

解答精簡摘要 (保留每題要點)

來源：老師提供的 Exercise Solution (ex_dataViz_pyjs_ch01 等)

Ex1-0aAdd1

題：Python 優點與缺點？

答：

- 優點：簡單易學、語法清晰、豐富第三方庫、開發效率高、應用廣泛、可移植性好。
- 缺點：速度較慢、對縮排敏感、功能多學習曲線長。

Ex1-0bAdd1

題：JS 最大優勢與潛在競爭者？

答：

- 優勢：在 Web 上的唯一主流語言，生態廣、應用多 (前/後端、行動、遊戲等)。
- 潛在競爭：WebAssembly (可用於性能關鍵部分)。

Ex1-0bAdd2

題：d3.js 與 3D 呈現是否合適？

答：

- d3.js 可呈現三種屬性 (例如 x, y, 大小/顏色)，對於 3 個維度可用氣泡圖等手法表示；若需複雜 3D，可考慮 vis.js 或專門 3D library。

Ex1-0bAdd3

題：WebAssembly 的風險 (惡意程式) ？

答：

- 由於二進位與高效能，可能被濫用做隱蔽行為 (如加密挖礦)，使惡意軟體檢測更困難。

Ex1-0cAdd1

題：結合 Python 與 JS 的優點與關鍵技術？

答：

- 優點：Python 擅資料處理；JS 擅互動呈現，結合可做資料分析並在瀏覽器互動展示。
- 關鍵資料交換技術：JSON (JavaScript Object Notation)。

Ex1-12Add1

題：如何印出 iris 的官方介紹？

答：

```
from sklearn import datasets
iris = datasets.load_iris()
print(iris['DESCR'])
```

Ex1-12Add2

題：Seaborn 的縮寫與如何畫 pairplot？

答：

- 縮寫：sns
- 範例：

```
import seaborn as sns
sns.set(style="ticks")
df = sns.load_dataset("iris")
sns.pairplot(df, hue="species")
```

- 任兩屬性可觀察：正相關、無關、負相關。

Ex1-12Add3

題：三種常用取得 iris 的來源與程式？ 答：

- sklearn: `datasets.load_iris()`
- seaborn: `sns.load_dataset("iris")`
- UCI 官方 CSV: 以 pandas 讀取 <https://archive.ics.uci.edu/.../iris.data>

Ex1-5Add1

題：資料視覺化工具鏈五步驟？

答：

1. 抓取資料 (Scraping)
2. 清洗資料 (Pandas)
3. 探索/處理 (IPython + Pandas + Matplotlib)
4. 提供資料 (Flask/Eve 建 RESTful API)
5. 前端視覺化 (D3 等轉換與呈現)

Ex1-5Add2

題：REST 與 RESTful？

答：

- REST：一套 Web 服務架構風格 (client-server, stateless, cache, layer, uniform interface...)。
 - RESTful：形容符合 REST 原則的服務 (-ful 為形容詞尾)。
-

Ex1.1Add1

題：Python 是解譯還是編譯？註解、多行、一行分多行符號？

答：

- Python 為解釋型 (interpreted) 。
- 註解：#；同一行寫多句用 ;；多行延續用 \。

Ex1.1Add2

題：常用 Python IDE？

答：

- IDLE, PyCharm, Eclipse+PyDev, VSCode, Spyder, Jupyter Notebook (資料科學常用) 。

Ex1.1Add3

題：Python 優點總結？

答：

- 簡潔、豐富庫、開發效率高、跨領域應用 (Web、AI、科學計算、爬蟲等) 。

Ex1.1Add4

題：為何 Python 流行？

答：

- 生態、庫豐富、AI 需求、可擴展 C/C++，多平台支援等。

Ex1.1Add5

題：結構化程式三種控制結構？

答：

- Sequence, Selection (if), Iteration (for/while)。簡單示例在原文。
-

Ex1.2Add1

題：如何查保留字？

答：

```
import keyword
print(keyword.kwlist)
```

Ex1.2Add2

題：Python 新增的兩個保留字？用途？

答：

- `async`, `await` (用於非同步協程) 。

Ex1.2Add3

題：四種不可變型別與常用可變序列？

答：

- 不可變：bool, int, float, str 。
- 常用可變序列：list (像可裝貨的火車) 。

Ex1.2Add4

題：list/tuple/dict 宣告符號與增刪方法？

答：

- list `[]` · dict `{}` · tuple `()` .
- list 加: `append` · 刪: `del list[idx]` ; dict 新增/修改: `d[key]=val` · 刪: `del d[key]` · 查: `d.get(key, default)` 。

Ex1.3Add1 / Add2

題：List 與 Dict 的常見操作 (存取/新增/刪除) 示例？

答：示例程式已在原文 (`append`、`pop`、`del`、dict 操作等) 。

```
lst = [1,2]; lst.append(3); del lst[0]
d = {}; d['a']=1; d.get('b', None); del d['a']
```

