



# 오늘의 학습

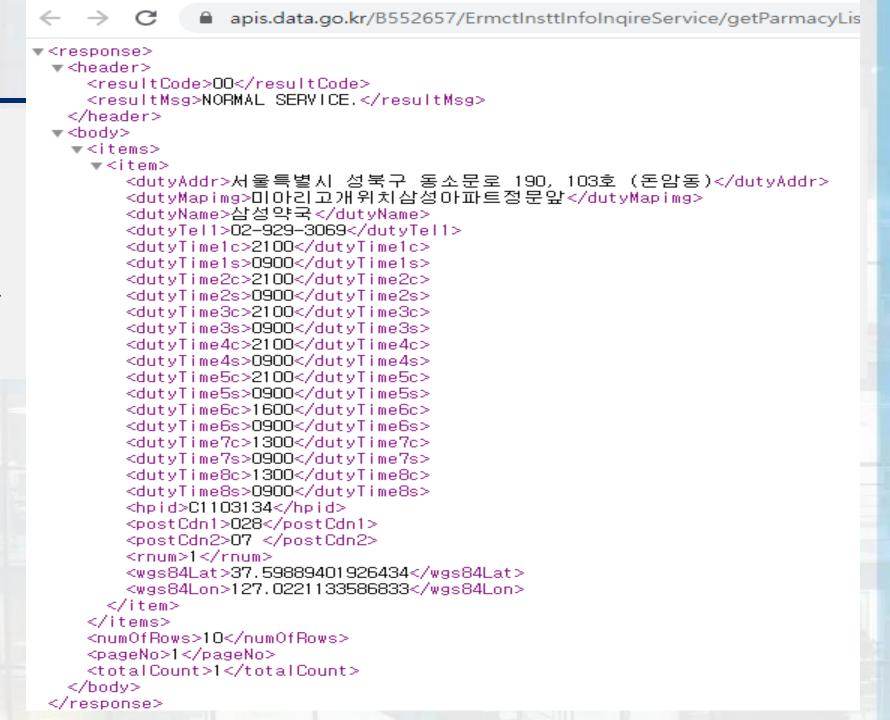
# 학습내용

- XML이해
- 공공 데이터 API기반 크롤링



• XML은 Extensible Markup Language의 약자로써, 데이터 교환에 목적을 두고 만들어진 마크업 언어

마크업 언어 : 태그 등을 이용하여 문서나 데이터의 구조를 나타내는 언어를 뜻한다.





# Requests 라이브러리 복습

```
1 # requests 복습
2 import requests
4 url = 'http://www.google.com'
6 7 response = requests.get(url)
8 9 print("정상응답 코드 =>",response.status_code)
10 response.text
```

정상응답 코드 => 200

'<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="ko"><head><meta content="text/html; charset="UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/googleg\_standard\_color\_128dp.png" itemprop="image"><title>Google</title><script nonce="8eTl4HdspoWJlxS-5drV2A">(function(){window.google={kEl:\delta}'e018YqnkOP-v2roP5uSnKA\delta', kEXPI:\delta'0,1302536,56873,1709,4349,207,2415,2389,2316,383,246,5,1354,4013,1237,1122516,1197713,688,380090,16114,28684,17572,4859,1361,9291,3027,17581,4020,978,13228,3847,4192,6430,14762,7979,5081,1594,1278,2451,291,149,1103,840,6297,108,3406,606,2023,2298,14669,3227,2845,7,4811,12639,16320,1851,15756,3,576,6459,151,13973,4,1528,2304,7039,25073,2658,7357,18095,16786,5827,2530,4094,4052,3,3541,1,42154,2,14022,1931,6332,2523,3330,11623,5679,1020,2381,2718,18279,2,6,7737,4567,6259,3020,333700,16695,1249,5838,14968,1554,2778,19,4220,1850,1395,445,2,2,1,6960,7816,6670,3333,1416,1,436,239,5537,2271,108,256

### Python requests 모듈

GET 요청(request)을 보냈고 서버에서는 그 요청을 받아 처리한 후 요청자에게 응답(response)

정상적인 응답 : 200 응답의 내용 : HTML 코드.

### # requests 기본 사용법

import requests
URL = 'http://www.google.com'
response = requests.get(URL)
response.status\_code
response.text

### GET 요청할 때 parameter 전달법

params = {'param1': 'value1', 'param2': 'value'}
res = requests.get(URL, params=params)



# BeautifulSoup

BeautifulSoup(response.content,"html.parser")

