勞動部產業新尖兵計畫

人工智慧金融應用與實務培訓班

Capital 4 Plus

課程模組： AI 金融科技課程 - 網路爬蟲技術

2.Python 與 JSON

葉建華 (Yeh, Jian-hua)

tdi.jhyeh@tdi.edu.tw
au4290@gmail.com

講次內容

- 認識 JSON 格式
- JSON 字串的處理
- JSON 的檔案輸出入

JSON 介紹

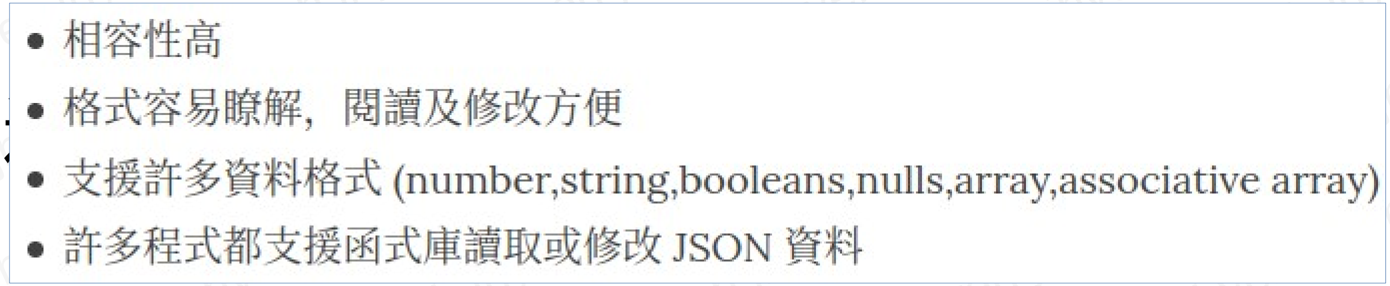
- JSON, JavaScript Object Notation
  - 就是 JavaScript 物件表示法
  - 經由 Douglas Crockford 推廣後普及化
  - 雖是以 JavaScript 語法為基礎，但可獨立使用

JSON 格式

- 以純文字為基底
  - 儲存和傳送簡單結構資料
  - 特定的格式，可儲存任何資料 (字串 , 數字 , 陣列 , 物件)
  - 也可透過物件或陣列來傳送較複雜的資料

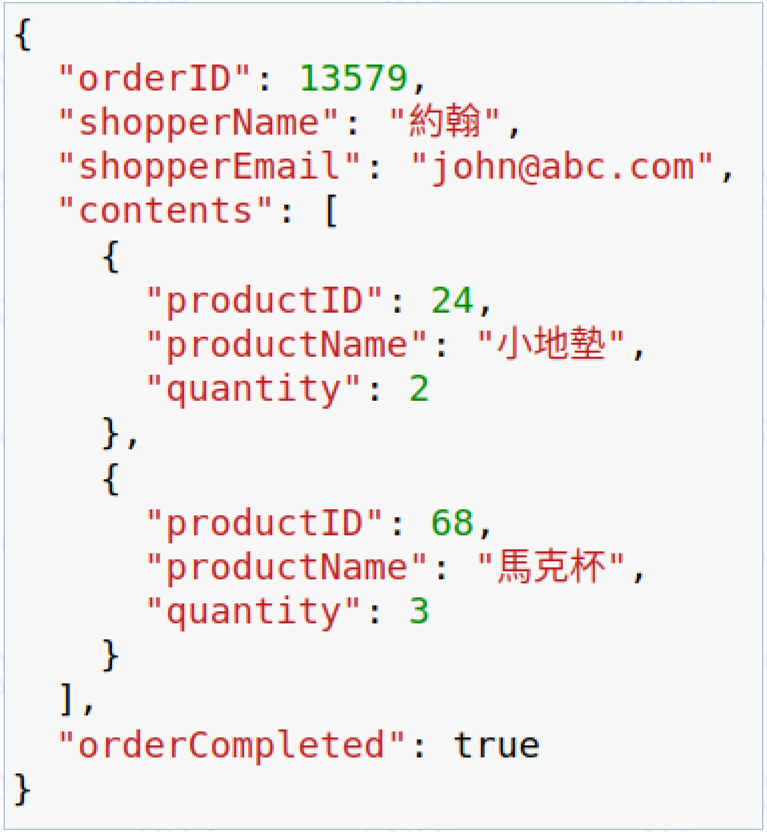
JSON 格式

- 以純文字為基底
    - 儲存和傳送簡單結構資料
    - 特定的格式，可儲存任何資料（字串，數字，陣列，物件）
    - 也可透



- 相容性高
- 格式容易瞭解，閱讀及修改方便
- 支援許多資料格式 (number,string,booleans,nulls,array,associative array)
- 許多程式都支援函式庫讀取或修改 JSON 資料

JSON 範例

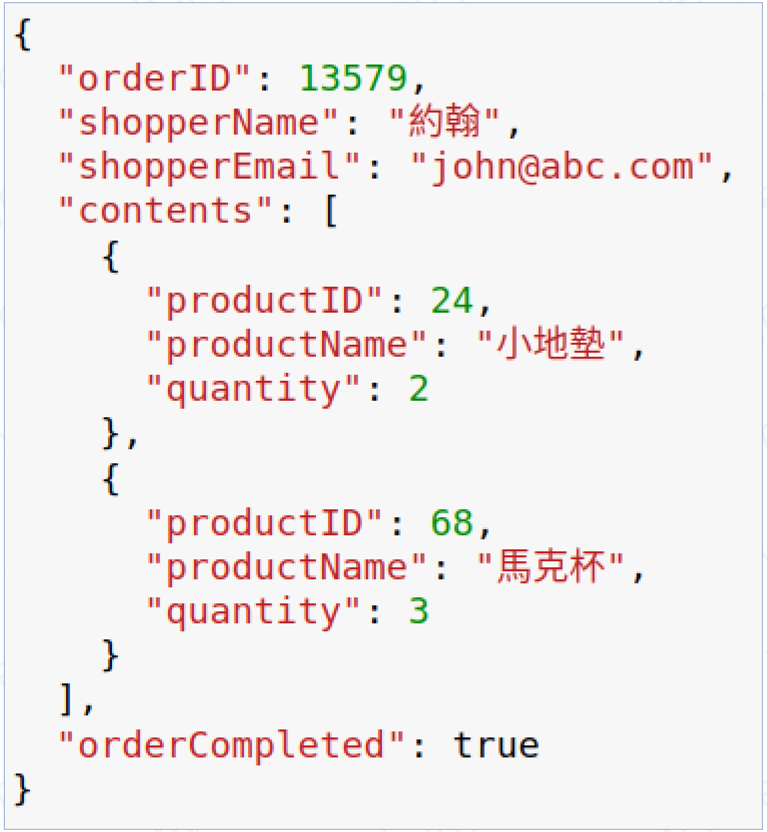


```
{
  "orderID": 13579,
  "shopperName": "約翰",
  "shopperEmail": "john@abc.com",
  "contents": [
    {
      "productID": 24,
      "productName": "小地墊",
      "quantity": 2
    },
    {
      "productID": 68,
      "productName": "馬克杯",
      "quantity": 3
    }
  ],
  "orderCompleted": true
}
```

JSON 範例

- JSON 類似 Python 的資料型別
  - Key-value pair
  - 物件 { } 表示 dict

    陣列 [ ] 表示 list



```
{
  "orderID": 13579,
  "shopperName": "約翰",
  "shopperEmail": "john@abc.com",
  "contents": [
    {
      "productID": 24,
      "productName": "小地墊",
      "quantity": 2
    },
    {
      "productID": 68,
      "productName": "馬克杯",
      "quantity": 3
    }
  ],
  "orderCompleted": true
}
```

JSON 的應用

- 做為資料交換用的 API
  - Bitcoin, Youtube, Facebook, ...
- 做為系統或程式的設定檔 (config)
  - Linux, Windows, OpenOffice, ...

