

(7주차) 데이터 분석

백석대학교 강윤희

6주차에서 7주차 넘어가기

6주차 2020-04-18 ~ 2020-04-25	NOW 7주차 2020-04-25 ~ 2020-05-02
(예습)5. 파일처리 39명 1명 0명 강의 보기	4. 리스트 컴프리헨션-5주차 실습내용 보충 6명 2명 32명 강의 보기
1. (실습) 데이터 수집 openAPI-openweather... 40명 0명 0명 강의 보기	2. lambda 표현식 4명 2명 34명 강의 보기
2. 예외처리-02 40명 0명 0명 강의 보기	1. 파일처리 4명 1명 35명 강의 보기
3. (실습) openweathermap 예외처리 40명 0명 0명 강의 보기	3. map 과 filter 함수 4명 1명 35명 강의 보기
2. 예외처리-01 39명 1명 0명 강의 보기	5. CSV 파일 프로그래밍 3명 2명 35명 강의 보기
4. CSV 파일이해하기 39명 1명 0명 강의 보기	화요일 5교시 0명 0명 40명

openAPI 자료처리

6주차 과제 정리

수업목표

- 데이터 수집 준비하기
- Open API를 사용하기 위한 과정을 익힘
- Open API의 사용을 위한 요청과 응답 처리를 이해함
- 파이썬으로 Open API의 요청과 응답 처리를 프로그래밍 함
- 예외처리 적용하기

4주차 실습

• openAPI 사용하기

 [일반 과제] 4주차-데이터수집 준비하기-openAPI [마감]

 수정  삭제  과제평가

참고 자료 :  4wk-데이터 준비하기-과제 .pdf

kmares@data.kma.go.kr
나에게 ▼

4월 11일 (토) 오후 10:11 (3월)

제출기간	성적반영	성적공개	성적공개일자	사용 O
2020-04-07 00:00 ~ 2020-04-07 23:59	반영	공개	2020-04-15 00:00	 사용

데이터 분석에 필요한 자료를 얻기 위한 방법인 OpenAPI 활용을 위해 데이터 준비 과정을 수행한다.

- 데이터 수집을 위해 사이트에 사용자 등록하기
- 사용자 키 획득하기
- 획득된 키를 사용하여 OpenAPI 사용해 보기 (웹브라우저)

* 프로젝트 사전 과제 준비

- data.go.kr 에서자신의 데이터 분석 필요한 자료 찾아보기
- data.go.kr에서 데이터 준비하기 결과제출

안녕하세요.
기상청 기상자료개방포털 관리자입니다.

기상자료 개방포털 회원가입 인증번호 안내


안녕하세요. 기상자료 개방포털입니다.
yhkang님의 회원가입 진행을 위한 인증번호를 안내드립니다.
아래의 인증번호를 회원가입 화면의 이메일 인증번호란에 입력해 주십시오.


이메일 인증번호 : 0851508984


openAPI 사용을 위한 준비과정
텀프로젝트 주제 선정


4주차 실습



• openAPI 사용하기

 [일반 과제] 4주차-데이터수집 준비하기-openAPI(2) [마감]



 수정

 삭제

 과제평가

참고 자료 :  Openweathermap_result.PNG  pip.PNG

90%


제출기간	성적반영	성적공개	성적공개일자	사용 여부	연장제출	제출자수 (제출 / 미제출 / 연장제출 / 수강인원)
2020-04-08 00:00 ~ 2020-04-12 23:59	반영	공개	2020-04-19 00:00	 사용 	미허용	36 명 / 3 명 / 1 명 / 40 명

openweathermap
테스트 해보세요.