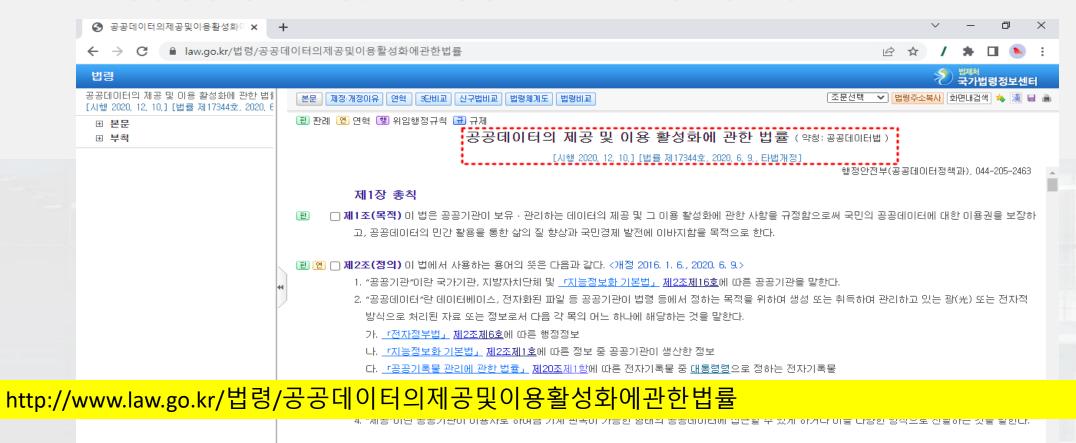
첫번째 인수는 받은텍스트 두번째 인수는 텍스트를 파싱할 파서

Parser는 데이터를 뽑아내는 프로그램
Parser종류에는 "html.parser","lxml","xml"등이 있음



# 공공데이터 포털(Data Portal)

- 공공데이터는 공공기관 업무를 수행하며 만들었거나 관리하는 다양한 형태의 데이터
- 우리나라는 정부3.0의 핵심 정책 중 하나로 공공데이터 정책을 추진
- 공공데이터의 제공 및 이용 활성화에 관한 법률을 제정하고 시행

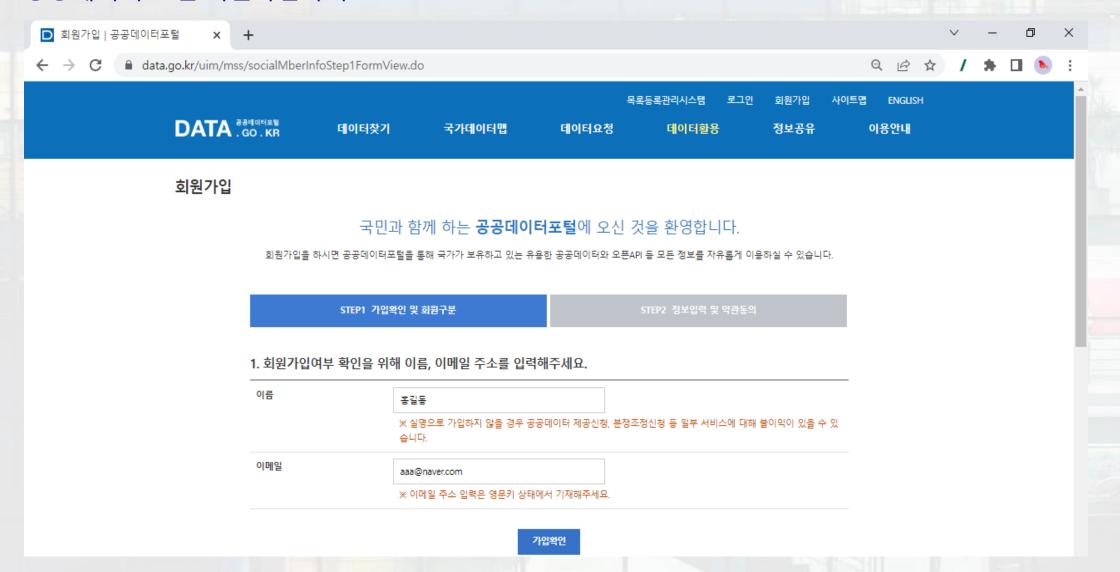


□ **제3조(기본원칙)** ① 공공기관은 누구든지 공공데이터를 편리하게 이용할 수 있도록 노력하여야 하며, 이용권의 보편적 확대를 위하여 필요한 조치를 취하여