講次內容

- 認識 JSON 格式
- JSON 字串的處理
- JSON 的檔案輸出入

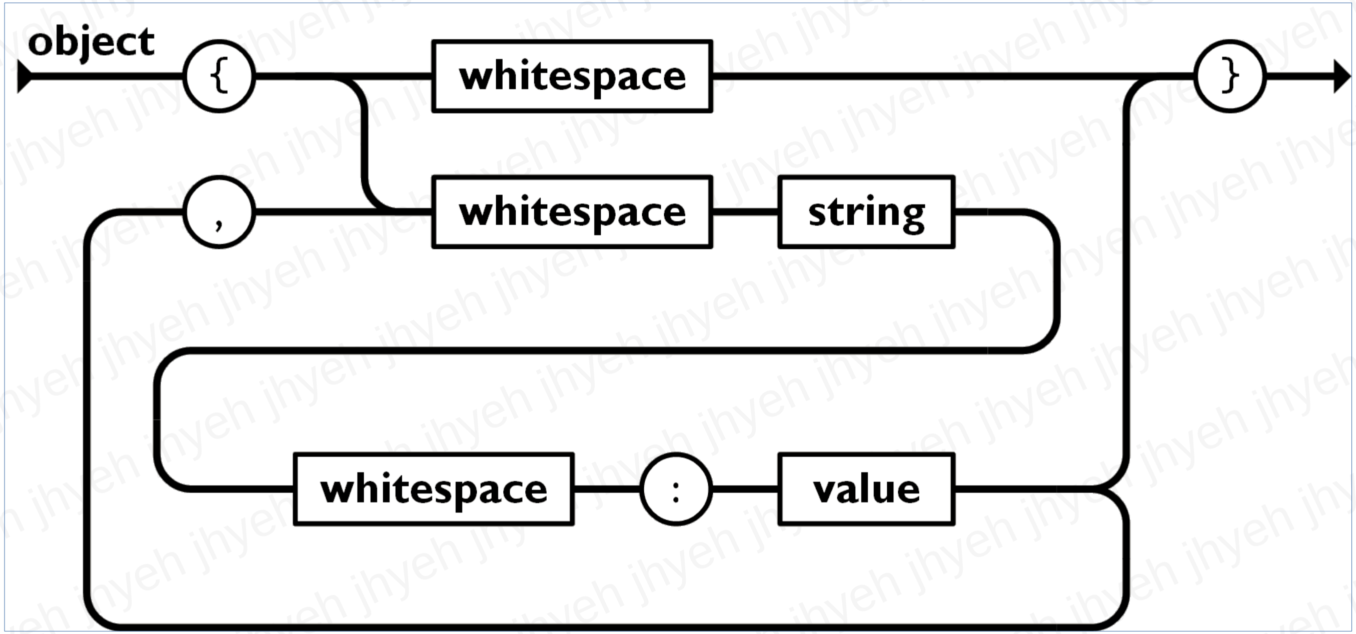
JSON 中的資料類型

- 字串 String
- 數字 Number
- 布林 Boolean
  - 小寫 true, false

- 物件 Object
- 陣列 Array
- 空值
  - 小寫 null

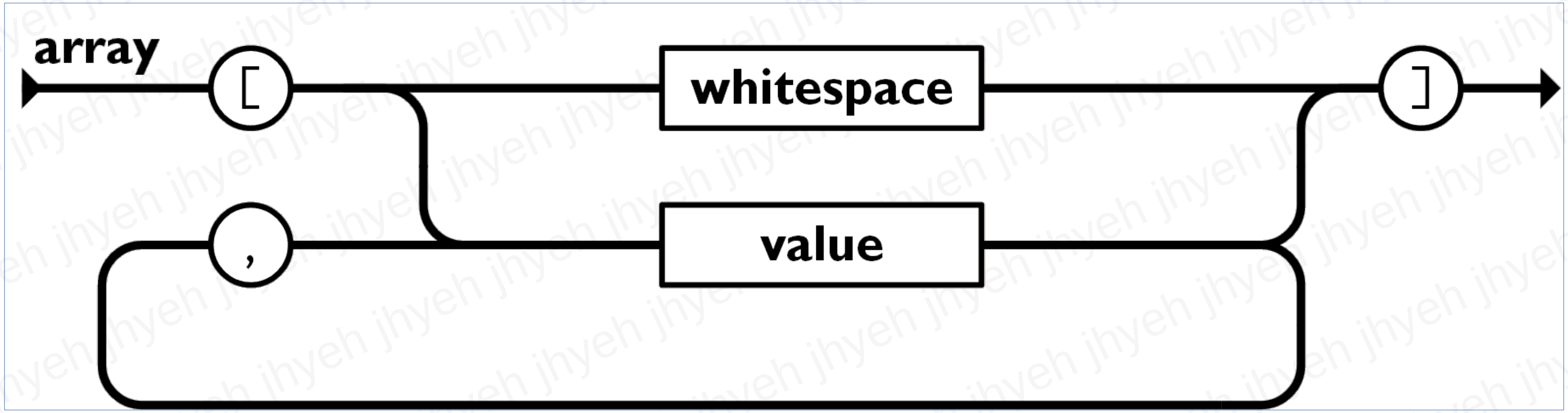
JSON Object

- 類似 Python dict，但限制鍵值是字串

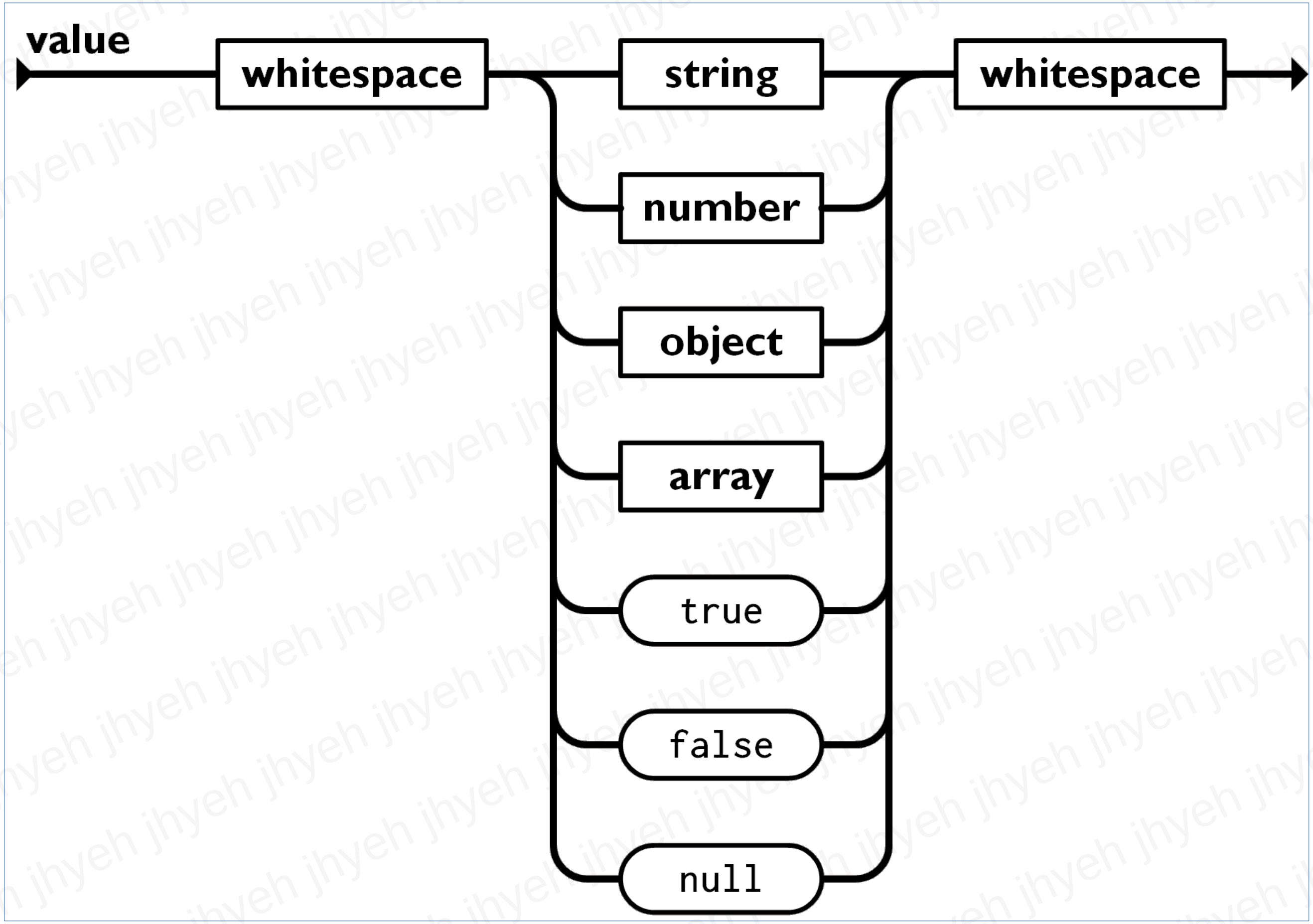


JSON Array

- 類似 Python list，可裝任意資料



各式資料


value
whitespace
string
number
object
array
true
false
null
whitespace

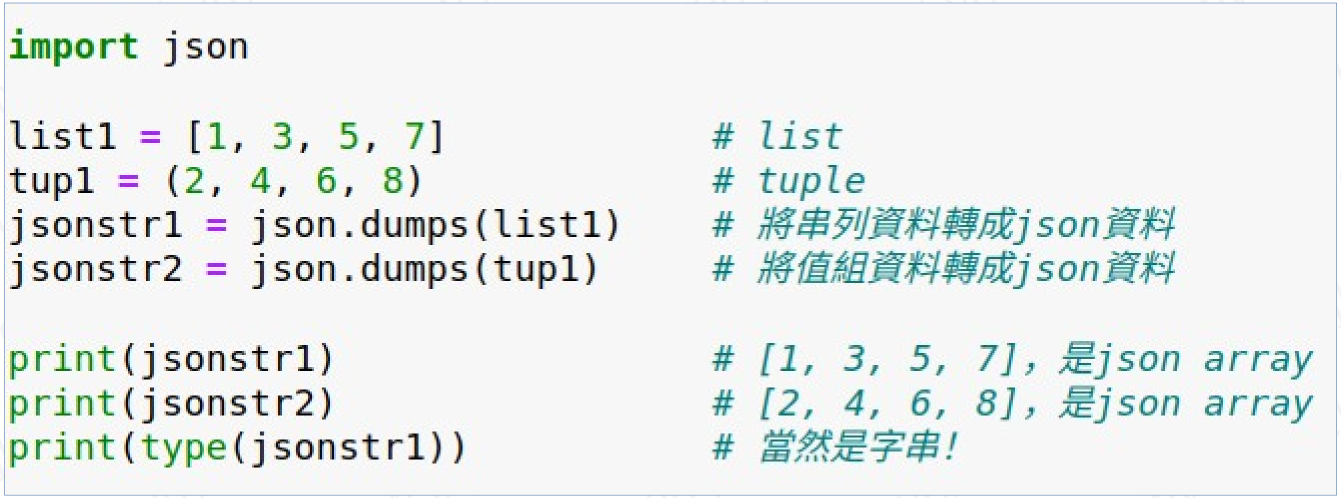
Python 資料轉 JSON

Python	JSON
dict	object
list, tuple	array
str, unicode	string
int, float, long(Python3 取消)	number
True, False	true, false
None	null

注意大小寫！

Python 資料轉 JSON

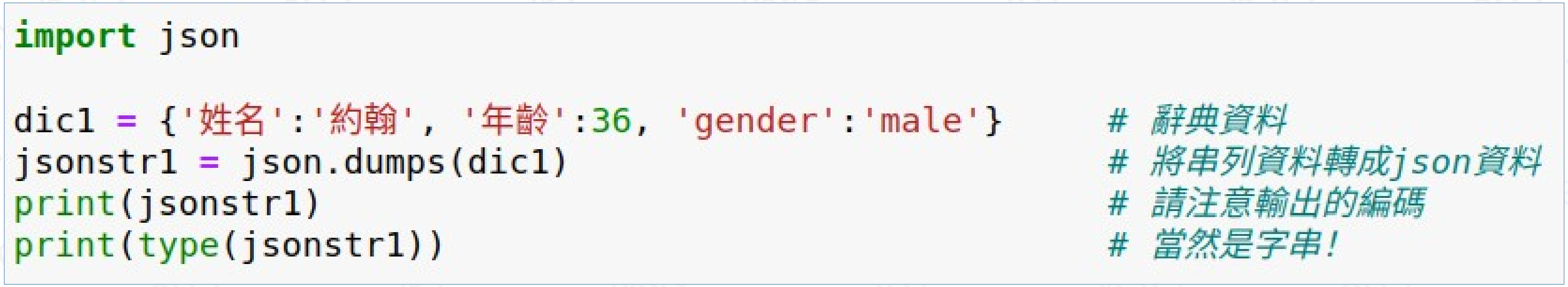
- json 模組， dumps() 函數
  - 直接將 Python 資料轉成 JSON 字串



