

Problem-Based Learning

Python 3

Lecture: 13-crawler

Online Python

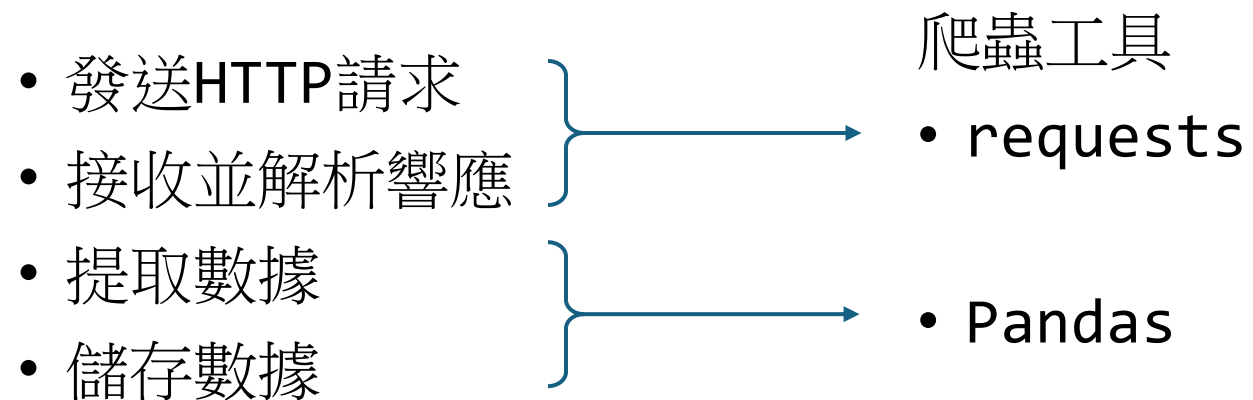
- <https://colab.research.google.com/>
- <https://www.onlinegdb.com/#>

Lecture Notes

- <https://web.phy.ntnu.edu.tw/~hongyi/?url=notes>



爬蟲



- 數據格式：

JSON

HTML

CSV

Accurate Weather Forecasts

- <https://open-meteo.com/>

`https://api.open-meteo.com/v1/forecast?latitude=25.0330&longitude=121.5654&daily=temperature_2m_max,temperature_2m_min`

URL解析

`https://api.open-meteo.com/v1/forecast?latitude=25.0330&longitude=121.5654&daily=temperature_2m_max,temperature_2m_min`

- `url = "https://api.open-meteo.com/v1/forecast"`
- `params = {
 "latitude": 25.0330,
 "longitude": 121.5654,
 "daily": "temperature_2m_max,temperature_2m_min"
}`

Request

- `response = requests.get(url, params=params)`
- `data = response.json()`
- `print(data)`  `dict in Python`

```
{"latitude":25,"longitude":121.625,"generationtime_ms":0.0259876251220703
,"utc_offset_seconds":28800,"timezone":"Asia/Taipei","timezone_abbreviati
on":"GMT+8","elevation":12,"daily_units":{"time":"iso8601","temperature_2
m_max":"°C","temperature_2m_min":"°C"},"daily":{"time":["2025-03-23","2025-
03-24","2025-03-25","2025-03-26","2025-03-27","2025-03-28","2025-03-
29"],"temperature_2m_max":[28.7,29.6,30.3,31.7,30.9,23.7,17.2],"temperatu
re_2m_min":[11.1,14.1,15.8,18.4,21.2,16.8,13.4]}}
```

JSON

- `response = requests.get(url, params=params)`
- `data = response.json()`

 dict in Python

```
{  
    "daily": {  
        "time": ["2025-03-23", "2025-03-24", "2025-03-25", "2025-03-  
26", "2025-03-27", "2025-03-28", "2025-03-29"],  
        "temperature_2m_max": [28.7, 29.6, 30.3, 31.7, 30.9, 23.7, 17.2],  
        "temperature_2m_min": [11.1, 14.1, 15.8, 18.4, 21.2, 16.8, 13.4]  
    }  
}
```

`data["daily"]["time"][0]`

Python程式

- `import requests`
- `import pandas as pd`
- `url = "https://api.open-meteo.com/v1/forecast"`
- `params = {`
 - `"latitude": 25.0330,`
 - `"longitude": 121.5654,`
 - `"daily": "temperature_2m_max,temperature_2m_min"``}`
- `response = requests.get(url, params=params)`
- `data = response.json()`

Python程式

- `d = pd.DataFrame({
 "日期": data["daily"]["time"],
 "最高溫度 (°C)": data["daily"]["temperature_2m_max"],
 "最低溫度 (°C)": data["daily"]["temperature_2m_min"]
})`
- `print("台北市一週天氣預報:")`
- `print(d)`

READ_HTML (1)

- `import pandas as pd`
- `url = "https://www.tiobe.com/tiobe-index/"`
- `d = pd.read_html(url, encoding="utf-8")`
- `print(d[0])`
- `# First table`

READ_HTML (2)

- `import pandas as pd`
- `url = "https://rate.bot.com.tw/xrt?Lang=zh-TW"`
- `data = pd.read_html(url, encoding="utf-8")`
- `d = data[0]`
- `print(d)`
- `d = d[[d.columns[0], d.columns[2]]]`
- `d.columns = ["幣別", "本行賣出"]`
- `d.set_index("幣別", inplace=True)`
- `print(d)`

幣別		現金匯率			Unnamed: 4_level_0
幣別	Unnamed: 1_level_1	本行買入	本行賣出	本行買入	
0	美金 (USD) 美金 (USD)	27.27	27.94	27.595	27.745
1	港幣 (HKD) 港幣 (HKD)	3.394	3.598	3.515	3.585
2	英鎊 (GBP) 英鎊 (GBP)	36.69	38.81	37.585	38.215
3	澳幣 (AUD) 澳幣 (AUD)	19.74	20.52	19.955	20.3

READ_CSV

- `import pandas as pd`
- `url = "https://rate.bot.com.tw/xrt/flcsv/0/day"`
- `d = pd.read_csv(url)`
- `d.set_index("幣別", inplace=True)`
- `print(d)`

Problem

- 自己找一個網頁下載資料，利用Pandas整理資料，最後輸出結果
- 科學數據：
- 電影時刻表：
- 一週預報(XHR)：<https://www.cwa.gov.tw/V8/C/W/week.html>
- 空氣品質指標(AQI)：<https://data.gov.tw/dataset/40448>
- 統一發票：<https://invoice.etax.nat.gov.tw/index.html>
- 臺灣銀行牌告匯率網頁：<https://rate.bot.com.tw/xrt>