# 공공데이터 활용 신청

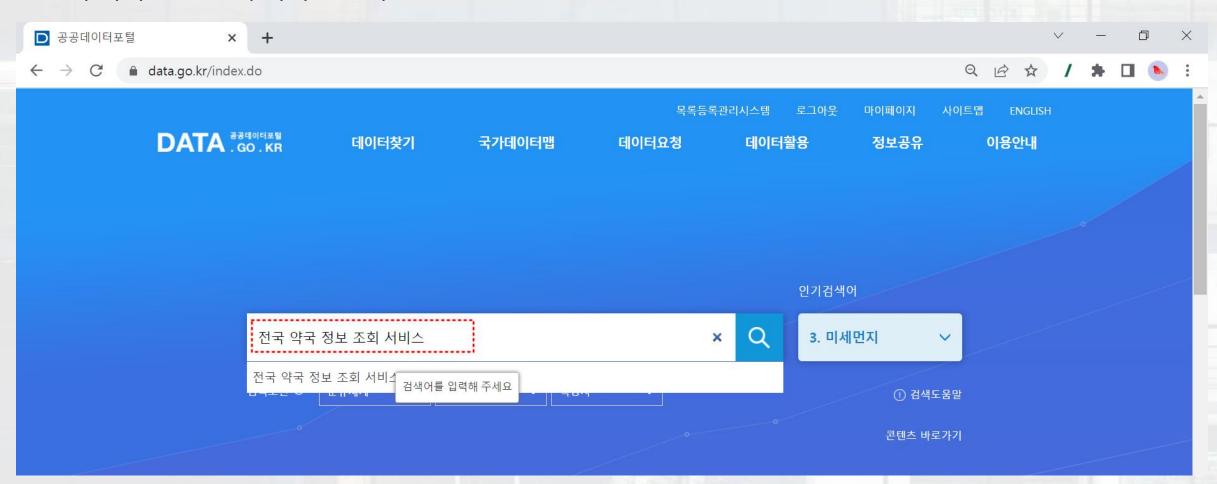
### •1. 공공데이터 포털 회원가입하기





# 공공데이터 활용 신청

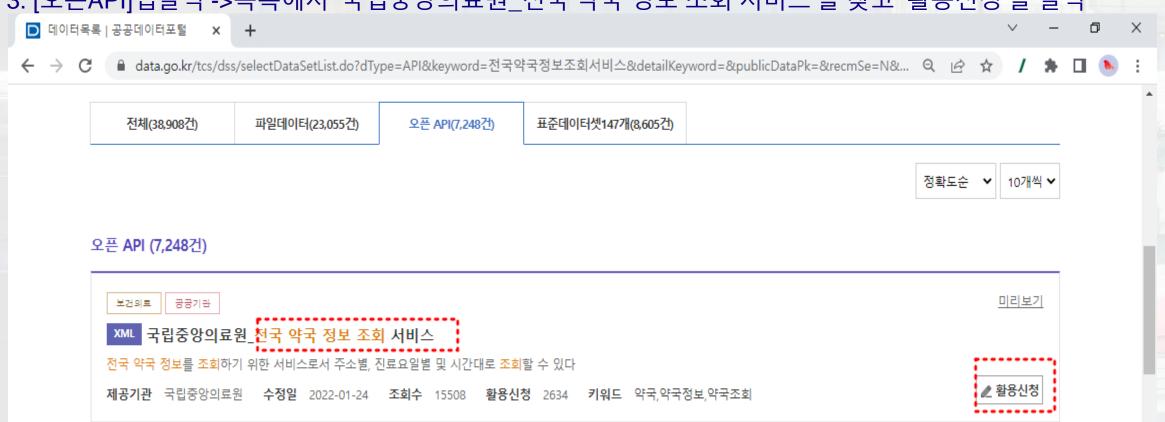
2. '전국 약국 정보 조회 서비스'검색





# 공공데이터 활용 신청

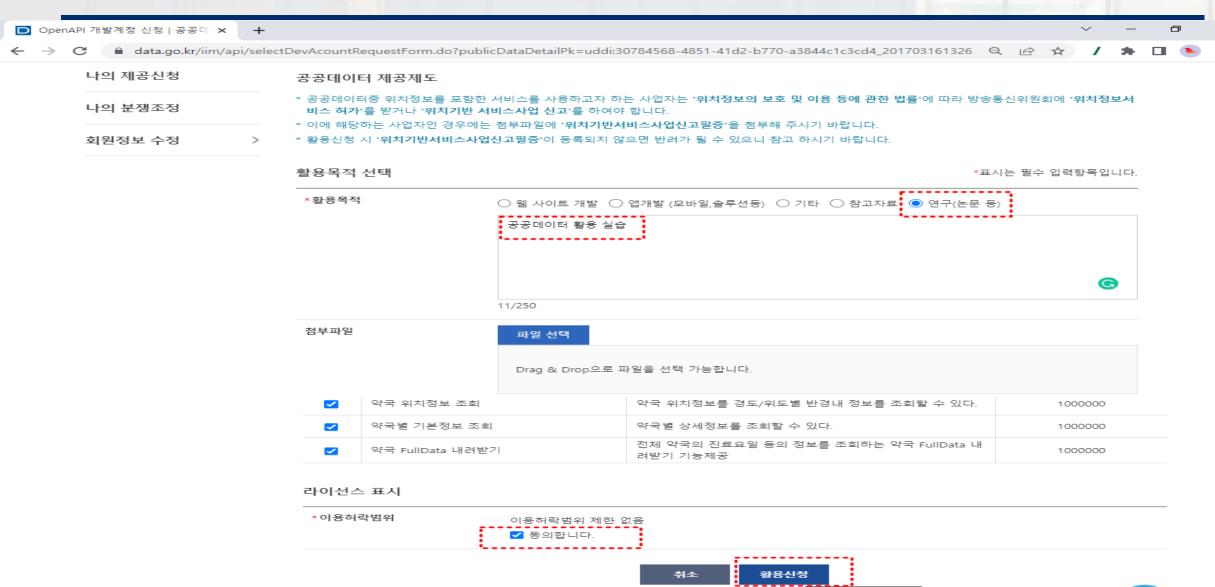
3. [오픈API]탭클릭 ->목록에서 '국립중앙의료원\_전국 약국 정보 조회 서비스'를 찾고 '활용신청'을 클릭





