

# 웹에서 가져오기

# 웹에서 가져오기

## ❖ 웹에서 가져오기

- ✓ 기본적으로 제공하는 urllib, urllib2 패키지를 이용해서 웹 문서를 가져올 수 있음
- ✓ 이 패키지에는 4개의 모듈이 존재하는데 웹에서 데이터를 읽어오는 모듈은 request
- ✓ request 모듈의 urlopen이라는 함수에 url을 문자열로 대입하면 url과 연결한 객체(response)를 리턴하는데 이 객체를 이용해서 getheaders()를 호출하면 서버의 정보를 읽을 수 있고 status 속성을 이용하면 서버의 상태 정보를 읽을 수 있으며 read()를 호출해서 내용을 읽을 수 있음
- ✓ 읽어온 내용에 한글이 포함된 경우 인코딩을 해야 하는 경우도 있는데 인코딩정보=response.info().get\_content\_charset() 을 호출해서 서버의 인코딩을 확인한 후 read()로 읽어온 내용.decode(인코딩정보) 를 호출해서 디코딩 된 결과를 리턴받으면 됨

# 웹에서 가져오기

## ❖ 웹에 요청하고 응답 받기

```
import urllib.request
```

```
#데이터 읽기
```

```
response = urllib.request.urlopen("https://www.kakao.com/")
```

```
data = response.read()
```

```
#읽어온 데이터의 인코딩 설정
```

```
encoding = response.info().get_content_charset()
```

```
html = data.decode(encoding)
```

```
#화면 출력
```

```
print(html)
```

# 웹에서 가져오기

## ❖ URL 인코딩

- ✓ URL에 한글이 포함된 경우는 인코딩이란 과정을 거쳐서 데이터를 요청해야 하는데 인코딩을 해주는 모듈이 `urllib.parse` 모듈
- ✓ 인코딩 해주는 함수 `quote_plus` 함수와 `quote`라는 함수가 있는데 이 함수와 `quote_plus` 함수의 차이점은 `quote_plus`는 공백을 `+` 기호로 처리하고 `quote` 함수는 공백을 `%20`으로 인코딩

# 웹에서 가져오기

## ❖ 데이터 읽기 -한겨레 신문사에서 국민카드로 검색한 결과 가져오기

#URL에 한글이 포함된 경우

```
import urllib.request
```

```
response =
```

```
urllib.request.urlopen("https://search.hani.co.kr/search?command=query&media=common  
&searchword=사이버가수아담&x=0&y=0&_ga=2.225940573.611667888.1689238630-  
1387075702.1689238630")
```

```
data = response.read()
```

#읽어온 데이터의 인코딩 설정

```
encoding = response.info().get_content_charset()
```

```
html = data.decode(encoding)
```

#화면 출력

```
print(html)
```

UnicodeEncodeError: 'ascii' codec can't encode characters in position 34-37: ordinal not in range(128)

# 웹에서 가져오기

## ❖ 데이터 읽기 -한겨레 신문사에서 국민카드로 검색한 결과 가져오기

#URL에 한글이 포함된 경우

```
import urllib.request
from urllib.parse import quote
```

```
keyword = quote("사이버가수아담")
```

```
response =
```

```
urllib.request.urlopen("https://search.hani.co.kr/search?command=query&media=common  
&searchword=" + keyword + "&x=0&y=0&_ga=2.225940573.611667888.1689238630-  
1387075702.1689238630")
```

```
data = response.read()
```

#읽어온 데이터의 인코딩 설정

```
encoding = response.info().get_content_charset()
```

```
html = data.decode(encoding)
```

#화면 출력

```
print(html)
```

# 웹에서 가져오기

## ❖ requests

- ✓ HTTP 라이브러리로 많이 쓰이고 있는 requests 를 설치해 웹에서 데이터를 가져올 수 있음
- ✓ 설치 - pip install requests
- ✓ requests는 HTTP GET, POST, PUT, DELETE 등을 사용할 수 있으며 편리한 데이터 인코딩 기능을 제공
- ✓ 파라미터로 넘겨줄 데이터를 dict로 만들어 GET, POST 등에서 사용하면 필요한 request 인코딩을 자동으로 처리
- ✓ requests.get(url) 함수를 사용하면 해당 웹 페이지 호출 결과를 가진 Response 객체를 리턴
- ✓ get이나 post 함수를 호출할 때 옵션으로 header를 이용하면 header를 포함해서 전송이 가능한데 이 때 데이터는 dict 형태로 생성해서 대입
- ✓ Response 객체는 여러 attribute들을 가지고 있는데 Response의 status\_code 속성을 확인해서 HTTP Status 결과를 체크할 수 있고 Response 에서 가져온 데이터를 문자열로 저장하고 있는 text 속성이 있으며 Response 데이터를 바이트(bytes)로 저장하고 있는 content 속성 등이 있음
- ✓ Response에서 에러가 있어서 중단하도록 할 때는 Response 객체의 raise\_for\_status() 함수를 호출