```
import json

list1 = [1, 3, 5, 7]            # list
tupl = (2, 4, 6, 8)             # tuple
jsonstr1 = json.dumps(list1)    # 將串列資料轉成json資料
jsonstr2 = json.dumps(tupl)     # 將值組資料轉成json資料

print(jsonstr1)                 # [1, 3, 5, 7], 是json array
print(jsonstr2)                 # [2, 4, 6, 8], 是json array
print(type(jsonstr1))           # 當然是字串!
```

Python 資料轉 JSON

- 用 dict 來舉例，使用 dumps() 函數

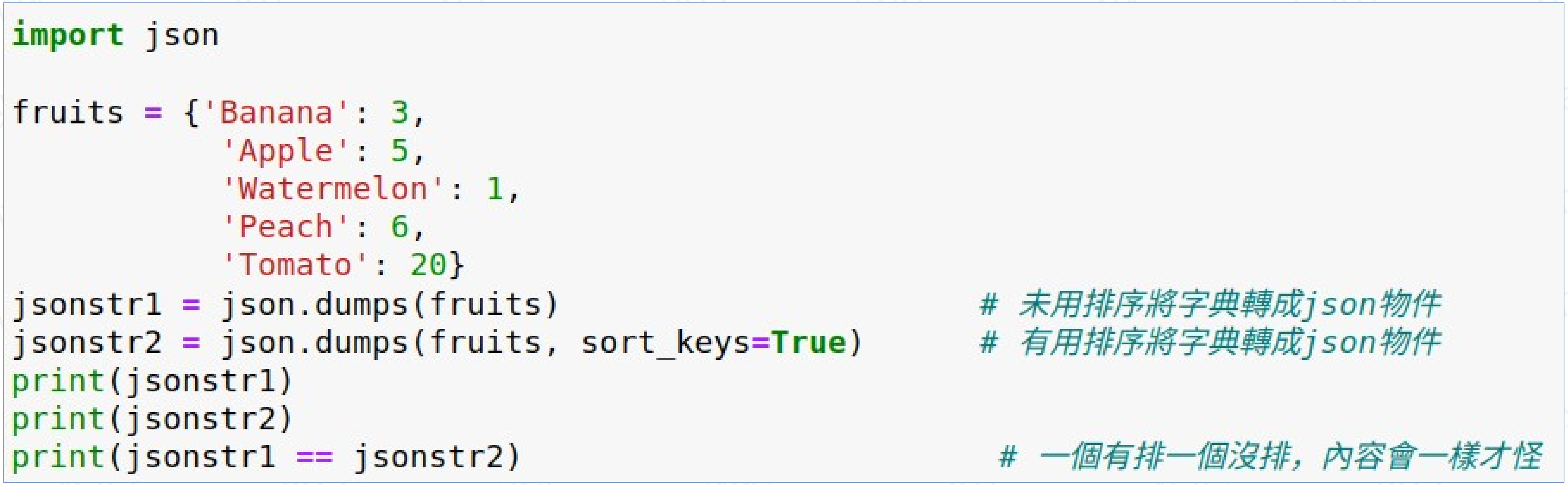


```
import json

dic1 = {'姓名':'約翰', '年齡':36, 'gender':'male'}     # 辭典資料
jsonstr1 = json.dumps(dic1)                          # 將串列資料轉成json資料
print(jsonstr1)                                      # 請注意輸出的編碼
print(type(jsonstr1))                                # 當然是字串!
```

請注意，輸出結果是 json object 哦!

