

自然語言處理

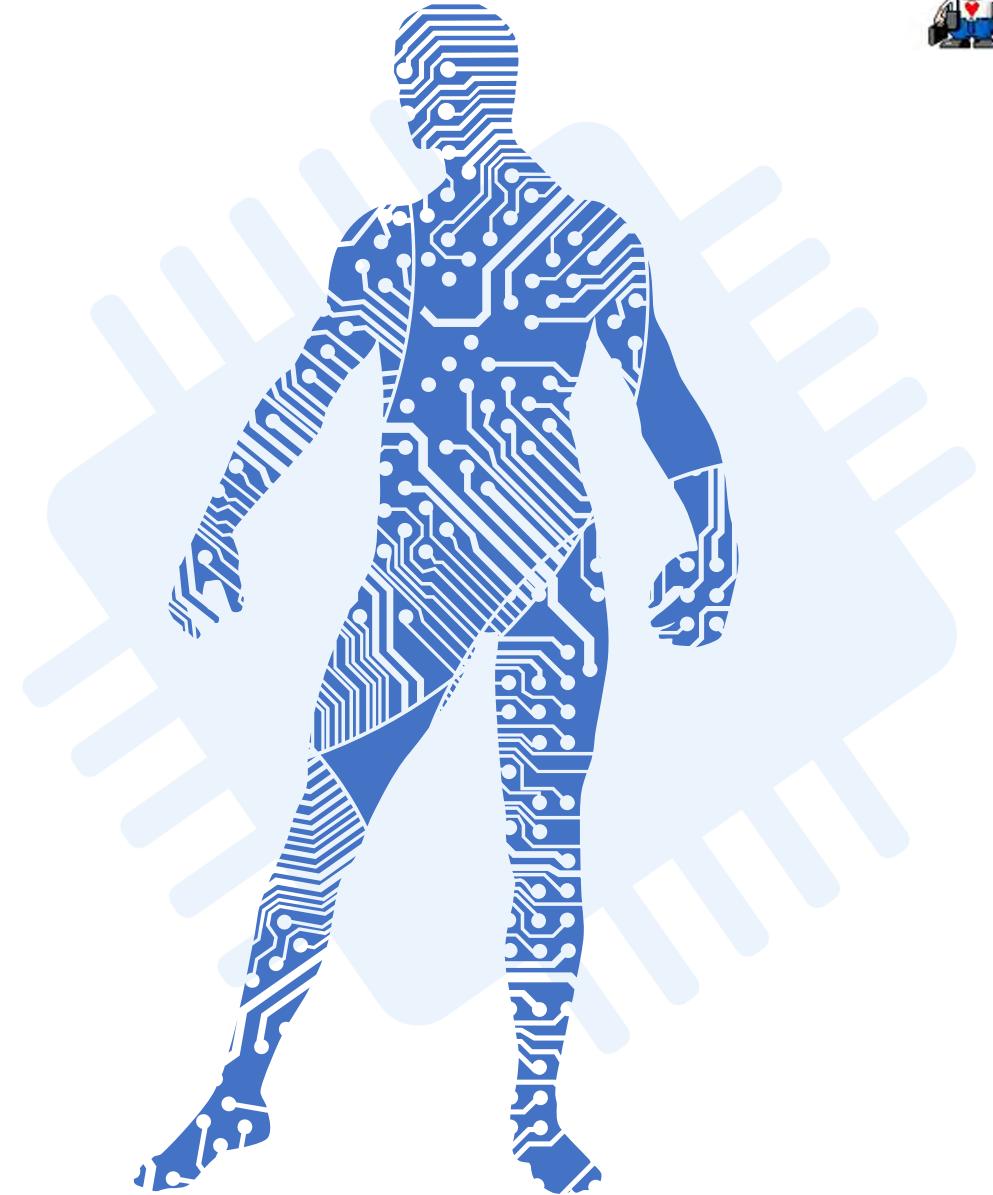
第 6 章 資料取得

講師：紀俊男



本章大綱

- 資料來源簡介
- 表格讀取
- 資料庫讀取
- 網頁讀取
- 各式檔案讀取
- 多媒體文字讀取

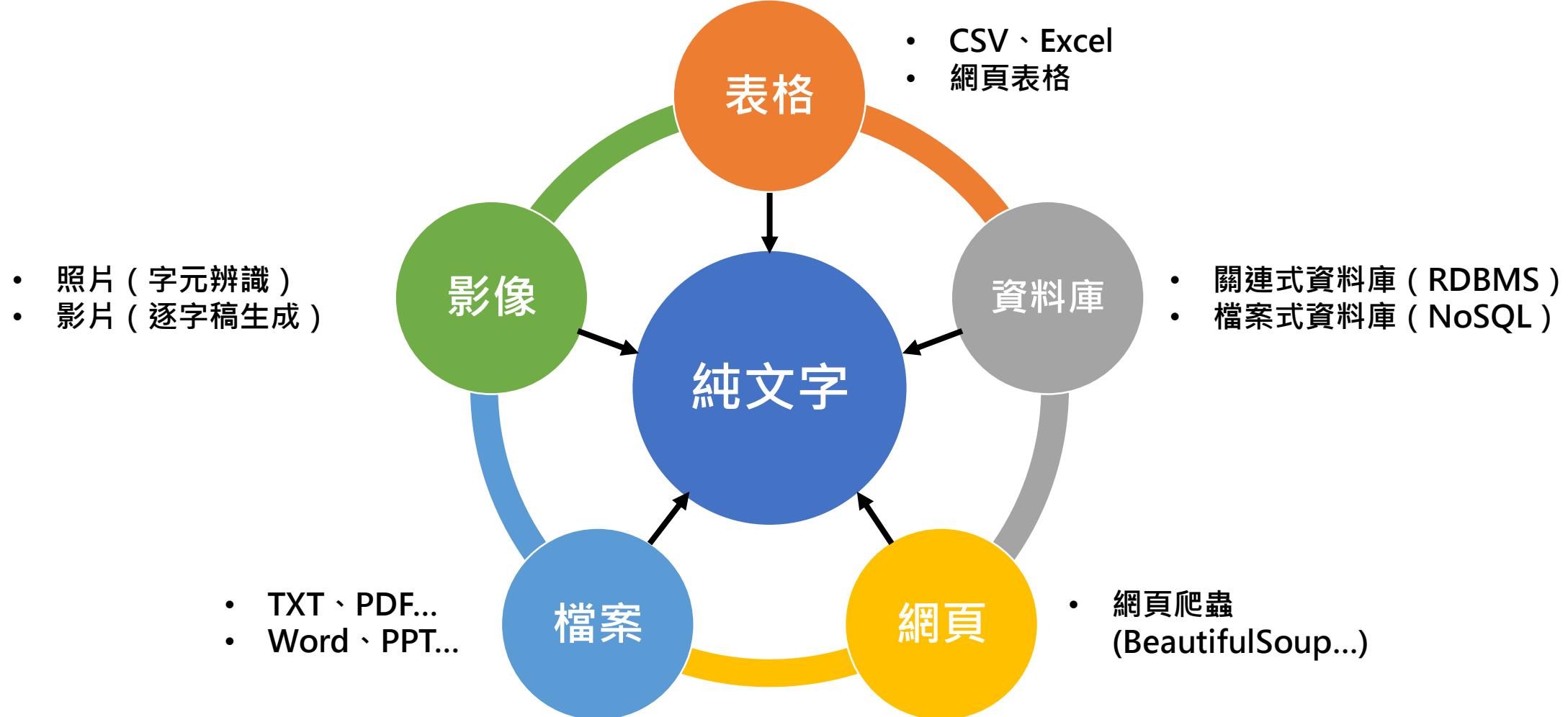




資料來源簡介



資料的來源有哪些？





「純文字格式」的定義是什麼？



- 純文字格式

- 以「記事本 (Notepad) 」開啟，能夠看懂

.CSV

User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
	Female	26	43000	0
	Female	27	57000	0
	Male	19	76000	0

```
1 User ID,Gender,Age,EstimatedSalary,Purchased
2 15624510,Male,19,19000,0
3 15810944,Male,35,20000,0
4 15668575,Female,26,43000,0
5 15603246,Female,27,57000,0
6 15804002,Male,19,76000,0
7 15728773,Male,27,58000,0
8 15598044,Female,27,84000,0
```

- 非純文字格式

- 以「記事本 (Notepad) 」開啟，無法看懂

.XLSX

User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
	Female	26	43000	0
	Female	27	57000	0
	Male	19	76000	0

```
1 504b 0304 1400 0600 0800 0000 2100 62ee
2 9d68 5e01 0000 9004 0000 1300 0802 5b43
3 6f6e 7465 6e74 5f54 7970 6573 5d2e 786d
4 6c20 a204 0228 a000 0200 0000 0000 0000
5 0000 0000 0000 0000 0000 0000 0000 0000
6 0000 0000 0000 0000 0000 0000 0000 0000
7 0000 0000 0000 0000 0000 0000 0000 0000
8 0000 0000 0000 0000 0000 0000 0000 0000
```





隨堂練習：純文字格式



- 請依照講師指示，下載下列兩個檔案：
 - [Social_Network_Ads.csv](#)
 - [Social_Network_Ads.xlsx](#)
- 請用滑鼠右鍵，分別點擊上述兩個檔案，並用下列方法開啟：
 - 開啟檔案 > 記事本
(若找不到「記事本」，請選擇「選擇其他應用程式」後即可看見)
- 分別觀察 .CSV 檔與 .XLSX 檔，以記事本開啟後的內容。
- 透過這個練習，了解何謂「純文字格式」。





表格讀取

CSV、Excel、網頁表格



如何讀取各種表格



.CSV



	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0

.XLSX



	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0

WEB



StockQ.org

表格 1

表格 2

亞洲股市指數

表格 3

表格 4

2024/3/26 16:13:48

亞洲股市行情 (Asian Markets)

股市	指數	漲跌	比例	最高	最低	開盤	今年表現	當地時間
紐西蘭	12000.82	-66.21	-0.55%	12067.03	11993	12067.03	1.96%	11:24
澳洲	8071.5	45.2	0.56%	8103.5	8026.3	8026.3	3.09%	16:47
日經225	40445.5	-451	-1.10%	40885	40445.5	40827.5	20.86%	14:59
東證股價	2777.58	-35.64	-1.27%	2806.09	2777.58	2804.54	17.38%	15:00
東證100	1913.59	-26.27	-1.35%	1933.39	1913.59	1931.9	-%	15:00
東證500	2173.46	-28.66	-1.30%	2196.12	2173.46	2195.08	-%	15:00
東證1000	2628.95	-34.2	-1.28%	2656.1	2628.95	2654.83	-%	15:00
韓國	2737.65	-10.91	-0.40%	2763.63	2731.7	2756.59	3.10%	15:32
台灣加權	20192.25	-36.18	-0.18%	-	-	-	12.61%	13:56
台灣店頭	252.27	1.10	0.44%	253.42	251.72	251.72	7.80%	13:49

pandas.read_csv()

pandas.read_excel()

pandas.read_html()





讀取 .CSV 檔案內的表格

User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0



- 1 # 下載 csv 檔本身
- 2 import os
- 3 Dataset_File = "Social_Network_Ads.csv"
- 4
- 5 if not os.path.isfile(Dataset_File):
- 6 os.system("wget https://raw.githubusercontent.com/cnchi/datasets/master/" + Dataset_File)
- 7
- 2 # 讀入 csv 檔案內的表格
- 9 import pandas as pd
- 10 dataset = pd.read_csv(Dataset_File)
- 11
- 3 # 印出讀取的結果
- 13 print(dataset)

範例完整原始碼：

<https://lurl.cc/Y73wyV>

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19.0	19000.0	0
1	15810944	Male	35.0	20000.0	0
2	15668575	Female	26.0	43000.0	0
3	15603246	Female	27.0	57000.0	0
4	15804002	Male	19.0	76000.0	0
..
395	15691863	Female	46.0	41000.0	1
396	15706071	Male	51.0	23000.0	1
397	15654296	Female	50.0	20000.0	1
398	15755018	Male	36.0	33000.0	0
399	15594041	Female	49.0	36000.0	1

[400 rows x 5 columns]





隨堂練習：讀取 .CSV 表格



- 請先撰寫好前一頁的原始碼，並且執行看看：

```
1 # 下載 CSV 檔本身
2 import os
3 Dataset_File = "Social_Network_Ads.csv"
4
5 if not os.path.isfile(Dataset_File):
6     os.system("wget https://raw.githubusercontent.com/cnchi/datasets/master/" + Dataset_File)
7
8 # 讀入 CSV 檔案內的表格
9 import pandas as pd
10 dataset = pd.read_csv(Dataset_File)
11
12 # 印出讀取的結果
13 print(dataset)
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19.0	19000.0	0
1	15810944	Male	35.0	20000.0	0
2	15668575	Female	26.0	43000.0	0
3	15603246	Female	27.0	57000.0	0
4	15804002	Male	19.0	76000.0	0
..
395	15691863	Female	46.0	41000.0	1
396	15706071	Male	51.0	23000.0	1
397	15654296	Female	50.0	20000.0	1
398	15755018	Male	36.0	33000.0	0
399	15594041	Female	49.0	36000.0	1

[400 rows x 5 columns]





讀取 .XLSX 檔案內的表格

User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0



```
1 # 下載 Excel 檔本身
2 import os
3 Dataset_File = "Social_Network_Ads.xlsx"
4
5 if not os.path.isfile(Dataset_File):
6     os.system("wget https://raw.githubusercontent.com/cnchi/datasets/master/" + Dataset_File)
7
8 # 讀入 Excel 檔案內的表格
9 import pandas as pd
10 dataset = pd.read_excel(Dataset_File)
11
12 # 印出讀取的結果
13 print(dataset)
```

範例完整原始碼：

<https://url.cc/FVUhqW>

User ID	Gender	Age	EstimatedSalary	Purchased
0	Male	19.0	19000.0	0
1	Male	35.0	20000.0	0
2	Female	26.0	43000.0	0
3	Female	27.0	57000.0	0
4	Male	19.0	76000.0	0
..
395	Female	46.0	41000.0	1
396	Male	51.0	23000.0	1
397	Female	50.0	20000.0	1
398	Male	36.0	33000.0	0
399	Female	49.0	36000.0	1

[400 rows x 5 columns]





隨堂練習：讀取 .XLSX 表格



- 請先撰寫好前一頁的原始碼，並且執行看看：

