

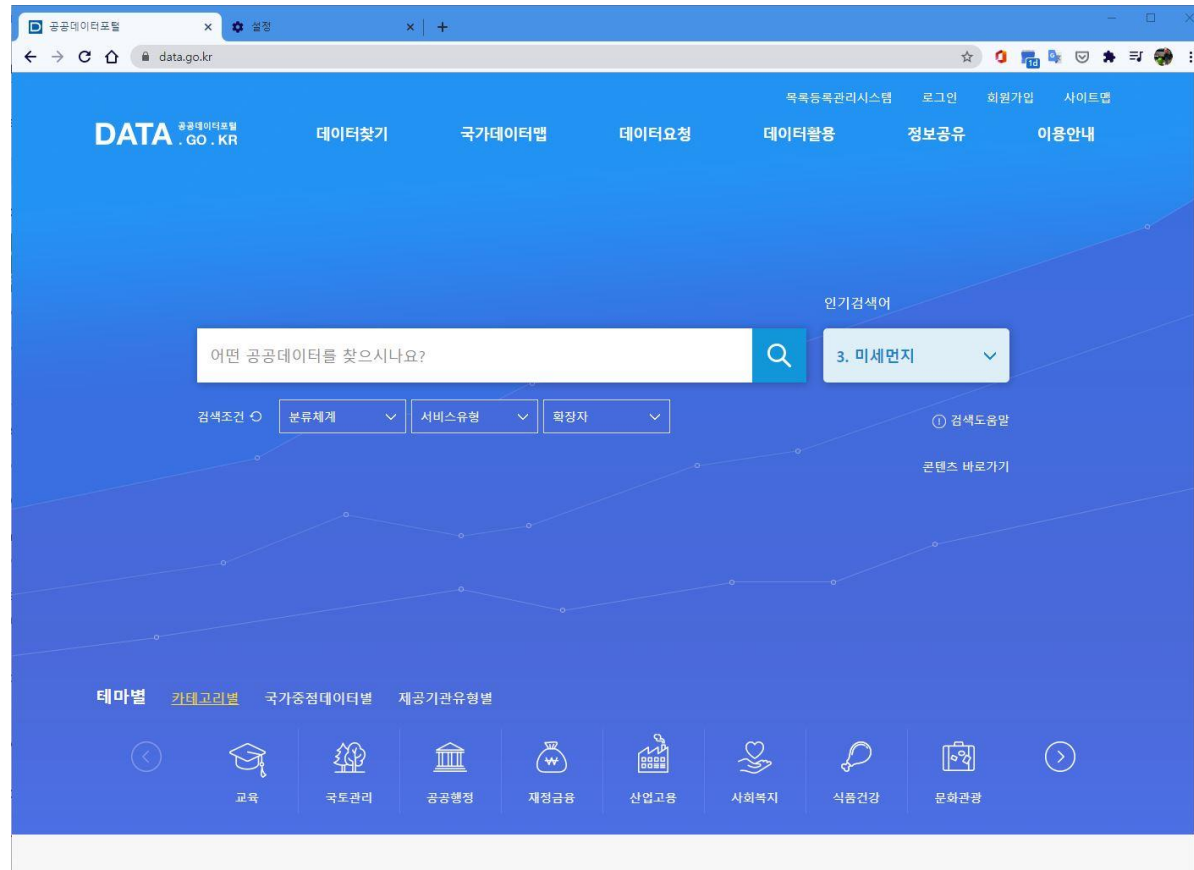
보강주
파이썬 기초 프로그래밍



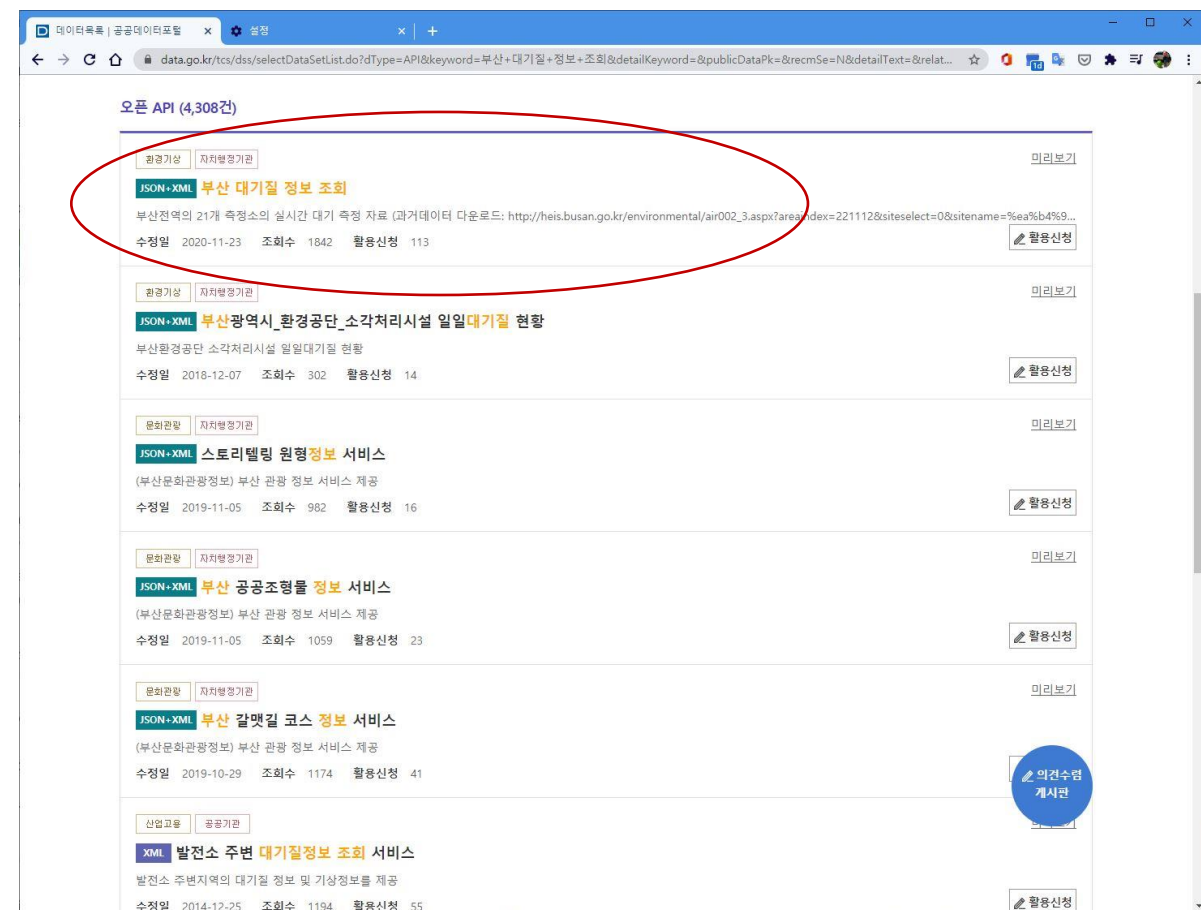