<https://github.com>

패키지 설치를 위해
사용하시면 됩니다

1. 윈도우의 명령프롬프트
2. 명령어 창에서
pip3 install requests

 명령 프롬프트

```
C:\Users\Yunhee>pip3 install requests
Requirement already satisfied: requests in c:\users\yunhee\miniconda3\lib\site-packages (2.21.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\yunhee\miniconda3\lib\site-packages (from requests) (2.21.0)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\yunhee\miniconda3\lib\site-packages (from requests) (2.8)
Requirement already satisfied: urllib3<1.25,>=1.21.1 in c:\users\yunhee\miniconda3\lib\site-packages (from requests) (2.4.1)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\yunhee\miniconda3\lib\site-packages (from requests) (3.0.4)
WARNING: You are using pip version 19.1.1, however version 20.0.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

패키지(또는 라이브러리)를 설치하고 JSON 자료를 가져오는 코드를 작성합니다.

4주차 실습

```
In [1]: import requests
```

```
In [2]: def search_city(city):  
  
    API_KEY = 'a070fcd8fc2db8d5d1f140466a2012b4' # initialize your key here  
    # call API and convert response into Python dictionary  
  
    url = f'http://api.openweathermap.org/data/2.5/weather?q={city}&APPID={API_KEY}'  
    response = requests.get(url).json()  
  
    print(response)  
    # error like unknown city name, invalid api key  
    if response.get('cod') != 200:  
        message = response.get('message', '')  
        return f'Error getting temperature for {city.title()}. Error message = {message}'  
  
    # get current temperature and convert it into Celsius  
    current_temperature = response.get('main', {}).get('temp')  
    if current_temperature:  
        current_temperature_celsius = round(current_temperature - 273.15, 2)  
        return f'Current temperature of {city.title()} is {current_temperature_celsius}'  
    else:  
        return f'Error getting temperature for {city.title()}'
```

```
In [3]: result = search_city('Seoul')
```

```
{'coord': {'lon': 126.98, 'lat': 37.57}, 'weather': [{'id': 800, 'main': 'Clear', 'description': 'clear sky', 'icon': '01d'}], 'base': 'stations', 'main': {'temp': 287.39, 'feels_like': 284.43, 'temp_min': 286.15, 'temp_max': 289.15, 'pressure': 1018, 'humidity': 47}, 'visibility': 10000, 'wind': {'speed': 2.1, 'deg': 230}, 'clouds': {'all': 1}, 'dt': 1586829293, 'sys': {'type': 1, 'id': 8117, 'country': 'KR', 'sunrise': 1586811505, 'sunset': 1586858750}, 'timezone': 32400, 'id': 1835848, 'name': 'Seoul', 'cod': 200}
```

```
In [4]: result
```

```
Out[4]: 'Current temperature of Seoul is 14.24'
```

```
In [ ]:
```

5주차 실습

• JSON 자료 다루기

[일반 과제] [일반 과제] 5주차-데이터 해석하기-openAPI(3) [마감] 수정 삭제

참고 자료 :

87%

제출기간	성적반영	성적공개	성적공개일자	사용 여부	연장제출	제출자수 (제출 / 미제출 / 연장제출)
2020-04-10 00:00 ~ 2020-04-17 23:59	반영	공개	2020-04-25 00:00	사용	미허용	35 명 / 4 명 / 1 명 / 40 명

```
In [18]: json = res.json()
         print(json)
```

```
{'coord': {'lon': 126.98, 'lat': 37.57}, 'weather': [{'id': 800, 'main': 'Clear', 'description': 'clear sky', 'icon': '01d'}], 'base': 'stations', 'main': {'temp': 286.33, 'feels_like': 283.01, 'temp_min': 285.15, 'temp_max': 287.15, 'pressure': 1022, 'humidity': 43}, 'visibility': 10000, 'wind': {'speed': 2.1, 'deg': 290}, 'clouds': {'all': 1}, 'dt': 1586511989, 'sys': {'type': 1, 'id': 8117, 'country': 'KR', 'sunrise': 1586466247, 'sunset': 1586512933}, 'timezone': 32400, 'id': 1835848, 'name': 'Seoul', 'cod': 200}
```

응답 JSON 자료 얻기

```
In [16]: print(json['coord'])
         print(json['coord']['lon'])
```

```
{'lon': 126.98, 'lat': 37.57}
126.98
```


6주차 실습

 [일반 과제] [일반 과제] [일반 과제] 6주차-데이터 해석하기-openAPI(3) 예외처리 [진행중]

 수정

 삭제

 과제평가

참고 자료 :  6주차-openweathermap-예외처리.pdf

2%

제출기간	성적반영	성적공개	성적공개일자	사용 여부	연장제출	제출자수 (제출 / 미제출 / 연장제출 / 수강인원)
2020-04-11 00:00 ~ 2020-04-24 23:59	반영	공개	2020-05-01 00:00	 사용 	미허용	1명 / 39명 / 0명 / 40명

openAPI 인 openweathermap API 수행을 위해 필요한 도시명에 오류가 있는 경우 이를 처리하도록 확장한다.

첨부된 내용은 도시명에 숫자 문자가 추가 되어진 경우 예외처리하도록 하였으며, 이를 확장하여 추가적인 예외처리를 위해 확장을 한다.