```
1 # 下載 Excel 檔本身
2 import os
3 Dataset_File = "Social_Network_Ads.xlsx"
4
5 if not os.path.isfile(Dataset_File):
6     os.system("wget https://raw.githubusercontent.com/cnchi/datasets/master/" + Dataset_File)
7
8 # 讀入 Excel 檔案內的表格
9 import pandas as pd
10 dataset = pd.read_excel(Dataset_File)
11
12 # 印出讀取的結果
13 print(dataset)
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19.0	19000.0	0
1	15810944	Male	35.0	20000.0	0
2	15668575	Female	26.0	43000.0	0
3	15603246	Female	27.0	57000.0	0
4	15804002	Male	19.0	76000.0	0
..
395	15691863	Female	46.0	41000.0	1
396	15706071	Male	51.0	23000.0	1
397	15654296	Female	50.0	20000.0	1
398	15755018	Male	36.0	33000.0	0
399	15594041	Female	49.0	36000.0	1

[400 rows x 5 columns]





讀取網頁上的表格

<http://www.stockq.org/market/asia.php>

亞洲股市指數										
2024/3/26 16:13:48										
股市	指數	漲跌	比例	最高	最低	開盤	今年表現	當地時間		
紐西蘭	12000.82	-66.21	-0.55%	12067.03	11993	12067.03	1.96%	11.24		
澳洲	8071.5	45.2	0.56%	8103.5	8026.3	8026.3	3.09%	16.47		
日經225	40445.5	-451	-1.10%	40885	40445.5	40827.5	20.86%	14.59		
全球指數寬螢幕版	2777.58	-35.64	-1.27%	2806.09	2777.58	2804.54	17.38%	15.00		
各國股市	東證100	1913.59	-26.27	-1.35%	1933.39	1913.59	1931.59	-.%	15.00	
亞洲股市指數	東證500	2173.46	-28.66	-1.30%	2196.12	2173.46	2195.08	-.%	15.00	
歐洲股市、非洲指數	東證1000	2628.95	-34.2	-1.28%	2656.1	2628.95	2654.83	-.%	15.00	
美洲股市指數	韓國	2737.65	-10.91	-0.40%	2763.63	2731.7	2756.59	3.10%	15.32	
臺灣加權	台灣加權	20192.25	-36.18	-0.18%	-	-	12.61%	13.56		
美洲股市指數	台灣店頭	252.27	1.10	0.44%	253.42	251.72	251.72	7.80%	13.49	

df = pandas.read_html()

Index	Type	Size	
0	DataFrame	(1, 1)	Column names: 0
1	DataFrame	(2, 1)	Column names: 0
2	DataFrame	(1, 2)	Column names: 0, 1
3	DataFrame	(1, 2)	Column names: 0, 1
4	DataFrame	(1, 2)	Column names: 0, 1
5	DataFrame	(1, 1)	Column names: 0
6	DataFrame	(1, 1)	Column names: 0
7	DataFrame	(1, 1)	Column names: 0
8	DataFrame	(1, 2)	Column names: 0, 1
9	DataFrame	(34, 9)	Column names: 0, 1, 2, 3, 4, 5, 6, 7, 8
10	DataFrame	(3, 1)	Column names: 0

亞洲股市指數										
2024/3/26 16:13:48										
紐西蘭										
8071.5										
澳洲										
8067.60										
日經225										
40500.50										
東證股價										
2776.10										
東證100										
1913.59										
東證500										
2173.46										
東證1000										
2628.95										
韓國										
2769.65										
台灣加權										
20192.25										
台灣店頭										

df[9]

Index	0	1	2
0	亞洲股市行情 (Asian Markets)	亞洲股市行情 (Asian Markets)	亞洲股市行情 (Asian Markets)
1	股市	指數	漲跌
2	紐西蘭	12029.67	-37.36
3	澳洲	8067.60	-3.90
4	日經225	40500.50	55.00
5	東證股價	2776.10	-1.54
6	東證100	1913.59	-26.27
7	東證500	2173.46	-28.66
8	東證1000	2628.95	-34.2
9	韓國	2769.65	32.08
10	台灣加權	20192.25	-36.18



讀取網頁上的表格

WEB



```

1 # 讀取指定網頁上的所有內容
2 import pandas as pd
3 dfHTML = pd.read_html("http://www.stockq.org/market/asia.php")
4
5 # 列出所有表格的「索引值」與「維度大小」
6 for i, df in enumerate(dfHTML):
7     print(i, df.shape)
8
9 # dfHTML[9] 是我們的目標，先印出前五筆看看
10 print(dfHTML[9].head())
11
12 # 我們只要 dfHTML[9] 的第二列之後、0~4 欄的所有內容
13 asia_stocks = dfHTML[9].loc[2:, :5]
14
15 # 印出擷取出來的表格內容
16 print(asia_stocks)

```

0 (1, 1)
1 (2, 1)
2 (1, 2)
...
9 (34, 9)
10 (3, 1)

StockQ.org								
表格 1								
亞洲股市指數								
2024/3/26 16:13:48								
亞洲股市行情 (Asian Markets)								
統計	指數	漲跌	比例	最高	最低	開盤	今收	當日變動
紐西蘭	12000.82	-66.21	-0.55%	12067.03	11993	12067.03	11995	11.24
澳洲	8071.5	45.2	0.56%	8103.5	8026.3	8026.3	8026.3	3.09%
日經225	40445.5	-451	-1.10%	40885	40445.5	40827.5	40827.5	20.88%
全指指數	2777.58	-35.64	-1.27%	2806.09	2777.58	2804.54	2777.58	17.38%
各國股市	1913.59	-26.27	-1.35%	1933.39	1913.59	1931.9	1931.9	15.00
新加坡	2173.46	-28.66	-1.30%	2195.12	2173.46	2195.08	2195.08	15.00
印度100	2628.95	-34.2	-1.28%	2656.1	2628.95	2654.83	2654.83	15.00
印度500	2737.65	-10.91	-0.40%	2763.63	2731.7	2756.59	2756.59	3.10%
韓國	20192.25	-36.16	-0.18%	-	-	-	-	12.01%
台灣加權	252.27	1.10	0.44%	253.42	251.72	251.72	251.72	13.49

表格 2
市場動向表格 3
各國股市表格 4
亞洲股市指數

enumerator: 窮舉器

(<6)

三 .iloc [2:, :6]

↓ row ↓ column (≤)

Index	0	1	2	3	4	5	6
0	亞洲股市行情 (Asian Marke...						
1	股市	指數	漲跌	比例	最高	最低	開盤
2	紐西蘭	10531.94	67.41	0.64%	10534.77	10464.53	10464.53
3	澳洲股市	6741.10	9.70	0.14%	6772.40	6731.40	18.07%
4	日經225	21754.27	24.30	0.11%	21784.22	21697.31	21699.43
5	東證一部	1589.84	4.99	0.31%	1590.79	1583.96	6.41%

範例完整原始碼：
<https://lurl.cc/tfiz2S>



隨堂練習：讀取網頁上的表格



- 請先撰寫好前一頁的原始碼，並且執行看看：

```
1 # 讀取指定網頁上的所有內容
2 import pandas as pd
3 dfHTML = pd.read_html("http://www.stockq.org/market/asia.php")
4
5 # 列出所有表格的「索引值」與「維度大小」
6 for i, df in enumerate(dfHTML):
7     print(i, df.shape)
8
9 # dfHTML[9] 是我們的目標，先印出前五筆看看
10 print(dfHTML[9].head())
11
12 # 我們只要 dfHTML[9] 的第二列之後、0~4 欄的所有內容
13 asia_stocks = dfHTML[9].loc[2:, :5]
14
15 # 印出擷取出來的表格內容
16 print(asia_stocks)
```

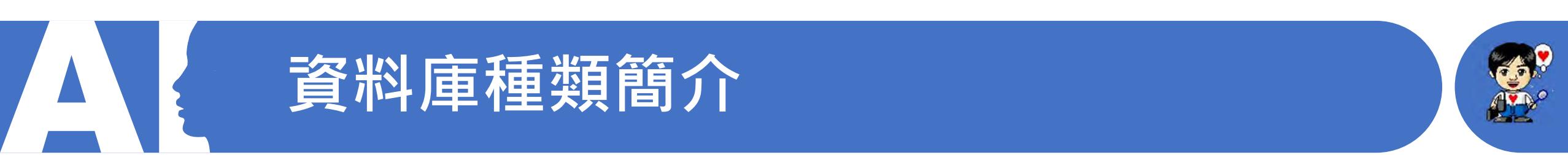
	0	1	2	3	4	5
2	紐西蘭	12025.02	-42.01	-0.35%	12067.03	11988.05
3	澳洲	8067.10	-4.40	-0.05%	8071.50	8034.00
4	日經225	40518.00	72.50	0.18%	40573.50	40318.50
5	東證股價	2779.89	2.25	0.08%	2779.89	2767.46
6	東證100	1915.97	2.76	0.14%	1915.97	1905.39
7	東證500	2175.26	1.73	0.08%	2175.58	2165.15
8	東證1000	2630.90	1.89	0.07%	2631.26	2619.17
9	韓國	2765.06	27.49	1.00%	2771.84	2753.60





資料庫讀取

關連式資料庫、檔案式資料庫

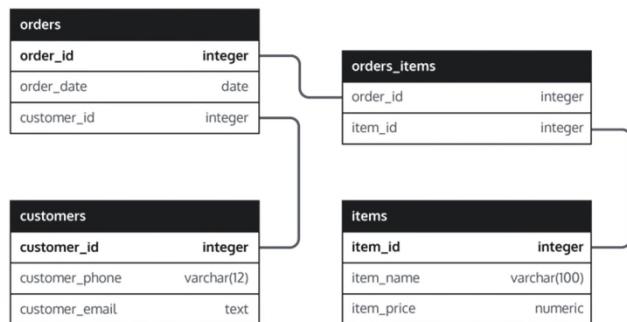


資料庫種類簡介



• 關連式資料庫 (RDBMS)

- 資料以「**表格**」方式儲存
- 每個欄位大小、資料型態皆**事先定義**，一旦決定之後，不得變更
- 需以 **SQL 語言**讀寫之 *Structural Query Language*
- 各表格以「**關鍵欄位 (Primary Key)**」相連，故稱「**關連式資料庫**」
- 同一個表格的**資料錄 (Records)**，一定要放在同一台伺服器，不易分散
- 底層以各家私有之**封閉式格式**儲存



效率好
彈性差

• 文件式資料庫 (NoSQL Databases)

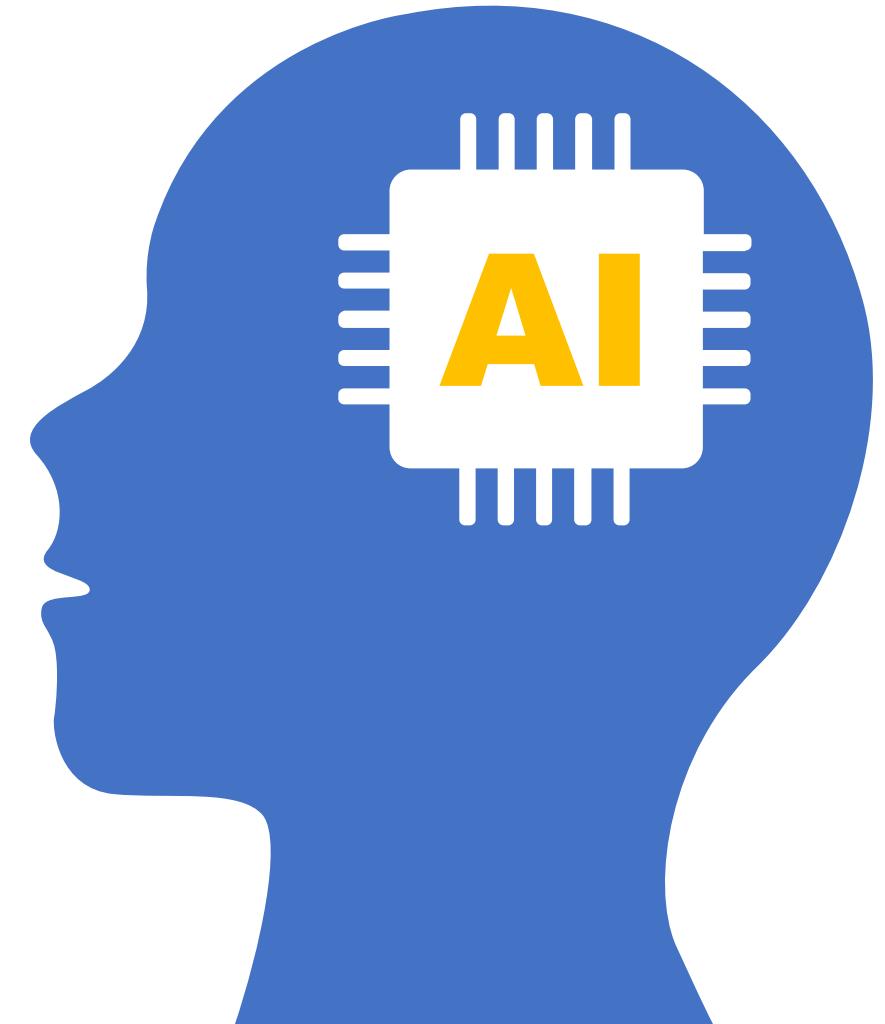
- 資料以「**一般檔案**」儲存
- **不需事先定義**每個欄位長相，同一欄位，還能儲存不同資料型態
- **不需以 SQL 語言**讀寫之，故稱「**NoSQL**」
- 各筆資料以「**索引值 (id)**」鬆散相連，彈性頗大
- 同一個資料集的「**記錄 (Records)**」，可以存放在不同伺服器上，易於分散運算
- 底層以 **JSON** 這種公開檔案格式儲存





資料庫讀取

- 讀寫「關連式資料庫」
(Relational Databases, RDBMS)
- 讀寫「文件式資料庫」
(NoSQL Databases)



範例完整原始碼：

<https://lurl.cc/yf5qrj>



常見的關連式資料庫



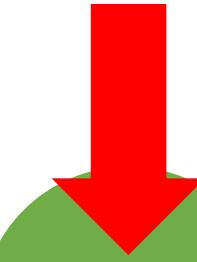
- 輕量型、開放式資料庫
- 只要一個檔案就可以跑
- 單板機等資源受限環境適用