## 4. Open API 개발계정 신청하기 공공데이터 활용 신청 활용목적 입력및 동의->활용신청(다른 공공 데이터 신청 예)

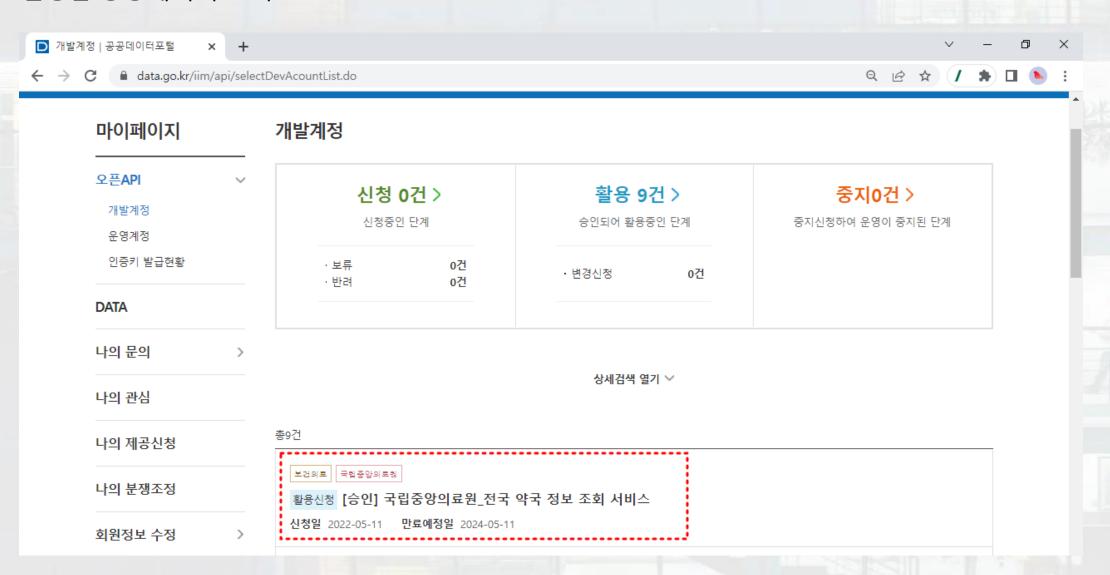
활용신청 바로가기





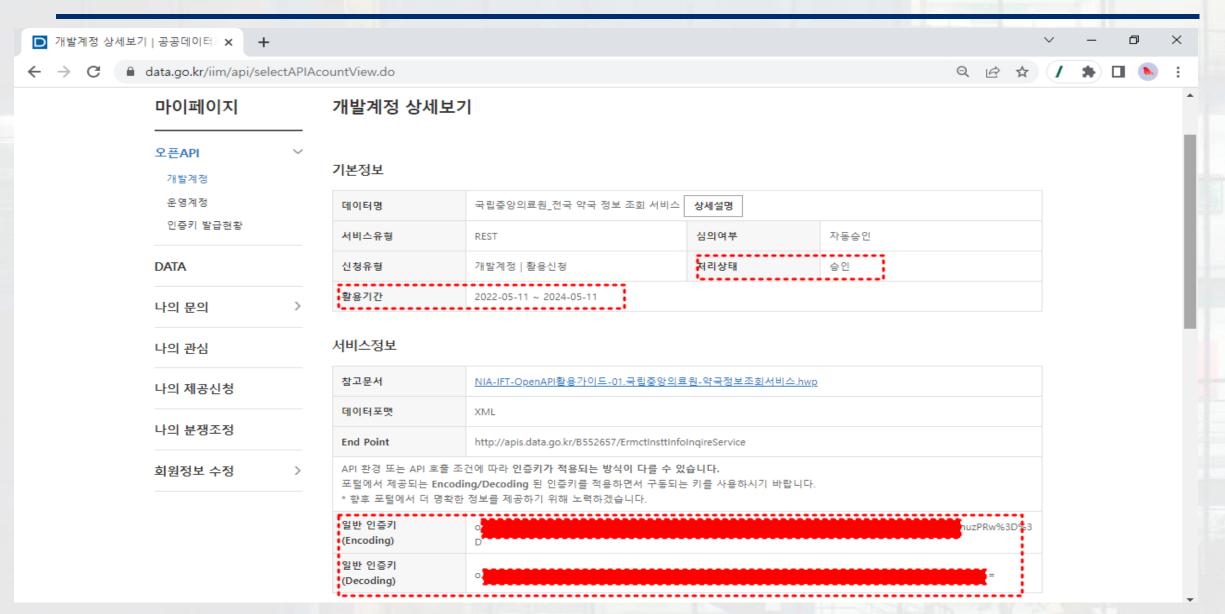
# 활용신청 한 공공데이터 보기

### 신청한 공공데이터 보기





# 개발계정 상세보기





### 요청변수(Request Parameter)

항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
주소(시도)	Q0	16	옵션	서울특별시	주소(시도)
주소(시군구)	Q1	60	옵션	강남구	주소(시군구)
진료요일	QT	1	옵션	1	월~일요일, 공휴일: 1~8
기관명	QN	100	옵션	삼성약국	기관명
순서	ORD	20	옵션	NAME	순서
페이지 번호	pageNo	100	옵션	1	페이지 번호
목록 건수	numOfRows	100	옵션	10	목록 건수



# 활용신청 상세기능정보

#### 활용신청 상세기능정보

NO	상세기능	설명	일일 트래픽	미리보기
1	약국 목록정보 조회	약국 정보를 시도/시군구/진료일자별 조회할 수 있다.	1000000	확인

#### 요청변수(Request Parameter)

닫기

항목명	샘플데이터	설명
Q0	서울특별시	주소(시도)
Q1	성북구	주소(시군구)
QT	1	월~일요일, 공휴일: 1~8
QN	삼성약국	기관명
ORD	NAME	순서
pageNo	1	페이지 번호
numOfRows	10	목록 건수

미리보기









This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<response>
 ▼<header>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL SERVICE.</resultMsg>
   </header>
 ▼<body>
   ▼<items>
     |▼<item>
        <dutyAddr>서울특별시 성북구 동소문로 190, 103호 (돈암동)</dutyAddr>
        <dutyMapimg>미아리고개위치삼성아파트정문앞</dutyMapimg>
        <dutyName>삼성약국</dutyName>
        <dutyTel1>02-929-3069</dutyTel1>
        <dutyTime1c>2100</dutyTime1c>
        <dutyTime1s>0900</dutyTime1s>
        <dutyTime2c>2100</dutyTime2c>
        <dutyTime2s>0900</dutyTime2s>
        <dutyTime3c>2100</dutyTime3c>
        <dutyTime3s>0900</dutyTime3s>
        <dutyTime4c>2100</dutyTime4c>
        <dutyTime4s>0900</dutyTime4s>
        <dutyTime5c>2100</dutyTime5c>
        <dutyTime5s>0900</dutyTime5s>
        <dutyTime6c>1600</dutyTime6c>
        <dutyTime6s>0900</dutyTime6s>
        <dutyTime7c>1300</dutyTime7c>
        <dutyTime7s>0900</dutyTime7s>
        <dutyTime8c>1300</dutyTime8c>
        <dutyTime8s>0900</dutyTime8s>
        <hpid>C1103134</hpid>
        <postCdn1>028</postCdn1>
