# Python爬蟲

黃彬華編撰

- 何謂爬蟲
- 爬蟲開發環境
- Beautiful Soup套件功能
- 爬蟲基本技巧
- 爬多頁內容
- 爬文存檔
- 爬圖存檔
- 爬到IP位置被封鎖
- 爬蟲遇到Cookies
- Selenium
- JSON資料格式
- 爬開放資料

# 何謂爬蟲

黃彬華編撰

- 使用者瀏覽網頁後會依照自己喜好將網頁資料下載
- 少量資料可以透過操作瀏覽器下載；如果要下載、篩選的資料有數十頁甚至更多，就需要使用工具或撰寫程式達成
- 爬蟲程式就是撰寫程式篩選網頁資料並下載，以達到特定目的
- Python撰寫爬蟲程式比其他程式語言更佳精簡且可讀性高

# 爬蟲開發環境 - 1

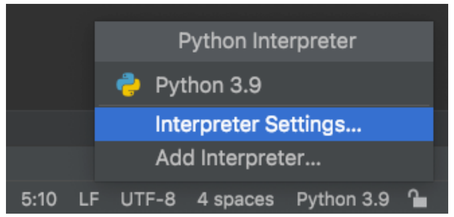
黃彬華編撰

- Python
  - python.org 官網下載
- IDE (Integrated Development Environment) 工具
  - PyCharm官網下載 > 下載 Community 版本 > 雙點即可安裝
- Python爬蟲相關套件
  - requests
  - beautifulsoup4
  - selenium

# 爬蟲開發環境 - 2

黃彬華編撰

- Python爬蟲相關套件可以直接透過PyCharm安裝
  - 開啟PyCharm
  - 點擊右下角Python版次 > Interpreter Settings > 開啟Preferences視窗 > 點擊「+」按鈕 > 開啟Available Packages視窗 > 輸入要安裝的package名稱後點擊「install Package」按鈕



# Beautiful Soup套件功能 - 1

黃彬華編撰

- 官方文件
- 讀取資料來源
  - 可以讀HTML文字內容
    - soup = BeautifulSoup(htmlDoc, "html.parser")
  - 搭配open()可以讀HTML檔案內容

```
with open("Index.html", "r") as fp:
    soup = BeautifulSoup(fp, "html.parser")
```

  - 搭配requests可以讀遠端網頁內容

```
response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")
```

# 範例

黃彬華編撰

- BeautifulSoup > DataSource.py

# Beautiful Soup套件功能 - 2

黃彬華編撰

- 取得指定標籤內容與屬性值
  - 範例：Basic.py
- 取得父標籤與子標籤內容
  - 範例：NavigateTree.py

# 範例



- BeautifulSoup > Basic.py
- BeautifulSoup > NavigateTree.py

# Beautiful Soup套件功能 - 3

黃彬華編撰

- findAll()搜尋功能
- 搭配BeautifulSoup提供的下列關鍵字並搭配正規表示式來搜尋資料
  - id：指定元素ID
    - soup.findAll(id='link1')
  - class_：指定類別名稱
    - 因為「class」為Python關鍵字，所以BeautifulSoup改為「 class_ 」
    - soup.findAll('p', class_='title')
  - href：搭配正規表示式指定href屬性值
    - soup.find_all(href=re.compile('frog'))  # 搜尋所有href含有"frog"標籤
  - attrs：指定任一屬性值
    - BeautifulSoup沒有提供name關鍵字來指定HTML的name屬性，所以要改用attrs來指定
    - soup.find_all(attrs={'name': 'comment'})

# Beautiful Soup套件功能 - 4

黃彬華編撰

- 加上函式功能，函式回傳True會將該元素留下

```
# 定義函式：搜尋有class屬性但沒有id屬性的標籤
def has_class_but_no_id(tag):
    return tag.has_attr("class") and not tag.has_attr('id')
# 呼叫函式
soup.find_all(has_class_but_no_id)
```

