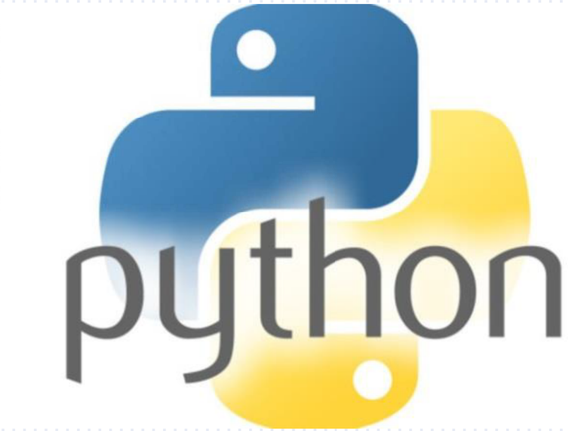
Python 資料儲存

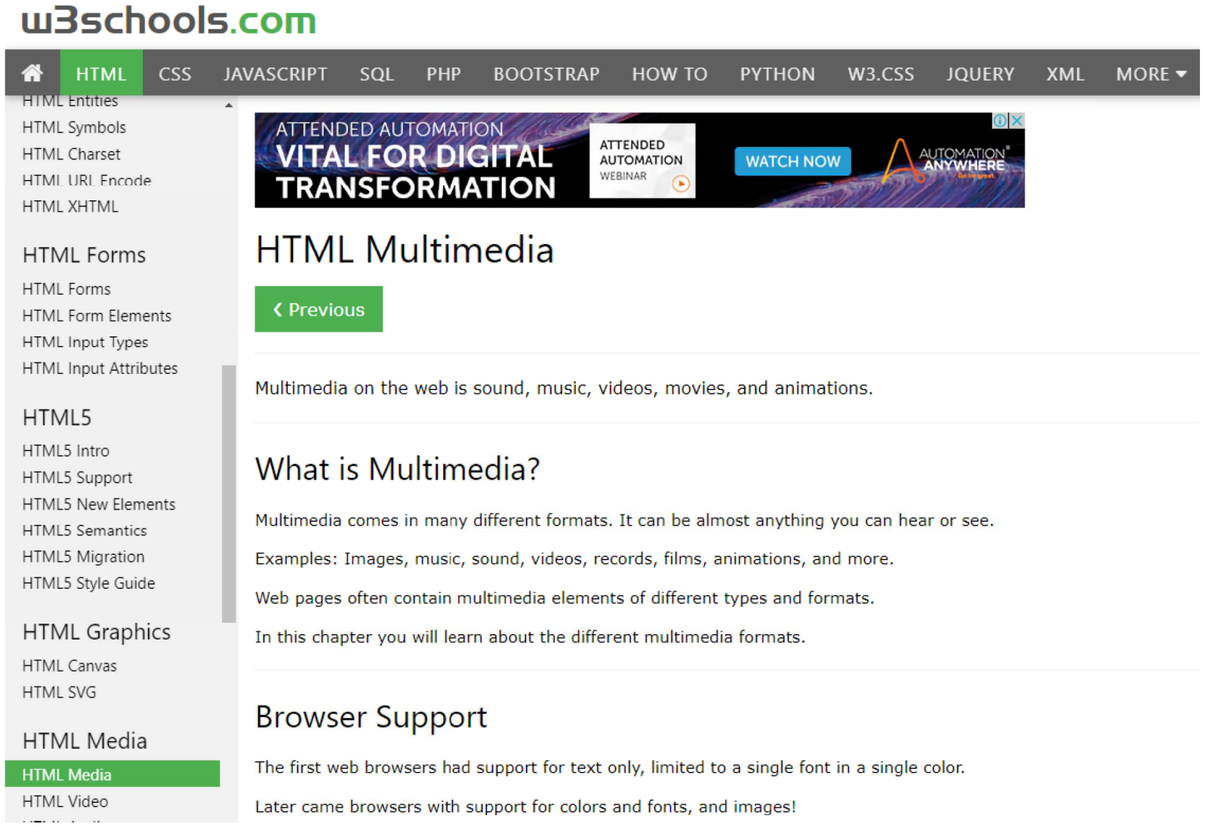
Python Data Storage



Python Web Crawler

爬蟲資料儲存成檔案

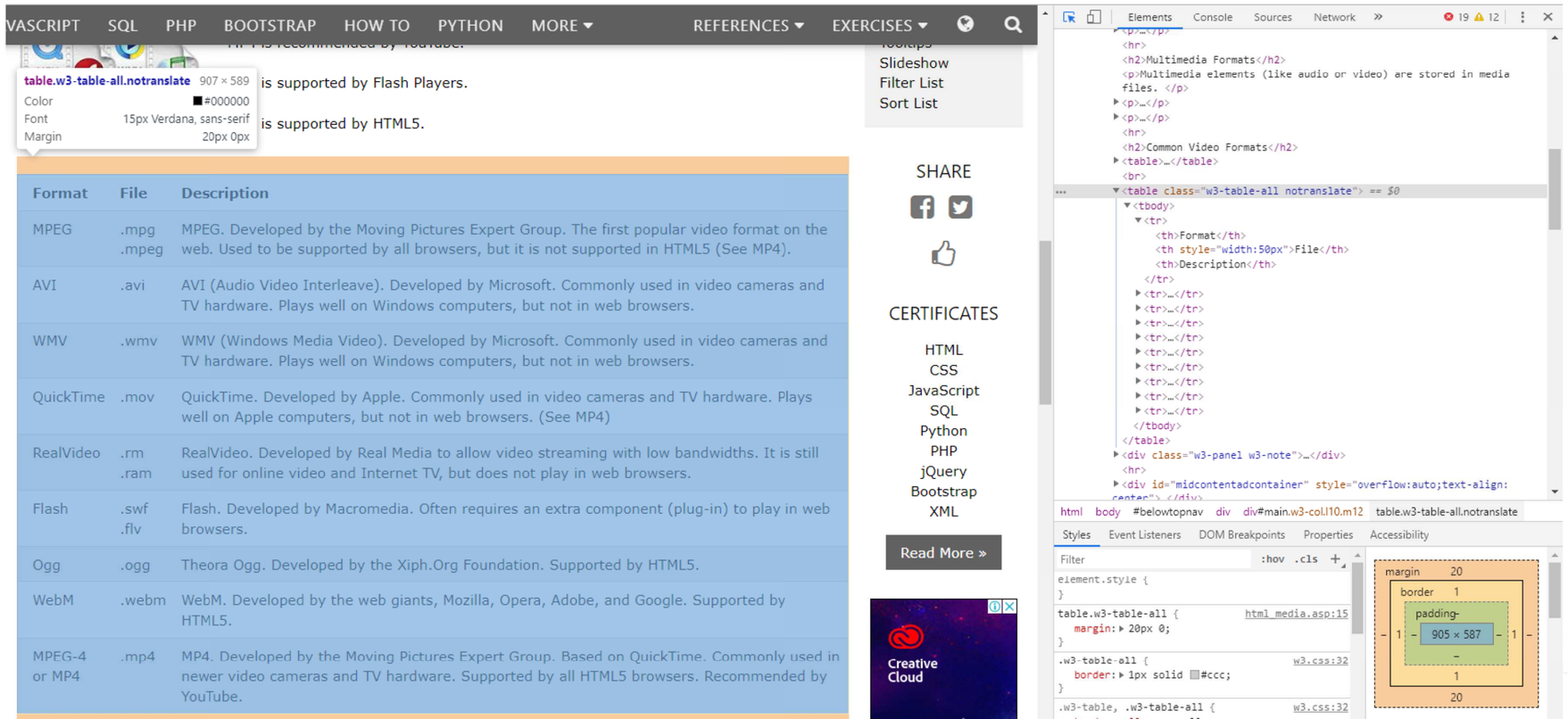
- 在成功從網路HTML網頁擷取出所需資料後，我們可以將整理好的資料儲存成檔案或是資料庫，常用的檔案格式有CSV和JSON檔案。以下將分別以W3schools與google圖書查詢舉例說明：



爬蟲資料儲存成檔案 - CSV

- 從W3School取出表格資料寫入CSV檔案
- 在本例，我們可以將網頁HTML表格資料存入CSV檔案，例如：W3School 的 Audio Format 說明表格如下。我們使用Chrome開發人員工具找出<table>表格標籤，可以看到 class 屬性是 "w3-table-all"，接著，我們可以搜尋HTML網頁取出表格資料來存入CSV檔案。

https://www.w3schools.com/html/html_media.asp



爬蟲資料儲存成檔案 - CSV

- 以下程式碼匯入相關模組後，使用 BeautifulSoup 物件的find()函數找到第一個 <table>標籤，然後使用 findAll() 函數找出表格的所有 <tr> 標籤。接著開啟CSV檔案準備寫入擷取出的資料。