        <postCdn2>07 </postCdn2>
        <rnum>1</rnum>
        <wgs84Lat>37.59889401926434/wgs84Lat>
        <wgs84Lon>127.0221133586833
      </item>
    </items>
    <numOfRows>10</numOfRows>
    <pageNo>1</pageNo>
```



# 출력결과\_크롤링 결과로 받을 데이터

### 응답 메시지 명세

항목명(영문)	항목명(국문)	항목 크기	항목 구분	샘플데이터	항목설명
resultCode	결과코드	2	1	00	00:성공
resultMag	결과메시지	10	1	N O R M A L SERVICE	
Items			0n		
mum	일련번호	10	1	1	item 번호
dutyAddr	주소	200	1	선 울 특 별 식 알 원 동 5 0 (일원로81)	주소
dutyEtc	비고	200	0	외래진료시간은 진 료 과 목 별 운영	비고
dutyMapimg	간이약도	100	0	-	간이약도
dutyName	기관명	100	1	(사) 삼성생명공 익 재 단 삼성서울병원	기관명
dutyTel1	대표전화1	14	1	02-3410-2114	대표전화1



# 출력결과\_크롤링 결과로 받을 데이터

응답 메시지 명세

항목명(영문)	항목명(국문)	항목 크기	항목 구분	샘플데이터	항목설명
dutyTime1c	진료시간(윌요일)C	4	0	1530	진료시간(윌요일)
dutyTime2c	진료시간(화요일)C	4	0	1530	진료시간(화요일)
dutyTime3c	진료시간(수요일)C	4	0	1530	진료시간(수요일)
dutyTime4c	진료시간(목요일)C	4	0	1530	진료시간(목요일)
dutyTime5c	진료시간(금요일)C	4	0	1530	진료시간(금요일)
dutyTime6c	진료시간(토요일)C	4	0	1530	진료시간(토요일)
dutyTime7c	진료시간(일요일)C	4	0	1530	진료시간(일요일)
dutyTime8c	진료시간(공휴일)C	4	0	1530	진료시간(공휴일)
dutyTime1s	진료시간(윌요일)S	4	0	0830	진료시간(윌요일)
dutyTime2s	진료시간(화요일)S	4	0	0830	진료시간(화요일)
dutyTime3s	진료시간(수요일)S	4	0	0830	진료시간(수요일)
dutyTime4s	진료시간(목요일)S	4	0	0830	진료시간(목요일)
dutyTime5s	진료시간(금요일)S	4	0	0830	진료시간(금요일)
dutyTime6s	진료시간(토요일)S	4	0	0830	진료시간(토요일)
dutyTime7s	진료시간(일요일)S	4	0	0830	진료시간(일요일)
dutyTime8s	진료시간(공휴일)S	4	0	0830	진료시간(공휴일)
hpid	기관ID	10	1	A0000028	기관ID : 동일테이블에 기관명을 포함하 있음
postCdn1	우편번호1	3	1	135	우편번호1
postCdn2	우편번호2	3	1	750	우편번호2
wgs84Lon	병원경도	8	1	127.0851566	병원경도
wgs84Lat 한목구부 · 필수	병원위도	8	1	37.48813256 서 또는 봉수거()	병원위도

※ 항목구분 : 필수(1), 옵션(0), 1건 이상 복수건(1..n), 0건 또는 복수건(0..n)





8

response = requests.get(url, params=params)

|print(response.content)

## Python 샘플코드 복사->JuperNotebook으로 가져오기

url = 'http://apis.data.go.kr/B552657/ErmctInsttInfolngireService/getParmacyListInfolngire'

#### 샘플코드