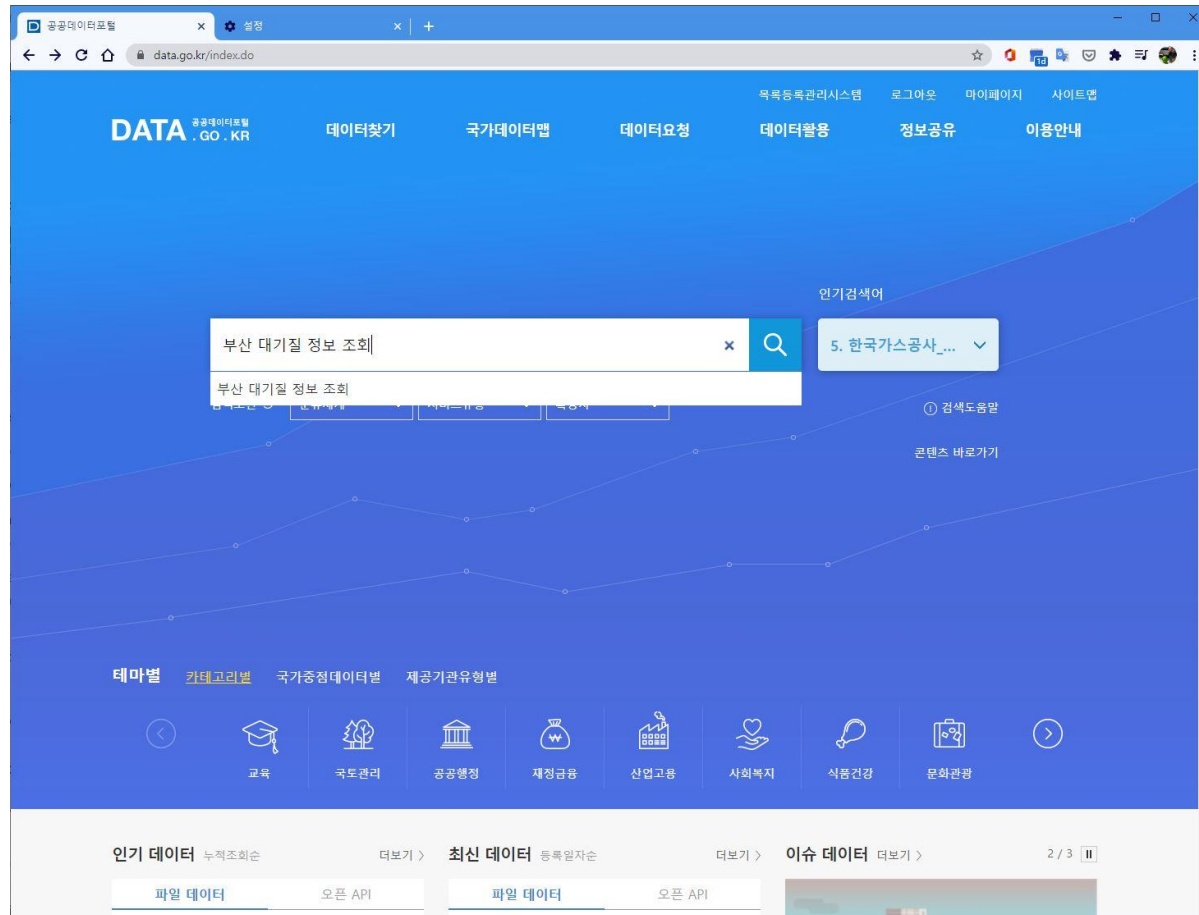
XML, JSON을 이용한
공공데이터 파싱