Python 資料轉 JSON

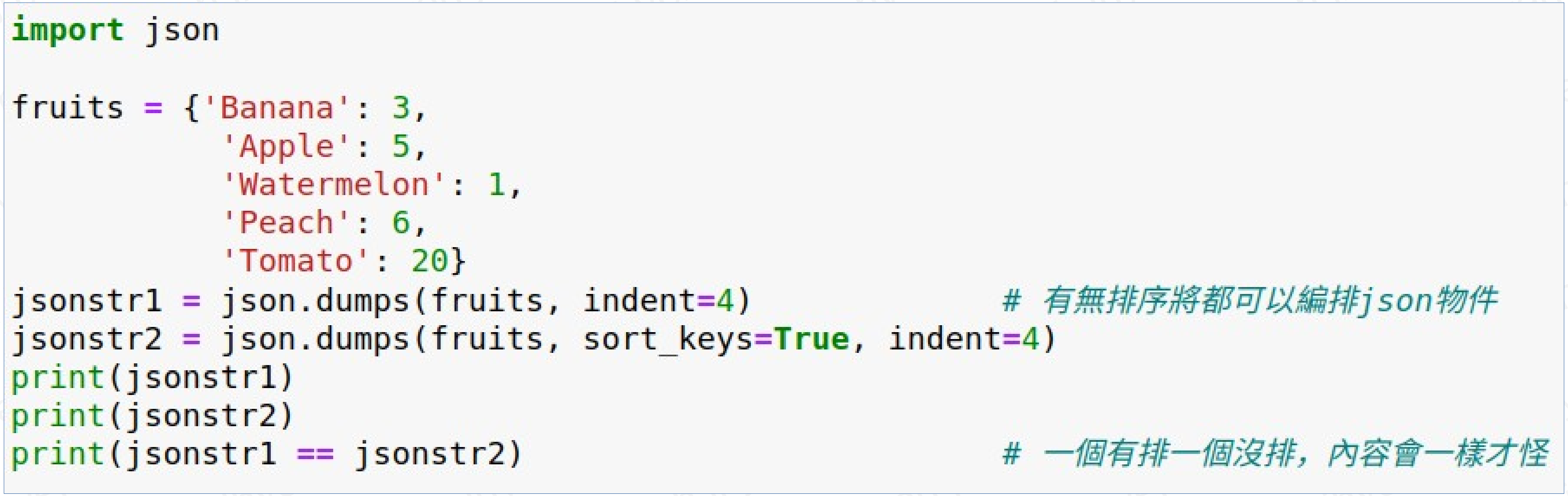
- dict 可以透過 dumps() 來對鍵值排序
  - 設定 sort_keys 參數



```
import json

fruits = {'Banana': 3,
          'Apple': 5,
          'Watermelon': 1,
          'Peach': 6,
          'Tomato': 20}
jsonstr1 = json.dumps(fruits)                       # 未用排序將字典轉成json物件
jsonstr2 = json.dumps(fruits, sort_keys=True)       # 有用排序將字典轉成json物件
print(jsonstr1)
print(jsonstr2)
print(jsonstr1 == jsonstr2)                          # 一個有排一個沒排，內容會一樣才怪
```

Python 資料轉 JSON

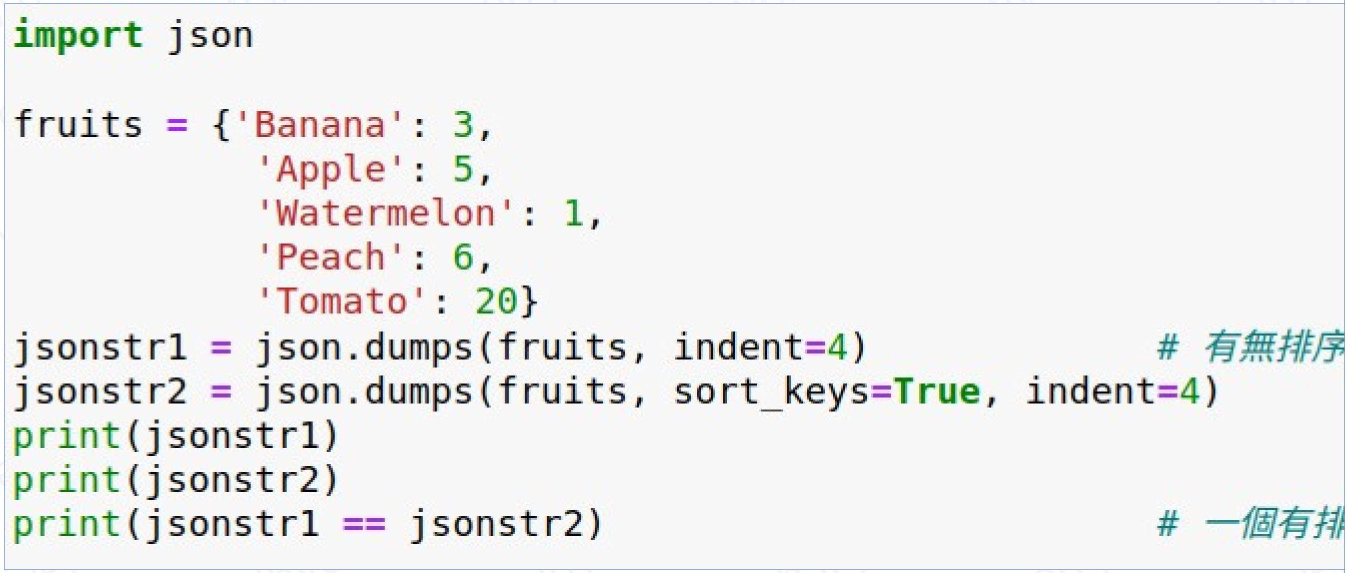
- dict 可以透過 dumps() 來對內容做對齊編排
  - 設定 indent 參數



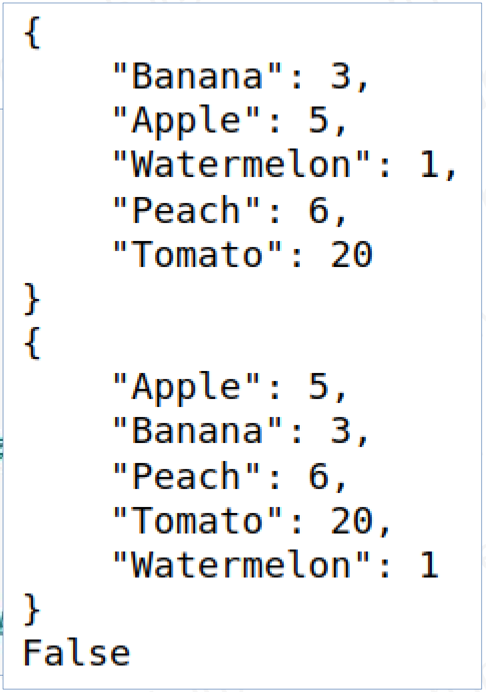
```
import json

fruits = {'Banana': 3,
          'Apple': 5,
          'Watermelon': 1,
          'Peach': 6,
          'Tomato': 20}
jsonstr1 = json.dumps(fruits, indent=4)                # 有無排序將都可以編排json物件
jsonstr2 = json.dumps(fruits, sort_keys=True, indent=4)
print(jsonstr1)
print(jsonstr2)
print(jsonstr1 == jsonstr2)                            # 一個有排一個沒排，內容會一樣才怪
```

