

# Python Practical Design © Enos Chou

## Cases

1. Open Data API Integration
2. Basic Crawler Integration
3. Integrity Consideration
4. Modularization for Delivery
5. Optimize Function Design to Improve Performance
6. Parallel Run to Improve Performance
7. Weather LINE Bot
8. Weather API
9. Configurable Code Design

## 1. Open Data API Integration

以中央氣象局之 Open Data API 實做查詢即時天氣模組

### SPEC

```
def grab(site):  
    ...  
    return {'S': '臺北', 'T': 25.3, 'H': 0.53, 'R': 0.5, 'O': '2024-08-02...'}
```

### Reference Output

```
print(grab('臺北'))  
  
{'S': '臺北', 'T': 35.7, 'H': 0.5, 'R': 0.0, 'O': '2024-08-02 14:20:00'}
```

### Hints

1. `requests` to simulate http(s) request

```
!pip install requests
```

```
r = requests.get(url)  
r.text  
r.json()
```

2. improve the code's fault tolerance

## 2. Basic Crawler Integration

爬取中央氣象局之氣象網站資訊實做查詢即時天氣模組

### SPEC

```
def grab(site):  
    ...  
    return {'S': '臺北', 'T': 25.3, 'H': 0.53, 'R': 0.5, 'O': '08/02 ...'}
```

### Reference Output

```
print(grab('臺北'))  
  
{'S':, '臺北', 'T': 32.2, 'H': 0.64, 'R': 0.0, 'O': '08/02 15:00'}
```

### Hints

1. `beautifulsoup4` for crawler

```
!pip install bs4
```

```
from bs4 import BeautifulSoup
```

```
soup = BeautifulSoup(text, 'html.parser')  
soup.find(headers="hum").text
```

## 3. Integrity Consideration

整合上述兩種取得天氣資訊方式至同個模組，取得最佳結果

### SPEC

查詢 五峰站、西拉雅、合歡山頂、梅山、中央大學、北投 等測站都需正確輸出天氣資訊

```
def grab(site, key=None):  
    ...  
    return {'S': '臺北', 'T': 25.3, 'H': 0.53, 'R': 0.5, 'O': '08/02 ...'}
```

### Notes

兩種方式取得的氣象資訊可能不同，須取得最佳資訊

### Reference Output

```
sites = ['五峰站', '西拉雅', '合歡山頂', '梅山', '中央大學', '北投']
for site in sites:
    print(site, grab(site, key='YOUR_KEY'))

五峰站 {'S': '五峰站', 'T': 24.1, 'H': 0.96, 'O': '2024-08-02 15:50:00'}
西拉雅 {'S': '西拉雅', 'T': 32.7, 'H': 0.63, 'O': '2024-08-02 15:50:00'}
合歡山頂 {'S': '合歡山頂', 'T': 14.3, 'H': 0.95, 'R': 0.0, 'O': '08/02 15:00'}
梅山 {'S': '梅山', 'T': 32.2, 'H': 0.64, 'R': 0.0, 'O': '08/02 15:00'}
中央大學 {'S': '中央大學', 'T': 26.4, 'H': 0.37, 'R': 0.0, 'O': '2024-08-02 15:50:00'}
北投 {}
```

## 4. Modularization for Delivery

### 模組交付準備，能夠被驗證與匯入

#### SPEC

1. 輸出天氣查詢模組為 wea.py
2. 能夠以模組的方式匯入 wea.py 後執行天氣查詢
3. 能夠直接於終端機執行 `python wea.py` 依指定測站名稱查詢天氣
4. 將輸出的天氣資訊轉譯為文字提升可讀性
5. 以 docstring 方式加入註解方便使用者理解

#### Reference Output

##### 1. in Command Line

```
python wea.py -k 'YOUR_KEY' 五峰站 西拉雅 合歡山頂 梅山 中央大學 北投
```

```
五峰站 測站: 五峰站, 時間: 2024-08-02 22:00:00, 溫度: 22.4°C, 濕度: 92%
西拉雅 測站: 西拉雅, 時間: 2024-08-02 22:10:00, 溫度: 28.7°C, 濕度: 85%
合歡山頂 測站: 合歡山頂, 時間: 08/02 22:00, 溫度: 9.5°C, 濕度: 94%, 雨量: 0.0mm
梅山 測站: 梅山, 時間: 08/02 22:00, 溫度: 27.7°C, 濕度: 86%, 雨量: 0.0mm
中央大學 測站: 中央大學, 時間: 2024-08-02 22:10:00, 溫度: 27.5°C, 濕度: 81%
北投 無觀測!
```

##### 2. in Code

```
import wea
```

```
sites = ['五峰站', '西拉雅', '合歡山頂', '梅山', '中央大學', '北投']
for site in sites:
```

```
print(site, wea.tostr(wea.grab(site, 'YOUR_KEY')))
```

五峰站 測站: 五峰站, 時間: 2024-08-02 22:30:00, 溫度: 22.7°C, 濕度: 91%  
西拉雅 測站: 西拉雅, 時間: 2024-08-02 22:30:00, 溫度: 28.5°C, 濕度: 86%  
合歡山頂 測站: 合歡山頂, 時間: 08/02 22:00, 溫度: 9.5°C, 濕度: 94%, 雨量: 0.0mm  
梅山 測站: 梅山, 時間: 08/02 22:00, 溫度: 27.7°C, 濕度: 86%, 雨量: 0.0mm  
中央大學 測站: 中央大學, 時間: 2024-08-02 22:30:00, 溫度: 27.6°C, 濕度: 81%, 雨量:  
0.0mm  
北投 無觀測!