- limit參數限縮資料筆數
  - soup.find_all('a', limit=2)  # 搜尋所有<a>標籤，但限縮在前2筆
- find()與findAll()差別在於find()搜尋到第一個符合的就停止
  - soup.find('a')

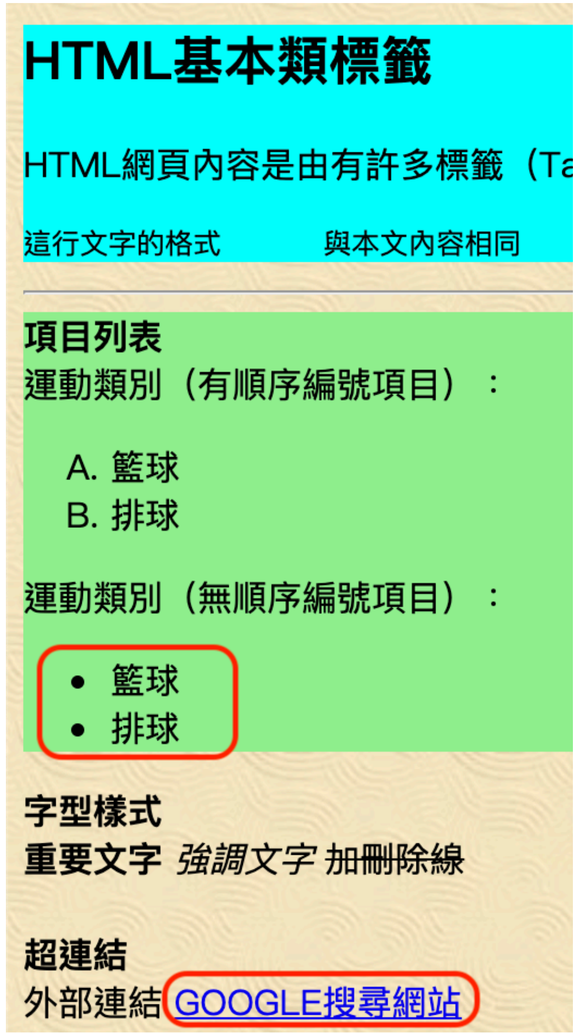
# 範例

黃彬華編撰

- BeautifulSoup > Search.py

# 練習1

黃彬華編撰




- 下載basic.html並置入Python專案內
- 以瀏覽器開啟上述檔案會呈現如左圖
- 請取得下列資訊
  - 取得無順序編號的2個項目「籃球」、「排球」並顯示
  - 取得「GOOGLE搜尋網站」的超連結並顯示
  - 最後顯示結果如下：

    籃球

    排球

    https://www.google.com

# Beautiful Soup套件功能 - 5

黃彬華編撰

- select()方法功能類似find_all()
- select_one()方法功能類似find()

# 範例



- BeautifulSoup > Select.py

# 練習2

黃彬華編撰

**HTML表格類標籤**

| name | telephone |
| --- | --- |
| Mary | 03-3456789 0911222333 |
| John | 02-23212321 0912333444 |

- 下載table.html並置入Python專案內
- 以瀏覽器開啟上述檔案會呈現如左圖
- 請取得標題下方的所有好友資訊並存放在二維List內，結果如下
  - [['Mary', '03-3456789', '0911222333'], ['John', '02-23212321', '0912333444']]

# 爬蟲基本技巧 - 1

黃彬華編撰

- 有些防爬蟲網站會檢查請求標頭 (request headers) 的user-agent是否有值，以辨識是一般使用者還是爬蟲程式訪問，來決定是否拒絕請求
  - 建議加上user-agent，值可以複製Chrome > F12 > Network > Headers > Request Headers > user-agent，來偽裝成一般使用者