# 웹에서 가져오기

## ❖ 웹에 요청하고 응답 받기

```
import requests
# GET
resp = requests.get('http://httpbin.org/get')
print(resp.text)

# POST
dic = {"id": "ggangpae1", "name": "박문석", "age": 40}
resp = requests.post('http://httpbin.org/post', data=dic)
print(resp.text)

resp = requests.put('http://httpbin.org/put')
resp = requests.delete('http://httpbin.org/delete')
```

## ✓ 경고가 발생하는 경우 아래 구문 수행

- pip uninstall urllib3
- pip install 'urllib3<2.0'



# 웹에서 가져오기

## ❖ 웹에서 문자열 읽어오기

```
import requests
```

```
res = requests.get( 'https://www.python.org/' )  
# resp.raise_for_status()
```

```
if (res.status_code == requests.codes.ok):  
    html = res.text  
    print(html)
```

# 인코딩

## ❖ 인코딩

- ✓ requests 에서 웹 호출을 진행한 후 결과는 Response 객체에 담기게 되는데 Response의 text 속성은 str 클래스 타입으로서 보통 requests 모듈에서 자동으로 데이터를 인코딩을 함
- ✓ requests는 HTTP 헤더를 통해 데이터의 인코딩 방식을 추측하여 Response 객체의 encoding 속성에 그 값을 지정하고 text 속성을 액세스 할 때 이 encoding 속성을 사용
- ✓ 인코딩 방식을 변경해야 한다면 text 속성을 읽기 전에 Response의 encoding 속성을 변경
- ✓ 인코딩이 유니코드 인코딩(예: UTF-8 등)이면 일반적으로 한글이 제대로 표현되지만 ISO-8859-1나 EUC-KR 같은 인코딩이면 한글이 제대로 표현되지 않을 수 있는데 해결하는 방법은 미리 Response 객체의 encoding을 한글인코딩(예: EUC-KR)이나 None(인코딩 추측을 하지 않도록) 으로 지정한 후 text 속성을 읽으면 가능

# 인코딩

## ❖ 웹에서 인코딩 설정해서 문자열 읽기

```
import requests
try:
    res = requests.get('https://www.naver.com') # 네이버 홈
    print(res.encoding)
    res = requests.get('http://www.tjoeun.co.kr/') # 더조은 홈페이지
    print(res.encoding)
except Exception as e:
    print("예외발생:", e)
```

# 파일 다운로드

## ❖ 웹에서 파일을 다운로드 받아서 저장하기

```
import requests
```

```
imgurl = 'http://www.onlifezone.com/files/attach/images/962811/376/321/005/2.jpg'
```

```
imgname = 'data/' + imgurl.split('/')[-1]
```

```
try:
```

```
    res = requests.get(imgurl)
```

```
    with open(imgname, 'wb') as h:
```

```
        img = res.content
```

```
        h.write(img)
```

```
except Exception as e:
```

```
    print(e)
```

# JSON

## ❖ JSON

- ✓ 속성-값 쌍으로 이루어진 데이터 오브젝트를 전달하기 위해 텍스트를 사용하는 개방형 표준 포맷
- ✓ 인터넷에서 자료를 주고 받을 때 그 자료를 표현하는 방법 중의 하나
- ✓ 자바스크립트 언어로부터 파생되어 자바스크립트의 구문 형식을 따르지만 언어 독립형 데이터 포맷
- ✓ 프로그래밍 언어나 플랫폼에 독립적이므로, 구문 분석 및 JSON 데이터 생성을 위한 코드는 C, C++, C#, 자바, 자바스크립트, 펄, 파이썬 등 수많은 프로그래밍 언어에서 쉽게 이용
- ✓ 파이썬에서는 `json.loads(문자열)`을 이용하면 파싱된 결과를 리턴
- ✓ 데이터 표현 방법

배열 : [ ]