Python 資料轉 JSON

- dict 可以透過 dumps() 來對內容做對齊編排
  - 設定 indent 參數



```
import json

fruits = {'Banana': 3,
          'Apple': 5,
          'Watermelon': 1,
          'Peach': 6,
          'Tomato': 20}
jsonstr1 = json.dumps(fruits, indent=4)              # 有無排序
jsonstr2 = json.dumps(fruits, sort_keys=True, indent=4)
print(jsonstr1)
print(jsonstr2)
print(jsonstr1 == jsonstr2)                          # 一個有排
```



```
{
    "Banana": 3,
    "Apple": 5,
    "Watermelon": 1,
    "Peach": 6,
    "Tomato": 20
}
{
    "Apple": 5,
    "Banana": 3,
    "Peach": 6,
    "Tomato": 20,
    "Watermelon": 1
}
False
```

JSON 轉 Python 資料

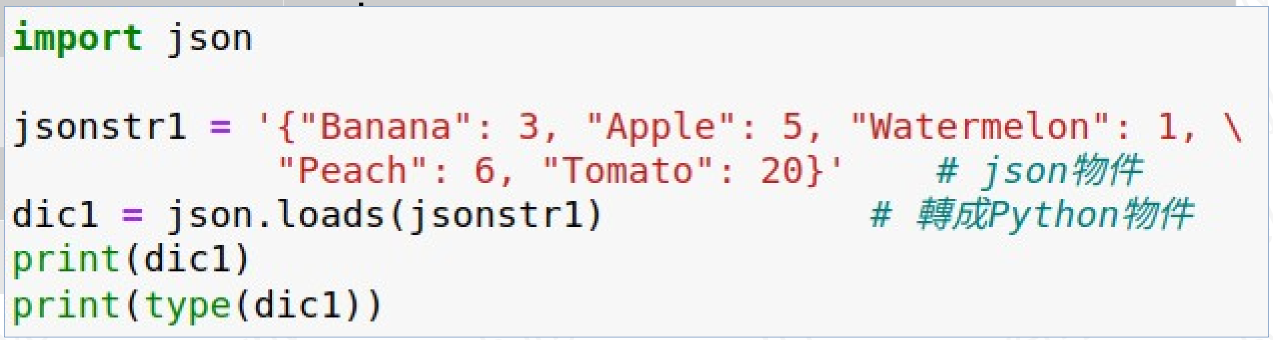
- JSON 字串可以用 loads() 轉入 Python
  - 要記得如何對應的

Python	JSON
dict	object
list, tuple	array
str, unicode	string
int, float, long(Python3 取消)	number
True, False	true, false
None	null

JSON 轉 Python 資料

- JSON 字串可以用 loads() 轉入 Python
  - 要記得如何對應的

Python	JSON
dict	object
list, tuple	array
str, unicode	[illegible]
int, float, long(Python3 取消)	
True, False	
None	



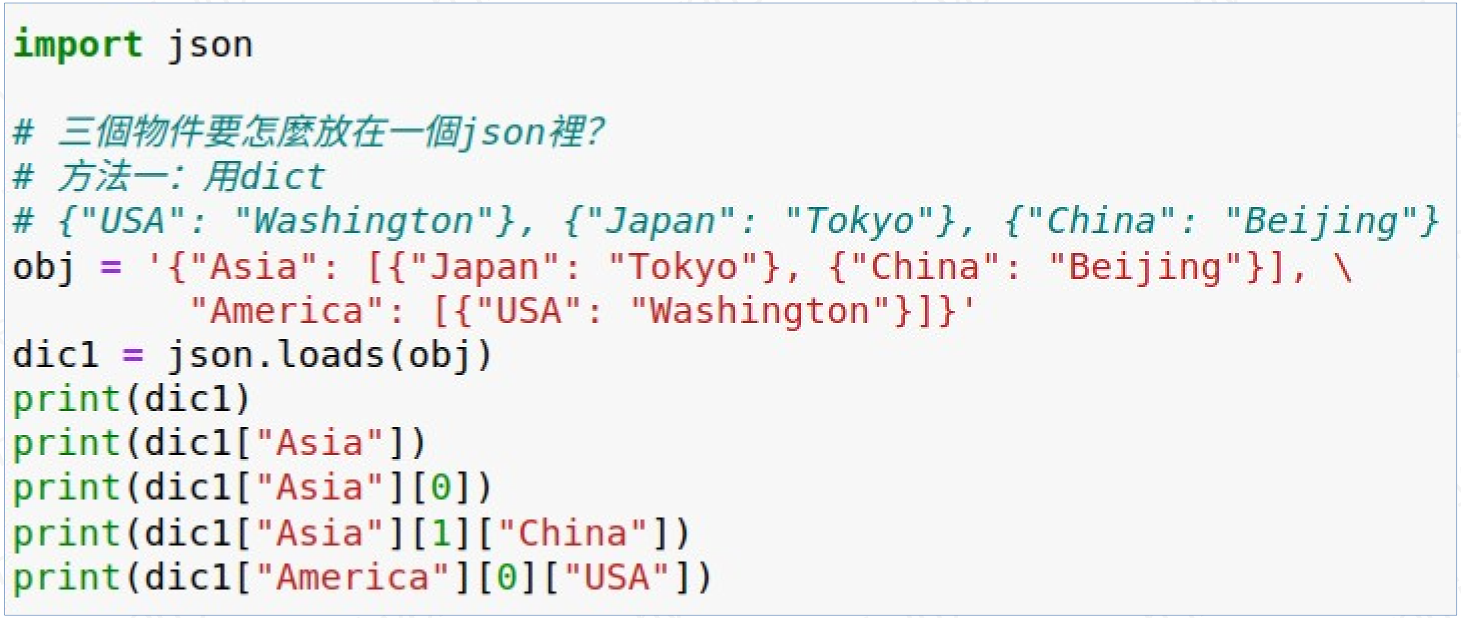
```
import json

jsonstr1 = '{"Banana": 3, "Apple": 5, "Watermelon": 1, \
            "Peach": 6, "Tomato": 20}'    # json物件
dic1 = json.loads(jsonstr1)                # 轉成Python物件
print(dic1)
print(type(dic1))
```