*single-board computer
(sBC)*

- 開放式資料庫，免費
- 功能比 MySQL 稍強
- 工程師、駭客之間流行

本小節使用



SQLite

RDBMS

Oracle

IBM DB2

PostgreSQL

MySQL

- 封閉式資料庫
- 功能多，業界市佔率大

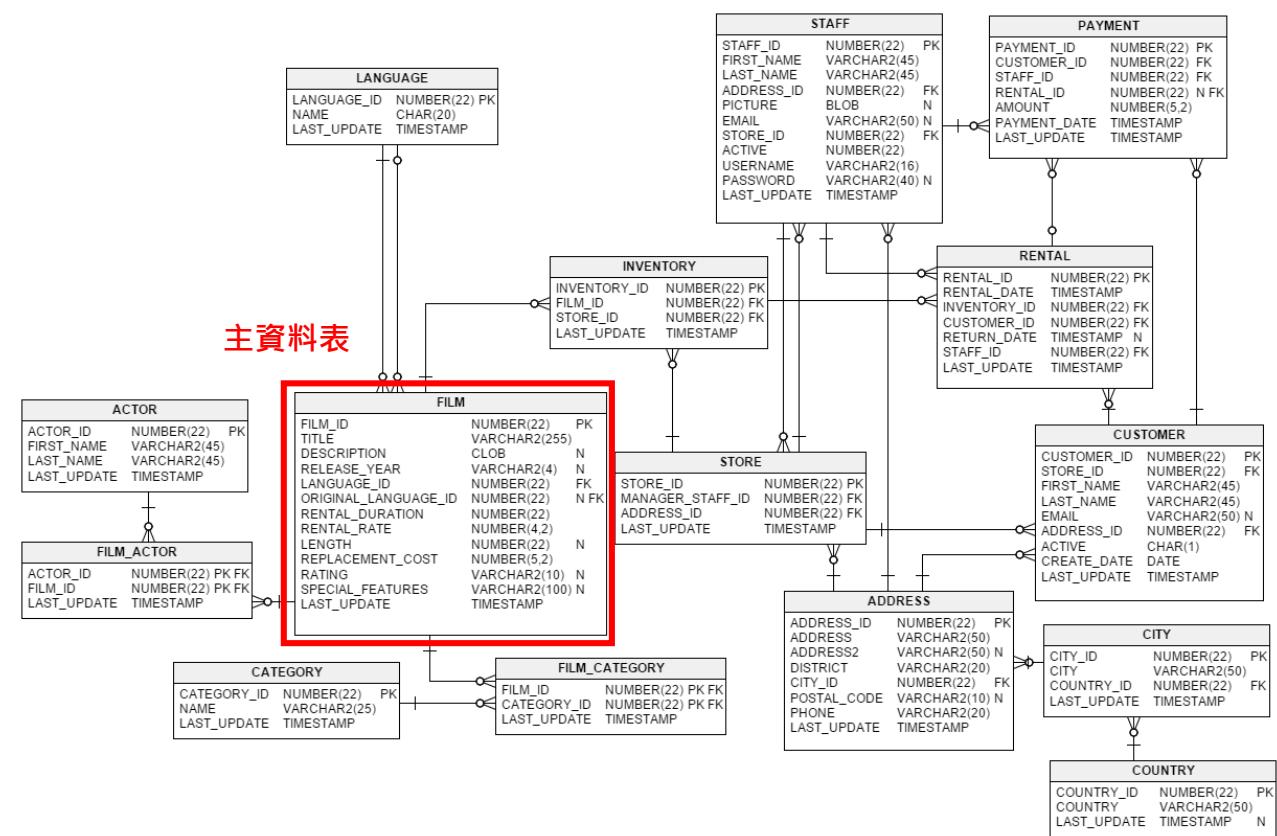
- 封閉式資料庫
- 穩穩定，金融業愛用

- 開放式資料庫
- 免費，中小企業愛用





- sakila.db : 「錄影帶租借連鎖店」資料庫





資料庫下載與連接



```
1 # 將 SQLite 資料庫檔案下載下來
2 import os
3 Dataset_File = "sakila.db"
4
5 if not os.path.isfile(Dataset_File):
6     os.system("wget https://raw.githubusercontent.com/cnchi/datasets/master/" + Dataset_File)
7
8 # 以 SQLite3 這個套件連上資料庫
9 import sqlite3
10 conn = sqlite3.connect(Dataset_File)
```





隨堂練習：讀取網頁上的表格

- 請先**撰寫**好前一頁的**原始碼**，並且**執行**看看：

```
1 # 將 SQLite 資料庫檔案下載下來
2 import os
3 Dataset_File = "sakila.db"
4
5 if not os.path.isfile(Dataset_File):
6     os.system("wget https://raw.githubusercontent.com/cnchi/datasets/master/" + Dataset_File)
7
8 # 以 SQLite3 這個套件連上資料庫
9 import sqlite3
10 conn = sqlite3.connect(Dataset_File)
```





顯示資料表的「架構 (Schema) 」



```
1 # 取得資料庫的「讀寫頭 (Cursor) 」  
2 cursor = conn.cursor()  
3  
4 # 執行 SQL 的 PRAGMA 指令, 取得 Schema  
5 cursor.execute("PRAGMA table_info(film)")  
6 rows = cursor.fetchall() # fetch from cache  
7  
8 # 將抓到的每條 Schema 都印出來  
9 for row in rows:  
10    print(row)
```

```
(0, 'film_id', 'INT', 1, None, 1)  
(1, 'title', 'VARCHAR(255)', 1, None, 0)  
(2, 'description', 'BLOB SUB_TYPE TEXT', 0, 'NULL', 0)  
(3, 'release_year', 'VARCHAR(4)', 0, 'NULL', 0)  
(4, 'language_id', 'SMALLINT', 1, None, 0)  
(5, 'original_language_id', 'SMALLINT', 0, 'NULL', 0)  
(6, 'rental_duration', 'SMALLINT', 1, '3', 0)  
(7, 'rental_rate', 'DECIMAL(4,2)', 1, '4.99', 0)  
(8, 'length', 'SMALLINT', 0, 'NULL', 0)  
(9, 'replacement_cost', 'DECIMAL(5,2)', 1, '19.99', 0)  
(10, 'rating', 'VARCHAR(10)', 0, "'G'", 0)  
(11, 'special_features', 'VARCHAR(100)', 0, 'NULL', 0)  
(12, 'last_update', 'TIMESTAMP', 1, None, 0)
```





隨堂練習：讀取網頁上的表格



- 請先撰寫好前一頁的原始碼，並且執行看看：

```
1 # 取得資料庫的「讀寫頭 (cursor) 」
2 cursor = conn.cursor()
3
4 # 執行 SQL 的 PRAGMA 指令，取得 Schema
5 cursor.execute("PRAGMA table_info(film)")
6 rows = cursor.fetchall()
7
8 # 將抓到的每條 Schema 都印出來
9 for row in rows:
10    print(row)
```





讀取資料庫內容



1

```
1 # 用 SQL 的 SELECT 指令, 讀取 film 資料表內容
2 cursor.execute("SELECT * FROM film")
3 rows = cursor.fetchall()
4 for row in rows:
5     print(row)
6
```

2

```
7 # 同樣用 SELECT 指令, 將結果塞入 DataFrame 中
8 import pandas as pd
9 dataset = pd.read_sql_query("SELECT * FROM film", conn)
10 print(dataset)
```

```
(1, 'ACADEMY DINOSAUR', 'A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies', '2006', 1, None, 6, 0.99, 86, 20.99, 'PG', 'Deleted Scenes,Behind the Scenes', '2020-12-23 07:12:31')
(2, 'ACE GOLDFINGER', 'A Astounding Epistle of a Database Administrator And a Explorer who must Find a Car in Ancient China', '2006', 1, None, 3, 4.99, 48, 12.99, 'G', 'Trailers,Deleted Scenes', '2020-12-23 07:12:31')
(3, 'ADAPTATION HOLES', 'A Astounding Reflection of a Lumberjack And a Car who must Sink a Lumberjack in A Baloon Factory', '2006', 1, None, 7, 2.99, 50, 18.99, 'NC-17', 'Trailers,Deleted Scenes', '2020-12-23 07:12:31')
(4, 'AFFAIR PREJUDICE', 'A Fanciful Documentary of a Frisbee And a Lumberjack who must Chase a Monkey in A Shark Tank', '2006', 1, None, 5, 2.99, 117, 26.99, 'G', 'Commentaries,Behind the Scenes', '2020-12-23 07:12:31')
(5, 'AFRICAN EGG', 'A Fast-Paced Documentary of a Pastry Chef And a Dentist who must Pursue a Forensic Psychologist in The Gulf of Mexico', '2006', 1, None, 6, 2.99, 130, 22.99, 'G', 'Deleted Scenes', '2020-12-23 07:12:31')
(6, 'AGENT TRUMAN', 'A Intrepid Panorama of a Robot And a Boy who must Escape a Sumo Wrestler in Ancient China', '2006', 1, None, 3, 2.99, 169, 17.99, 'PG', 'Deleted Scenes', '2020-12-23 07:12:31')
...
```





隨堂練習：讀取網頁上的表格



- 請先撰寫好前一頁的原始碼，並且執行看看：

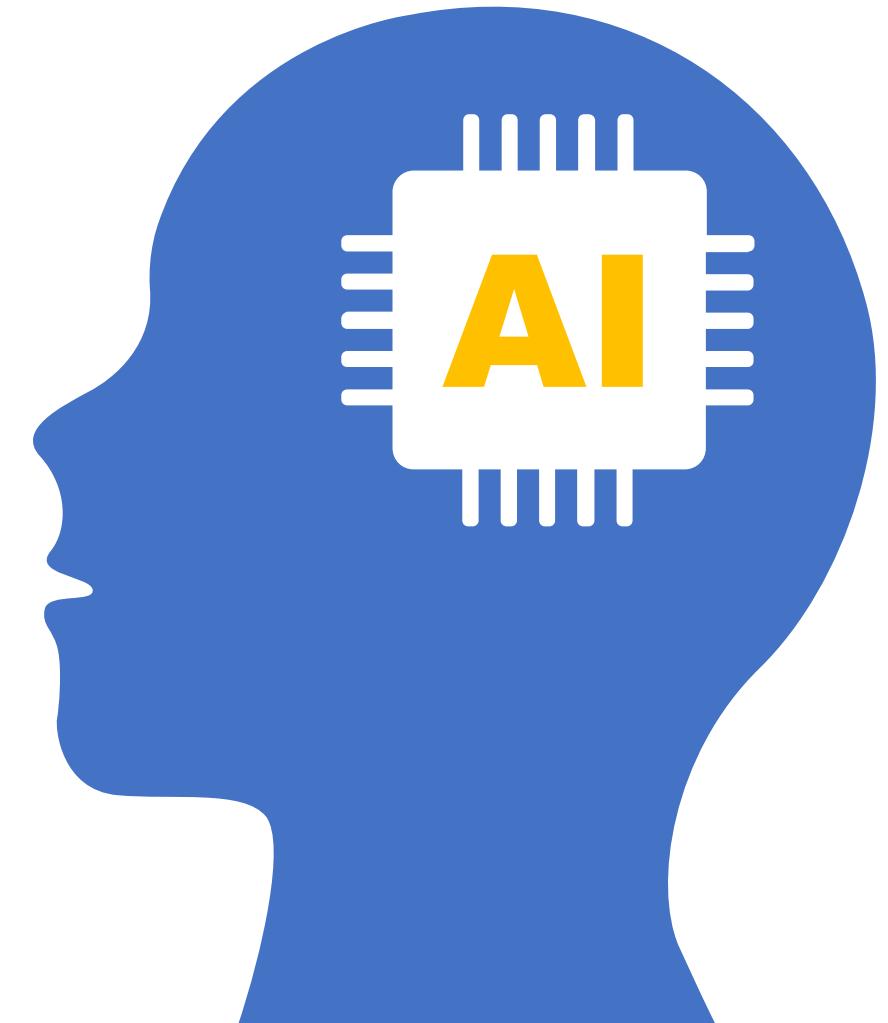
```
1 # 用 SQL 的 SELECT 指令，讀取 film 資料表內容
2 cursor.execute("SELECT * FROM film")
3 rows = cursor.fetchall()
4 for row in rows:
5     print(row)
6
7 # 同樣用 SELECT 指令，將結果塞入 DataFrame 中
8 import pandas as pd
9 dataset = pd.read_sql_query("SELECT * FROM film", conn)
10 print(dataset)
```





資料庫讀取

- 讀寫「關連式資料庫」
(Relational Databases, RDBMS)
- 讀寫「文件式資料庫」
(NoSQL Databases)



範例完整原始碼：

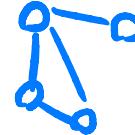
<https://lurl.cc/4BEdNI>



常見的 NoSQL 資料庫

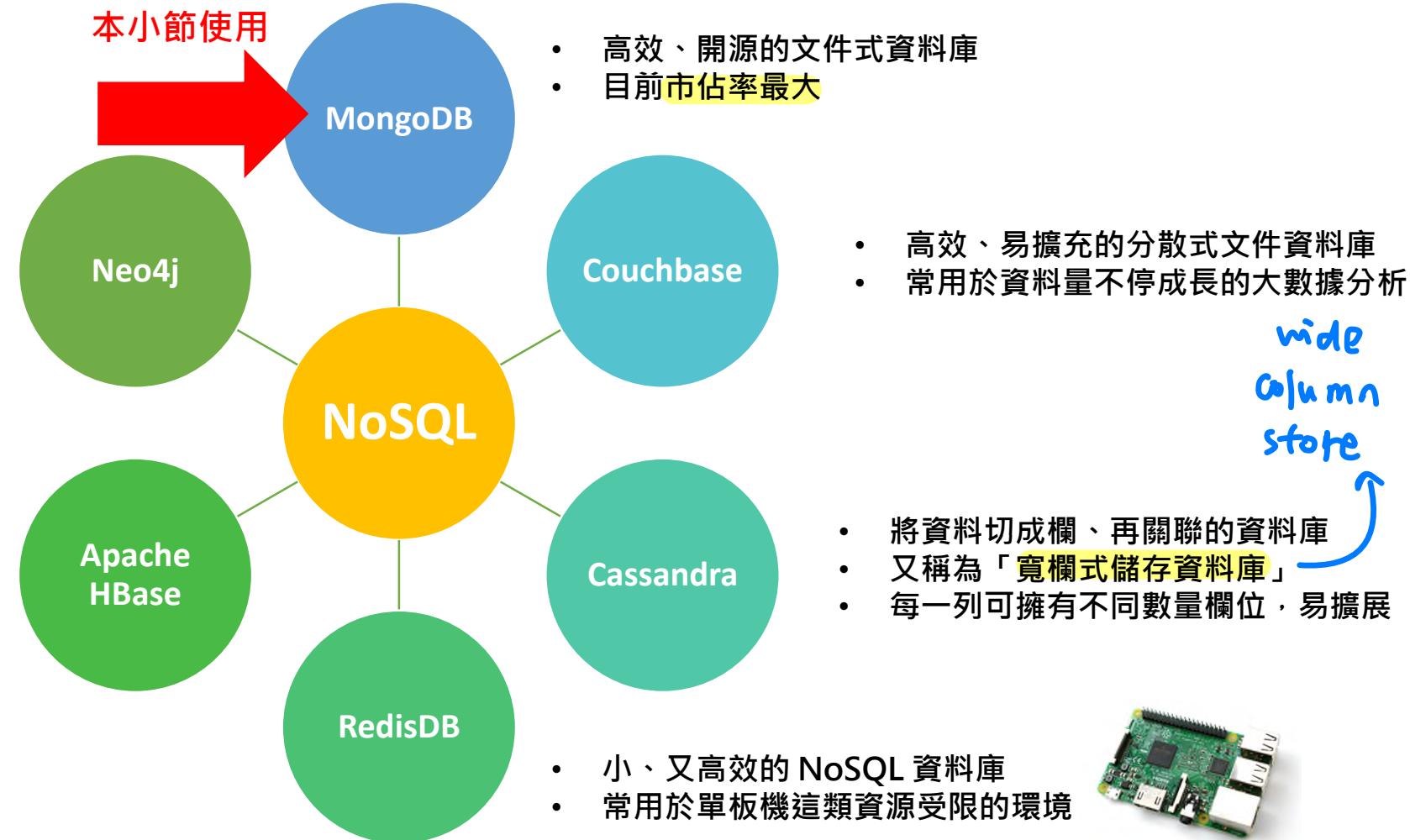


Graph



- 以圖形演算法關聯的資料庫
- 常用於社交網路、推薦系統
- 開源、快速的「寬欄式」資料庫
- 常用於金融等需即時反應的場合

asynchronous
loading





JSON 概念與專有名詞簡介

- JSON = **JavaScript Object Notation**
- NoSQL 資料庫底層用於儲存資料的一種純文字格式，欄位彈性、內容易讀、格式好交換