# 爬蟲基本技巧 - 2

黃彬華編撰

- 觀察
  - 使用Chrome開發者工具觀察一篇文章範圍
- 一元復始，萬象更新
  - 用select_one()或find()查看單筆資料
  - OK後再使用select()或findAll()取得符合條件的所有資料

# 範例



- WebCrawler > Basic01Demo.py
- WebCrawler > Basic02Demo.py

# 練習3

黃彬華編撰

- 爬天瓏網路書店的Python書
  - https://www.tenlong.com.tw/search?availability=buyable&display=list&keyword=python&langs%5B%5D=all
    - 網址中的"display=list"代表橫式條列，方能顯示完整書名
- 只需爬第一頁所有書的售價、出版日期、書名並顯示如下

售價: $616  出版日期：2019-10-11  Python 技術者們

售價: $562  出版日期：2021-12-06  Python 出神入化

# 爬多頁內容

黃彬華編撰

- 想要有穩定資料來源以提供服務，需要爬大量資料或是類似索引資料 (例如每頁檔案名稱或超連結) 存放在自家資料庫內
- 觀察
  - 使用Chrome開發者工具觀察去上、下頁的超連結

# 範例

黃彬華編撰

- WebCrawler > MultiPages01Demo.py
- WebCrawler > MultiPages02Demo.py

# 練習4

黃彬華編撰

- 爬天瓏網路書店的Python書
  - https://www.tenlong.com.tw/search?availability=buyable&display=list&keyword=python&langs%5B%5D=all
    - 網址中的"display=list"代表橫式條列，方能顯示完整書名
- 讓使用者輸入需要爬幾頁內容，並且顯示每一頁所有書的售價、出版日期、書名

要顯示幾頁? 3

https://www.tenlong.com.tw/search?availability=buyable&display=list&keyword=python&langs%5B%5D=all

售價: $616　出版日期：2019-10-11　　Python 技術者們

售價: $562　出版日期：2021-12-06　　Python 出神入化

# 爬文存檔

黃彬華編撰

- 爬到的網頁內容可以暫時存檔，方便之後應用；也可減輕佔用過大記憶體空間的問題
- 步驟
  - 建立一個目錄存放下載的檔案
  - 存檔名稱可以直接使用原始檔案名稱，或其他具有識別效果的檔案名稱

# 範例



- WebCrawler > MultiPagesSaveDemo.py

# 爬圖存檔

黃彬華編撰

- 圖片只要有來源網址就可下載該圖片並存檔
- 圖片內容屬於二進位格式，所以存檔格式也要指定為二進位格式
- 步驟大致與前述「爬文存檔」相同

# 範例



- WebCrawler > MultiImagesSaveDemo.py

# 爬到IP位置被封鎖 - 1

黃彬華編撰

- 爬多頁時，一般會使用迴圈，因為電腦執行速度快，一秒鐘就會對目標網站發出n個請求，有時會被該網站認為是惡意程式，輕則停止回應一段時間，重則封鎖爬蟲程式電腦的IP位置
- 要避免這個問題最簡單方式就是使用time套件降低請求頻率
  - time.sleep(3)  # 停3秒後繼續