一個文件只能有一個物件

- JSON 文件就是這個特性
- 如果想多個，就製作巢狀 JSON 物件
  - 使用 object 包住多個物件
  - 使用 array 包住多個物件

巢狀物件：使用 object

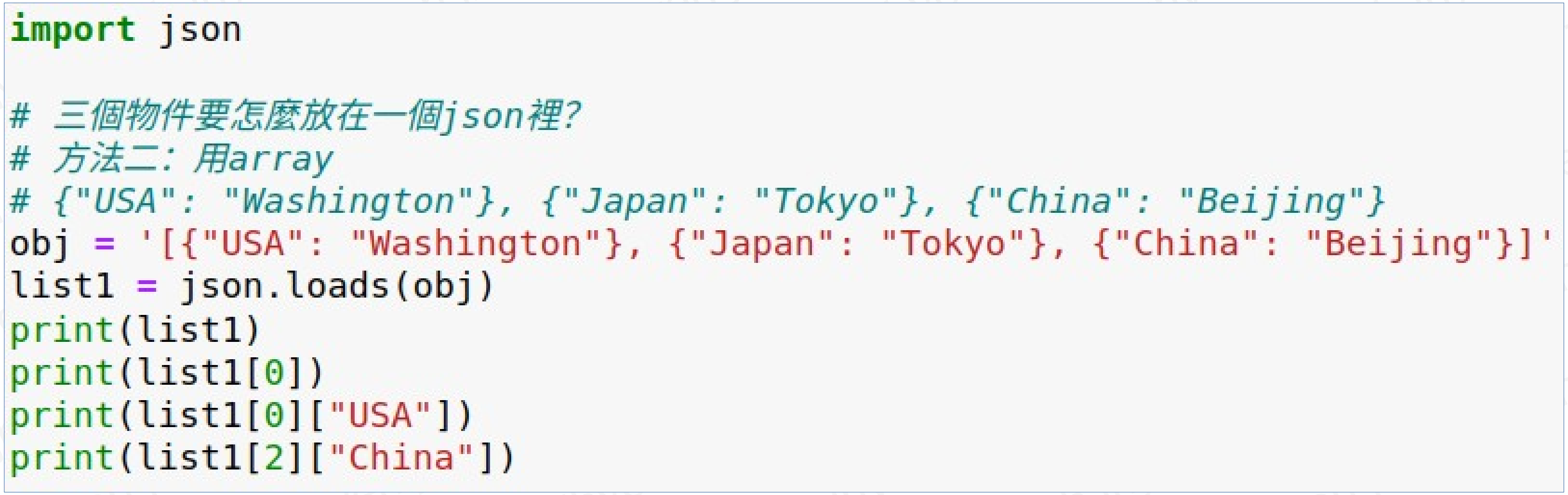
- 用 JSON object 來包住其他物件
  - 不保證有序，要使用鍵值



```
import json

# 三個物件要怎麼放在一個json裡?
# 方法一: 用dict
# {"USA": "Washington"}, {"Japan": "Tokyo"}, {"China": "Beijing"}
obj = '{"Asia": [{"Japan": "Tokyo"}, {"China": "Beijing"}], \
        "America": [{"USA": "Washington"}]}'
dic1 = json.loads(obj)
print(dic1)
print(dic1["Asia"])
print(dic1["Asia"][0])
print(dic1["Asia"][1]["China"])
print(dic1["America"][0]["USA"])
```

巢狀物件：使用 array

- 用 JSON array 來包住其他物件
  - 有序，可以用索引值存取



```
import json

# 三個物件要怎麼放在一個json裡?
# 方法二: 用array
# {"USA": "Washington"}, {"Japan": "Tokyo"}, {"China": "Beijing"}
obj = '[{"USA": "Washington"}, {"Japan": "Tokyo"}, {"China": "Beijing"}]'
list1 = json.loads(obj)
print(list1)
print(list1[0])
print(list1[0]["USA"])
print(list1[2]["China"])
```

講次內容

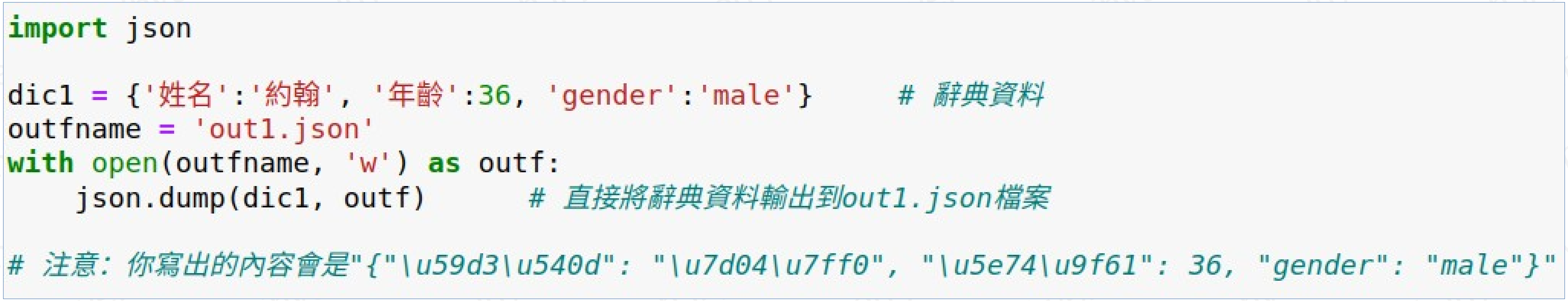
- 認識 JSON 格式
- JSON 字串的處理
- JSON 的檔案輸出入

JSON 內容與檔案

- dumps()、 loads() 是針對 JSON 字串
- dump()、 load() 則是針對 JSON 格式的檔案

Python 資料轉 JSON 檔案

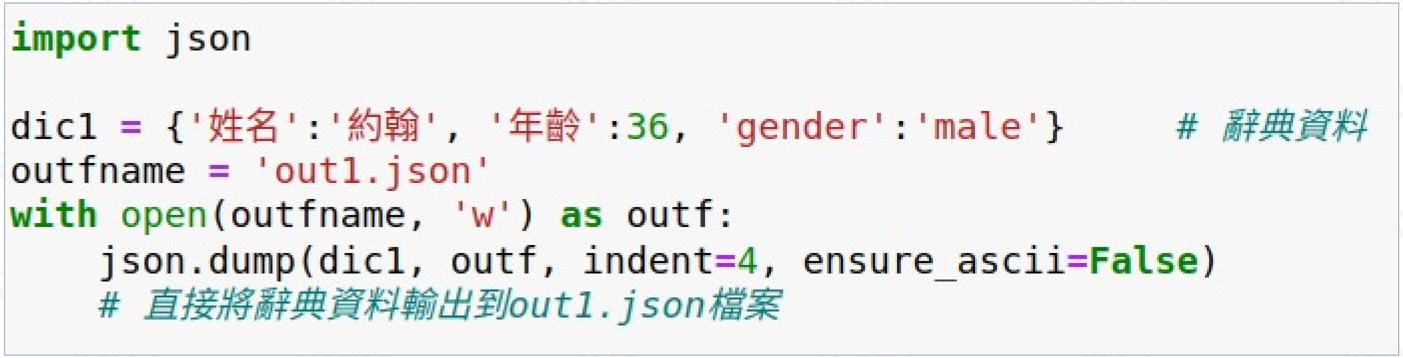
- dump()
  - 可指定檔名參數



```
import json

dic1 = {'姓名':'約翰', '年齡':36, 'gender':'male'}     # 辭典資料
outfname = 'out1.json'
with open(outfname, 'w') as outf:
    json.dump(dic1, outf)      # 直接將辭典資料輸出到out1.json檔案

# 注意: 你寫出的內容會是"{"\u59d3\u540d": "\u7d04\u7ff0", "\u5e74\u9f61": 36, "gender": "male"}"
```

Python 資料轉 JSON 檔案

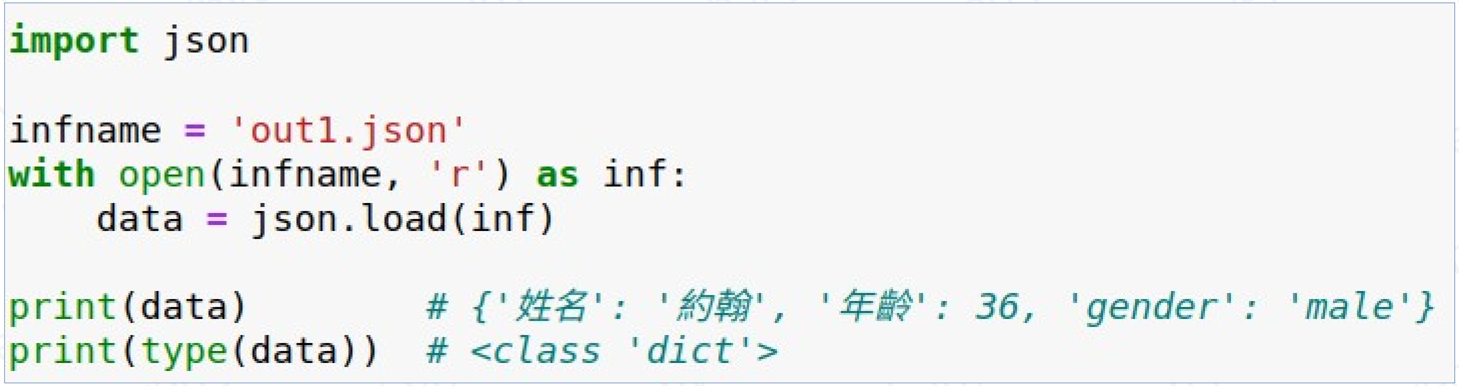
- dump()
  - 還可以設定 indent 參數來對齊編排
  - 設定 ensure_ascii 可以強制輸出 utf-8 編碼



```python
import json

dic1 = {'姓名':'約翰', '年齡':36, 'gender':'male'}     # 辭典資料
outfname = 'out1.json'
with open(outfname, 'w') as outf:
    json.dump(dic1, outf, indent=4, ensure_ascii=False)
    # 直接將辭典資料輸出到out1.json檔案
```