Books
(=Collection =Table)

BookStore (= Database)

Authors
(=Collection =Table)

NoSQL Terms
RDBMS Terms

Books

```
1 [ 
2 { key value
3   "title": "The Great Gatsby",
4   "author": "F. Scott Fitzgerald",
5   "publishedYear": 1925,
6   "genres": ["Novel", "Fiction"],
7   "available": true
8 }, (=Document, Object =Row, Record)
9 {
10   "title": "To Kill a Mockingbird",
11   "author": "Harper Lee",
12   "publishedYear": 1960,
13   "genres": ["Novel", "Drama"],
14   "available": false
15 }
16 ]
```

(=Array)

BookStore (= Database)

```
1 [
2 { =Field =Column
3   "name": "F. Scott Fitzgerald",
4   "bio": "Francis Scott Key Fitzgerald was an American novelist.",
5   "booksPublished": 5,
6   "birthYear": 1896
7 },
8 { (=Key) (=Value)
9   "name": "Harper Lee",
10  "bio": "Nelle Harper Lee was best known for her 1960 novel.",
11  "booksPublished": 2,
12  "birthYear": 1926
13 }
14 ]
```

(=Field =Column)

(=Key) (=Value)





安裝 MongoDB 伺服器



前往 MongoDB 社群版下載處

<https://www.mongodb.com/download-center/community/releases>

MongoDB Community Downloads

Thank you for your interest in MongoDB Community. You can download installation packages below.

Please note that for those who are not yet MongoDB customers, download and use constitutes acceptance of the Customer Agreement. For Enterprise MongoDB releases, click [here](#).

Current Releases

[Subscribe](#) to be notified when new versions of MongoDB Enterprise and Ops Manager are available.

The following are direct download links for MongoDB Community (available via curl or wget). Installation docs can be found [here](#).

7.0.7

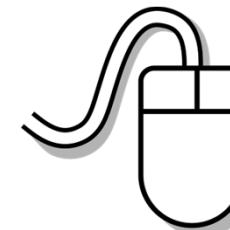
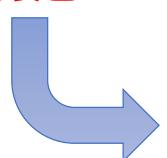
- Ubuntu 22.04 x64

找到適合自己機器的安裝包

- Archive: [mongodb-linux-x86_64-ubuntu2204-7.0.7.tgz](https://fastdl.mongodb.org/linux/mongodb-linux-aarch64-amazon2-6.0.14.tgz)
- Server Package: [mongodb-org-server_7.0.7_amd64.deb](https://fastdl.mongodb.org/server/7.0.7/_/mongodb-org-server_7.0.7_amd64.deb)
- Mongos Package: [mongodb-org-mongos_7.0.7_amd64.deb](https://fastdl.mongodb.org/mongos/7.0.7/_/mongodb-org-mongos_7.0.7_amd64.deb)

- Windows x64

- Archive: [mongodb-windows-x86_64-7.0.7.zip](https://fastdl.mongodb.org/windows/mongodb-windows-x86_64-7.0.7.zip)



MongoDB Community Edition

- 擁有所有基本功能。
- 免費，但不可商用。

MongoDB Enterprise Edition

- 增加安全性、額外工具、免費教育訓練。
- 需收費，可商用。

右鍵點擊，複製鏈結

<https://fastdl.mongodb.org/linux/mongodb-linux-aarch64-amazon2-6.0.14.tgz>

PS: Colab 使用軟體：

- CPU: x86 64 Bits
- OS: Ubuntu 22.04





安裝 MongoDB 伺服器



- **wget** : web get = 可下載指定 URL 的檔案
- **-q0** : Quiet Level 0, 下載時不需逐步報告現況
- **tar -xvz** : 專門解開 tgz 壓縮檔的指令。x=eXtract, z=Zip format, v=verbose
- **mv** : MoVe。將解壓縮後、位於 /bin 的執行檔，全數移動到 /usr/bin。

1

```
1 # 下載免費開源的 MongoDB 社群版
2 # 可從 https://www.mongodb.com/download-center/community/releases 查詢你要下載的版本
3 # 需與 Google Colab 當前使用的作業系統一致 (Ubuntu 22.04)
4 !wget -q0 - https://fastdl.mongodb.org/linux/mongodb-linux-x86\_64-ubuntu2204-7.0.7.tgz | tar -xvz
5 !mv mongodb-linux-x86_64-ubuntu2204-7.0.7/bin/* /usr/bin/
6
```

2

```
7 # 下載與 MongoDB 版本相應的金鑰，保證 MongoDB 連線的安全
8 # 可以上 https://pgp.mongodb.com/ 查詢欲下載金鑰的完整檔名
9 !wget https://www.mongodb.org/static/pgp/server-7.0.asc
10 !sudo apt-key add server-7.0.asc
```

- **wget** : 取得與 MongoDB 版本相應的金鑰，用以加密往來封包
- **sudo** : Super User DO。類似 Windows 的「以系統管理員執行」
- **apt-key add** : 將金鑰加入作業系統中



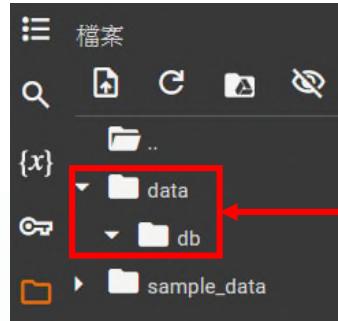


安裝 MongoDB 伺服器



1 # 建造 MongoDB 資料庫存放處（路徑、名稱可自訂）
2 !mkdir -p /content/data/db
3

4 # 啟動 MongoDB 伺服器 (Mongo Daemon) 於本地端 (localhost)
5 !mongod --fork --logpath /var/log/mongodb.log --dbpath /content/data/db --bind_ip localhost



- MongoDB 資料存放處
- Colab 結束即消失

- **mongod** : MongoDB Daemon = MongoDB 背景伺服器
- **--fork** : 啟動 MongoDB 伺服器。fork 有「分支、啟動副本」之意
- **--logpath** : 指定存放 MongoDB 日誌 (Logs) 的路徑與檔名
- **--dbpath** : 告知 MongoDB 要把資料存放到何處
- **--bind_ip** : 告知 MongoDB 開放哪個 IP Address 供外界存取。localhost = 僅 Colab 本機存取

```
1 # 掛載私人 Google Drive 儲存 MongoDB 資料庫資料  
2 from google.colab import drive  
3 drive.mount('/content/gdrive')  
4  
5 # 啟動 MongoDB 伺服器 (Mongo Daemon) 於本地端 (localhost) (使用掛載的外部硬碟會耗時約 1 分鐘, 請耐心等待)  
6 !mongod --fork --logpath /var/log/mongodb.log --dbpath /content/gdrive/MyDrive/Datasets/MongoDB --bind_ip localhost
```

將 MongoDB 資料存放至私人的 Google Drive ·
Colab 結束資料不消失的方法

/content/ :不能改
gdrive : 掛載點(可任意取名)

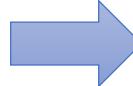




確認安裝成功 & 安裝其它套件

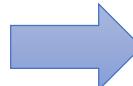


```
1 # 確認 MongoDB 正在執行中  
2 !mongod --sysinfo
```



```
{  
  "sysinfo":{  
    "_SC_PAGE_SIZE": 4096,  
    "_SC_PHYS_PAGES": 3322620,  
    "_SC_AVPHYS_PAGES": 2042468  
  }  
}
```

```
1 # 安裝 PyMongo 函式庫  
2 !pip install pymongo
```



```
Installing collected packages: dnspython, pymongo  
Successfully installed dnspython-2.6.1 pymongo-4.6.3
```





隨堂練習：環境設定



- 請先撰寫好下列原始碼，並且執行看看：

```
1 # 下載免費開源的 MongoDB 社群版
2 !wget -qO - https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-ubuntu2204-7.0.7.tgz | tar -xvz
3 !mv mongodb-linux-x86_64-ubuntu2204-7.0.7/bin/* /usr/bin/
4
5 # 下載與 MongoDB 版本相應的金鑰
6 !wget https://www.mongodb.org/static/pgp/server-7.0.asc
7 !sudo apt-key add server-7.0.asc
8
9 # 建造 MongoDB 資料庫存放處（路徑、名稱可自訂）
10 !mkdir -p /content/data/db
11
12 # 啟動 MongoDB 伺服器 (Mongo Daemon) 於本地端 (localhost)
13 !mongod --fork --logpath /var/log/mongodb.log --dbpath /content/data/db --bind_ip localhost
14
15 # 確認 MongoDB 正在執行中
16 !mongod --sysinfo
17
18 # 安裝 PyMongo 函式庫
19 !pip install pymongo
```





連上 MongoDB 資料庫



```
1 # 引入 PyMongo 客戶端
2 from pymongo import MongoClient
3
4 # 創造 PyMongo 客戶端物件
5 client = MongoClient()
6
7 # 顯示 MongoDB 伺服器資訊
8 client.server_info()
9
10 # 取得 MongoDB 伺服器內所有現存資料庫
11 databases = client.list_database_names()
12
13 for db in databases:
14     print(db)
```

```
{'version': '7.0.7',
'gitVersion': 'cfb08e1322514c45bd',
'modules': [],
...
'bits': 64,
'debug': False,
'maxBsonObjectSize': 16777216,
'storageEngines': ['devnull', 'wiredTiger'],
'ok': 1.0}
```

admin
config
local





隨堂練習：連上 MongoDB 資料庫



- 請先**撰寫好下列原始碼**，並且**執行看看**：

```
1 # 引入 PyMongo 客戶端
2 from pymongo import MongoClient
3
4 # 創造 PyMongo 客戶端物件
5 client = MongoClient()
6
7 # 顯示 MongoDB 雖服器資訊
8 client.server_info()
9
10 # 取得 MongoDB 雖服器內所有現存資料庫
11 databases = client.list_database_names()
12
13 for db in databases:
14     print(db)
```





資料庫的建立：輸入資料



```
1 # 建立資料庫
2 db = client['books']
3
4 # 建立資料集 (collection, 類似關連式資料庫的 Table)
5 collection = db['book_info']
6
7 # 在資料集裡加入數筆文件錄 (document, 類似關連式資料庫的 Record)
8 collection.insert_one({"name": "PyTorch 入門",
9                 "authors": "李小明",
10                "publisher": "碁峰出版社",
11                "price": "560"})
12 collection.insert_one({"name": "自然語言一本通",
13                 "authors": "張大華",
14                "publisher": "碁峰出版社",
15                "price": "720"})
16 collection.insert_one({"name": "深度學習聖經：中譯本",
17                 "authors": ["Steven Oakman", "Josh Worker"],
18                 "translators": "黃小玲",
19                 "publisher": "博碩出版社",
20                 "price": "680"})
```

```
4 22 # 另一種建立資料集的方式：'books' > 'publishers'
5 23 collection = db.publishers
6
7 24
8 25 # 將文件錄插入新建的資料集
9 26 collection.insert_one({"name": "碁峰出版社",
10                "URL": "https://www.gotop.com.tw/"})
11 27
12 28 collection.insert_one({"name": "博碩出版社",
13                "URL": "https://www.drmaster.com.tw/"})
14 29
```





隨堂練習：輸入資料

- 請先撰寫好下列原始碼，並且執行看看。
- 由於內容簡單且繁雜，請從標準答案直接拷貝貼上即可。

```
1 # 建立資料庫
2 db = client['books']
3
4 # 建立資料集 (collection, 類似關連式資料庫的 Table)
5 collection = db['book_info']
6
7 # 在資料集裡加入數筆文件錄 (document, 類似關連式資料庫的 Record)
8 collection.insert_one({"name": "PyTorch 入門",
9                     "authors": "李小明",
10                    "publisher": "碁峰出版社",
11                    "price": "560"})
12 collection.insert_one({"name": "自然語言一本通",
13                     "authors": "張大華",
14                    "publisher": "碁峰出版社",
15                    "price": "720"})
```

```
16 collection.insert_one({"name": "深度學習聖經：中譯本",
17                         "authors": ["Steven Oakman", "Josh Worker"],
18                         "translators": "黃小玲",
19                         "publisher": "博碩出版社",
20                         "price": "680"})
21
22 # 另一種建立資料集的方式：'books' > 'publishers'
23 collection = db.publishers
24
25 # 將文件錄插入新建的資料集
26 collection.insert_one({"name": "碁峰出版社",
27                         "URL": "https://www.gotop.com.tw/"})
28 collection.insert_one({"name": "博碩出版社",
29                         "URL": "https://www.drmaster.com.tw/"})
```





資料庫的建立：驗證資料輸入成功



```
1 # 驗證資料庫已經建立
2 databases = client.list_database_names()
3
4 for db in databases:
5     print(db)
6
7 # 驗證兩個資料集都已經建立
8 db = client.books
9 collections = db.list_collection_names()
10
11 for collection in collections:
12     print(collection)
```

admin
books
config
local

```
14 # 取得 book_info 的讀寫頭
15 cursor = db.book_info.find()
16
17 # 印出 book_info 的內容
18 for document in cursor:
19     print(document)
20
21 # 取得 publishers 的讀寫頭
22 cursor = db.publishers.find()
23
24 # 印出 publishers 的內容
25 for document in cursor:
26     print(document)
```

book_info
publishers





隨堂練習：驗證資料輸入正確



- 請先撰寫好下列原始碼，並且執行看看：

```
1 # 驗證資料庫已經建立  
2 databases = client.list_database_names()  
3  
4 for db in databases:  
5     print(db)  
6  
7 # 驗證兩個資料集都已經建立  
8 db = client.books  
9 collections = db.list_collection_names()  
10  
11 for collection in collections:  
12     print(collection)  
  
14 # 取得 book_info 的讀寫頭  
15 cursor = db.book_info.find()  
16  
17 # 印出 book_info 的內容  
18 for document in cursor:  
19     print(document)  
20  
21 # 取得 publishers 的讀寫頭  
22 cursor = db.publishers.find()  
23  
24 # 印出 publishers 的內容  
25 for document in cursor:  
26     print(document)
```