- open() 函數指定編碼是utf-8，row 清單變數是所有 <tr> 標籤的表格列，第一層 for 迴圈取出每一列，第二層 for 迴圈取出每一個儲存格。

```
import requests
from bs4 import BeautifulSoup
import csv

url = "https://www.w3schools.com/html/html_media.asp"
csvfile = "VideoFormat.csv"
r = requests.get(url)
r.encoding = "utf8"
soup = BeautifulSoup(r.text, "lxml")
tag_table = soup.find(class_="w3-table-all")  # 找到<table>
rows = tag_table.findAll("tr")   # 找出所有<tr>
# 開啟CSV檔案寫入截取的資料
with open(csvfile, 'w', newline='', encoding="utf-8") as fp:
    writer = csv.writer(fp)
    for row in rows:
        rowList = []
        for cell in row.findAll(["td", "th"]):
            rowList.append(cell.get_text().replace("\n", "").replace("\r", ""))
        writer.writerow(rowList)
```

爬蟲資料儲存成檔案 - CSV

- 在內層 for 迴圈的 findAll() 函數可以找出此列的所有 <td> 和 <th> 標籤，我們使用 append()函數將 get_text()函數取得的標籤內容新增至串列，replace()函數刪除 "\n" 和 "\r\" 字元，最後呼叫 writeow() 函數寫入每一列資料至CSV檔案：VideoFormat.csv。檔案開啟如下：

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Format	File	Description																	
2	MPEG	.mpg.mpeg	MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Used to be supported by all browsers, but it is not supported in HTML5 (See MP4).																	
3	AVI	.avi	AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.																	
4	WMV	.wmv	WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.																	
5	QuickTime	.mov	QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers. (See MP4)																	
6	RealVideo	.rm.ram	RealVideo. Developed by Real Media to allow video streaming with low bandwidths. It is still used for online video and Internet TV, but does not play in web browsers.																	
7	Flash	.swf.flv	Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers.																	
8	Ogg	.ogg	Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5.																	
9	WebM	.webm	WebM. Developed by the web giants, Mozilla, Opera, Adobe, and Google. Supported by HTML5.																	
10	MPEG-4or	.mp4	MP4. Developed by the Moving Pictures Expert Group. Based on QuickTime. Commonly used in newer video cameras and TV hardware. Supported by all HTML5 browsers. Recommended by YouTube.?																	

```python
import requests
from bs4 import BeautifulSoup
import csv

url = "https://www.w3schools.com/html/html_media.asp"
csvfile = "VideoFormat.csv"
r = requests.get(url)
r.encoding = "utf8"
soup = BeautifulSoup(r.text, "lxml")
tag_table = soup.find(class_="w3-table-all")  # 找到<table>
rows = tag_table.findAll("tr")   # 找出所有<tr>
# 開啟CSV檔案寫入截取的資料
with open(csvfile, 'w+', newline='', encoding="utf-8") as fp:
    writer = csv.writer(fp)
    for row in rows:
        rowList = []
        for cell in row.findAll(["td", "th"]):
            rowList.append(cell.get_text().replace("\n", "").replace("\r", ""))
        writer.writerow(rowList)
```

爬蟲資料儲存成檔案 - JSON

- 從Google圖書查詢的JSON資料寫入JSON檔案

https://www.googleapis.com/books/v1/volumes?maxResults=5&q=Python&projection=lite


```
{
  "kind": "books#volumes",
  "totalItems": 449,
  "items": [
    {
      "kind": "books#volume",
      "id": "YEoiYr4H2A0C",
      "etag": "qkNXYS0XMQ8",
      "selfLink": "https://www.googleapis.com/books/v1/volumes/YEoiYr4H2A0C",
      "volumeInfo": {
        "title": "Python Scripting for Computational Science",
        "authors": [
          "Hans Petter Langtangen"
        ],
        "publisher": "Springer Science & Business Media",
        "publishedDate": "2009-01-09",
        "description": "With a primary focus on examples and applications of relevance to computational scientists, this brilliantly useful book sh
        "readingModes": {
          "text": false,
          "image": true
```

爬蟲資料儲存成檔案 - JSON

- Google圖書查詢的Web服務可以讓我們輸入書名的關鍵字來查詢圖書資料。以下URL參數的 maxResults參數是最多傳回幾筆；q是關鍵字；projection參數值lite是傳回精簡版的查詢資料。

```
1 import json
2 import requests
3
4 url = "https://www.googleapis.com/books/v1/volumes?maxResults=5&q=Python&projection=Lite"
5 jsonfile = "Books.json"
6 r = requests.get(url)
7 r.encoding = "utf8"
8 json_data = json.loads(r.text)
9 with open(jsonfile, 'w') as fp:
10     json.dump(json_data, fp)
```