공공데이터 포털 : <http://data.go.kr>

- 회원 가입



오픈 API : 부산 대기질 정보 조회



활용 신청

JSON+XML 부산 대기질 정보 조회

부산전역의 21개 측정소의 실시간 대기 측정 자료 (과거데이터 다운로드: http://heis.busan.go.kr/environmental/air002_3.aspx?areaindex=221112&siteselect=0&sitenamename=%ea%b4%91%eb%b3%b5%eb%8f%99&yearselect=0&monthselect=9&year=2010&month=10)

👍 2 🗨️ 0 □ 관심

활용신청
유류신고 및
담당자 문의

OpenAPI 정보

| | | | |
|--------|----------------------------------|-----------|--------------------------------------|
| 분류체계 | 환경 - 대기 | 제공기관 | 부산광역시 |
| 관리부서명 | 통계빅데이터담당관 | 관리부서 전화번호 | 051-888-2364 |
| API 유형 | REST | 데이터포맷 | JSON+XML |
| 활용신청 | 113 | 키워드 | 실시간 대기질, 미세먼지, 부산측정소 정보, 실시간 데이터, 환경 |
| 등록 | 2019-11-08 | 수정 | 2020-11-24 |
| 심의유형 | 개발단계 : 허용 / 운영단계 : 허용 | | |
| 비용부과유무 | 무료 | | |
| 이용허락범위 | 이용허락범위 제한 없음 | | |
| 참고문서 | 부산 대기질 정보조회.docx | | |