Ex1.4Add1-3

題：控制區塊起始符號、縮排、For/While 使用時機？

答：

- 起始符號：· 縮排建議 4 空格 (不要混用 Tab) 。
- `for` 適合已知迴圈次數；`while` 適合條件驅動直到滿足停止 。

Ex1.5Add1-3

題：參數傳遞(Call by Sharing)與效果、clear 範例、scale 函式的正確寫法？

答：

- Python 採 Call-by-Sharing (物件參考傳遞) 。
- 想要 value-like 行為用不可變類型；想要 ref-like 行為用可變類型 (list) 。
- `scale` 應修改 list 元素：

```
def realscale(data, factor):  
    for i in range(len(data)):  
        data[i] *= factor
```

- `val *= factor` 在迴圈內對值副本無效 (bad 設計) 。

Ex1C1.17 (縮放函數)

題： `for val in data: val *= factor` 為何不變更原 list ?

答：因為 `val` 是元素值的複製 (名稱綁定) ，非改變 list 的參考，需用 `data[i] *= factor` 才能修改原元素。

Ex2- 系列 (Python 與 JS 比較、ES 規格、PEP、async/await 等)

- Python 與 JS 相似：直譯語言、互動式 shell、垃圾回收、函數一等公民等。
- 差異：JS 為 single-thread 非阻塞、Python 廣泛應用於科學/AI、JS 在瀏覽器為唯一語言等。
- JS 標準：ECMAScript (ES5, ES6/2015,...) 。
- Python PEP (例如 PEP8) 為風格指南；PEP3107 引入 `->` 函數返回註解；3.5 加入 `async/await` 。
- `async/await`：定義非同步函數與等待任務完成 (兩語言語義相似) 。

Ex2-2、Ex2-3 等 (線上編輯器、容器比較、map/filter/reduce)

- 線上 JS 編輯器方便，但不適合大型多檔案專案 (缺少 IDE 工具) 。
- Python list ~ JS array；Python dict ~ JS object；語法/內建函數略有差異。
- Filter/Map/Reduce 範例：Python 可用 list comprehension 或 `functools.reduce`；JS 用 `array.filter/map/reduce` 。

Ex3.0 - 檔案操作、CSV、JSON、with 好處

- 檔案基本三步驟：open (寫入)、write/print、close；推薦用 `with open(...) as f:` 自動 close。
- CSV 常見且有變種 (TSV、分號分隔等)；Pandas 讀取比內建 csv 更能自動推測型別。
- JSON 寫入用 `json.dump()`，讀用 `json.load()`；時間類型需先轉為字串或自訂編碼。

示範寫入文字檔：

```
with open('out.txt','w') as fp:  
    print("學號:912345678", file=fp)  
    print("座號:99", file=fp)  
    print("姓名:江俊廷", file=fp)
```

Ex3.2 / Ex3.3 (CSV/JSON 與 Pandas/型別注意)

- Pandas 讀 CSV 常能正確推斷欄位型別；內建 csv 會回傳字串。
- JSON dump 會把 Python None/True/False 轉為 null/true/false；key 會被雙引號包住。

範例 JSON dumps 結果：

```
'{"today": "9", "is": "7", "nice": 8, "day": [6, 5, 0.4, true, null]}'
```

Ex3.4 - ORM / SQLAlchemy / Django / CRUD

- ORM 定義：以物件操作資料庫（優：可讀性與防 injection；缺：效能）。
- SQLAlchemy 常用操作：`session.add()`、`session.query()`、`commit()`（commit 應盡量減少以提升效能）。
- Django 提供 `inspectdb` 轉換既有 DB 為 ORM model。
- 將 SQLAlchemy instance 轉 dict 範例：

```
def inst_to_dict(inst, delete_id=True):
    dat = {}
    for column in inst.__table__.columns:
        dat[column.name] = getattr(inst, column.name)
    if delete_id:
        dat.pop('id', None)
    return dat
```

ExA-1 (資料移動、JSON/CSV 與 DB)

- 常見資料存放：資料庫（SQL/NoSQL）與檔案（CSV/JSON）。
- CSV 輸出示例（由 stock_yahoo 物件）已在原文；可用 `with open(...,'w')` 寫入。

Ex3 範例與實作題 (檔案 I/O、CSV、json 實作)

- 範例程式包含：以 pandas/sklearn 取得 iris、用 seaborn 畫 pairplot、建立 iris CSV、用 json.dump/load、示範 open/with、seek/tell 等。

小結與建議

- 我已將每一題保留並濃縮成一行或幾行重點（並在必要處保留短程式片段）。
- 如果你要我輸出成 Word / PDF / 純文字（txt）或再更精簡成一頁 A4 的版本我可以轉檔。
- 若要「每題保留但更具體程式碼版本」也可指定題號，我會把該題展開成可執行範例檔案。