```
In [5]: import requests
```

```
In [2]: def search_city(city):

    API_KEY = 'a070fcd8fc2db8d5d1f140466a2012b4' # initialize your key here
    # call API and convert response into Python dictionary

    url = f'http://api.openweathermap.org/data/2.5/weather?q={city}&APPID={API_KEY}'
    response = requests.get(url).json()

    # error like unknown city name, invalid api key
    if response.get('cod') != 200:
        message = response.get('message', '')
        return f'Error getting temperature for {city.title()}. Error message = {message}'

    # get current temperature and convert it into Celsius
    current_temperature = response.get('main', {}).get('temp')
    if current_temperature:
        current_temperature_celsius = round(current_temperature - 273.15, 2)
        return f'Current temperature of {city.title()} is {current_temperature_celsius}'
    else:
        return f'Error getting temperature for {city.title()}'
```

```
In [10]: city = input("Enter city name:")
```

```
try:
    for char in city:
        if char in "0123456789":
            raise ValueError
except ValueError:
    print("Invalid city name")
else:
    print("city name %s" % city)
    result = search_city(city)
    print(result)
```

```
Enter city name:cheonan1
Invalid city name
```

 [02-1-XML-응용-openAPI.pdf](#) Add files via upload

 [Cheonan.csv](#) Add files via upload

 [OpenAPI JSON Deep Dive.ipynb](#) Add files via upload

 [OpenWeather_JSON_mini.py](#) Add files via upload

 [Openweathermap_JSON.ipynb](#) Add files via upload

 [Openweathermap_JSON.py](#) Add files via upload

 [Openweathermap_JSON_except.ipynb](#) Add files via upload

 [README](#) Update README

 [WeatherSightSeeing.py](#) Add files via upload

(7주) CSV 파일 다루기

Public_OpenAPI/OpenWeather_JSON_csv_writer.py

```
import csv
import time
from datetime import datetime
if __name__ == '__main__':
    city = input("Enter city name:")
    try:
        for char in city:
            if char in "0123456789":
                raise ValueError
    except ValueError:
        print ("Invalid city name")
    else:
        print("city name %s" % city)
        count = int(input("# of service rquests:"))

        with open("{}_csv".format(city), "w") as csv_file:
            fieldnames = ['Date', 'Temp']
            writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
            writer.writeheader()
            for i in range(count):
                result = search_city(city)
                print(result)
                now = datetime.now()
                date_time = now.strftime("%m/%d/%Y %H:%M:%S")
                writer.writerow({'Date': date_time, 'Temp': result})
                time.sleep(5)
```