상세기능 활용사례

상세기능

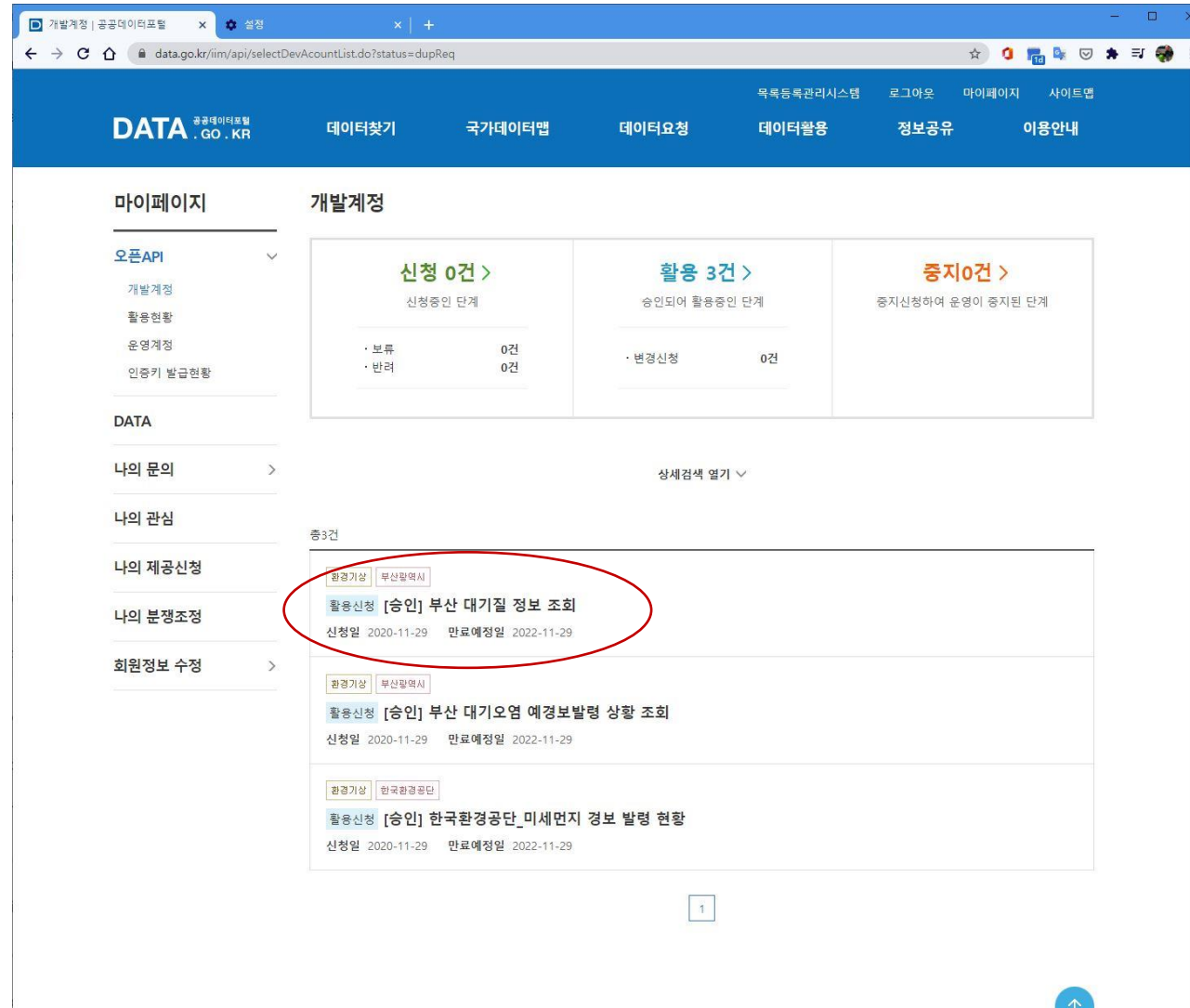
목록 부산 대기질 측정소별 정보 조회 조회

부산시 대기질 정보(지역이름, 일산화탄소, 오존, 미세먼지 등)를 측정소별로 제공

- 활용승인 절차 개발단계 : 허용 / 운영단계 : 허용
- 신청가능 트래픽 1000000 / 운영계정은 활용사례 등록시 신청하면 트래픽 증가 가능
- 요청주소 <http://apis.data.go.kr/6260000/AirQualityInfoService/getAirQualityInfoClassifiedByStation>
- 서비스URL <http://apis.data.go.kr/6260000/AirQualityInfoService>

활용신청

활용 승인 -> 마이페이지



The screenshot shows the '마이페이지' (My Page) section of the DATA GO.KR portal. The left sidebar contains navigation links: '오픈API' (Open API), 'DATA', '나의 문의' (My Inquiry), '나의 관심' (My Interest), '나의 제공신청' (My Application), '나의 분쟁조정' (My Dispute Resolution), and '회원정보 수정' (Edit Member Information). The main content area is titled '개발계정' (Developer Account) and displays three summary boxes: '신청 0건' (0 Applications), '활용 3건' (3 Applications), and '중지0건' (0 Suspended). Below these, a table lists the applications. The first application, '활용신청 [승인] 부산 대기질 정보 조회' (Application for Use [Approved] Busan Air Quality Information Inquiry), is circled in red. It shows a status of '승인' (Approved) and a date of '2020-11-29'. The second application is '활용신청 [승인] 부산 대기오염 예경보발령 상황 조회' (Application for Use [Approved] Busan Air Pollution Forecast Alert Status Inquiry), and the third is '활용신청 [승인] 한국환경공단_미세먼지 경보 발령 현황' (Application for Use [Approved] Korea Environment Corporation - PM2.5 Alert Status).

| 신청일 | 만료예정일 |
|------------|------------|
| 2020-11-29 | 2022-11-29 |
| 2020-11-29 | 2022-11-29 |
| 2020-11-29 | 2022-11-29 |

데이터명 : 부산 대기질 정보 조회

- 서비스정보 -> 참고문서 : 부산 대기질 정보조회.doc 다운로드

The screenshot displays the 'DATA.GOV.KR' portal interface. The left sidebar contains navigation links such as '마이페이지', '오픈API', 'DATA', '나의 문의', '나의 관심', '나의 제공신청', '나의 분쟁조정', and '회원정보 수정'. The main content area is titled '개발계정 상세보기' (Developer Account Detail View) and includes a '기본정보' (Basic Information) table, a '서비스정보' (Service Information) table, a '활용신청 상세기능정보' (Application Detail Function Information) table, and a '활용정보' (Usage Information) table.