객체: {속성:값, 속성:값...}

# JSON

## ❖ 카카오의 카테고리 검색 API 결과

```
{'documents': [{ 'address_name': '서울 강남구 삼성1동 155-11', 'category_group_code':  
'PM9', 'category_group_name': '약국', 'category_name': '의료,건강 > 약국', 'distance': '',  
'id': '16059439', 'phone': '02-555-5505', 'place_name': '삼성수약국', 'place_url':  
'http://place.map.kakao.com/16059439', 'road_address_name': '서울 강남구 삼성로96길  
11', 'x': '127.056836574631', 'y': '37.5089448116358'}, { 'address_name': '서울 강남구 대  
치동 943-16', 'category_group_code': 'PM9', 'category_group_name': '약국',  
'category_name': '의료,건강 > 약국', 'distance': '', 'id': '16618597', 'phone': '02-558-5476',  
'place_name': '장생당약국', 'place_url': 'http://place.map.kakao.com/16618597',  
'road_address_name': '서울 강남구 테헤란로84길 17', 'x': '127.058969686607', 'y':  
'37.5060533424248'}, { 'address_name': '서울 강남구 삼성동 158-10',  
'category_group_code': 'PM9', 'category_group_name': '약국', 'category_name': '의료,건  
강 > 약국', 'distance': '', 'id': '13508696', 'phone': '02-508-1363', 'place_name': '참좋은온  
누리약국', 'place_url': 'http://place.map.kakao.com/13508696', 'road_address_name': '서  
울 강남구 테헤란로87길 17', 'x': '127.05852732111232', 'y': '37.508759280055614'},  
{ 'address_name': '서울 강남구 삼성동 159', 'category_group_code': 'PM9',  
'category_group_name': '약국', 'category_name': '의료,건강 > 약국', 'distance': '', 'id':  
'22547729', 'phone': '02-554-4906', 'place_name': '도심공항약국',
```

# JSON

## ❖ 카카오의 카테고리 검색 API 결과

|          |                    |
|----------|--------------------|
| 삼성수약국    | 서울 강남구 삼성1동 155-11 |
| 장생당약국    | 서울 강남구 대치동 943-16  |
| 참좋은온누리약국 | 서울 강남구 삼성동 158-10  |
| 도심공항약국   | 서울 강남구 삼성동 159     |
| 미소약국     | 서울 강남구 대치2동 944    |
| 이도건강약국   | 서울 강남구 삼성동 154-12  |
| 소망약국     | 서울 강남구 삼성동 158-12  |
| 해성약국     | 서울 강남구 대치2동 942    |
| 팜엑스약국    | 서울 강남구 삼성동 159     |
| 삼성하나약국   | 서울 강남구 삼성동 158-16  |
| 사랑온누리약국  | 서울 강남구 삼성동 158-14  |

# JSON

## ❖ 카카오의 카테고리 검색 API 결과

|          |                    |
|----------|--------------------|
| 삼성수약국    | 서울 강남구 삼성1동 155-11 |
| 장생당약국    | 서울 강남구 대치동 943-16  |
| 참좋은온누리약국 | 서울 강남구 삼성동 158-10  |
| 도심공항약국   | 서울 강남구 삼성동 159     |
| 미소약국     | 서울 강남구 대치2동 944    |
| 이도건강약국   | 서울 강남구 삼성동 154-12  |
| 소망약국     | 서울 강남구 삼성동 158-12  |
| 해성약국     | 서울 강남구 대치2동 942    |
| 팜엑스약국    | 서울 강남구 삼성동 159     |
| 삼성하나약국   | 서울 강남구 삼성동 158-16  |
| 사랑온누리약국  | 서울 강남구 삼성동 158-14  |



# JSON

## ❖ 카카오의 카테고리 검색 API 결과

```
import requests
import json
url =
    'https://dapi.kakao.com/v2/local/search/category.json?category_group_code=PM9&re
    ct=126.95,37.55,127.0,37.60'
headers = {'Authorization': 'KakaoAK {}'.format('c454c0e64688ce2bde2dfff9cceced87')}
data = requests.post(url, headers=headers)
result = json.loads(data.text) #파싱한 결과
#print(result)
documents = result['documents']
for data in documents:
    print(format(data['place_name'], '30s'), end='Wt')
    print(format(data['address_name'], '30s'))
```

# HTML 파싱

## ❖ HTML 파싱: BeautifulSoup 패키지

- ✓ 웹 페이지 HTML 파싱 문서를 파싱(Parsing)하기 위해서 BeautifulSoup 라는 모듈을 사용
- ✓ 설치(Anaconda에서는 설치할 필요가 없음)  
`pip install beautifulsoup4`