## Hints

1. `argparse` to parse various user input

```
import argparse

parser = argparse.ArgumentParser()
parser.add_argument('site', nargs='+')
parser.add_argument('--key', '-k')
args = parser.parse_args()
```

2. `docstring` to note for public functions only

## Notes

command "python wea.py" can not be executed under Jupyter Notebook

# 5. Optimize Function Design to Improve Performance

## 改善程式邏輯以提升執行效率

## Hints

1. `time` to measure the time consumption

```
import time

start = time.time()
...
end = time.time()
print(f'{end-start:.3f}s elapsed')
```

## 6. Parallel Run to Improve Performance

同時存取三組不同的天氣觀測 API 以提升執行效率

### Reference Output

#### 1. Single-Thread(Sequential)

```
> python wea.py 五峰站 西拉雅 合歡山頂 梅山 中央大學 北投
```

```
五峰站 觀測時間: 2024-08-03 14:40:00, 溫度: 27.4°C, 濕度: 81%
```

```
西拉雅 觀測時間: 2024-08-03 14:40:00, 溫度: 33.7°C, 濕度: 61%
```

```
合歡山頂 觀測時間: 08/03 14:00, 溫度: 15.5°C, 濕度: 90%, 雨量: 0.0mm
```

```
梅山 觀測時間: 08/03 14:00, 溫度: 34.5°C, 濕度: 54%, 雨量: 0.0mm
```

```
中央大學 測站: 中央大學, 時間: 2024-08-03 14:40:00, 溫度: 28.6°C, 濕度: 80%, 雨量:  
0.0mm
```

```
北投 無觀測!
```

```
6.631s elapsed
```

#### 2. Multi-Thread

```
> python wea.py 五峰站 西拉雅 合歡山頂 梅山 中央大學 北投
```

```
五峰站 觀測時間: 2024-08-03 14:40:00, 溫度: 27.4°C, 濕度: 81%
```

```
西拉雅 觀測時間: 2024-08-03 14:40:00, 溫度: 33.7°C, 濕度: 61%
```

```
合歡山頂 觀測時間: 08/03 14:00, 溫度: 15.5°C, 濕度: 90%, 雨量: 0.0mm
```

```
梅山 觀測時間: 08/03 14:00, 溫度: 34.5°C, 濕度: 54%, 雨量: 0.0mm
```

```
中央大學 測站: 中央大學, 時間: 2024-08-03 14:40:00, 溫度: 28.6°C, 濕度: 80%, 雨量:  
0.0mm
```

```
北投 無觀測!
```

```
3.615s elapsed
```

### Hints

#### 1. `threading` to run parallelly

```
import threading
```

```
def thread_fn(arg, i):
```

```
    global results
```

```
    ...
```

```
    results[i] = f'some result'
```

```
results = [None, None]
```

```
t1 = threading.Thread(target=thread_fn, args=('test1', 0), daemon=True)
t2 = threading.Thread(target=thread_fn, args=('test2', 1), daemon=True)
t1.start()
t2.start()
t1.join()
t2.join()
return results
```

## Notes

daemon behavior would operate correctly only in command line

## 7. Weather LINE Bot

整合即時天氣查詢模組，製作天氣查詢 LINE Bot

### Reference Output



## 8. Weather API

整合即時天氣查詢模組，製作天氣查詢 API

### Reference Output

```
\> curl --location 'https://weapi.enadv.online' --header 'Content-Type: application/json' --data '{"site": "苗栗"}'
```

測站: 苗栗, 時間: 07/03 11:00, 溫度: 31.1°C, 濕度: 68.0%, 雨量: 0.0mm

### Hints

1. refer to the usage of `Flask` in LINE Bot

## 9. Configurable Code Design

### 查詢天氣 LINE Bot 能夠彈性指定各種 Keys

#### Hints

1. `dotenv` ft. INI

**CONFIGURATION FILE NAME:** `.env`

```
YOUR_KEY=aaabbbcccd  
YOUR_...=...
```

#### USAGE

```
!pip install python-dotenv  
  
import os  
from dotenv import load_dotenv  
load_dotenv()  
key = os.environ.get('YOUR_KEY')
```

2. `json` ft. JSON

**CONFIGURATION FILE NAME:** `env.json`

```
{  
    "YOUR_KEY": "aaabbbcccd",  
    "YOUR_...": "..."  
}
```

#### USAGE

```
import json  
with open('env.json') as f:  
    env = json.load(f)  
key = env.get('YOUR_KEY')
```

3. `module` ft. Python

#### 4. default value design