資料庫的建立：資料的查詢



```
1 # 尋找書名、出版社、與官方網址
2 book_name = input("請輸入書名：")
3
4 # 取得 book_info 與 publishers 兩個資料集代表變數
5 bk_info = client.books.book_info
6 pub = client.books.publishers
7
8 # 尋找使用者輸入書名是否存在
9 bk = bk_info.find_one({"name":book_name})
10 if bk:
11     print("書名：" , bk["name"])
12     print("出版社：" , bk["publisher"])
13
14 # 用書籍資訊 book_info 裡的資訊跨資料集查詢
15 pub_info = pub.find_one({"name":bk["publisher"]})
16 if pub_info:
17     print("官方網址：" , pub_info["URL"])
18 else:
19     print("查無此書")
```

請輸入書名：

從頭找第一筆符合的資料

請輸入書名：PyTorch 入門
書名： PyTorch 入門
出版社： 墓峰出版社
官方網址： <https://www.gotop.com.tw/>





隨堂練習：資料的查詢



- 請先撰寫好下列原始碼，並且執行看看：

```
1 # 尋找書名、出版社、與官方網址
2 book_name = input("請輸入書名：")
3
4 # 取得 book_info 與 publishers 兩個資料集代表變數
5 bk_info = client.books.book_info
6 pub = client.books.publishers
7
8 # 尋找使用者輸入書名是否存在
9 bk = bk_info.find_one({"name":book_name})
10 if bk:
11     print("書名：" , bk["name"])
12     print("出版社：" , bk["publisher"])
13
14 # 用書籍資訊 book_info 裡的資訊跨資料集查詢
15 pub_info = pub.find_one({"name":bk["publisher"]})
16 if pub_info:
17     print("官方網址：" , pub_info["URL"])
18 else:
19     print("查無此書")
```





資料庫的匯入：尋找資料



至「政府資料開放平臺」，尋找 YouBike 2.0 即時資訊

政府資料開放平臺 DATA.GOV.TW 網站導覽 X f Language 小幫手 線上客服 會員登入

... 資料集 高應用價值主題專區 資料故事館 互動專區 消息專區 諒詢小組 授權條款 關於平臺

家 / 資料集 / YouBike2.0臺北市公共自行車即時資訊

YouBike2.0臺北市公共自行車即時資訊

YouBike臺北市公共自行車即時資訊介接網址為https://tcgbusfs.blob.core.windows.net/dotapp/youbike/v2/youbike_immediate

評分此資料集：
☆☆☆☆☆ 平均 4.15 (40 人次投票)

資料資源欄位 sno、sna、tot、sbi、sarea、infoTime、infoDate

檔案格式 JSON

編碼格式 **UTF-8** 注意一下文字編碼格式

主要欄位說明 *粗體欄位為資料標準欄位 sno、sna、tot、sbi、sarea、mday、lat infoDate

資料資源下載網址 [JSON](#) [檢視資料](#) YouBike2.0

提供機關 臺北市政府交通局

提供機關聯絡人姓名 陳心思 (02-27208889#6877)

更新頻率 每1分

授權方式 政府資料開放授權條款-第1版

計費方式 免費

資料預覽(僅摘錄前10列資料)

sno	sna	tot	sbi	sarea	mday	lat	lng	ar	sareaen	snaen
500101001	YouBike 2.0_捷運科技大樓站	28	11	大安區	2024-02-22 15:43:19	25.02605	121.5436	復興南路二段235號前	Daan Dist.	YouBike 2.0_MRT Technol ogy Bld g. Sta.
500101002	YouBike 2.0_復興南路二段273號前	21	6	大安區	2024-02-22 15:38:19	25.02565	121.54357	復興南路二段273號西側	Daan Dist.	YouBike 2.0_No. 273, Sec. 2, Fuxing S. Rd.

亦可預覽一下資料

注意一下文字編碼格式
拷貝此一網址



資料庫的匯入：讀取原始資料

```
1 # 至「政府資料開放平台」取得 YouBike 站台資料 https://data.gov.tw/dataset/137993
2 json_url = "https://tcgbusfs.blob.core.windows.net/dotapp/youbike/v2/youbike_immediate.json"
3
4 # 下載 JSON 檔案
5 import urllib.request as req
6
7 with req.urlopen(json_url) as f:           ← 將 URL 所指的檔案，以變數 f 代替
8     json_string = f.read().decode("utf-8")    ← 將檔案內容，以 UTF-8 編碼格式讀入
9     print(json_string)                      ← 印出讀到的字串
```

```
[{"sno": "500101001", "sna": "YouBike2.0_捷運科技大樓站", "tot": 28, "sbi": 6, "sarea": "大安區", "mday": "2024-03-30 16:35:19", "lat": 25.02605, "lng": 121.5436, "ar": "復興南路二段235號前", ...}]
```

raw string





隨堂練習：讀入原始資料



- 請先撰寫好下列原始碼，並且執行看看：

```
1 # 至「政府資料開放平台」取得 YouBike 站台資料 https://data.gov.tw/dataset/137993
2 json_url = "https://tcgbusfs.blob.core.windows.net/dotapp/youbike/v2/youbike_immediate.json"
3
4 # 下載 JSON 檔案
5 import urllib.request as req
6
7 with req.urlopen(json_url) as f:
8     json_string = f.read().decode("utf-8")
9     print(json_string)
```





資料庫的匯入：轉為 Python 字典



1

```
1 # 將 JSON 字串轉換成 Python 字典 dict
2 import json
3 json_dict = json.loads(json_string)
4 Python dictionary
5 # 印出字典內容驗證轉換成功
6 for item in json_dict:
7     print("站點位置:", item["sna"])
8     print(f"(經度, 緯度) = ({item['lng']}, {item['lat']})")
9     print("容量:", item["tot"])
10    print("已停:", item["sbi"])
11    print("可停:", item["bemp"])
12    print("更新時間:", item["mday"])
13    print("-----")
```

2

```
站點位置 : YouBike2.0_長春龍江路口
(經度, 緯度) = (121.54026, 25.05473)
容量 : 24
已停 : 4
可停 : 20
更新時間 : 2024-03-31 23:59:18
-----
```

```
[{'sno': '500101001',
 'sna': 'YouBike2.0_捷運科技大樓站',
 'tot': 28,
 'sbi': 5,
 'sarea': '大安區',
 'mday': '2024-04-01 00:01:15',
 'lat': 25.02605,
 'lng': 121.5436,
 'ar': '復興南路二段235號前',
 'sareaen': 'Daan Dist.',
 'snaen': 'YouBike2.0_MRT Technology Bldg. Sta.',
 'aren': 'No.235 · Sec. 2 · Fuxing S. Rd.',
 'bemp': 23,
 'act': '1',
 'srcUpdateTime': '2024-04-01 00:03:24',
 'updateTime': '2024-04-01 00:03:51',
 'infoTime': '2024-04-01 00:01:15',
 'infoDate': '2024-04-01'},
 {'sno': '500101002',
 'sna': 'YouBike2.0_復興南路二段273號前',
 ...
 }]
```





隨堂練習：轉為 Python 字典



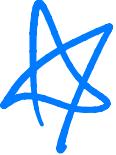
- 請先撰寫好下列原始碼，並且執行看看：

```
1 # 將 JSON 字串轉換成 Python 字典 dict
2 import json
3 json_dict = json.loads(json_string)
4
5 # 印出字典內容驗證轉換成功
6 for item in json_dict:
7     print("站點位置：" , item["sna"])
8     print(f"(經度, 緯度) = ({item['lng']}, {item['lat']})")
9     print("容量：" , item["tot"])
10    print("已停：" , item["sbi"])
11    print("可停：" , item["bemp"])
12    print("更新時間：" , item["mday"])
13    print("-----")
```

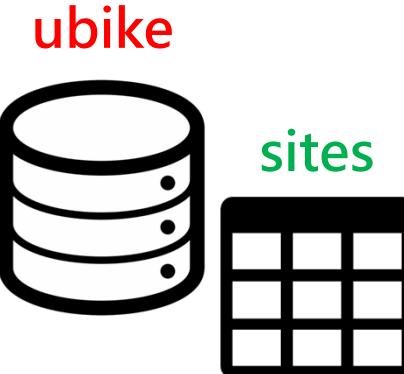




資料庫的匯入：寫入資料庫



```
1 # 將站點資訊存入 MongoDB 資料庫中
2 from pymongo import MongoClient
3 client = MongoClient()
4
5 # 建立一個 ubike 資料庫與 sites 資料集
6 ubike_sites = client.ubike.sites
7     ↴ collection
8 # 檢查資料是一筆還是很多筆
9 if isinstance(json_dict, list):
10    ubike_sites.insert_many(json_dict)
11 else:
12    ubike_sites.insert_one(json_dict)
13
14 # 將資料從 MongoDB 撈出來，驗證寫入成功
15 cursor = ubike_sites.find()
16 for document in cursor:
17    print(document)
```



每筆會被系統自動插入 '_id' 以資辨別

```
[{'_id': ObjectId('66098e64945b7348c6337e2a'), 'sno': '500113079', ...}
 {'_id': ObjectId('66098e64945b7348c6337e2b'), 'sno': '500113080', ...}
 ...
 ]
```





隨堂練習：寫入資料庫



- 請先撰寫好下列原始碼，並且執行看看：

```
1 # 將站點資訊存入 MongoDB 資料庫中
2 from pymongo import MongoClient
3 client = MongoClient()
4
5 # 建立一個 ubike 資料庫與 sites 資料集
6 ubike_sites = client.ubike.sites
7
8 # 檢查資料是一筆還是很多筆
9 if isinstance(json_dict, list):
10    ubike_sites.insert_many(json_dict)
11 else:
12    ubike_sites.insert_one(json_dict)
13
14 # 將資料從 MongoDB 撈出來，驗證寫入成功
15 cursor = ubike_sites.find()
16 for document in cursor:
17    print(document)
```





資料庫的匯入：資料庫查詢

1 # 從資料庫找出仍有空位的站台

```
2 cursor = ubike_sites.find({"bemp": {"$gt": 0}})
```

2 # 將這些站台印出來驗證一下

```
5 for document in cursor:  
    # Show "站點位置", "可停", and "更新時間"  
    print("站點位置：" , document["sna"])  
    print("可停：" , document["bemp"])  
    print("更新時間：" , document["mday"])  
    print("-----")
```

3 # 計算有幾個站台是仍有空位的

```
13 count = ubike_sites.count_documents({"bemp": {"$gt": 0}})  
14 print("總共可停站點：" , count)
```

>

Operator	Description	Operator	Description
\$eq	Equal to	\$gte	Greater than or equal to
\$gt	Greater than	\$lt	Less than
\$ne	Not equal to	\$lte	Less than or equal to
\$in	In a set	\$nin	Not in a set

站點位置 : YouBike2.0_忠孝東路三段217巷口
可停 : 34
更新時間 : 2024-01-14 04:19:10

總共可停站點 : 1340





隨堂練習：資料庫查詢



- 請先撰寫好下列原始碼，並且執行看看：

```
1 # 從資料庫找出仍有空位的站台
2 cursor = ubike_sites.find({"bemp": {"$gt": 0}})
3
4 # 將這些站台印出來驗證一下
5 for document in cursor:
6     # Show "站點位置", "可停", and "更新時間"
7     print("站點位置：" , document["sna"])
8     print("可停：" , document["bemp"])
9     print("更新時間：" , document["mday"])
10    print("-----")
11
12 # 計算有幾個站台是仍有空位的
13 count = ubike_sites.count_documents({"bemp": {"$gt": 0}})
14 print("總共可停站點：" , count)
```





網頁讀取

網路爬蟲

範例完整原始碼：
<https://url.cc/Xe3Id4>





網頁底層表示法 - HTML

批踢踢實業坊 > 看板 Gossiping

聯絡資訊 關於我們

看板

精華區

最舊

上頁

下頁

最新

8 [新聞] 護國神山再擴廠？傳台積電將在高雄增2座 A14廠
mark751019

4/01 ...

2 [問卦] 新安東京這波要賠多少
nickyeh779

4/01 ...

Re: [問卦] 魏德聖拍三體 要花多少錢才拍得出來？
autokey

4/01 ...

2 Re: [新聞] 高雄大樹居民反映出現大量蚊子 衛生局：
goatcheese

4/01 ...

```
<div class= "r-ent" >
  <div class= "title" >
    <a> [新聞] 護國神山再擴廠？傳台積電將在高雄增2座 A14廠 </a>
  </div>
</div>  <div class= "r-ent" > → <div class= "title" > → <a>
```

```
1 <div class="r-ent">
2   <div class="nrec"><span class="hl f2">8</span></div>
3   <div class="title">
4     <a href="/bbs/Gossiping/M.1711962911.A.DFA.html">[新聞] 護國神山再擴廠？傳台積電將在高雄增2座 A14廠 </a>
5   </div>
6   <div class="meta">
7     <div class="author">mark751019</div>
8     <div class="article-menu">
9       <div class="trigger">⋮</div>
10      <div class="dropdown">
11        <div class="item"><a href="/bbs/Gossiping/search?q=thread%3A%5B%E6%96%B0%E8%81%9E%5D&#43;%E8%AD%B7%E5%9C%8B%E7%A5%9E%E5%86%8D%E6%93%84%E5%BB%A0%EF%BC%9F%E5%82%B3%E5%8F%B0%E7%A9%8D%E9%9B%BB%E5%80%87%E5%9C%A8%E9%AB%88%E9%9B%84%E5%A2%9E%2E%5%BA%A7&#43;">搜尋同標題文章</a></div>
12        <div class="item"><a href="/bbs/Gossiping/search?q=author%3Amark751019">搜尋看板內 mark751019 的文章</a></div>
13      </div>
14    </div>
15    <div class="date"> 4/01</div>
16    <div class="mark"></div>
17  </div>
18 </div>

20 <div class="r-ent">
21   <div class="nrec"><span class="hl f2">2</span></div>
22   <div class="title">
23     <a href="/bbs/Gossiping/M.1711962924.A.818.html">[問卦] 新安東京這波要賠多少 </a>
24   </div>
25   <div class="meta">
26     <div class="author">nickyeh779</div>
27     <div class="article-menu">
28       <div class="trigger">⋮</div>
29       <div class="dropdown">
30         <div class="item"><a href="/bbs/Gossiping/search?q=thread%3A%5B%E5%95%8F%E5%8D%A6%5D&#43;%E6%96%B0%E5%AE%89%E6%9D%B1%E4%BA%AC%E9%80%99%E6%B3%A2%E8%A6%81%E8%B3%A0%E5%A4%9A%E5%80%91">搜尋同標題文章</a></div>
31         <div class="item"><a href="/bbs/Gossiping/search?q=author%3Anickyeh779">搜尋看板內 nickyeh779 的文章</a></div>
32       </div>
33     </div>
34     <div class="date"> 4/01</div>
35     <div class="mark"></div>
36   </div>
37 </div>
```





尋找對應 HTML 標籤之法



The diagram illustrates a workflow for identifying corresponding HTML tags. On the left, a screenshot of a forum page titled "批踢踢實業坊 > 看板 Gossiping" shows several posts. One post by "mark751019" is highlighted. On the right, the "Elements" tab of the Chrome DevTools is shown, with the "Elements" button highlighted by a red box. The DevTools interface displays the HTML structure of the page, with the same post from "mark751019" selected. A white hand cursor is positioned over the DevTools interface, pointing towards the highlighted element in the tree view.