```
Objective-C
                                                                                                   Python
                                                                                                                   Nodeis
    Java
                  Javascript
                                    C#
                                                    PHP
                                                                   Curl
                                                                                                                                     R
# Python3 샘플 코드 #
import requests
url = 'http://openapi.tour.go.kr/openapi/service/EdrcntTourismStatsService/getEdrcntTourismStatsList'
params ={'serviceKey' : '서비스키', 'YM' : '201201', 'NAT_CD' : '112', 'ED_CD' : 'D' }
response = requests.get(url, params=params)
print(response.content)
  # Pvthon3 샘플 코드 #
  import requests
```

b'<OpenAPI\_ServiceResponse>\m\t<cmmMsgHeader>\m\t\terrMsg>SERVICE ERROR</errMsg>\m\t\t<returnAuthMsg>SERVICE\_KEY\_IS\_NOT\_REG ISTERED\_ERROR</returnAuthMsg>\m\t\terrure

params ={'serviceKey': '서비스키', 'QO': '서울특별시', 'Q1': '강남구', 'QT': '1', 'QN': '삼성약국', 'ORD': 'NAME',

```
# Python3 샘플 코드#
```

import requests

```
url = 'http://apis.data.go.kr/B552657/ErmctInsttInfoInqireService/getParmacyListInfoInqire'
params ={'serviceKey' : '서비스키', 'Q0' : '서울특별시', 'Q1' : '강남구', 'QT' : '1', 'QN' : '삼성약국', 'ORD' : 'NAME', 'pageNo' : '1', 'numOfRows' : '10' }
```

response = requests.get(url, params=params)
print(response.content)

공공데이터는 <item></item>항목 안에 데이터들이 태그로 저장되어 있습니다. 가장먼저 item들을 다 찾아줍니다.

items = soup.find\_all('item')을 통해 찾은 item들을 저장해줍니다.

#### [실습 1-1] 발급받은 인증키 넣기

```
import requests

url = 'http://apis.data.go.kr/B552657/ErmctInsttInfoInqireService/getParmacyListInfoInqire'