```
Enter city name: london
city name london
# of service rquests:5
10.41
10.41
10.41
10.41
10.41
>>> |
```

1	Date	Temp
2		
3	04/28/202	10.41
4		
5	04/28/202	10.41
6		
7	04/28/202	10.41
8		
9	04/28/202	10.41
10		
11	04/28/202	10.41

- CSV 파일 읽기

기상청에서 다운로드 받은 CSV 의 자료 중 100개의 자료를 읽어 최대값, 최소값, 평균을 구한다.

- 자료는 날짜, 시간과 기온값으로 구성되어 있음
- 자료값은 기온값만을 대상으로 자료의 최대, 최소 및 평균을 구한다.

텀프로젝트 진행

• 수정계획서 제출 및 발표

 [일반 과제] 7-8주차-텀프로젝트 계획서 발표 [진행중]

 수정  삭제  과제평가

참고 자료 :

2%

제출기간	성적반영	성적공개	성적공개일자	사용 여부	연장제출	제출자수 (제출 / 미제출 / 연장제출 / 수강인원)
2020-04-20 00:00 ~ 2020-05-08 23:59	반영	공개	2020-05-11 00:00	 사용 	미허용	1명 / 39명 / 0명 / 40명

텀프로젝트 계획서 발표

- 발표 자료를 ppt 로 제출함
- 발표 자료에는 제출한 계획서를 중심으로 데이터분석을 하고자 하는 개방형 공공데이터 소개 및 데이터 분석 (예상)내용 및 (예상) 결과를 포함하여 작성함
- 발표 자료는 별도의 동영상으로 제작하여 Youtube 에 올린후 해당 Youtube 주소를 제출한다.
- 동영상 제작은 자신의 스마트폰 또는 zoom 기록를 통해 작성한다.
- 발표 동영상은 3분을 넘지않도록 하여야 하며 계획서를 중심으로 구성하여야 함

텀프로젝트 진행

• 평가

① 텀프로젝트 동료평가

 텀프로젝트-동료평가.xlsx

작성자 : 강윤희

* 목적 *

- (동료)제출 텀프로젝트의 공정한 평가를 위해 동료평가를 도입한다.
- (동료)평가의 표준화를 위해 상중하의 3단계로 평가하며, 상(2), 중(2), 하(1)개를 선정한다.
- (동료)단, 자신의 텀프로젝트는 평가하지 않는다.
- (교수)최종적으로 중복평가결과에 기준으로 제출 프로젝트의 점수를 부여한다.

* 수행 방법 *

- 제출된 텀프로젝트 계획서 중 5개를 선정하여 상중하를 평가한다.
상(2개) 중(2개) 하(1개)를 최종 결정한다.
- 동료평가는 제출된 텀프로젝트 만을 대상으로 한다.
- 평가를 위해 필요한 추가자료(동영상 발표자료는 Youtube 공지) 을 참고하며, 단, 추가자료가 제시하지 못한 경우
계획서만을 기준으로 평가를 수행한다.
- 평가를 위한 5개 프로젝트는 출석부에서 자신의 위치를 기준으로 위에서 2개 아래에서 3개를 선정한다.
단, 위에 2개가 없는 경우 마지막의 2개를 선택하며, 아래에 3개가 없는 경우 앞에서 3개를 선정한다.
전체적으로 출석부를 원형큐로 이해하여 앞에서 2개 뒤에서 3개를 선정한다.
- 제출된 텀프로젝트의 계획서를 검토하여 4가지 항목에 대해 평가를 수행한다.
- 해당 항목은 1에서 5점을 부여한다. (상: 4-5, 중:3-2, 하: 1점)
- 4가지 항목에 대한 평균값으로 최종 상중하를 부여한다.

10주 pandas 사용

- Pandas CSV Read

지역별 온도데이터

<https://data.kma.go.kr/>

```
In [1]: import pandas as pd
```

```
In [2]: try:
df = pd.read_csv('400_20200411_221745769.csv')
except OSError as err:
    print("OS error: {0}".format(err))
df.head()
```

```
Out [2]:
```

	format: day	hour	value location:63_108 Start : 20190401
0	1	0.0	8.1
1	1	100.0	8.8
2	1	200.0	8.3
3	1	300.0	9.8
4	1	400.0	9.5

```
In [3]: df.columns
```

```
Out [3]: Index(['format: day', 'hour', 'value location:63_108 Start : 20190401'], dtype='object')
```

```
In [4]: df = df.rename(columns={'value location:63_108 Start : 20190401': 'temp'})
```

```
In [5]: df.head()
```

```
Out [5]:
```

	format: day	hour	temp
0	1	0.0	8.1
1	1	100.0	8.8
2	1	200.0	8.3
3	1	300.0	9.8
4	1	400.0	9.5

```
In [9]: import matplotlib.pyplot as plt
plt.style.use('seaborn-whitegrid')
```

```
In [10]: x = list(range(len(df_temp)))
```

```
In [11]: plt.plot(x, df_temp, ':b', color='g')
plt.xlim(0, 5)
plt.ylim(15, 50);
plt.axis('tight');
plt.title("63_108")
plt.ylabel("Temp");

plt.show()
```

