

- ExA-1Add1:「能夠移動數據」是數據可視化基本技能之一，請問可放數據的地方，主要有那兩類？分別又可概分為什麼？簡單說明即可。
 - ExA-1Add2:關於數據庫的 SQL 與 NoSQL 請簡單說明與比較，而且指出 python 在處理這些數據時常用的工具庫是什麼？
 - ExA-1Add3:如果在 python 已經有一個符合 JSON 格式的資料物件稱為 stock_yahoo,請寫出可以將資料寫入 CSV 檔的程式碼。
-

ExA-1Add1:「能夠移動數據」是數據可視化基本技能之一，請問可放數據的地方，主要有那兩類？分別又可概分為什麼？簡單說明即可。

Ans:

主要放數據的地方：

數據庫：資料庫(databases) 包括 SQL 與 NoSQL

數據檔：檔案格式(file formats)包括 JavaScript 物件表示法格式(JSON)與 逗號分隔值格式(CSV, Comma-separated values) 或其他更複雜的格式.

對數據處理人員而言，都應該都可以輕鬆讀取數據並進行轉換並將其寫成更方便的形式。

ExA-1Add2:關於數據庫的 SQL 與 NoSQL 請簡單說明與比較，而且指出 python 在處理這些數據時常用的工具庫是什麼？

Ans:

關於數據庫：

(第 1 類)SQL 資料庫--關聯式資料庫(Relational database)：

可用標準的資料結構化查詢語言(SQL, Structured Query Language) 結構化查詢語言),對關聯式資料庫中資料的檢索和操作。 python 有一些很棒的庫可以與 SQL 數據庫進行交互，例如 SQLAlchemy，是不錯的入門指南。

(第 2 類)NoSQL 非關聯式資料庫(Not Only SQL)：

數據儲存可以不需要固定的表格模式，也可認為就是「無綱要 SQL (No Schema SQL)」(或「無表格架構資料庫」)

MongoDB 是一種較新的 NoSQL 資料庫。配合 Python 的 pymongo 庫，適合新手。

ExA-1Add3:python 的 Dict 與 Json 格式，有何異同？

Python 的字典(dict)和 JSON (JavaScript Object Notation) 格式：

關於相似之處：

結構類似：JSON 格式的資料結構是以鍵-值對的方式組織的，與 Python 的字典結構相似。在 JSON 中，物件是由鍵-值對組成的，而在 Python 中，字典也是由鍵-值對組成的。

支援嵌套：JSON 和 Python 的字典都支援嵌套結構，這意味著你可以在其中嵌套其他的字典、串列或其他資料型別。

易於讀寫：由於結構的相似性，你可以將 Python 的字典很容易地轉換成 JSON 格式，反之亦然。

關於差異之處：

語法： JSON 的語法是一種用於數據交換的輕量級文本格式，它有一些特定的語法規則，例如鍵和值「必須使用雙引號」包裹。而 Python 的字典使用的是 Python 的語法，鍵可以使用引號或不使用引號。

數據類型限制： JSON 只支援有限的基本數據類型，如字串、數字、布林值、物件、陣列（串列），並且不能包含自定義的函數或其他 Python 特定的類型。然而，Python 的字典可以容納更多各種類型的數據，包括自定義的類型和函數。

命名空間： 在 Python 中，字典的鍵可以是任何不可變的數據類型，如字串、數字等。然而，在 JSON 中，物件的鍵必須是字串類型。

特殊方法和屬性： Python 的字典有許多內建的特殊方法和屬性，可以用來操作和查詢字典。JSON 是一個單純的數據格式，沒有這些特殊方法和屬性。

儘管有這些差異，Python 的字典和 JSON 之間的相似性使得它們可以方便地互相轉換，這對於數據序列化、儲存和交換非常有用。通常情況下，你可以使用 Python 的 json 模組來實現字典和 JSON 之間的轉換。

ExA-1Add4:如果在 python 已經有一個符合 JSON 格式的資料物件稱為 stock_yahoo,請寫出可以將資料寫入 CSV 檔的程式碼。

Ans:

參考 Lab#3-13 建立一個簡單的資料檔案 CSV-以諾貝爾獎得主為例(stock_yahoo)

```
cols = stock_yahoo[0].keys()
with open('./ stock_yahoo.csv', 'w') as f:
    f.write(','.join(cols) + '\n')
    for o in stock_yahoo:
        row = [str(o[col]) for col in cols]
        f.write(','.join(row) + '\n')
```

Lab# 3-1-3 實作nobel的CSV檔

- (行號1)從csv檔的第一列資料，取出當作物件的鍵（即“category”，“name”...）
- (行號3)使用Python的with語句來確保文件在離開該代碼塊或發生任何異常時均已關閉。
- (行號4) join目的是把列表（此處為cols）轉為字串，並用,串接。即“category.name,”。當作key.
- (行號6)使用nobel_winners中物件的列(column)，建立list，然後，就可在行號7再用join轉為字串，即可寫到CSV檔。

```
1 cols = nobel_winners[0].keys()
2 ### sorted(cols) sorted by keys 此時應不需排序。所以省略
3 with open('./nobel_winners.csv', 'w') as f:
4     f.write(','.join(cols) + '\n')
5     for o in nobel_winners:
6         row = [str(o[col]) for col in cols]
7         f.write(','.join(row) + '\n')
```