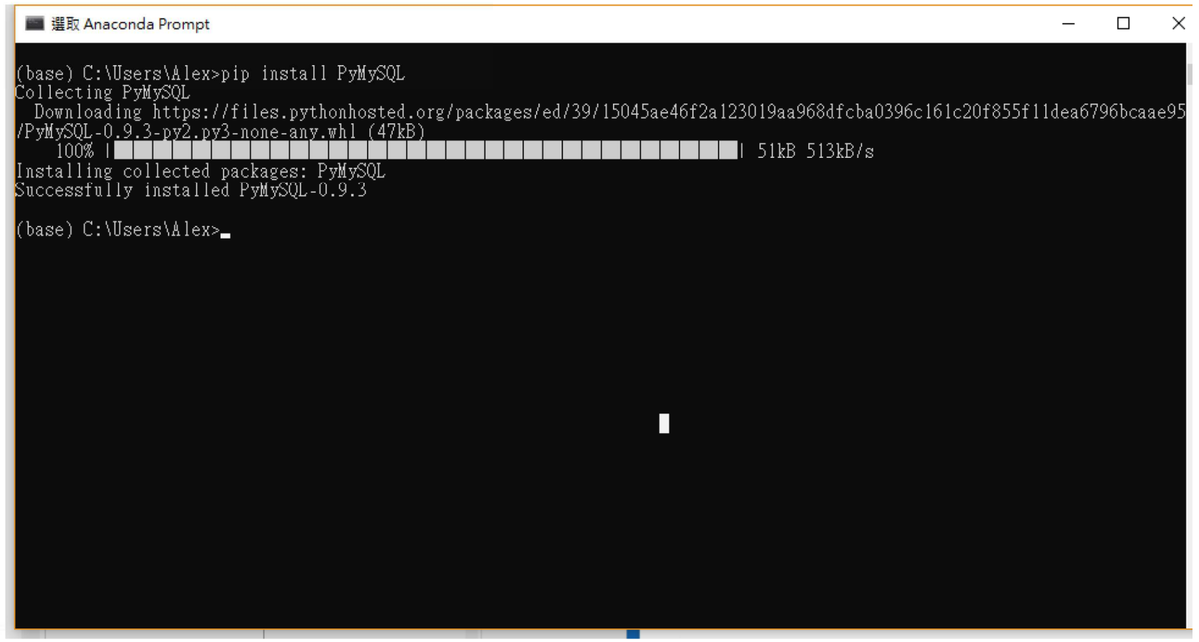
爬蟲資料存入MySQL資料庫

- Python支援MySQL資料庫模組很多，我們使用PyMySQL模組，因為 Anaconda 預設並沒有安裝此模組，因此我們需要先自行安裝。

- 安裝PyMySQL模組

  開啟 Anaconda Prompt 命令提示字元視窗後，輸入指定安裝 PyMySQL 模組。

  pip install PyMySQL



爬蟲資料存入MySQL資料庫

查詢MySQL資料庫

```python
import pymysql

# 建立資料庫連接
db = pymysql.connect("localhost", "root", "", "mybooks", charset="utf8")
cursor = db.cursor()  # 建立cursor物件
# 執行SQL指令SELECT
cursor.execute("SELECT * FROM books")
data = cursor.fetchall()   # 取出所有記錄
# 取出查詢結果的每一筆記錄
for row in data:
    print(row[0], row[1])
db.close()  # 關閉資料庫連接
```

- Python程式在使用MySQL資料庫的第一步需要匯入PyMySQL模組。
- 匯入PyMySQL模組後，可以建立資料庫連接來執行SQL指令。
- Connected函數建立資料庫連接，參數依序是MySQL主機名稱、使用者名稱、密碼和資料庫名稱，最後指定編碼是utf8，在成功建立資料庫連接後，呼叫cursor()函數建立cursor物件，即可呼叫execute()函數執行SQL指令來查詢MySQL資料庫。
- 因為是查詢資料，我們需要呼叫 fetchall() 函數取回第1筆記錄，fetchall()函數可以取回所有記錄，然後使用 for 迴圈取出查詢結果的每一筆記錄，row[0]和row[1]是前2個欄位，即id和title欄位，最後呼叫close()函數關閉資料庫連接，執行結果如下：

```
D0001 Access入門與實作
P0001 資料結構 - 使用C語言
P0002 Java程式設計入門與實作
P0003 Scratch+fChart程式邏輯訓練
W0001 PHP與MySQL入門與實作
W0002 jQuery Mobile與Bootstrap網頁設計
```