# 範例

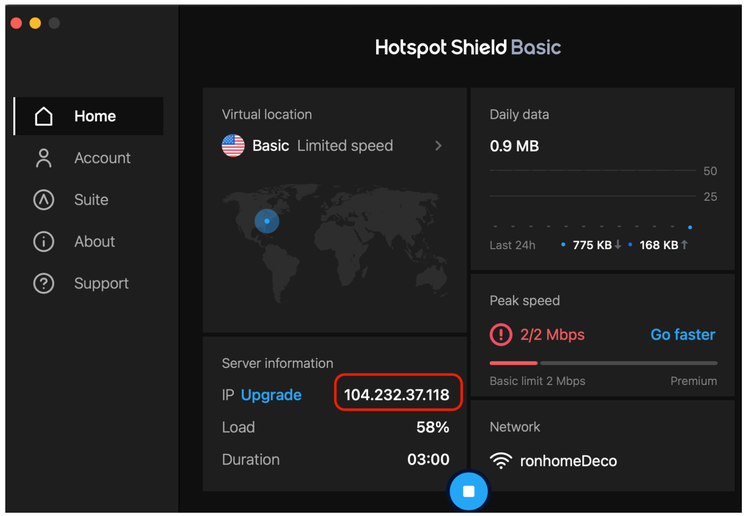


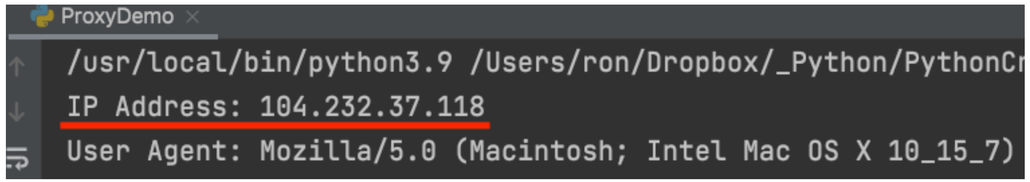
- WebCrawler > MultiPagesTimeDemo.py

# 爬到IP位置被封鎖 - 2

黃彬華編撰

- 如果真的被網站封鎖IP位置，可採用下列方式解決
  - 安裝 VPN軟體：例如 Hotspot Shield Free版
  - 設定 代理伺服器 (proxy server)：https://free-proxy-list.net ，但多半無效
- VPN伺服器與代理伺服器都可產生不同於原先的IP位置來連線



# 範例



- WebCrawler > ProxyDemo.py

# 爬蟲遇到Cookies - 1

黃彬華編撰

- 先使用瀏覽器拜訪PTT八卦版 (https://www.ptt.cc/bbs/Gossiping/index.html)
- 跟之前去PTT手機版不同，八卦版會檢查是否有點擊滿18歲的按鈕，如果沒有會導至同意畫面，而無法進一步取得文章資料