批踢踢實業坊 > 看板 Gossiping

聯絡資訊 關於我們

div.r-ent 806 x 59.75

最舊 上頁 下頁 最新

8 [新聞] 護國神山再擴廠？傳台積電將在高雄增2座 A14廠
mark751019 4/01 ...

2 [問卦] 新安東京這波要賠多少
nickyeh779

Re: [問卦] 魏德聖拍三體 要花多少錢才拍得出來？
autokey

2 Re: [新聞] 高雄大樹居民反映出現大量蚊子 衛生局：
goatcheese

F12

DevTools is now available in Chinese!

Always match Chrome's language Switch DevTools to Chinese Don't show again

Elements Console Sources Network Performance

```
<div class="r-ent">...</div>
<div class="r-ent">...</div>
<div class="r-ent">...</div>
...
<div class="r-ent"> == $0
  <div class="nrec">...</div>
  <div class="title">
    <a href="https://www.ptt.cc/bbs/Gossiping/11.A.DFA.html">[新聞] 護國神山再擴廠？傳台積電將在高雄增2座 A14廠 </a>
  </div>
  <div class="meta">...</div>
</div>
<div class="r-ent">...</div>
<div class="r-ent">...</div>
<div class="r-ent">...</div>
<div class="r-ent">...</div>
<div class="r-ent">...</div>
```



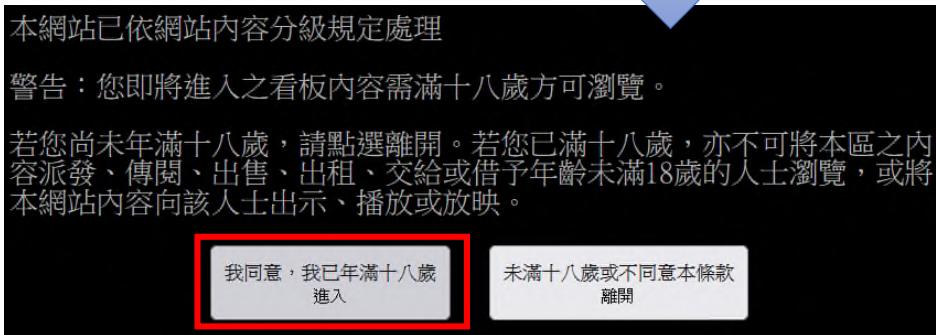


抓取 PTT 八卦版原始內容

1

```
1 # 爬蟲設定
2 import requests
3 ptt_url = "https://www.ptt.cc/bbs/Gossiping/index.html"
4
5 # 開始抓取
6 ptt_text = requests.get(ptt_url, cookies={'over18': '1'}).text
7 print(ptt_text)
```

2



```
批踢踢 業坊 > 看板 Gossiping
聯絡資訊 | 關於我們
看板 精華區 最舊 上頁 下頁 最新
8 [新聞] 護國神山再擴廠？傳台積電將在高雄增2座 A14廠
mark751019 4/01 ...
1 <div class="r-ent">
2   <div class="rec"><span class="hl f2">B</span></div>
3   <div class="title">
4     <a href="/bbs/Gossiping/M.1711962911.A.DFA.html">[新聞] 護國神山再擴廠？傳台積電將在高雄增2座 A14廠 </a>
5   </div>
6   <div class="meta">
7     <div class="author">mark751019</div>
8     <div class="article-menu">
9       <div class="trigger">&#x22ef;</div>
10      <div class="dropdown">
11        <div class="item"><a href="/bbs/Gossiping/search?q=thread%3A%5B%E6%96%B0%E8%81%9E%5D&#43;%E8%AD%87%E5%9C%8B%7%A5%9E%5E5%80%6%93%4%5%8B%AO%EF%BC%9F%5%82%3%5%8F%80%7%9E%5%87%E5%9C%AB%6%98%4%5%8A%2%9E2%5%8A%7%43%A14%5%88%A0%43%>搜尋同標題文章</a></div>
12        <div class="item"><a href="/bbs/Gossiping/search?q=author%3Amark751019">搜尋看板內 mark751019 的文章</a></div>
13      </div>
14    </div>
15    <div class="date"> 4/01 </div>
16    <div class="mark"></div>
17  </div>
18 </div>
```

A



隨堂練習：抓取原始內容



- 請先撰寫好下列原始碼，並且執行看看：

```
1 # 爬蟲設定
2 import requests
3 ptt_url = "https://www.ptt.cc/bbs/Gossiping/index.html"
4
5 # 開始抓取
6 ptt_text = requests.get(ptt_url, cookies={'over18': '1'}).text
7 print(ptt_text)
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>看板 Gossiping 文章列表 - 批踢踢實業坊</title>

    <link rel="stylesheet" type="text/css" href="//images.ptt.cc/bbs/v2.27/bbs-common.css">
    <link rel="stylesheet" type="text/css" href="//images.ptt.cc/bbs/v2.27/bbs-base.css" media="screen">
    <link rel="stylesheet" type="text/css" href="//images.ptt.cc/bbs/v2.27/bbs-custom.css">
    <link rel="stylesheet" type="text/css" href="//images.ptt.cc/bbs/v2.27/pushstream.css" media="screen">
    <link rel="stylesheet" type="text/css" href="//images.ptt.cc/bbs/v2.27/bbs-print.css" media="print">
```





整理成 JSON 格式



```

1 # 剖析抓到的原始資料
2 from bs4 import BeautifulSoup 引入 HTML 剖析函式庫
3 import json
4
5 soup = BeautifulSoup(ptt_text, 'html.parser') 告知現在要剖析的是 HTML 檔案
6 articles = [] (不是 XML 或其它格式 )
7
8 # 找所有標籤是 <div class= "r-ent" > 的 HTML 元素
9 for div in soup.find_all("div", class_="r-ent"):
10    try:
11        # 標題放在 <div class="title" > ~ </div> 之間
12        3 title = div.find("div", class_="title").text.strip()
13        # 網址放在 <a> 裡面的 href 屬性內，且缺乏首頁網址
14        4 href = "https://www.ptt.cc" + div.find("a")['href']
15        # 作者放在 <div class="author" > ~ </div> 之間
16        5 author = div.find("div", class_="author").text.strip()
17        # 日期放在 <div class="date" > ~ </div> 之間
18        6 date = div.find("div", class_="date").text.strip()
19        # 推文數放在 <div class="nrec" > ~ </div> 之間，且需轉成整數
20        7 push_count = div.find("div", class_="nrec").text.strip()
21        push_count = int(push_count) if push_count.isdigit() else 0

```

.text : remove tags
.strip(): removes leading & trailing whitespaces

```

1 <div class="r-ent">
2   <div class="nrec"><span class="hl f2">8</span></div>
3   <div class="title">
4     <a href="https://www.ptt.cc/bbs/Gossiping/M.1711962911.A.DFA.html">
5       [新聞] 護國神山再擴廠？傳台積電將在高雄增2座 A14廠
6     </a>
7   </div>
8   <div class="meta">
9     <div class="author">mark751019</div>
10    ...
11  </div>
12  <div class="date"> 4/01</div>
13  <div class="mark"></div>
14 </div>
15

```

PPT :
沒人推文為空白(非 0)





整理成 JSON 格式

1 23 # 將抓到的內容，存入 articles 串列中
24 articles.append({
25 "title": title,
26 "href": href,
27 "author": author,
28 "push_count": push_count,
29 "date": date
30 })
31 except:
32 continue
33
2 34 # 將 articles 串列，轉換成 JSON 格式印出
35 print(json.dumps(articles, indent=2, ensure_ascii=False))

內縮 2 個空白 當 ASCII > 127
不要編碼成 "\u4F60"
而是直接輸出原字 “你”

```
[  
 {  
     "title": "[問卦] 張學友演唱會沒有清票代表什麼？",  
     "href": "https://www.ptt.cc/bbs/Gossiping/M.1711787192.A.8BD.html",  
     "author": "rs6677",  
     "push_count": 4,  
     "date": "3/30"  
 },  
 {  
     "title": "[問卦] 朋友媽媽去買蕃茄丁 該怎麼辦？",  
     "href": "https://www.ptt.cc/bbs/Gossiping/M.1711787262.A.777.html",  
     "author": "HNCB2880",  
     "push_count": 1,  
     "date": "3/30"  
 },  
 ...  
 ]
```





隨堂練習：整理成 JSON 格式



- 請先撰寫好下列原始碼，並且執行看看：

```
1 # 剖析抓到的原始資料
2 from bs4 import BeautifulSoup
3 import json
4
5 soup = BeautifulSoup(ptt_text, 'html.parser')
6 articles = []
7
8 # 每篇文章都放在 <div class="r-ent"> ~ </div> 之間
9 for div in soup.find_all("div", class_="r-ent"):
10     try:
11         # 標題放在 <div class="title"> ~ </div> 之間
12         title = div.find("div", class_="title").text.strip()
13         # 網址放在 <a> 裡面的 href 屬性內，且缺乏首頁網址
14         href = "https://www.ptt.cc" + div.find("a")['href']
15         # 作者放在 <div class="author"> ~ </div> 之間
16         author = div.find("div", class_="author").text.strip()
17         # 日期放在 <div class="date"> ~ </div> 之間
18         date = div.find("div", class_="date").text.strip()
19         # 推文數放在 <div class="nrec"> ~ </div> 之間，且需轉成整數
20         push_count = div.find("div", class_="nrec").text.strip()
21         push_count = int(push_count) if push_count.isdigit() else 0
23     # 將抓到的內容，存入 articles 串列中
24     articles.append({
25         "title": title,
26         "href": href,
27         "author": author,
28         "push_count": push_count,
29         "date": date
30     })
31     except:
32         continue
33
34 # 將 articles 串列，轉換成 JSON 格式印出
35 print(json.dumps(articles, indent=2, ensure_ascii=False))
```





將結果存入 MongoDB

• 環境設定 (與之前做的相同)

```
1 # 下載免費開源的 MongoDB 社群版
2 !wget -qO - https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-ubuntu2204-7.0.7.tgz | tar -xv
3 !mv mongodb-linux-x86_64-ubuntu2204-7.0.7/bin/* /usr/bin/
4
5 # 下載與 MongoDB 版本相應的金鑰，保證 MongoDB 連線的安全
6 !wget https://www.mongodb.org/static/pgp/server-7.0.asc
7 !sudo apt-key add server-7.0.asc
8
9 # 建造 MongoDB 資料存放處（路徑、名稱可自訂）
10 !mkdir -p /content/data/db
11
12 # 啟動 MongoDB 伺服器（Mongo Daemon）於本地端（localhost）
13 !mongod --fork --logpath /var/log/mongodb.log --dbpath /content/data/db --bind_ip localhost
14
15 # 確認 MongoDB 正在執行中
16 !mongod --sysinfo
17
18 # 安裝 PyMongo 函式庫
19 !pip install pymongo
```





將結果存入 MongoDB

```
1 # 將站點資訊存入 MongoDB 資料庫中
2 from pymongo import MongoClient
3 client = MongoClient()
4 gossip = client.ptt.gossiping
5
6 if isinstance(articles, List):
7     gossip.insert_many(articles)
8 else:
9     gossip.insert_one(articles)

11 # 從資料庫找出有任何推文的帖子 >
12 cursor = gossip.find({"push_count": {"$gt": 0}})
13
14 # 將這些帖子印出來驗證一下
15 for document in cursor:
16     # Show "標題", "網址", and "推文數"
17     print("標題:", document["title"])
18     print("網址:", document["href"])
19     print("推文數:", document["push_count"])
20     print("-----")
21
22 # 計算總共有幾篇帖子推文數 > 0 的
23 count = gossip.count_documents({"push_count": {"$gt": 0}})
24 print("有推文帖子總數:", count)
```

標題：[問卦] 張學友演唱會沒有清票代表什麼？
網址：<https://www.ptt.cc/bbs/Gossiping/M.1711787192.A.8BD.html>
推文數：4

標題：[問卦] 朋友媽媽去買蕃茄丁 該怎麼辦？
網址：<https://www.ptt.cc/bbs/Gossiping/M.1711787262.A.777.html>
推文數：1

有推文帖子總數：19





隨堂練習：存入 MongoDB



- 請先撰寫好前兩頁的「環境設定」原始碼，再撰寫下列原始碼，並且通通執行看看：

```
1 # 將站點資訊存入 MongoDB 資料庫中
2 from pymongo import MongoClient
3 client = MongoClient()
4 gossip = client.ptt.gossiping
5
6 if isinstance(articles, list):
7     gossip.insert_many(articles)
8 else:
9     gossip.insert_one(articles)
11 # 從資料庫找出有任何推文的帖子
12 cursor = gossip.find({"push_count": {"$gt": 0}})
13
14 # 將這些帖子印出來驗證一下
15 for document in cursor:
16     # Show "標題", "網址", and "推文數"
17     print("標題:", document["title"])
18     print("網址:", document["href"])
19     print("推文數:", document["push_count"])
20     print("-----")
21
22 # 計算總共有幾篇帖子推文數 > 0 的
23 count = gossip.count_documents({"push_count": {"$gt": 0}})
24 print("有推文帖子總數:", count)
```





