

(4주차) 데이터 분석

백석대학교 강윤희

4주차 수업

2. 복습-3주차-파이썬교재실습	4명	2명	34명	 (0)	강의 보기
1. 2-3주차 과제문제풀이	3명	1명	36명	 (0)	강의 보기
3. 리스트 컴프리헨션	3명	1명	36명	 (0)	강의 보기
4. 반복문-while	3명	0명	37명	 (0)	강의 보기

3주차 zoom 실습

- 문자열 결합하기

```
>>> x = "awesome"
>>>
>>> print("Python is " + x)
Python is awesome
>>> |
```

3주차 zoom 실습

- 문자열 리스트로 구성하기

```
a = "Life is too short"
```

```
a.split()
```

```
a = "Art is long. Life is too short"
```

```
a.split('.')    #부분에서는 . 문자를 기준으로 문자열을 나눈다.
```

```
['Art is long', ' Life is too short']
```

3주차 for문 실습

- 다음을 실행하여 처리과정의 오류 메시지를 이해한다.

```
for i in 3:  
    print(i)
```

```
Traceback (most recent call last):  
  File "C:/Users/Yunhee/Desktop/for_range_test.py", line 2, in <module>  
    for i in 3:  
TypeError: 'int' object is not iterable  
>>> |
```

3주차 for문 실습

- 다음은 for와 range 함수를 사용하면 1부터 10까지 더하는 것을 보인 것이다.

```
s = 0
```

```
for x in range(1, 11):
```

```
    s = s + x
```

```
print (" x:", x, "sum : ", s)
```



3주차과제- for문

- for 문과 if 문 으로 자료 추출

다음 리스트에서 50점 이상의 점수들의 총합을 구하시오.
marks=[90,25,67,45,80]

```
total = 0
i = 0
higher_cnt = 0
for i in range(len(marks)):
    if marks[i] >= 50:
        higher_cnt += 1
        total = total + marks[i]
    i = i + 1
```

```
print ("50점 이상의 점수들의 총합 = %d" % total)
print ("50점 이상의 점수들의 평균 = %d" % (total/higher_cnt))
```

3주차과제-리스트 활용

- 자료 추출하여 새로운 리스트 만들기

```
marks=[90,25,67,45,80]
```

```
higher = []  
for i in range(len(marks)):  
    print ("%d 번째 학생 점수 %d" % (i, marks[i]))  
    if marks[i] > 50:  
        higher.append(marks[i])
```

```
total = sum(higher)  
average = total/len(higher)
```

```
print ("50점 이상 학생 점수 평균 %d" % average)
```


(4주차) 리스트 활용

- 리스트 컴프리헨션 활용 자료 추출

```
higher = [x for x in marks if x > 50]  
print ("50점 이상 학생 점수 평균 = {}".format(sum(higher)/len(higher)))
```

3주차 실습

- 점수별 평균구하기

```
-- 학생 축하합니다. 합격입니다.
1 90 79
2 25 35
3 67
4 45
5 80
>>>

File Edit Format Run Options Window Help
passCount = 0
passSum = 0
failCount = 0
failSum = 0
passAvg = 0
failAvg = 0

marks=[90,25,67,45,80]

for idx,mark in enumerate(marks):
    if(mark>=60):
        passCount += 1
        passSum += mark
        print("%d번 학생 축하합니다. 합격입니다."%(idx+1))
    else:
        failCount += 1
        failSum += mark
|
passAvg = (passSum)/(passCount)
failAvg = (failSum)/(failCount)

print("합격학생의 평균 %7d"%passAvg)
print("불합격학생의 평균 %5d"%failAvg)
```

3주차 실습

```
marks = [90, 25, 67, 45, 80]

pass_cnt = fail_cnt = 0 #학생수 변수
pass_sum = fail_sum = 0 #합격/불합격자 점수 합계 변수
pass_avg = fail_avg = 0 #합격/불합격자 점수 평균 변수
number = 0

for i in marks:
    number = number + 1
    if i >= 60:
        print("%d번 학생 축하합니다. 합격입니다." % number)
        pass_cnt += 1
        pass_sum += i
        pass_avg = pass_sum / pass_cnt
    else:
        fail_cnt += 1
        fail_sum += i
        fail_avg = fail_sum / fail_cnt

print("합격한 사람의 평균 : %d" % pass_avg)
print("불합격한 사람의 평균 : %d" % fail_avg)
```

4주차 과제

(데이터 수집) openAPI 활용

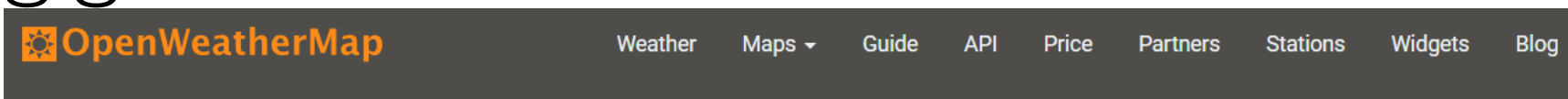
데이터 준비하기
(XML 및 JSON 자료 다루기)

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

데이터 준비

• 계정 생성



Weather API

[Home](#) / [Weather API](#)

Please [sign up](#) and use our fast and easy-to-work weather APIs for free. Look at our [monthly subscriptions](#) for more options than Free account can provide you. Read [How to start](#) first and enjoy using our powerful weather APIs.

