行 get 的http， ##另一個常見的是post
9 dic=response.json()
10 for i in range(3):
11     #print(dic[i])
12     print(dic[i]['district'], " 地段：" , dic[i]['rps02'][:16], " 平方公尺=",dic[i]['rps03'], " 總
```

中和區 地段： 錦和段316地號 平方公尺= 14.4 總價= 1380960 每平方公尺價格= 1380960  
中和區 地段： 新北市中和區興南路一段105巷2 平方公尺= 12.87 總價= 6900000 每平方公尺價格= 6900000  
中和區 地段： 新北市中和區國光街102巷27號 平方公尺= 16.98 總價= 8100000 每平方公尺價格= 8100000

```
1 # 只要爬蟲JSON成功即完成。
2 # ●ch03_上傳作業2： 爬蟲相關 --- (1)爬JSON數據 (2)爬CSV數據 ----- end ----- 實驗者：江俊廷99號
```

### P9(實作題 9)： P9\_ai 工具\_生成模擬某次月考全班成績的資料集

可考慮使用的提示(Prompt):

「請幫忙製作一組數據，模擬某次月考全班的成績有「英文、數學、計概」三科，用 python 來寫，並且可放在 dataframe，然後，增加修改程式，可讓第一位學生姓名改為「江俊廷」，學號是「987654321」，成績可修改為 99 分,98 分,97 分。完成後，全部放在一個 JSON 的檔案中。」

※本題額外要求 1： # 額外要求：請用自己熟練的 AI 工具，生成這段程式，功能要求是「要有修改資料的程式」，並且讓第一位學生姓名改為「江俊廷」，學號是「987654321」，成績可修改為 99 分,98 分,97 分

參考畫面如下：

The screenshot displays a Jupyter Notebook interface with two main panels. The left panel shows the code editor with Python code for generating student data and modifying the first student's record. The right panel shows a file view of the generated JSON file, `student_grades.json`.

**Code Editor (Left Panel):**

```
70
71 # 生成原始數據
72 df = generate_student_data()
73
74 # 修改第一位學生的資料
75 new_scores = {'英文': 99, '數學': 98, '計概': 97}
76 df = modify_student_data(df, student_index=0, new_scores=new_scores)
77 save_to_json(df) # 保存為JSON
78 # 顯示結果
79 print("學生成績數據__目前只顯示前面3個人的狀況。")
80 print(df[:3])
81 #print("\n成績統計:")
82 #print(df.describe())
83 print("\n數據已保存至 student_grades.json")
```

**Output (Below Code Editor):**

學生成績數據\_\_目前只顯示前面3個人的狀況。

|   | 學號        | 姓名  | 英文 | 數學 | 計概 | 平均    |
|---|-----------|-----|----|----|----|-------|
| 0 | 987654321 | 江俊廷 | 99 | 98 | 97 | 98.00 |
| 1 | 000000002 | 周小婷 | 56 | 67 | 83 | 68.67 |
| 2 | 000000003 | 黃小明 | 75 | 57 | 93 | 75.00 |

數據已保存至 student\_grades.json

**File View (Right Panel):**

```
1 {
2   "metadata": {
3     "generated_time": "2024-10-24 13:25:25",
4     "total_students": 30,
5     "subjects": [
6       "英文",
7       "數學",
8       "計概"
9     ]
10  },
11  "data": [
12    {
13      "學號": "987654321",
14      "姓名": "江俊廷",
15      "英文": 99,
16      "數學": 98,
17      "計概": 97,
18      "平均": 98.0
19    },
20    {
21      "學號": "000000002"
```

◎ P10(實作題 10)： Numpy\_補充的實作練習 EXercise\_Numpy\_補充的實作 Exercise(2025au)

本題請參考 pdf 檔： Quiz01 的 prob 之 1\_\_EXercise\_Numpy\_補充的實作\_probOnly.pdf

說明：這個 pdf 內有 numpy 實作題 7 題。請找出全部解答，並且上傳「可正確執行後的 ipynb」，與「最後面的照相畫面」

※本題額外要求 1：可正確執行後的 ipynb」檔名是 p10.ipynb

※本題額外要求 2：「最後面的照相畫面」檔名是 p10.jpg (或 p10.png)

參考畫面如下：

請自行判斷。

P11(實作題 11)：有機會額外加分的特別題，(若選本題，有機會額外加分)

儲存資料的四種方式：csv, json, sqllite3 與 mongodb

# QA: 請幫我製作中文 Notebook 版本的 ipynb 檔案 (若無法產生下載檔案，請改為產生產生完整 Notebook 原始 JSON 的內容，我就可以複製到文字檔，並可自己改為 ipynb 來執行。)

，在前面格子要加上下列三行程式碼呈現正確中文字。

```
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['Microsoft JhengHei', 'SimHei', 'Taipei Sans TC Beta', 'PingFang TC']
plt.rcParams['axes.unicode_minus'] = False # 避免負號顯示錯誤
```

假設，我已經安裝好 monogodb 在 27017 的 port。

現在的工作目的，是可以讓初學者，知道python+資料庫，並且了解核心操作中 CRUD 四件重要函數。預設一個只有 4 個學生的「學號，姓名、住址、電話」的資料表格。同時用 csv, json, sqllite3 與 mongodb 來存資料。儘量讓學者「可視化內容」。並在每種方式上方加上格子，呈現標題與簡單易懂的中文註解。

※※※注意，基本要求是 前三種方式 csv, json, sqllite3 可正確工作即可!

若連 mongodb 都有在執行前，正確安裝，且得到完全正確的結果，有額外加分。

參考畫面如下：

請自行判斷。

### 3-5-1: 資料庫，要解決的CRUD的問題

#### ●【check2】SQL如何處理資料庫CRUD的問題:

- (Create)新增資料
  - insert into 表名 values('資料1', '資料2', '資料3');
  - insert into 表名 (資料項1, 資料項2, 資料項3) values ('資料1', '資料2', '資料3');
- (Read)讀取資料
  - select 資料項 from 表名 where 條件;
- (Update)更新資料
  - update 表名 set 屬性項 where 表名;
- (Delete)刪除資料
  - delete from 表名 where 條件;
  - delete from 表名; //表中資料全部刪除