# 발급받은 의증가 넣기
serviceKey = 'o

params ={'serviceKey' : serviceKey, 'QO' : '서울특별시', 'Q1' : '성북구', 'QT' : '1', 'QN' : '서울약국', 'ORD' : 'NAME', response = requests.get(url,params=params)
print(response.content)
```

b'<?xmI version="1.0" encoding="UTF-8" standalone="yes"?><response><header><resultCode>OO</resultMcode><resultMsg>NORMAL SERV ICE.</resultMsg></header><bddy><items><item><dutyAddr>#xec#x84#x9c#xec#x9a#xb8#xed#x8a#xb9#xeb#xb3#x84#xec#x8b#x9c #xec#x84 #xb1#xeb#xb6#x81#xea#xb5#xac #xeb#x8f#x99#xec#x86#x8c#xeb#xac#xb8#xeb#xa1#x9c 181, 1#xec#xb8#xb5 (#xeb#x8f#x88#xec#x95#x94#xeb#x8f#x99, #xec#x84#xb1#xeb#xb6#x8f#x88#xec#x84#xb1#xec#x8b#xac#xeb#xb9#x8c#xeb#x99
181, 1#xec#xb8#xb9#xb6#xb6#x86#x86#x88#xec#x85#x95#x94#xeb#x9c#x95#x94#xeb#x9c#x9c#x95#x94#xeb#x9c#x95#x94#xeb#x96#x86#xac#xeb#x96#xac#xeb#xb6#xac#xeb#xb9#xac#xeb#xb9#xac#xeb#xb9#xac#xeb#xb9#xac#xeb#xb9#x901#xec#xb8#xec#x84#xb1#xec#x84#xb1#xec#x84#xb1#xec#x84#xb1#xec#x84#xb1#xec#x84#xb1#xec#x84#xb9#xac#xeb#xb9#xac#xeb#xb9#xac#xeb#xb9#xac#xeb#xb9#xac#xeb#xb9#xad
\dutyAddr><dutyMapimg>\dutyMapimg>\dutyMapimg>\dutyName>\dutyTime2\solonooon=\dutyTime2\solonooon=\dutyTime2\solonooon=\dutyTime2\solonooon=\dutyTime2\solonooon=\dutyTime2\solonooon=\dutyTime2\solonooon=\dutyTime2\solonooon=\dutyTime2\solonooon=\dutyTime2\solonooon=\dutyTime3\solonooon=\dutyTime3\solonooon=\dutyTime3\solonooon=\dutyTime4\solonooon=\dutyTime4\solonooon=\dutyTime4\solonooon=\dutyTime6\solo



[요청예>

요청 예제

#### REST(URI)

http://apis.data.go.kr/B552657/ErmctInsttInfoIngireService/getParmacyListInfoIngire?Q0=%EC%84%9 C%EC%9A%B8%ED%8A%B9%EB%B3%84%EC%8B%9C&Q1=%EC%A2%85%EB%A1%9C%EA%B5%AC& ORD=ADDR&pageNo=1&numOfRows=1

# [실습 1-2] decode('utf-8')

<?xml version="1.0" encoding="UTF-8" standalone="yes"?><response><header><resultCode>00</resultCode><resultMsg>NORMAL SERVICE.</resultMsg></header><body><items><item><dutyAddr>서울특별시 성북구 동소문로 181, 1층
(돈암동, 성북성심빌딩)</dutyAddr><dutyMapimg>미아리고개성북성심병원1층</dutyMapimg><dutyName>서울약국</dut
yName><dutyTel1>02-924-5191</dutyTel1><dutyTime1c>1900</dutyTime1c><dutyTime1s>0900</dutyTime1s><dutyTime2</pre>

#### [실습 1-3] 성북구의 삼성약국을 검색하는 코드 작성

```
import urllib.request
 2 import bs4
   endpoint = 'http://apis.data.go.kr/B552657/ErmctInsttInfolngireService/getParmacyListInfolngire?'
   serviceKev = 'o.
 6
   QD = urllib.parse.quote("서울특별시")
 8 Q1 = urllib.parse.quote("설북구")
10 QN = urllib.parse.guote("삼성약국")
   ORD = "NAME"
12 | pageNo = "1"
13 | startPage = "1"
14 | numofRows = "10"
   pageSize = "10"
16
   params ="serviceKey=" + serviceKey + #
18
             "&OO=" + OO + #
19
             "&Q1=" + Q1 + #
20
21
             "&QN=" + QN + #
22
             "&ORD=" + ORD + #
23
            "&pageNo=" + pageNo + #
24
            "&startPage=" + startPage + #
25
            "&numofRows=" + numofRows + #
26
             "&pageSize=" + pageSize
27
28 url = endpoint + params
29 | print("url:"+ url)
   result = requests.get(url)
31 bs_obj = bs4.BeautifulSoup(result.content, "html.parser")
   print(bs_obj)
33
   items = bs_obj.findAll("item")
35
   for item in items:
37
       tagged_item = item.find("dutyname")
38
       print(tagged_item.text)
```

20

url:http://apis.data.go.kr/B552657/ErmctInsttInfoInqireService/getParmacyListInfoInqire?serviceKey=oJG8Ljp5EXdpKTX7w0MzhaocaELQLRTU67PV3af
XoNjTFnuCeb4rmkgVNCB0NArLx4vD38BfcIMzSgN/nuzPRw==&Q0=%EC%84%9C%EC%9A%B8%ED%8A%B9%EB%B3%84%EC%8B%9C&Q1=%EC%84%B1%EB%B6%81%EA%B5%AC&QT=2&QN
=%EC%82%BC%EC%84%B1%EC%95%BD%EA%B5%AD&ORD=NAME&pageNo=1&startPage=1&numofRows=10&pageNize=10

</ml version="1.0" encoding="UTF-8" standalone="yes"?><response><header><resultcode>OC/resultcode><resultmsg>NORMAL SERVICE.</resultmsg>
</header><body><item><dutyaddr>서울특별시 성북구 동소문로 190, 103호 (돈암동)</dutyaddr><dutymapimg>□lo라고개위치삼성이파트정문앞
</dutymapimg><dutyname>삼성약국</dutyname><dutytel1>O2-929-3069</dutytime1c>2100</dutytime1c>20100</dutytime1c>20utytime1s>0900</dutytime1s>0900</dutytime1s>0900</dutytime2s><dutytime3c>2100</dutytime3c>2000</dutytime3s>0900</dutytime3s>0900</dutytime3s>0900</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000</dutytime3c>2000<

### ■ [실습 1-4] input()함수를 사용하여 서울에 있는 모든 약국보기