讀取 JSON 檔案

- load()
  - 可指定檔名參數



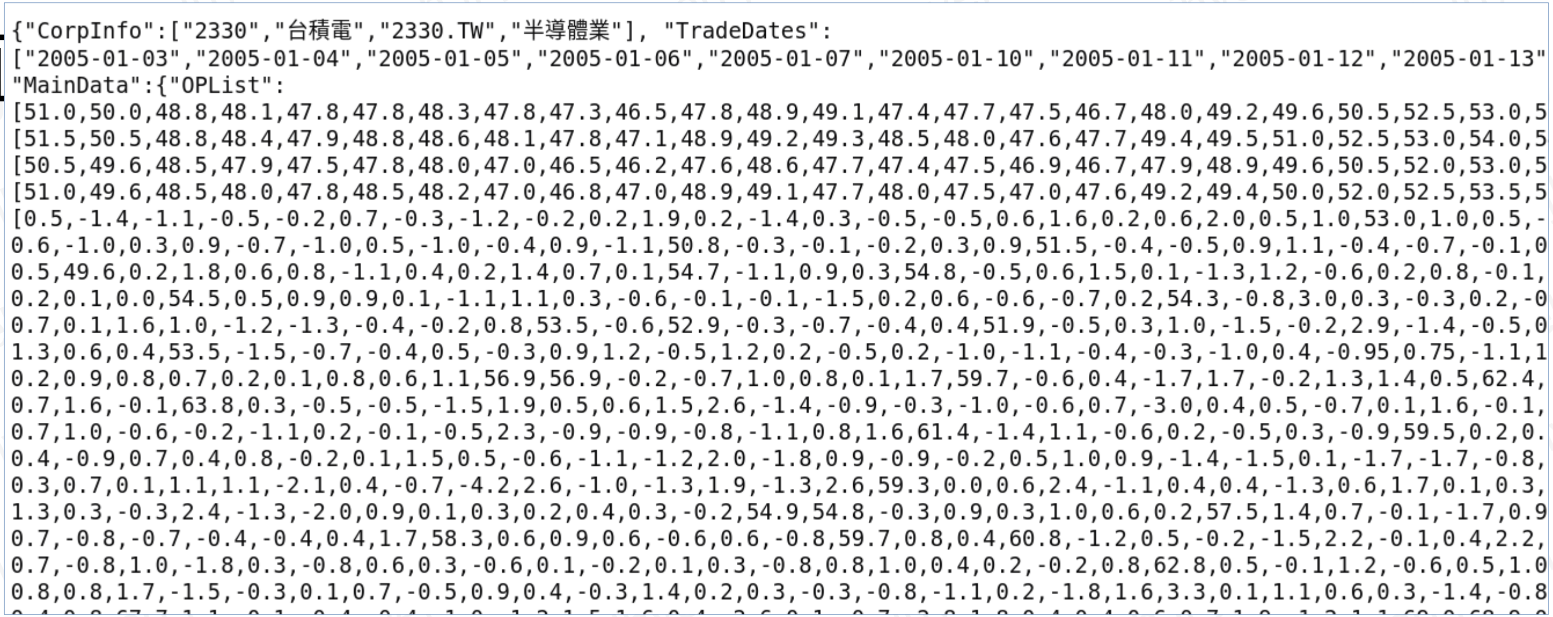
```
import json

infname = 'out1.json'
with open(infname, 'r') as inf:
    data = json.load(inf)

print(data)         # {'姓名': '約翰', '年齡': 36, 'gender': 'male'}
print(type(data))   # <class 'dict'>
```

練習：讀取「 2330 」檔案

- 2330.json 是什麼?
- 輸出鍵值來看看
- 怎麼做?

練習：讀取「2330」檔案

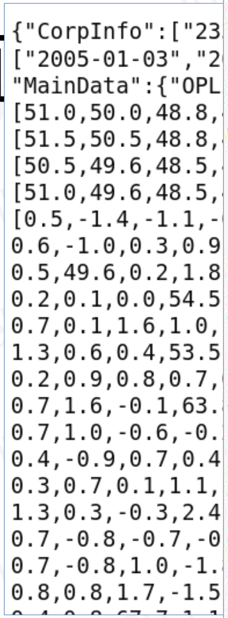
- 2330.json 是什麼?
- 輸出鍵值
- 怎麼做?



```
{"CorpInfo":["2330","台積電","2330.TW","半導體業"], "TradeDates":
["2005-01-03","2005-01-04","2005-01-05","2005-01-06","2005-01-07","2005-01-10","2005-01-11","2005-01-12","2005-01-13"
"MainData":{"OPList":
[51.0,50.0,48.8,48.1,47.8,47.8,48.3,47.8,47.3,46.5,47.8,48.9,49.1,47.4,47.7,47.5,46.7,48.0,49.2,49.6,50.5,52.5,53.0,5
[51.5,50.5,48.8,48.4,47.9,48.8,48.6,48.1,47.8,47.1,48.9,49.2,49.3,48.5,48.0,47.6,47.7,49.4,49.5,51.0,52.5,53.0,54.0,5
[50.5,49.6,48.5,47.9,47.5,47.8,48.0,47.0,46.5,46.2,47.6,48.6,47.7,47.4,47.5,46.9,46.7,47.9,48.9,49.6,50.5,52.0,53.0,5
[51.0,49.6,48.5,48.0,47.8,48.5,48.2,47.0,46.8,47.0,48.9,49.1,47.7,48.0,47.5,47.0,47.6,49.2,49.4,50.0,52.0,52.5,53.5,5
[0.5,-1.4,-1.1,-0.5,-0.2,0.7,-0.3,-1.2,-0.2,0.2,1.9,0.2,-1.4,0.3,-0.5,-0.5,0.6,1.6,0.2,0.6,2.0,0.5,1.0,53.0,1.0,0.5,-
0.6,-1.0,0.3,0.9,-0.7,-1.0,0.5,-1.0,-0.4,0.9,-1.1,50.8,-0.3,-0.1,-0.2,0.3,0.9,51.5,-0.4,-0.5,0.9,1.1,-0.4,-0.7,-0.1,0
0.5,49.6,0.2,1.8,0.6,0.8,-1.1,0.4,0.2,1.4,0.7,0.1,54.7,-1.1,0.9,0.3,54.8,-0.5,0.6,1.5,0.1,-1.3,1.2,-0.6,0.2,0.8,-0.1,
0.2,0.1,0.0,54.5,0.5,0.9,0.9,0.1,-1.1,1.1,0.3,-0.6,-0.1,-0.1,-1.5,0.2,0.6,-0.6,-0.7,0.2,54.3,-0.8,3.0,0.3,-0.3,0.2,-0
0.7,0.1,1.6,1.0,-1.2,-1.3,-0.4,-0.2,0.8,53.5,-0.6,52.9,-0.3,-0.7,-0.4,0.4,51.9,-0.5,0.3,1.0,-1.5,-0.2,2.9,-1.4,-0.5,0
1.3,0.6,0.4,53.5,-1.5,-0.7,-0.4,0.5,-0.3,0.9,1.2,-0.5,1.2,0.2,-0.5,0.2,-1.0,-1.1,-0.4,-0.3,-1.0,0.4,-0.95,0.75,-1.1,1
0.2,0.9,0.8,0.7,0.2,0.1,0.8,0.6,1.1,56.9,56.9,-0.2,-0.7,1.0,0.8,0.1,1.7,59.7,-0.6,0.4,-1.7,1.7,-0.2,1.3,1.4,0.5,62.4,
0.7,1.6,-0.1,63.8,0.3,-0.5,-0.5,-1.5,1.9,0.5,0.6,1.5,2.6,-1.4,-0.9,-0.3,-1.0,-0.6,0.7,-3.0,0.4,0.5,-0.7,0.1,1.6,-0.1,
0.7,1.0,-0.6,-0.2,-1.1,0.2,-0.1,-0.5,2.3,-0.9,-0.9,-0.8,-1.1,0.8,1.6,61.4,-1.4,1.1,-0.6,0.2,-0.5,0.3,-0.9,59.5,0.2,0.
0.4,-0.9,0.7,0.4,0.8,-0.2,0.1,1.5,0.5,-0.6,-1.1,-1.2,2.0,-1.8,0.9,-0.9,-0.2,0.5,1.0,0.9,-1.4,-1.5,0.1,-1.7,-1.7,-0.8,
0.3,0.7,0.1,1.1,1.1,-2.1,0.4,-0.7,-4.2,2.6,-1.0,-1.3,1.9,-1.3,2.6,59.3,0.0,0.6,2.4,-1.1,0.4,0.4,-1.3,0.6,1.7,0.1,0.3,
1.3,0.3,-0.3,2.4,-1.3,-2.0,0.9,0.1,0.3,0.2,0.4,0.3,-0.2,54.9,54.8,-0.3,0.9,0.3,1.0,0.6,0.2,57.5,1.4,0.7,-0.1,-1.7,0.9
0.7,-0.8,-0.7,-0.4,-0.4,0.4,1.7,58.3,0.6,0.9,0.6,-0.6,0.6,-0.8,59.7,0.8,0.4,60.8,-1.2,0.5,-0.2,-1.5,2.2,-0.1,0.4,2.2,
0.7,-0.8,1.0,-1.8,0.3,-0.8,0.6,0.3,-0.6,0.1,-0.2,0.1,0.3,-0.8,0.8,1.0,0.4,0.2,-0.2,0.8,62.8,0.5,-0.1,1.2,-0.6,0.5,1.0
0.8,0.8,1.7,-1.5,-0.3,0.1,0.7,-0.5,0.9,0.4,-0.3,1.4,0.2,0.3,-0.3,-0.8,-1.1,0.2,-1.8,1.6,3.3,0.1,1.1,0.6,0.3,-1.4,-0.8
```