기본정보

| | | | |
|-------|-------------------------|------|------|
| 데이터명 | 부산 대기질 정보 조회 | | |
| 서비스유형 | REST | 심의여부 | 자동승인 |
| 신청유형 | 개발계정 활용신청 | 처리상태 | 승인 |
| 활용기간 | 2020-11-29 ~ 2022-11-29 | | |

서비스정보

| | |
|----------------|---|
| 일반 인증키 (UTF-8) | %2FxdTzUirW%2Bz9e39M6z8BXMIcO3NoCn5YKg2HKcLLN%2FVODtKBpk5m0jRr0MEHDon1cU9SPfqjFMEEMaxpxEw%3D%3D |
| End Point | http://apis.data.go.kr/6260000/AirQualityInfoService |
| 데이터포맷 | JSON+XML |
| 참고문서 | 부산 대기질 정보조회.doc |

활용신청 상세기능정보

| NO | 상세기능 | 설명 | 일일 트래픽 | 미리보기 |
|----|-------------------|---|--------|--------------------|
| 1 | 부산 대기질 항목별 정보 조회 | 부산시 대기질 정보를(측정항목, 측정시간, 좌동, 연산 등) 항목별로 제공 | 10000 | 확인 |
| 2 | 부산 대기질 측정소별 정보 조회 | 부산시 대기질 정보(지역이름, 일산화탄소, 오존, 미세먼지 등)를 측정소별로 제공 | 10000 | 확인 |

활용정보

| | |
|------|------------------|
| 활용목적 | 앱개발 (모바일, 솔루션 등) |
| 활용내용 | 교육 |

활용신청 상세기능 정보 -> 2. 부산 대기질 측정소별 정보 조회

- getAirQualityInfoClassifiedByStation

개발계정 상세보기 | 공공데이터 x 설정

data.go.kr/iim/api/selectAPIAccountView.do

활용신청 상세기능정보

| NO | 상세기능 | 설명 | 일일 트래픽 | 미리보기 |
|----|-------------------|---|--------|------|
| 1 | 부산 대기질 항목별 정보 조회 | 부산시 대기질 정보를(측정항목, 측정시간, 좌표, 연산등) 항목별로 제공 | 10000 | 확인 |
| 2 | 부산 대기질 측정소별 정보 조회 | 부산시 대기질 정보(지역이름, 일산화탄소, 오존, 미세먼지 등)를 측정소별로 제공 | 10000 | 확인 |

요청변수(Request Parameter)

닫기

| 항목명 | 샘플데이터 | 설명 |
|---------------|-----------------|---------------------------------------|
| serviceKey | 인증키(URL Encode) | 공공데이터포털에서 받은 인증키 |
| pageNo | 1 | 페이지번호 |
| numOfRows | 5 | 한 페이지 결과 수 |
| resultType | json | JSON방식으로 호출 시 파라미터 resultType=json 입력 |
| areaIndex | 221251 | 측정소코드 |
| controlnumber | 2019101109 | 측정시간 |

미리보기

활용정보

| | |
|------|----------------|
| 활용목적 | 앱개발 (모바일,솔루션등) |
| 활용내용 | 교육 |

라이선스표시

| | |
|--------|--------------|
| 이용허락범위 | 이용허락범위 제한 없음 |
|--------|--------------|

운영계정 신청

변경 신청

연장 신청

증지 신청

일반 인증키 재발급

목록

부산 대기질 정보 조회

- 부산 대기질 측정소별 정보 조회 : `getAirQualityInfoClassifiedByStation`
- 요청 메시지 정보

| 항목명(영문) | 항목명(국문) | 항목크기 | 항목구분 | 샘플데이터 | 항목설명 |
|---------------|---------------|------|------|---------------------|--|
| serviceKey | 인증키 | 100 | 1 | 인증키 (URL Encode) | 공공데이터포털에서 발급 받은 인증키 |
| numOfRows | 한 페이지 결과 수 | 4 | 1 | 5 | 한 페이지 결과 수 |
| pageNo | 페이지 번호 | 4 | 1 | 1 | 페이지 번호 |
| resultType | JSON 방식 호출 | 4 | 0 | json | JSON방식으로 호출 시 파라미터 resultType=json 입력 |
| areaIndex | 측정소코드 | 10 | 0 | 221251 | 측정소코드 |
| controlnumber | 측정시간 | 10 | 0 | 2019101109 | 측정시간 |

• 측정소 코드

| 분류 | 측정소 | 입력코드값 | 측정소 | 입력코드값 |
|-----------|-----|--------|-----|--------|
| areaIndex | 광복동 | 221112 | 장림동 | 221202 |
| | 초량동 | 221131 | 대저동 | 221211 |
| | 태종대 | 221141 | 녹산동 | 221212 |
| | 전포동 | 221152 | 연산동 | 221221 |
| | 온천동 | 221162 | 기장읍 | 221231 |
| | 명장동 | 221163 | 용수리 | 221233 |
| | 대연동 | 221172 | 수정동 | 221241 |
| | 학장동 | 221181 | 부곡동 | 221251 |
| | 덕천동 | 221182 | 광안동 | 221271 |
| | 청룡동 | 221191 | 대신동 | 221281 |
| | 좌동 | 221192 | | |