```
import urllib.request
   import bs4
   endpoint = 'http://apis.data.go.kr/B552657/ErmctInsttInfoIngireService/getParmacyListInfoIngire?'
   serviceKey = 'oJ
   QD = urllib.parse.quote(input("시 입력=>"))
   ORD = "NAME"
                          # 정렬순서
   pageNo = "10"
10 | startPage = "1"
11 numofRows = "10"
                       #목록건수
   pageSize = "10"
13
   params = "serviceKey=" + serviceKey + #
15
            "&QO=" + QO + #
            "&ORD=" + ORD + #
16
17
            "&pageNo=" + pageNo + #
18
            "&startPage=" + startPage + #
            "&numofRows=" + numofRows + #
19
20
            "&pageSize=" + pageSize
22 | url = endpoint + params
   result = requests.get(url)
   bs_obj = bs4.BeautifulSoup(result.content,"html.parser")
   print(bs_obj)
   items = bs_obj.findAll("item")
   for item in items:
31
       tagged_item = item.find("dutyname")
       print(tagged_item.text)
```

# ■ [실습 1-5] input()함수를 시/구를 입력하여 약국보기

#### [실습 1-5] input()함수를 시/구를 입력하여 약국보기

print(tagged item.text)

```
import urllib.request
 2 import bs4
   endpoint = 'http://apis.data.go.kr/B552657/ErmctInsttInfoIngireService/getParmacyListInfoIngire?'
 5 serviceKey =
 7¦QO = urllib.parse.quote(input("시 입력=>"))
 8! Q1 = urllib.parse.quote(input("구 입력=>"))
10 \mid ORD = "NAME"
                        # 정렬순서
   pageNo = "1"
12 | startPage = "1"
13 | numofRows = "10"
                       #목록건수
                                                                 시 입력=>서울특별시 .
14 | pageSize = "10"
   params = "serviceKev=" + serviceKev + #
           "&OO=" + OO + #
18
           "&O1=" + O1 + \#
                                                                21세기고려약국
19
           "&ORD=" + ORD + #
                                                                365약국
20
           "&pageNo=" + pageNo + #
                                                                ₩스토어참좋은약국
21
           "&startPage=" + startPage + #
22
           "&numofRows=" + numofRows + #
                                                                가까운약국
23
           "&pageSize=" + pageSize
                                                                 가까운행복약국
24
25 url = endpoint + params
                                                                가정약국
26 | result = requests.get(url)
                                                                강원약국
   bs_obj = bs4.BeautifulSoup(result.content,"html.parser")
28
                                                                 개미약국
   items = bs_obj.findAll("item")
                                                                건강약국
   print("="*50)
                                                                건강한약국
31
   for item in items:
      tagged_item = item.find("dutyname")
```

### <u>■ [실습 1-6] 평일(월요일) 오후 8시 이후까지 영업하는 약국 수 세어보기</u>

[실습 1-6] 평일(월요일) 오후 8시 이후까지 영업하는 약국 수 세어보기

37 print("월요일 8시 이후까지 영업하는 약국수:" + str(count))

36

```
import urllib.request
2 import bs4
   endpoint = 'http://apis.data.go.kr/B552657/ErmctInsttInfoInqireService/getParmacyListInfoInqire?'
 5 serviceKev = '
   QD = urllib.parse.quote(input("시 입력=>"))
 8 ORD = "NAME"
                          # 정렬순서
9 | pageNo = "1"
10 | startPage = "1"
  numofRows = "5003"
   pageSize = "10"
   | params = "serviceKey=" + serviceKey + #
15
            "&OO=" + OO + #
16
            "&ORD=" + ORD + #
17
            "&pageNo=" + pageNo + #
18
            "&startPage=" + startPage + #
19
            "&numofRows=" + numofRows + #
20
            "&pageSize=" + pageSize
21
22 url = endpoint + params
23 result = requests.get(url)
   bs_obj = bs4.BeautifulSoup(result.content,"html.parser")
   |items = bs_obj.findAll("item")
26
   count = 0 # 초깃값 0
28
   for item in items:
30
        tagged_item = item.find("dutytime1c")
31
32
       if (tagged_item != None):
33
           close_time = int(tagged_item.text)
34
           if(close_time > 2000):
35
              count += 1
```

시 입력=>서울특별시 월요일 8시 이후까지 영업하는 약국수:2



[참고] &job=programmer

### 설명(Description)

웹을 통해서 데이터를 전송할 때 특정 문자들은 특수한 기능으로 사용된다. 예를들어, http://a.com?name=egoing&job=programmer 에서 &job=programmer 중 '&'는 하나의 파라미터가 끝나고 다음 파라미터가 온다는 의미