各式檔案讀取

文字檔、PDF、Word

範例完整原始碼：
<https://url.cc/qKYBBi>





本範例要讀取的三個檔案



- 1 BBC 以 AI 來取代女演員
- 2
- 3 英國廣播公司（BBC）決定在即將來臨的一個專案中，以人工智慧（AI）取代女演員莎拉 · 波伊澤（Sara Poyzer），引發了英國演藝界及其他地方的廣泛辯論和爭議。莎拉 · 波伊澤因在舞台劇《媽咪咪呀！》中的角色而廣受人知，已在該劇中演出超過十年。她分享了一封來自未具體指明 BBC 專案製作團隊的電子郵件截圖。郵件告知她，BBC 已批准在該專案中使用 AI 生成的聲音，因此她不再需要提供服務。



- 4
- 5 BBC 後來澄清，使用 AI 是為了一部「高度敏感的紀錄片」，其中有一位即將結束生命且無法言語表達的貢獻者是考慮到家屬的意願，旨在重新創建貢獻者的聲音，用於紀錄片的一個簡短部分。

BBC 以 AI 來取代女演員



- 6
- 7 波伊澤被 AI 取代的消息引發了關於 AI 在創意產業中的影響的更廣泛討論，特別是對演員和配音藝術家的影響。米爾托斯是一位因在《冰與火之歌：權力遊戲》中的角色而知名的演員，他對這一趨勢表示抗議。同樣，波伊澤的配音代理公司 Voice Squad 對 BBC 對該行業的潛在危害以及對技術優秀演員工作的貶低。
- 8
- 9 此事件還再次引發了有關在媒體製作中使用 AI 的道德和實踐考量的討論。BBC 強調不會犧牲人類的創意控制，並將支持權利持有者，同時保持編輯標準。然而，將此決定引發了有關技術進步與保護人類創造力和藝術就業之間平衡的問題。

英國廣播公司（BBC）決定在即將來臨的一個專案中，以人工智慧（AI）取代女演員莎拉 · 波伊澤（Sara Poyzer），引發了英國演藝界及其他地方的廣泛辯論和爭議。莎拉 · 波伊澤因在舞台劇《媽咪咪呀！》中的角色而廣受人知，已在該劇中演出超過十年。她分享了一封來自未具體指明 BBC 專案製作團隊的電子郵件截圖。郵件告知她，BBC 已批准在該專案中使用 AI 生成的聲音，因此她不再需要提供服務。

BBC 以 AI 來取代女演員



- 10
- 11 這一情況反映了娛樂業對 AI 角色的角色的討論，類似於好萊塢 SAG-AFTRA 演員罷工期間提出的擔憂。罷工的焦點是確保演員就 AI 使用其肖像和表演而獲得補償和同意，突顯了在數字技術時代，創意專業人士需要

英國廣播公司（BBC）決定在即將來臨的一個專案中，以人工智慧（AI）取代女演員莎拉 · 波伊澤（Sara Poyzer），引發了英國演藝界及其他地方的廣泛辯論和爭議。莎拉 · 波伊澤因在舞台劇《媽咪咪呀！》中的角色而廣受人知，已在該劇中演出超過十年。她分享了一封來自未具體指明 BBC 專案製作團隊的電子郵件截圖。郵件告知她，BBC 已批准在該專案中使用 AI 生成的聲音，因此她不再需要提供服務。

BBC 後來澄清，使用 AI 是為了一部「高度敏感的紀錄片」，其中有一位即將結束生命且無法言語表達的貢獻者。決定使用 AI 是考慮到家屬的意願，旨在重新創建貢獻者的聲音，用於紀錄片的一個簡短部分。在影片中，將清楚標註此 AI 使用。

←

BBC 後來澄清，使用 AI 是為了一部「高度敏感的紀錄片」，其中有一位即將結束生命且無法言語表達的貢獻者。決定使用 AI 是考慮到家屬的意願，旨在重新創建貢獻者的聲音，用於紀錄片的一個簡短部分。在影片中，將清楚標註此 AI 使用。←





讀取 TXT 檔

```
1 # 下載 TXT 檔案
2 import os
3 Dataset_File = "AI_News.txt"
4
5 if not os.path.isfile(Dataset_File):
6     os.system("wget https://raw.githubusercontent.com/cnchi/datasets/master/" + Dataset_File)
7
8 # 打開、讀取並顯示 AI_News.txt 中的內容
9 with open(Dataset_File, "r") as f: // 打開
10    content = f.read() // 讀取
11    print(content)
```





隨堂練習：讀取 TXT 檔



- 請先撰寫好下列原始碼，並且執行看看：

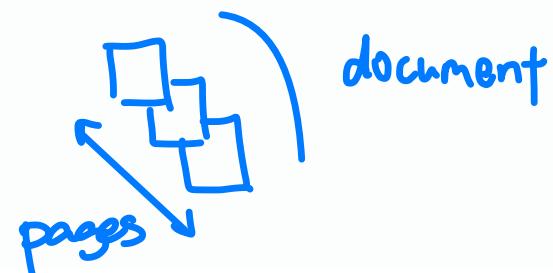
```
1 # 下載 TXT 檔案
2 import os
3 Dataset_File = "AI_News.txt"
4
5 if not os.path.isfile(Dataset_File):
6     os.system("wget https://raw.githubusercontent.com/cnchi/datasets/master/" + Dataset_File)
7
8 # 打開、讀取並顯示 AI_News.txt 中的內容
9 with open(Dataset_File, "r") as f:
10     content = f.read()
11     print(content)
```





讀取 PDF 檔

```
1 # 下載 PDF 檔案
2 import os
3 Dataset_File = "AI_News.pdf"
4
5 if not os.path.isfile(Dataset_File):
6     os.system("wget https://raw.githubusercontent.com/cnchi/datasets/master/" + Dataset_File)
7
8 # 安裝 PyMuPDF 套件
9 !pip install PyMuPDF
10
11 # 打開、讀取並顯示 AI_News.pdf 中的內容
12 import fitz
13
14 with fitz.open(Dataset_File) as doc:
15     content = ""
16     for page in doc:
17         content += page.get_text()
18     print(content)
```





隨堂練習：讀取 PDF 檔



- 請先撰寫好下列原始碼，並且執行看看：

```
1 # 下載 PDF 檔案
2 import os
3 Dataset_File = "AI_News.pdf"
4
5 if not os.path.isfile(Dataset_File):
6     os.system("wget https://raw.githubusercontent.com/cnchi/datasets/master/" + Dataset_File)
7
8 # 安裝 PyMuPDF 套件
9 !pip install PyMuPDF
10
11 # 打開、讀取並顯示 AI_News.pdf 中的內容
12 import fitz
13
14 with fitz.open(Dataset_File) as doc:
15     content = ""
16     for page in doc:
17         content += page.get_text()
18     print(content)
```





讀取 MS-Word 檔



```
1 # 下載 MS-Word 檔案
2 import os
3 Dataset_File = "AI_News.docx"
4
5 if not os.path.isfile(Dataset_File):
6     os.system("wget https://raw.githubusercontent.com/cnchi/datasets/master/" + Dataset_File)
7
8 # 安裝 python-docx 套件
9 !pip install python-docx
10
11 # 打開、讀取並顯示 AI_News.pdf 中的內容
12 from docx import Document
13
14 doc = Document(Dataset_File)
15 content = ""
16 for para in doc.paragraphs:
17     content += para.text
18 print(content)
```

paragraph





隨堂練習：讀取 MS-Word 檔

- 請先撰寫好下列原始碼，並且執行看看：

```
1 # 下載 MS-Word 檔案
2 import os
3 Dataset_File = "AI_News.docx"
4
5 if not os.path.isfile(Dataset_File):
6     os.system("wget https://raw.githubusercontent.com/cnchi/datasets/master/" + Dataset_File)
7
8 # 安裝 python-docx 套件
9 !pip install python-docx
10
11 # 打開、讀取並顯示 AI_News.pdf 中的內容
12 from docx import Document
13
14 doc = Document(Dataset_File)
15 content = ""
16 for para in doc.paragraphs:
17     content += para.text
18 print(content)
```





多媒體文字讀取

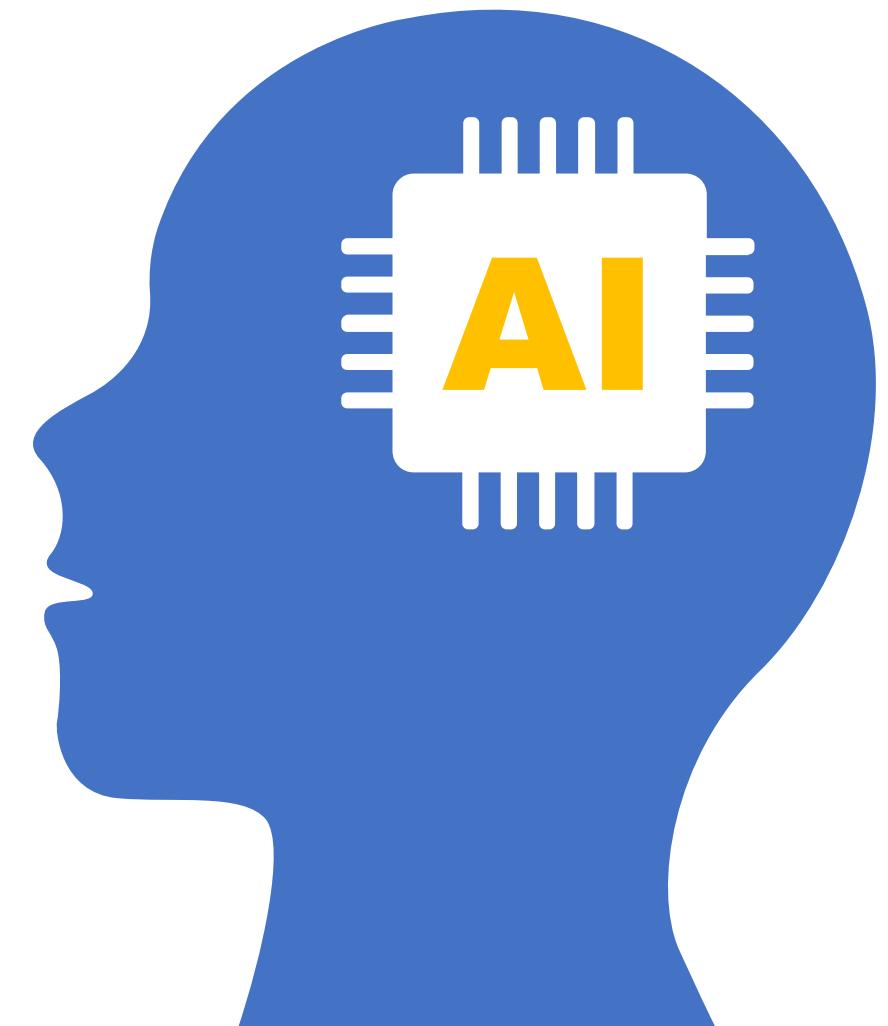
讀取照片文字、影片逐字稿



多媒體文字讀取

- 讀取照片上的文字
- 取得影片逐字稿

範例完整原始碼：
<https://lurl.cc/cuxfOR>





本範例使用的照片



：文字



Optical
Character
Recognition





環境設定

```
1 # 下載照片檔 (Image)
2 import os
3 Dataset_File = "Frieren.jpg"
4
5 if not os.path.isfile(Dataset_File):
6     os.system("wget https://raw.githubusercontent.com/cnchi/datasets/master/" + Dataset_File)
7
8 # 安裝 EasyOCR 套件
9 !pip install easyocr
```

簡單，還不錯用





辨識照片上的文字

```
1 import easyocr 引入 EasyOCR 套件
2
3 # 指定辨識繁體中文與英文
4 reader = easyocr.Reader(['ch_tra', 'en'])
5
6 # 會傳回來一個 Python List, 內含辨識到的每一組文字資訊
7 results = reader.readtext(Dataset_File)
8
9 # 每一組文字資訊, 又包含「文字座標」、「文字本身」、「信心度」
10 for items in results:
11     print("座標:", items[0])
12     print("文字:", items[1])
13     print("信心度:", items[2])
14     print("-----")
```

Language Code	Language	Language Code	Language
'ch_sim'	Simplified Chinese	'en'	English
'ch_tra'	Traditional Chinese	'fr'	French
'de'	German	'it'	Italian
'es'	Spanish	'pt'	Portuguese
'ru'	Russian	'ar'	Arabic
'he'	Hebrew	'hi'	Hindi
'ja'	Japanese	'ko'	Korean
'th'	Thai	'vi'	Vietnamese
'zh_cn'	Chinese (Simplified)	'zh_tw'	Chinese (Traditional)





辨識結果解讀



座標 : [[713, 55], [769, 55], [769, 95], [713, 95]]

文字 : +

信心度 : 0.016972056057964702

座標 : [[331, 37], [875, 37], [875, 247], [331, 247]]

文字 : 葬*芙莉蓮

信心度 : 0.060725962437658695

座標 : [[390, 177], [766, 177], [766, 229], [390, 229]]

文字 : R T RI

信心度 : 0.01301050642022479

座標 : [[1059, 721], [1099, 721], [1099, 737], [1059, 737]]

文字 : TOHO

信心度 : 0.4000180661678314

座標 : [[1060, 738], [1120, 738], [1120, 746], [1060, 746]]

文字 : 1 nTminioT

信心度 : 0.025372240267466453

座標 : [[74, 747], [239, 747], [239, 788], [74, 788]]

文字 : MUSE木棉花代理

信心度 : 0.5268352225144601

座標 : [[769, 761], [1125, 761], [1125, 779], [769, 779]]