• 응답 메시지

| 항목명(영문) | 항목명(국문) | 항목크기 | 항목구분 | 샘플데이터 | 항목설명 |
|---------------|------------|------|------|------------|-----------------------------------|
| numOfRows | 한 페이지 결과 수 | 4 | 1 | 10 | 한 페이지당 표출 데이터 수 |
| pageNo | 페이지 수 | 4 | 1 | 1 | 페이지 수 |
| resultCode | 결과코드 | 2 | 1 | 00 | 결과코드 |
| resultMsg | 결과메시지 | 50 | 1 | OK | 결과메시지 |
| totalCount | 데이터 총 개수 | 4 | 1 | 17 | 데이터 총 개수 |
| site | 지역이름 | 255 | 0 | 부곡동 | 지역이름 |
| areaIndex | 측정소코드 | 10 | 0 | 221251 | 측정소코드 |
| controlnumber | 측정시간 | 10 | 0 | 2019101109 | 측정시간 |
| repltem | 대표항목 | 10 | 0 | PM25 | 대표항목 |
| repVal | 대표항목 값 | 50 | 0 | 66 | 대표항목 값 |
| repCai | 대표항목 지수 | 10 | 0 | 2 | 대표항목 지수 |
| so2 | 아황산가스 | 50 | 0 | 0.004 | 아황산가스(ppm) |
| so2Cai | 아황산가스 지수 | 10 | 0 | 1 | 아황산가스 지수 |
| no2 | 이산화질소 | 50 | 0 | 0.031 | 이산화질소(ppm) |
| no2Cai | 이산화질소 지수 | 10 | 0 | 2 | 이산화질소 지수 |
| o3 | 오존 | 50 | 0 | 0.012 | 오존(ppm) |
| o3Cai | 오존 지수 | 10 | 0 | 1 | 오존 지수 |
| co | 일산화탄소 | 50 | 0 | 0.4 | 일산화탄소(ppm) |
| coCai | 일산화탄소 지수 | 10 | 0 | 1 | 일산화탄소 지수 |
| pm25 | 초미세먼지 | 50 | 0 | 23 | 초미세먼지($\mu\text{g}/\text{m}^3$) |
| pm25Cai | 초미세먼지 지수 | 10 | 0 | 2 | 초미세먼지 지수 |
| pm10 | 미세먼지 | 50 | 0 | 38 | 미세먼지($\mu\text{g}/\text{m}^3$) |
| pm10Cai | 미세먼지 지수 | 10 | 0 | 2 | 미세먼지 지수 |

```

import requests
from bs4 import BeautifulSoup

M = '&numOfRows=1&pageNo=1&areaIndex=221152'
# M = '&numOfRows=1&pageNo=1&areaIndex=221152&resultType=json'

key = '%2FxDJTZuIRw%2Bz9e39M6z8BXMic03NoCn5YKg2HKcLLN%2FVOD' \
      + 'tKBepk5m0jjRr0MEHDon1cU9SPfqjFMEeMaxpxEw%3D%3D'

url = 'http://apis.data.go.kr/6260000/AirQualityInfoService/' \
      + 'getAirQualityInfoClassifiedByStation?serviceKey=' + key + M

response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")

ItemList = soup.findAll('item')

for item in ItemList:
    a = item.find('areaindex').text
    g = item.find('controlnumber').text
    i = item.find('pm10').text
    s = item.find('pm25').text
    t = item.find('pm10cai').text
    u = item.find('pm25cai').text

    print('측정소: ' + a)
    print('측정시간: ' + g)
    print('미세먼지 농도: ' + i + 'µg/m³ ( ' + t + ' )')
    print('초미세먼지 농도: ' + s + 'µg/m³ ( ' + u + ' )')
    print('( 좋음: 1 ),( 보통: 2 ),( 나쁨: 3 ),( 매우나쁨: 4)')
    print('\n')

```

실습 03 : XML 파싱

- BeautifulShop 모듈을 사용한 XML 파싱
- 결과 값

```

측정소: 221152
측정시간:2020120310
미세먼지 농도:32µg/m³ ( 1 )
초미세먼지 농도:20µg/m³ ( 2 )
( 좋음: 1 ),( 보통: 2 ),( 나쁨: 3 ),( 매우나쁨: 4)

