多媒體程式設計
文字資料處理

Instructor: 馬豪尚

JSON資料格式

- 瀏覽器和網站伺服器之間交換資料，資料只能是文字形式，JSON就是一種文字資料格式，最初是為了JavaScript開發的
- 這種資料格式常被應用在Web開發和大數據資料庫(NoSQL)，Python也有採用與支援這種格式，可以將資料以JSON的格式做儲存

JSON資料格式

- 物件
  - 在json中的物件採用key : value的方式配對儲存
  - 物件內容用左右大括弧{ }來包住
  - key和value中間用冒號 ":" 來區隔
  - 每一組key : value以逗號 "," 來區隔
  - key必須是一個文字字串
  - value可以是數值、字串、布林值、陣列、null
- 陣列
  - 陣列的值可以是數值、字串、布林值、陣列、null

JSON資料樣式

```
{
    "id": 123,
    "Name": "wsrsw",
    "Email": "wsrsw@example.com",
    "contents": [
        {
            "subject":"Math",
            "score":80
        },
        {
            "subject":"English",
            "score":90
        }
    ]
}
```

JSON字串轉Python dict

- Impost json
- loads(json_str)
  - 回傳一個python字典物件

json與python資料類型轉換關係表

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

Python dict轉JSON字串再寫入

- Json_str = json.dumps(dict)
- dumps中的sort_keys參數
  - Python字典是無序的資料，使用sort_keys=True可以將轉成的json進行排序
  - json.dumps(dict, sort_keys=True)
- dumps中的indent參數
  - indent可以讓轉成json格式進行縮排排版，讓json格式比較好閱讀
  - json.dumps(dict, indent=4)

將Python資料直接輸出成json檔

- dump()函數, 需要兩個參數
  - json.dump(data, jsonfile)
  - 第一個參數為python的資料，想要序列化的目標
    - dictObj = {'b':80, 'a':25, 'c':60}
  - 第二個參數為開啟的json寫入物件
    - with open('mock_data.json', 'w', newline='') as jsonfile:

練習1

- 參考populations.json檔案，該檔案為世界各國人口數的資料
  - 將2000年的相關資料取出並存入populations_2000.json檔案

維基百科中文文本

- 下載文本
  - https://dumps.wikimedia.org/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2
  - wget https://dumps.wikimedia.org/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2
- 文本是一個xml檔案
  - 可延伸標記式語言(Extensible Markup Language, XML)
  - XML設計是用來傳送和攜帶資料資訊
  - 可以自定義結構化標籤

WikiExtractor套件

- 下載
  - pip install wikiextractor
- 使用Extractor將載來的維基文件做資料清理並存成檔案
  - python Wiki_Extractor.py -b 1024M -o extracted zhwiki-latest-pages-articles.xml.bz2
  - -b 設定使用的記憶體
  - -o 輸出檔案的路徑
  - 輸入處理的檔案

WikiExtractor套件

- 儲存的檔案格式
  - <doc id=“文章id” url=“文章網址” title=“文章標題">
    文章內容
  - </doc>

Open CC 開放中文轉換套件

- 安裝套件
  - pip install opencc
- 載入套件
  - from opencc import OpenCC
- 簡體轉繁體
  - opencc = OpenCC('s2twp')
  - raw_data_changed = opencc.convert(raw_data)
  - raw_data為要轉的資料

Open CC轉換模式

- hk2s: 繁體中文 (香港) -> 簡體中文
- s2hk: 簡體中文 -> 繁體中文 (香港)
- s2t: 簡體中文 -> 繁體中文
- s2tw: 簡體中文 -> 繁體中文 (台灣)
- s2twp: 簡體中文 -> 繁體中文 (台灣, 包含慣用詞轉換)
- t2hk: 繁體中文 -> 繁體中文 (香港)
- t2s: 繁體中文 -> 簡體中文
- t2tw: 繁體中文 -> 繁體中文 (台灣)
- tw2s: 繁體中文 (台灣) -> 簡體中文
- tw2sp: 繁體中文 (台灣) -> 簡體中文 (包含慣用詞轉換)

練習2

- 將wiki文本載下來
  - 文本太大，提供給大家一個小的sample檔案
  - wiki_sample_SC.txt
- 做簡體轉繁體
- 將文本存成JSON格式

```
[
    {
        “id”: 文章編號,
        “title”: 文章標題,
        “articles”: 文章內容
    },
]
```

Wordcloud文字雲應用

- 安裝載入模組
  - pip install wordcloud
  - From wordcloud import WordCloud
- 用jieba模組斷詞和統計詞頻
  - dict={‘word1’: 10, ‘word2’: 9, ‘word3’:3, …}
- 按字詞頻率排序
  - 字典用value排序

Wordcloud文字雲應用

- 宣告物件
  - wc =WordCloud(參數1, 參數2, 參數3, 參數4...)
- 較重要的三個參數
  - background_color: 設定背景顏色，預設是黑色
  - font_path: 設定文字字型，預設字型不能顯示中文，必須設定為中文字型，比較簡單的方法是將中文字型檔跟程式碼放在一起就可以直接載入
  - mask: 設定文字雲形狀，預設是長方形，可以用任意圖形做為遮罩繪圖，圖形格式必須是numpy格式
    - np.array(Image.open('圖檔'))

Wordcloud文字雲應用

- 產生文字雲
  - wc = wc.generat_from_frequencies(dictionary)
- 繪圖
  - Import matplotlib.pyplot as plt
  - plt.figure(figsize=(寬度, 高度))
  - plt.imshow(wordcloud物件)
  - plt.axis(“off”)
  - plt.show()
- 存檔
  - wc.to_file(“檔名”)