# 範例

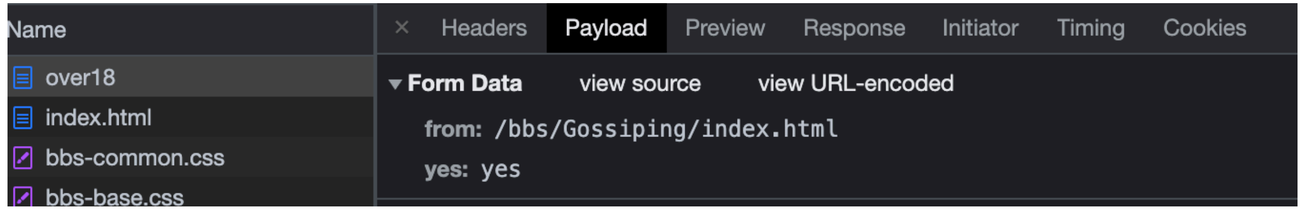


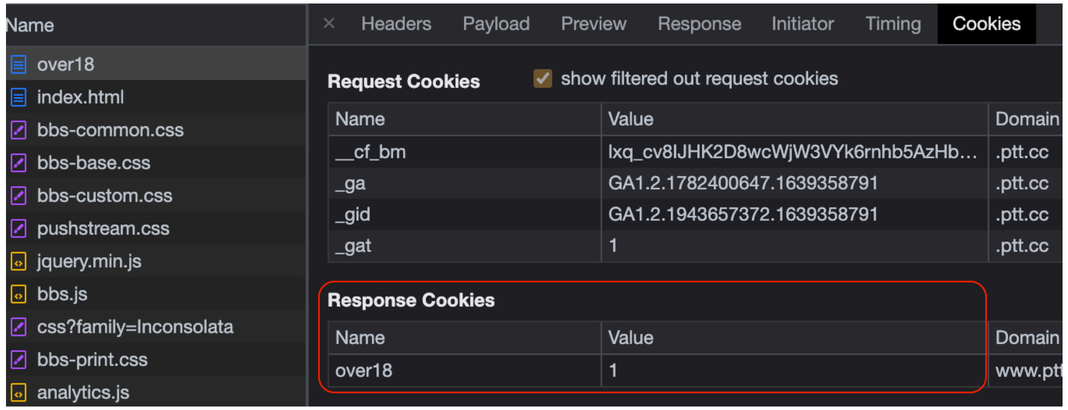
- WebCrawler > Cookies01Demo.py

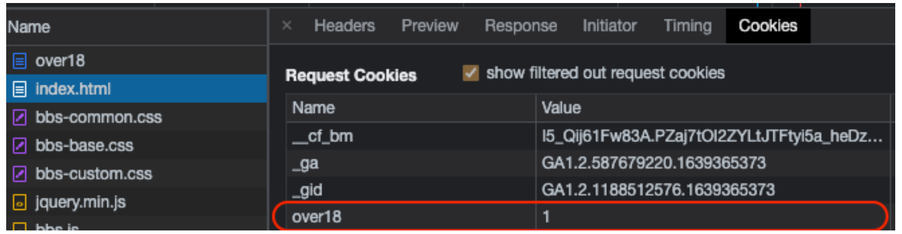
# 爬蟲遇到Cookies - 2

黃彬華編撰

- 使用Chrome開發者工具觀察點擊「我同意，我已年滿十八歲」的結果
  - Payload記錄送出資訊"from"與"yes"參數與值
  - Cookies記錄Response Cookies有"over18=1"資訊
  - 之後再次進入八卦版，Request Cookies就會帶有"over18=1"資訊，也不再有要求同意的畫面



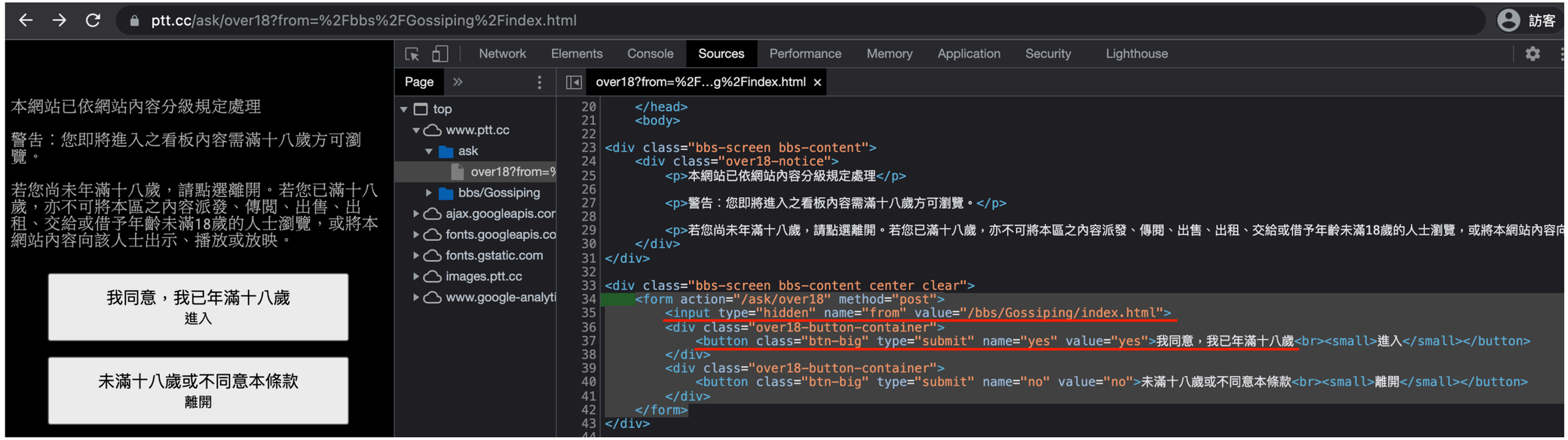



# 爬蟲遇到Cookies - 3

黃彬華編撰

- 使用Chrome開發者工具觀察「我同意，我已年滿十八歲」頁面原始碼
  - form表使用POST請求
  - hidden標籤、「我同意」按鈕參數名稱與值都與前述Payload內容相同