爬蟲資料存入MySQL資料庫

將CSV資料存入MySQL資料庫

- 當我們將爬取的資料建立成CSV字串後，就可以將CSV資料存入MySQL資料庫。
- 首先將CSV字串book轉換成串列(list)，程式碼建立資料庫連接後，使用 format()函數建立 SQL 插入記錄的SQL指令字串，在字串中的6個參數值 '{0}'，'{1}'，'{2}'，'{3}'，'{4}'，'{5}' 是對應清單的6個項目。
- 程式碼建立 SQL 指令後，使用 try/except 呼叫 execute()函數執行新增記錄，接著執行 commit()函數確認交易來真正變更資料庫，如果失敗，就執行 rollback() 函數回復交易，即回復成沒有執行SQL指令前的資料庫內容，執行結果新增一筆記錄：

```python
import pymysql

book = "P0004,Python程式設計,g,550,程式設計,2018-01-01"
f = book.split(",")

# 建立資料庫連接
db = pymysql.connect("localhost", "root", "", "mybooks", charset="utf8")
cursor = db.cursor()  # 建立cursor物件
# 建立SQL指令INSERT字串
sql = """INSERT INTO books (id,title,author,price,category,pubdate)
         VALUES ('{0}','{1}','{2}',{3},'{4}','{5}')"""
sql = sql.format(f[0], f[1], f[2], f[3], f[4], f[5])
print(sql)
try:
    cursor.execute(sql)   # 執行SQL指令
    db.commit() # 確認交易
    print("新增一筆記錄...")
except:
    db.rollback() # 回復交易
    print("新增記錄失敗...")
db.close()  # 關閉資料庫連接
```

```
INSERT INTO books (id,title,author,price,category,pubdate)
         VALUES ('P0004','Python程式設計','g',550,'程式設計','2018-01-01')
新增一筆記錄...
```

id	title	author	price	category	pubdate
D0001	Access入門與實作	A	450	資料庫	2016-06-01
P0001	資料結構 - 使用C語言	B	520	資料結構	2016-04-01
P0002	Java程式設計入門與實作	B	550	程式設計	2017-07-01
P0003	Scratch+fChart程式邏輯訓練	D	350	程式設計	2017-04-01
W0001	PHP與MySQL入門與實作	E	550	網頁設計	2016-09-01
W0002	jQuery Mobile與Bootstrap網頁設計	F	500	網頁設計	2017-10-01
P0004	Python程式設計	g	550	程式設計	2018-01-01

爬蟲資料存入MySQL資料庫

將JSON資料存入MySQL資料庫

- 同樣的，我們可以將JSON資料存入MySQL資料庫，首先將JSON資料轉換成的Python字典d。程式碼建立資料庫連接後，使用format()函數建立SQL插入記錄的SQL指令字串，在try/except呼叫execute()函數執行新增記錄，接著執行commit()函數真正變更資料庫，如果失敗，就執行rollback()函數回復交易，執行結果可以新增一筆記錄如下：

```python
import pymysql

d = {
    "id": "P0005",
    "title": "Node.js程式設計",
    "author": "h",
    "price": 650,
    "cat": "程式設計",
    "date": "2018-02-01"
}

# 建立資料庫連接
db = pymysql.connect("localhost", "root", "", "mybooks", charset="utf8")
cursor = db.cursor()  # 建立cursor物件
# 建立SQL指令INSERT字串
sql = """INSERT INTO books (id,title,author,price,category,pubdate)
         VALUES ('{0}','{1}','{2}',{3},'{4}','{5}')"""
sql = sql.format(d['id'],d['title'],d['author'],d['price'],d['cat'],d['date'])
print(sql)
try:
    cursor.execute(sql)   # 執行SQL指令
    db.commit() # 確認交易
    print("新增一筆記錄...")
except:
    db.rollback() # 回復交易
    print("新增記錄失敗...")
db.close()  # 關閉資料庫連接
```

```
INSERT INTO books (id,title,author,price,category,pubdate)
         VALUES ('P0005','Node.js程式設計','h',650,'程式設計','2018-02-01')
新增一筆記錄...
```

id	title	author	price	category	pubdate
D0001	Access入門與實作	A	450	資料庫	2016-06-01
P0001	資料結構 - 使用C語言	B	520	資料結構	2016-04-01
P0002	Java程式設計入門與實作	B	550	程式設計	2017-07-01
P0003	Scratch+fChart程式邏輯訓練	D	350	程式設計	2017-04-01
W0001	PHP與MySQL入門與實作	E	550	網頁設計	2016-09-01
W0002	jQuery Mobile與Bootstrap網頁設計	F	500	網頁設計	2017-10-01
P0004	Python程式設計	g	550	程式設計	2018-01-01
P0005	Node.js程式設計	h	650	程式設計	2018-02-01

Q & A