Current weather data

[API doc](#) [Subscribe](#)

- Access current weather data for any location including over 200,000 cities
- Current weather is frequently updated based on global models and data from more than 40,000 weather stations
- Data is available in JSON, XML, or HTML format
- Available for Free and all other paid accounts

Hourly forecast ^{NEW}

[API doc](#) [Subscribe](#)

- Hourly forecast is available for 4 days
- Forecast weather data for 96 timestamps
- Higher geographic accuracy
- Forecast is available in JSON and XML
- Available for Developer, Professional and Enterprise accounts
- **Free trial for all users until 1st May 2019!**

16 day / daily forecast

[API doc](#) [Subscribe](#)

- 16 day forecast is available at any location or city
- 16 day forecast includes daily weather
- Forecast is available in JSON and XML
- Available for all paid accounts

데이터 준비

• API 키 얻기



2019-03-06 (수) 오후 2:49

OWM Team <robot2@openweathermap.org>

OpenWeatherMap Create Account

받는 사람 Customer


이 메시지가 표시되는 방식에 문제가 있으면 여기를 클릭하여 웹 브라우저에서 메시지를 확인하십시오.

Thank you for subscribing to OpenWeather API!

Dear Customer!

Thank you for subscribing to Free [OpenWeather API!](#)

API key:

- Your API key is **a070**  **012b4**
- Within the next couple of hours, it will be activated and ready to use
- You can later create more API keys on your [account page](#)
- Please, always use your API key in each API call

응답 등록키

데이터 준비

- 도시 날씨 정보를 openweathermap에 요청함
 - (사전작업) 요청을 위해 id를 얻어야 함

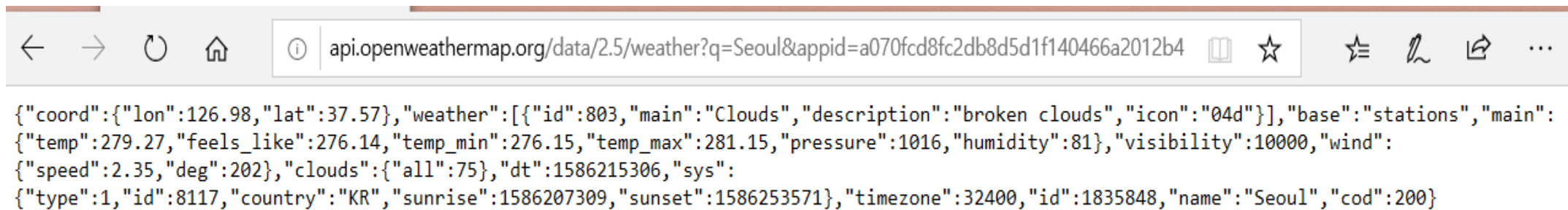
요청

<http://api.openweathermap.org/data/2.5/weather?q=Seoul&appid=등록id>

응답

```
{"coord":{"lon":126.98,"lat":37.57},"weather":[{"id":501,"main":"Rain","description":"moderate rain","icon":"10n"}, {"id":701,"main":"Mist","description":"mist","icon":"50n"}],"base":"stations","main":{"temp":277.91,"pressure":1014,"humidity":93,"temp_min":276.15,"temp_max":279.15,"visibility":6000,"wind":{"speed":2.6,"deg":10,"gust":6.2},"rain":{"1h":1.78},"clouds":{"all":90},"dt":1554817200,"sys":{"type":1,"id":5509,"message":0.0044,"country":"KR","sunrise":1554757599,"sunset":1554804038},"id":1835848,"name":"Seoul","cod":200}}
```

(예상)과제 결과



The screenshot shows a web browser window with the address bar displaying the URL: `api.openweathermap.org/data/2.5/weather?q=Seoul&appid=a070fcd8fc2db8d5d1f140466a2012b4`. The page content displays a JSON object representing weather data for Seoul. The JSON includes coordinates, weather conditions (broken clouds), temperature, pressure, humidity, visibility, wind speed, and location details like country and sunrise/sunset times.

```
{
  "coord": {
    "lon": 126.98,
    "lat": 37.57
  },
  "weather": [
    {
      "id": 803,
      "main": "Clouds",
      "description": "broken clouds",
      "icon": "04d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 279.27,
    "feels_like": 276.14,
    "temp_min": 276.15,
    "temp_max": 281.15,
    "pressure": 1016,
    "humidity": 81,
    "visibility": 10000,
    "wind": {
      "speed": 2.35,
      "deg": 202
    },
    "clouds": {
      "all": 75
    },
    "dt": 1586215306,
    "sys": {
      "type": 1,
      "id": 8117,
      "country": "KR",
      "sunrise": 1586207309,
      "sunset": 1586253571,
      "timezone": 32400,
      "id": 1835848,
      "name": "Seoul",
      "cod": 200
    }
  }
}
```