```

웹 브라우저에서 테스트

- XML 데이터 받기
 - <http://apis.data.go.kr/6260000/AirQualityInfoService/getAirQualityInfoClassifiedByStation?serviceKey=%2FxDJTZuIRw%2Bz9e39M6z8BXMIcO3NoCn5YKg2HKcLLN%2FVODtKBEpk5m0jjRr0MEHDon1cU9SPfqjFMEeMaxpxEw%3D%3D>

```

▼<item>
  <site>청룡동</site>
  <no2Cai>1</no2Cai>
  <o3Cai>1</o3Cai>
  <so2>0.003</so2>
  <pm25Cai>1</pm25Cai>
  <repItem>PM25</repItem>
  <o3>0.025</o3>
  <repCai>1</repCai>
  <so2Cai>1</so2Cai>
  <coCai>1</coCai>
  <controlNumber>2020120210</controlNumber>
  <co>0.4</co>
  <repVal>47</repVal>
  <no2>0.02</no2>
  <pm10Cai>1</pm10Cai>
  <pm10>20</pm10>
  <pm25>15</pm25>
  <areaIndex>221191</areaIndex>
</item>
▼<item>
  <site>전포동</site>
  <no2Cai>1</no2Cai>
  <o3Cai>1</o3Cai>
  <so2>0.004</so2>
  <pm25Cai>2</pm25Cai>
  <repItem>PM25</repItem>
  <o3>0.012</o3>
  <repCai>2</repCai>
  <so2Cai>1</so2Cai>
  <coCai>1</coCai>
  <controlNumber>2020120210</controlNumber>
  <co>0.5</co>
  <repVal>69</repVal>
  <no2>0.025</no2>
  <pm10Cai>2</pm10Cai>
  <pm10>55</pm10>
  <pm25>27</pm25>
  <areaIndex>221152</areaIndex>
</item>
▼<item>
  <site>태종대</site>
  <no2Cai>1</no2Cai>
  <o3Cai>1</o3Cai>
  <so2>0.004</so2>
  <pm25Cai>1</pm25Cai>
  <repItem>PM25</repItem>
  <o3>0.018</o3>
  <repCai>1</repCai>
  <so2Cai>1</so2Cai>
  <coCai>1</coCai>
  <controlNumber>2020120210</controlNumber>
  <co>0.3</co>
  <repVal>47</repVal>
  <no2>0.021</no2>
  <pm10Cai>1</pm10Cai>
  <pm10>28</pm10>
  <pm25>15</pm25>
  <areaIndex>221141</areaIndex>
</item>
</items>
<numOfRows>10</numOfRows>
<pageNo>1</pageNo>
<totalCount>985</totalCount>
</body>
</response>

```

웹 브라우저에서 테스트

- JSON 데이터 받기

<http://apis.data.go.kr/6260000/AirQualityInfoService/getAirQualityInfoClassifiedByStation?serviceKey=%2FxDJTZuIRw%2Bz9e39M6z8BXMlcO3NoCn5YKg2HKcLLN%2FVODtKBEpk5m0jjRr0MEHDon1cU9SPfqjFMEeMaxpxEw%3D%3D&resultType=json>

```
← → ↺ ⌂ ⚠ 주의 요함 | apis.data.go.kr/6260000/AirQualityInfoService/getAirQ
{
  "getAirQualityInfoClassifiedByStation": {
    "header": {
      "code": "00",
      "message": "NORMAL_CODE"
    },
    "item": [
      {
        "site": "기장읍",
        "areaIndex": "221231",
        "controlnumber": "2020120312",
        "repltem": "03",
        "repVal": "55",
        "repCai": "2",
        "so2": "0.004",
        "so2Cai": "1",
        "no2": "0.006",
        "no2Cai": "1",
        "o3": "0.036",
        "o3Cai": "2",
        "co": "0.3",
        "coCai": "1",
        "pm25": "15",
        "pm25Cai": "2",
        "pm10": "17",
        "pm10Cai": "1"
      },
      {
        "site": "광복동",
        "areaIndex": "221112",
        "controlnumber": "2020120312",
        "repltem": "PM25",
        "repVal": "54",
        "repCai": "2",
        "so2": "0.003",
        "so2Cai": "1",
        "no2": "0.008",
        "no2Cai": "1",
        "o3": "0.033",
        "o3Cai": "2",
        "co": "0.4",
        "coCai": "1",
        "pm25": "24",
        "pm25Cai": "2",
        "pm10": "34",
        "pm10Cai": "1"
      }
    ]
  }
}
```

JSON 데이터 파싱하기

- JSON 인코딩

- 파이썬 객체(Dictionary, List, tuple 등)을 JSON 문자열(str)로 변환

- **pip install json**

- JSON 인코딩/ 디코딩 표준 라이브러리

```
import json

# 테스트용 Python Dictionary
customer = {
    'id': 152352,
    'name': 'Hong Gil Dong',
    'history': [
        {'date': '2015-03-11', 'item': 'iPhone'},
        {'date': '2016-02-23', 'item': 'Monitor'},
    ]
}

# JSON 인코딩
jsonString = json.dumps(customer, indent=4)

# 문자열 출력
print(jsonString)
print(type(jsonString))  # class str
```

Dictionary

- 결과 값 :