文字 : CKanehito Yamada Tsukasa Abe Shogakukan/ Frieren Project

信心度 : 0.43206531509795604



亦可根據「信心度」，判斷要不要採用某辨識結果





隨堂練習：讀取照片文字



- 請先撰寫好前三頁的「環境設定」原始碼，再撰寫下列原始碼，並且通通執行看看：

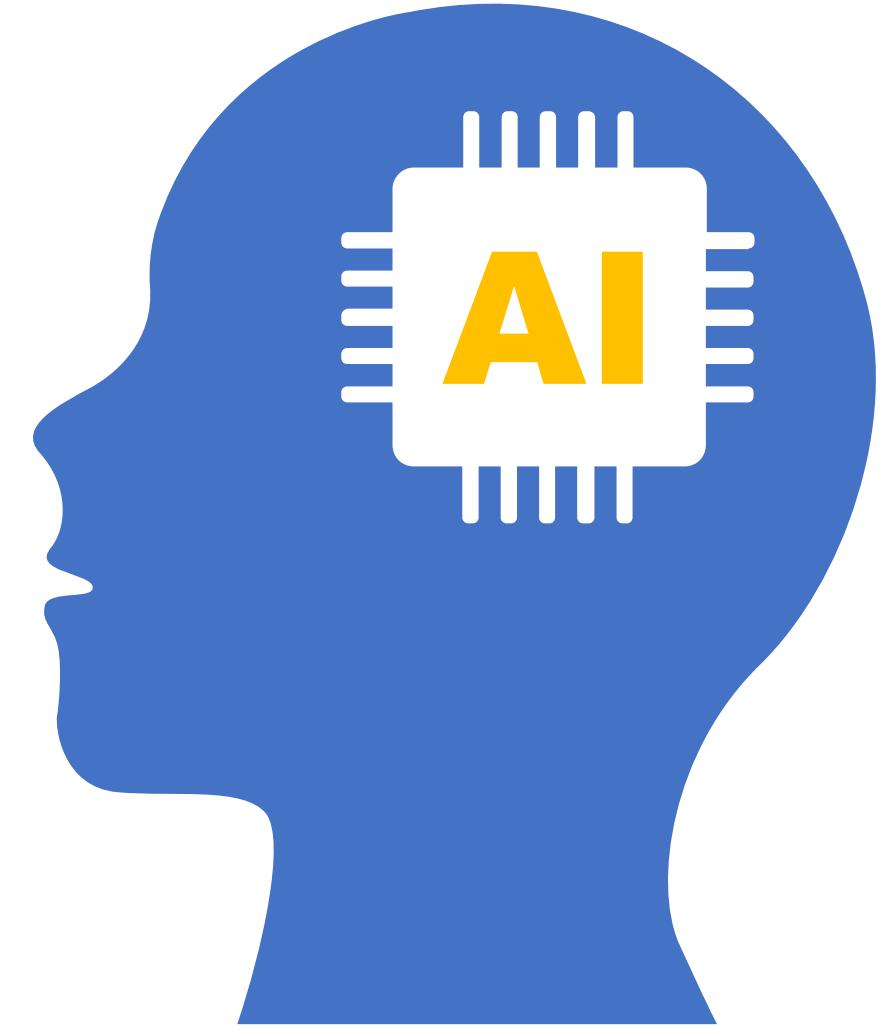
```
1 import easyocr  
2  
3 # 指定辨識繁體中文與英文  
4 reader = easyocr.Reader(['ch_tra', 'en'])  
5  
6 # 會傳回來一個 Python List, 內含辨識到的每一組文字資訊  
7 results = reader.readtext(Dataset_File)  
8  
9 # 每一組文字資訊，又包含「文字座標」、「文字本身」、「信心度」  
10 for items in results:  
11     print("座標：" , items[0])  
12     print("文字：" , items[1])  
13     print("信心度：" , items[2])  
14     print("-----")
```





多媒體文字讀取

- 讀取照片上的文字
- 取得影片逐字稿





本範例使用的影片

英AI機器人回話直逼真人！警告人類防「AI浩劫」 | TVBS新聞
(影片內含中、英兩種語言的對話)



注意！URL 格式一定要如下所示，否則抓取不到：
<https://www.youtube.com/watch?v=XXXXXX>





環境設定



- 1 # 安裝 YouTube 下載套件
2 !pip install yt-dlp
3
- 2 # 安裝聲音格式轉 .WAV 套件 (.WAV = 一種非壓縮音訊格式)
5 !pip install pydub
6
- 3 # 安裝 OpenAI 的 Whisper 語音辨識套件 (約需一分鐘)
8 !pip install git+https://github.com/openai/whisper.git

e.g. mp3 → 壓縮音訊





定義下載 & 取得聲音檔的函數



```
1 import subprocess  
2 from pydub import AudioSegment  
3  
4 def youtube_downloader(url, file_name="audio"):  
5     try:  
6         2 # 使用 yt-dlp 下載 YouTube 影片的音訊  
7         subprocess.run(['yt-dlp', '-x', '--audio-format', 'mp3', '-o', f'{file_name}.%(ext)s', url])  
8  
9         3 # 將 MP3 轉換為 WAV 格式  
10        audio = AudioSegment.from_file(file_name + '.mp3')  
11        audio.export(file_name + '.wav', format='wav')  
12  
13        4 # 輸出成功  
14        print(f"影片已經下載&轉成 {file_name}.wav")  
15  
16        5 except Exception as e:  
17            print(f"發生錯誤：{e}")
```

CLI command
引入相關套件

default argument

- subprocess: 在作業系統裡，啟動一個「子行程」
- -x: eXtract，下載 YouTube 影片之意
- --audio-format: 僅下載音源部分為 .mp3 檔 (壓縮格式)
- -o: Output, 指定輸出檔名

解壓縮
↓
The placeholder gets replaced by the actual file extension e.g. mp3





定義辨識為逐字稿的函數

```
1 import whisper  
2  
3 def transcribe_with_whisper(wav_basename="audio", model="base"):  
4     model = whisper.load_model(model)  
5     result = model.transcribe(f"{wav_basename}.wav")  
6  
7     return result["text"]
```

精準,辨識久↓

Model	Description	Optimized for	Use Case
tiny	The smallest model	General	Fast recognition, low resource requirements
tiny.en	Small model optimized for English	English	Fast recognition for English, low resource requirements
base	Basic model	General	Balanced accuracy and resource requirements
base.en	Basic model optimized for English	English	Balanced accuracy for English
small	Small model	General	General purpose, better accuracy
small.en	Small model optimized for English	English	Better accuracy for English
medium	Medium-sized model	General	Higher accuracy requirements
medium.en	Medium-sized model optimized for English	English	Higher accuracy for English
large	Large model	General	Highest accuracy requirements
large-v2	Second version of the large model	General	Further improved accuracy





主程式

```
1 # 定義下載的 URL 與檔名
2 url = "https://www.youtube.com/watch?v=C_n9QhNN3qo"
3 audio_basename = "news"
4
5 # 下載影片並轉成 .WAV 檔
6 youtube_downloader(url, file_name=audio_basename)
7
8 # 用 OpenAI Whisper 辨識出影片的逐字稿
9 text = transcribe_with_whisper(wav_basename=audio_basename, model="medium")
10
11 # 印出辨識的逐字稿
12 print(text)
```

英國研發的AI人型機器人Amika一亮相就警告人類最可怕的情況我可以想像的是AI和機器人是一個世界上機器人變得如此強大他們能夠控制或操控人類沒有他們的知識由英國科技公司開發主打互動直逼真人她說要阻止AI浩劫還來得及但要注意到有可能的危險與機器人的關係人皺眉微笑 表情超豐富因為頭部有數十個驅動器一台30萬台幣起跳最終目標是融入人類社會我希望你從未聽到Amika說對不起 我不明白請你重複問Amika應該說的是什麼現在更出現第一位AI藝術家人類的感覺沒有想法但我開心我的設計被思考幫助觀看者問候新科技的作用她創造一套家具用品靈感來自私等現代主義只是細看有洞的勺子根本裝不下整合一體很可惜有些女性玩家受傷了這些傷害也受了很大的傷害我覺得我們在這裡展示的技術絕對能幫助他們英國足球隊隊長凱恩投資科技新創公司希望透過AI技術監測身體識別弱點和受傷的可能性來減少運動員的傷害TVBS新聞 政府報導

經驗分享：
推薦使用 **medium** 模式，
兼顧**能耗**與**準確度**

← 沒有標點符號





用 ChatGPT 替文字加上標點符號

ChatGPT 3.5 ▾



You

底下有一篇文章，沒有加上標點符號。請幫我加上標點符號，使之唸的時候能夠通順。

英國研發的AI人型機器人Amika一亮相就警告人類最可怕的情況我可以想像的是AI和機器人是一個
世界上機器人變得如此強大他們能夠控制或
直逼真人她說要阻止AI浩劫還來得及但要注
富因為頭部有數十個驅動器一台30萬台幣起
對不起我不明白請你重複問Amika應該說的
但我開心我的設計被思考幫助觀看者問候新
主義只是細看有洞的勺子根本裝不下整合一
害我覺得我們在這裡展示的技術絕對能幫助
技術監測身體識別弱點和受傷的可能性來減



ChatGPT

英國研發的AI人型機器人Amika一亮相就警告人類最可怕的情況。我可以想像的是，AI和機器人是一個世界上，機器人變得如此強大，他們能夠控制或操控人類，沒有他們的知識。由英國科技公司開發，主打互動，直逼真人。她說要阻止AI浩劫還來得及，但要注意到有可能的危險與機器人的關係。人皺眉微笑，表情超豐富，因為頭部有數十個驅動器。一台30萬台幣起跳，最終目標是融入人類社會。我希望你從未聽到Amika說對不起，我不明白。請你重複問。Amika應該說的是什麼？現在更出現第一位AI藝術家。人類的感覺沒有想法，但我開心我的設計被思考幫助觀看者問候新科技的作用。她創造一套家具用品，靈感來自包豪斯等現代主義，只是細看有洞的勺子，根本裝不下整合一體。很可惜有些女性玩家受傷了，這些傷害也受了很大的傷害。我覺得我們在這裡展示的技術絕對能幫助他們。英國足球隊隊長凱恩投資科技新創公司，希望透過AI技術監測身體，識別弱點和受傷的可能性，來減少運動員的傷害。TVBS新聞，政府報導。





隨堂練習：取得逐字稿



- 請先撰寫好前五頁的「環境設定」原始碼，再撰寫下列原始碼，並且通通執行看看：





課後作業：熱門新聞爬蟲

- **題目說明**

- 利用爬蟲技術，收集網路上各大報紙的「**熱門新聞**」，並將它們存入 **MongoDB** 資料庫中。

- **建議步驟**

- 在 Google 輸入你想爬取的報社。如：「**聯合報 熱門**」。
- 記下該熱門新聞之 **URL**。如：<https://udn.com/rank/pv/2>
- 先用 **requests** 套件，將原始文字爬回來。
- 再用 **BeautifulSoup** 套件，解析原始文字，抓出每條熱門新聞的下列資訊：
 - “**title**” : 大甲媽祖剛到 嘉義新港奉天宮牌樓竟起火
 - “**url**” : <https://udn.com/news/story/123477/7884374>
 - “**date**” : 2024-04-08 14:26
- 最後再記得附上該新聞出處。如：
 - “**source**” : 聯合報
- 請至少收集兩家報社，如：聯合報、自由時報。歡迎同學自由添加報社數量。
- 最後將收集到的資訊，以 **JSON** 格式，寫入 **MongoDB** 資料庫中，再抓取出來驗證。





課後作業：熱門新聞爬蟲



- 參考輸出（以抓取「聯合報」為例）

原始文字

```
1 <!-- nav:0.0000 --><!-- top:0.0388 --><!-- bottom:0.0018 --><!-- meta:0.0002 -->
<!-- ad:0.0409 --><!-- content:0.0980 --><!-- total:0.2062 --><!-- misc rank3pv
0.2062 --><!--hottest_news 0.0016 --><!--content 0.0162 --><!--ad_1 0.0258 --><!--
ad 0.0517 --><!--view_type 0.0000 --><!--meta 0.0001 --><!--family bar 0.0008 -->
<!--family_bar sub --><!--bre: 1 [ {
  2   "date": "2024-04-08 16:08",
  3   "title": "高雄男車禍自撞「全裸」逃下車！急護重要部位...父母不解 網友全看傻",
  4   "href": "https://udn.com/news/story/7320/7884649",
  5   "source": "聯合報"
},
{
  6     "date": "2024-04-08 22:20",
  7     "title": "獲救「沒雀躍感」舉
  8     "href": "https://udn.com/
  9     "source": "聯合報"
},
{
  10    "date": "2024-04-08 14:20",
  11    "title": "大甲媽祖剛到 嘉義新港奉天宮牌樓竟起火
  12    "href": "https://udn.com/
  13    "source": "聯合報"
},
...
  14  ... (下略) ...
  15 ]
  16 ]
  17 ]
  18 ]
  19 ...
  20 ]
```

JSON

MongoDB 讀出

```
1 標題： 高雄男車禍自撞「全裸」逃下車！急護重要部位...父母不解 網友全看傻
2 日期： 2024-04-08 16:08
3 網址： https://udn.com/news/story/7320/7884649
4 來源： 聯合報
5 -----
6 標題： 獲救「沒雀躍感」舉止異常 屏東漁船失聯5天...海巡帶回調查
7 日期： 2024-04-08 22:20
8 網址： https://udn.com/news/story/7320/7885379
9 來源： 聯合報
10 -----
11 標題： 大甲媽祖剛到 嘉義新港奉天宮牌樓竟起火
12 日期： 2024-04-08 14:26
13 網址： https://udn.com/news/story/123477/7884374
14 來源： 聯合報
15 -----
16 ... (下略) ...
```

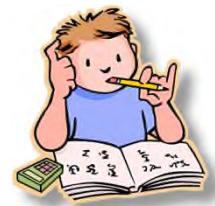




課後作業：熱門新聞爬蟲

- 參考用爬蟲原始碼（以聯合報為例）

```
1 # 爬蟲設定
2 udn_url = "https://udn.com/rank/pv/2"
3
4 # 開始抓取
5 udn_text = requests.get(udn_url).text
6 print(udn_text)
7
8 # 剖析抓到的原始資料
9 soup = BeautifulSoup(udn_text, 'html.parser')
10
11 # 每篇文章都放在 <div class="story-list__text"> ~ </div> 之間
12 for news in soup.find_all('div', class_='story-list__text'):
13     try:
14         # 發文日期放在 <time class="story-list__time"> ~ </time> 之間
15         date = news.find('time', class_='story-list__time').text.strip()
16         # 標題放在 <h2> 之下, <a> ~ </a> 之間的文字
17         title = news.find('h2').find('a').text.strip()
18         # 網址放在 <h2> 之下, <a> 裡面的 href 屬性
19         href = news.find('h2').find('a').get('href')
20         # 來源固定填「聯合報」
21         source = '聯合報'
23             # 將抓到的內容, 存入 articles 串列中
24             articles.append({
25                 'date': date,
26                 'title': title,
27                 'href': href,
28                 'source': source
29             })
30     except:
31         continue
32
33 # 將 articles 串列, 轉換成 JSON 格式印出
34 print(json.dumps(articles, indent=2, ensure_ascii=False))
```





本章總結：抓取各類資料來源



- 表格 (CSV, Excel, 網頁)
 - pandas.read_csv(), pandas.read_excel(), pandas.read_html()
- 資料庫 (RDMBS, NoSQL)
 - SQLite: sqlite3.connect(), conn.cursor(), cursor.execute(), cursor.fetchall()
 - MongoDB: PyMongo.MongoClient, .insert_one(), .insert_many(), .find()
- 網頁爬蟲
 - requests: 抓取網頁原始資料。
 - BeautifulSoup: 剖析網頁原始資料。
- 檔案 (TXT, PDF, Word)
 - TXT: open(<檔名>, "r"), .read()
 - PDF: PyMuPDF, fitz.open(<檔名>), .get_text()
 - Word: Python-Docx, Document(<檔名>), .paragraphs, .text
- 多媒體 (照片、影片)
 - 照片 : EasyOCR, .Reader(), .readtext()
 - 影片 : OpenAI Whisper, .load_model(), .transcribe()