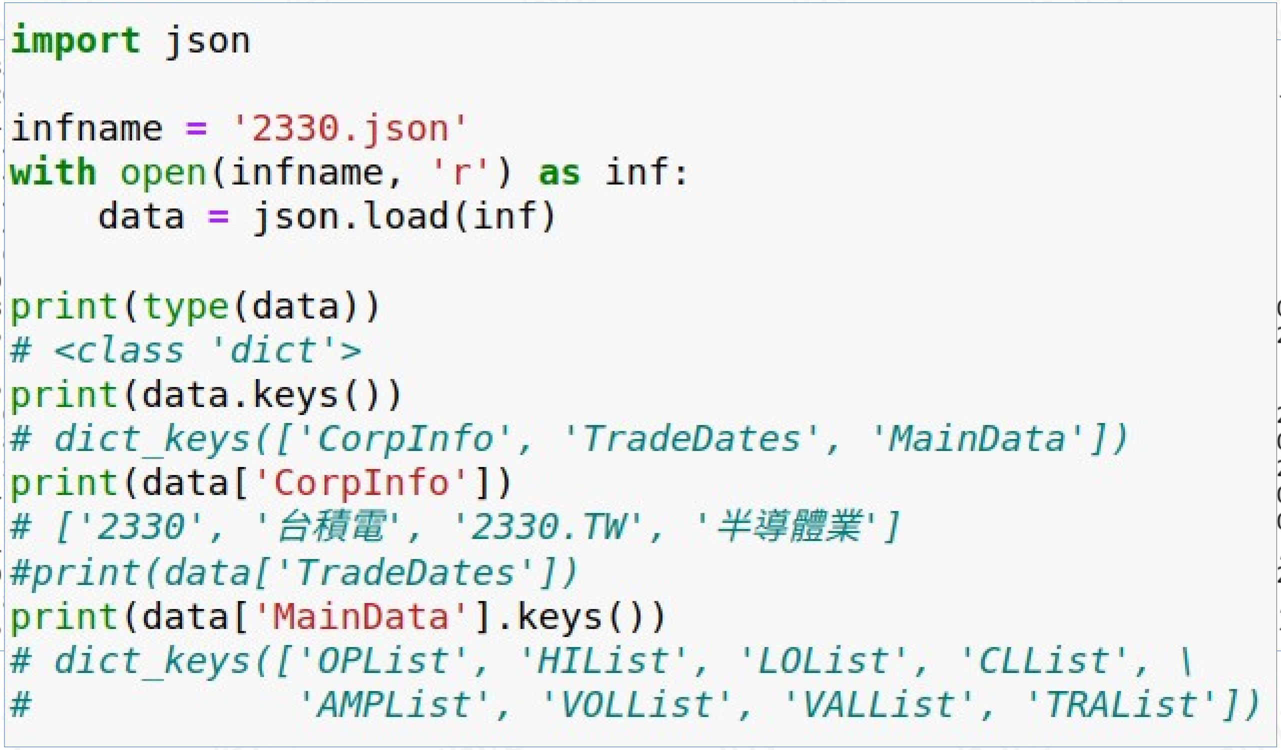
練習：讀取「2330」檔案

- 2330.json 是什麼?
- 輸出鍵值
- 怎麼做?



```
{"CorpInfo":["23
["2005-01-03","2
"MainData":{"OPL
[51.0,50.0,48.8,
[51.5,50.5,48.8,
[50.5,49.6,48.5,
[51.0,49.6,48.5,
[0.5,-1.4,-1.1,-
0.6,-1.0,0.3,0.9
0.5,49.6,0.2,1.8
0.2,0.1,0.0,54.5
0.7,0.1,1.6,1.0,
1.3,0.6,0.4,53.5
0.2,0.9,0.8,0.7,
0.7,1.6,-0.1,63.
0.7,1.0,-0.6,-0.
0.4,-0.9,0.7,0.4
0.3,0.7,0.1,1.1,
1.3,0.3,-0.3,2.4
0.7,-0.8,-0.7,-0
0.7,-0.8,1.0,-1.
0.8,0.8,1.7,-1.5
```

```

-13"

.0,5
.0,5
.0,5
.5,5
.5,-
.1,0
0.1,
2,-0
.5,0
.1,1
2.4,
0.1,
2,0.
0.8,
0.3,
,0.9
2.2,
,1.0
-0.8
```



```python
import json

infname = '2330.json'
with open(infname, 'r') as inf:
    data = json.load(inf)

print(type(data))
# <class 'dict'>
print(data.keys())
# dict_keys(['CorpInfo', 'TradeDates', 'MainData'])
print(data['CorpInfo'])
# ['2330', '台積電', '2330.TW', '半導體業']
#print(data['TradeDates'])
print(data['MainData'].keys())
# dict_keys(['OPList', 'HIList', 'LOList', 'CLList', \
#            'AMPList', 'VOLList', 'VALList', 'TRAList'])
```

這個講次中，你應該學到了 ...

- 了解 JSON 資料格式
- Python 與 JSON 字串之間的轉換
- Python 與 JSON 檔案之間的轉換