# 爬蟲遇到Cookies - 4

黃彬華編撰

- 解決方式
- 使用session會儲存發送請求後收到的cookies資訊，方便再次送出請求時可以將儲存的cookies一起送出
  - session = requests.Session()
- 視情況將參數名稱與值包成POST請求的data

```
data = {
    "from": "/bbs/Gossiping/index.html",
    "yes": "yes"
}
```

- 送出POST請求後server會回傳cookies，並且會儲存至session
  - session.post(https://www.ptt.cc/ask/over18", headers=headers, data=data)
- 因為session內儲存cookies資訊，之後每次以session發出請求，server都會收到cookies，所以會回傳網頁內容
  - response = session.get("https://www.ptt.cc/bbs/Gossiping/index.html")

# 範例



- WebCrawler > Cookies02Demo.py

# Selenium - 1

黃彬華編撰

- Selenium是一個自動化測試工具
- 可以模仿真人操作網頁的各種行為 (例如：點擊按鈕、輸入資料)
- 因為擬真度很高，目標網站無從阻擋；因為如果阻擋意味著也很可能將真人操作阻擋在外

# Selenium - 2

黃彬華編撰

- 安裝步驟
  - 安裝web driver
    - 要讓Selenium控制哪一個瀏覽器，就需要安裝該瀏覽器對應的web driver
    - Chrome需要安裝Chrome Driver，先查詢Chrome版次 (查看 關於Google Chrome)，然後下載適當版本
      - http://chromedriver.chromium.org/downloads
  - PyCharm安裝selenium套件
- 官方文件 Selenium with Python

# Selenium - 3

黃彬華編撰

- 建立Chrome Driver並指定位置 (可放在Python程式同目錄內)
  - driver = Chrome(service=Service("./chromedriver"))
- 連結到目標網站
  - driver.get(url)
- 搜尋參數名稱為"yes"的按鈕，然後模擬點擊該按鈕
  - driver.find_element(By.NAME, "yes").click()
- 取得網頁內容，並建立soup
  - page_source = driver.page_source
  - soup = BeautifulSoup(page_source, "html.parser")

# 範例



- WebCrawler > SeleniumDemo.py

# JSON資料格式

黃彬華編撰

- JSON (JavaScript Object Notation) 資料格式已經成為資料交換的主流格式，而且有許多網站提供的開放資料 (open data) 也是JSON格式
- Python附有json套件可轉換JSON與Python資料格式
  - 呼叫json.dumps()可以將list、dictionary轉成JSON字串
    - jsonString = json.dumps(list_or_dic)
  - 也可以將JSON字串轉成list、dictionary
    - list_or_dic = json.loads(jsonString)
- 如果有自訂物件，需要透過"__dict__"轉成dictionary，方能轉成JSON字串，否則會產生"TypeError"錯誤
  - jsonString = json.dumps(customObj.__dict__)

# 範例



- OpenData > JsonDemo.py

# 爬開放資料

黃彬華編撰

- 有許多網站提供開放資料 (open data) 供開發者抓取，例如：政府資料開放平臺 (https://data.gov.tw)
- 取得資料方式也是使用requests套件
- 提供的資料內容，一般都會有XML與JSON格式
  - XML：可以繼續使用BeautifulSoup套件
  - JSON：使用Python內附的json套件

# 範例



- OpenData > OpenDataDemo.py