문자열(str)

```
{
  "history": [
    {
      "date": "2015-03-11",
      "item": "iPhone"
    },
    {
      "date": "2016-02-23",
      "item": "Monitor"
    }
  ],
  "id": 152352,
  "name": "hong Gil Dong"
}
```


- JSON 디코딩

- JSON 문자열(str)을 파이썬 객체(Dictionary, List, Tuple 등)으로 변환

```
import json

# 테스트용 JSON 문자열
jsonString = '{"name": "Hong Gil Dong", \
              "id": 152352, \
              "history": [{"date": "2015-03-11", \
                           "item": "iPhone"}, \
                           {"date": "2016-02-23", \
                           "item": "Monitor"}]}'

# JSON 디코딩
dict = json.loads(jsonString)

# Dictionary 데이터 출력
print(dict['name'])

for h in dict['history']:
    print(h['date'], h['item'])
```

문자열(str)

- 결과값 :

Hong Gil Dong
2015-03-11 iPhone
2016-02-23 Monitor

Dictionary

공공데이터 JSON 파싱

- API 요청 명세 메시지 : **resultType=json**

```
import requests
import json
```

```
M = '&numOfRows=5&pageNo=1&resultType=json'
key = '%2FxDJTZuIRw%2Bz9e39M6z8BXMlcO3NoCn5YKg2HKcLLN%2FVODtKBepk5m0jjRr0MEHDon1cU9SPfqjFMEeMaxpxEw%3D%3D' + M
url = 'http://apis.data.go.kr/6260000/AirQualityInfoService/getAirQualityInfoClassifiedByStation?serviceKey=' + key
```

```
result = requests.get(url)
print(result)
```

```
# json 문자열(str)을 파이썬 객체(dictionary)로 변환
js = json.loads(result.content)
print(js)
```

```
data = js['getAirQualityInfoClassifiedByStation']['item']
print(data)
print(data[0])
```

```
outVal = data[0]
for key, val in outVal.items():
    print(key + ': ' + val)
```

```
print('Cai : ( 좋음: 1 ),( 보통: 2 ),( 나쁨: 3 ),( 매우나쁨: 4)')
```

결과 값

<Response [200]>

```
{'getAirQualityInfoClassifiedByStation': {'header': {'code':  
[{'site': '기장읍', 'areaIndex': '221231', 'controlnumber': '  
{'site': '기장읍', 'areaIndex': '221231', 'controlnumber': '2
```

site: 기장읍

areaIndex: 221231

controlnumber: 2020121113

repItem: PM25

repVal: 64

repCai: 2

so2: 0.003

so2Cai: 1

no2: 0.006

no2Cai: 1

o3: 0.046

o3Cai: 2

co: 0.3

coCai: 1

pm25: 22

pm25Cai: 2

pm10: 21

pm10Cai: 1

Cai : (좋음: 1),(보통: 2),(나쁨: 3),(매우나쁨: 4)

실습 04 : BeautifulSoup 모듈을 이용한 XML 파싱

- 실습 03를 참고하여, 부산 대기질 항목별(Item) 정보를 출력하는 프로그램을 작성하시오.
 - `getAirQualityInfoClassifiedByItem`
 - `item : 'site', 'val', 'controlnumber', 'item'`
 - 부산시 대기질 정보를(항목, 측정시간, 측정소, 측정값 등) 항목별로 제공
 - Callback URL :
 - <http://apis.data.go.kr/6260000/AirQualityInfoService/getAirQualityInfoClassifiedByItem>
 - 부산 대기질 정보 조회 문서 참고(.doc)

실습 05 : json 표준 라이브러리를 이용한 JSON 파싱

- 실습 04를 참고하여, 부산 대기질 항목별(Item) 정보를 출력하는 프로그램을 작성하시오.
 - `getAirQualityInfoClassifiedByItem`
 - `item : 'site', 'val', 'controlnumber', 'item'`
 - 부산시 대기질 정보를(항목, 측정시간, 측정소, 측정값 등) 항목별로 제공
 - Callback URL :
 - <http://apis.data.go.kr/6260000/AirQualityInfoService/getAirQualityInfoClassifiedByItem>
 - 부산 대기질 정보 조회 문서 참고(.doc)

과제

- 실습 01, 02, 03, 04, 05를 작성하여 구글 클래스룸에 올리시오.
 - 기한 : 2020년 12월 20(일) 까지 : 제출 기간 연장
 - 성적처리 관계로 기한 엄수
- 성적 처리 기준
 - 출석 : 20%
 - 중간 시험 : 20%
 - 기말 시험 : 20%
 - 과제 : 30 % -> 주석문을 주로 평가 함, copy 하는 것은 0점 처리 함
-> [15주차] 실습 04, 실습 05 문제 보너스 점수 부여

Python Roadmap