# HTML 파싱

## ❖ HTML 파싱: BeautifulSoup

- ✓ HTML 파싱을 위한 라이브러리로 데이터를 쉽게 탐색할 수 있도록 파이썬의 XML 객체로 변환
- ✓ BeautifulSoup를 사용하기 위해서는 먼저 BeautifulSoup 모듈을 import하여야 하는데 모듈명은 bs4
- ✓ bs4 모듈이 import 된 후, bs4.BeautifulSoup(HTML 문서, “html.parser”)를 호출하여 BeautifulSoup 객체를 생성

```
import bs4
```

```
html텍스트 = “<html><body>...생략...</body></html>”
```

```
변수명 = bs4.BeautifulSoup(html텍스트, 'html.parser')
```

- ✓ prettify() 함수를 호출하면 저장하고 있는 내용을 들여쓰기를 적용해서 문자열로 리턴
- ✓ 변수명.태그 이름으로 태그에 접근한 후 get\_text()를 호출하면 원하는 태그 내의 데이터만 추출

# HTML 파싱

## ❖ HTML Parsing

```
import requests, bs4
try:
    resp = requests.get('http://finance.daum.net/')
    resp.raise_for_status()

    html = resp.text
    bs = bs4.BeautifulSoup(html, 'html.parser')

except Exception as e:
    print("예외 발생:", e)

else:
    print(bs.body.span.get_text())
```

금융

# HTML 파싱

## ❖ find 함수

- ✓ find(tag, attributes, recursive, text, limit, keywords)
- ✓ HTML 코드 안에서 원하는 항목 1개 가져올 수 있음  
bs.find('title'): title 태그를 가져오는 코드
- ✓ HTML 코드 안에서 어떤 특정 태그를 가져오고 싶다면 위와 같이 find 함수의 인수에 태그의 이름을 전달
- ✓ 여러 개의 태그를 찾을 때는 {태그 나열}
- ✓ bs.find('p', align="center"): 태그 중에서 align 속성의 값이 center 인 태그를 찾음
- ✓ 여러 개의 속성을 적용할 때는 dict 로 설정 : {"class":{"green", "red"}, "align":"center"}
- ✓ text는 태그에 둘러싸인 내용을 찾을 때 사용
- ✓ recursive의 값이 False 라면 최상위 태그에서만 찾고 True 라면 재귀적으로 탐색해서 찾음
- ✓ keywords 는 별도의 속성을 제공해서 찾고자 할 때 사용

# HTML 파싱

## ❖ find\_all 함수

- ✓ find\_all() 함수는 find() 함수와는 다르게 원하는 태그가 몇 개가 있던 한꺼번에 가져옴  
bs.find\_all('p')
- ✓ find\_all() 함수를 사용해서 HTML 코드 안에 있는 모든 p 태그를 가져오는 문장
- ✓ 앞에서 봤던 find() 함수는 찾는 태그가 여러 개가 있어도 첫 번째 발견되는 1개 태그만 가져오지만 find\_all() 함수는 여러 개를 한꺼번에 다 가져올 수 있어서 편리
- ✓ limit 옵션을 이용해서 가져올 수 있는 개수 설정 가능

# HTML 파싱

## ❖ 데이터 탐색 속성

- ✓ 자손을 탐색할 때는 .을 입력하고 태그 이름 입력
- ✓ children 속성을 이용하면 바로 아래 자식 태그들만 리턴
- ✓ descendants를 호출하면 모든 자손 태그들을 반환
- ✓ next\_siblings를 호출하면 동일한 레벨의 태그들 중에서 뒤에 있는 태그들을 반환하고 previous\_siblings 를 호출하면 이전에 있는 태그들을 반환
- ✓ next\_sibling, previous\_sibling 는 1개만 반환
- ✓ parent 와 parents는 부모 태그를 반환
- ✓ attrs는 속성을 반환하고 attrs['속성이름']을 이용하면 속성의 값만 반환

# HTML 파싱

- ❖ 위키피디아에서 기계\_학습으로 검색한 결과의 링크만 가져와서 출력하기

```
/wiki/%ED%86%A0%EB%A1%A0:%EA%B8%B0%EA%B3%84_%ED%95%99%EC%8A%B5
/wiki/%EC%9C%84%ED%82%A4%EB%B0%B1%EA%B3%BC:%EC%B6%9C%EC%B2%98_%EB%B0%9D%ED%9E%88%EA%B8%B0
/wiki/%EC%9C%84%ED%82%A4%EB%B0%B1%EA%B3%BC:%EC%8B%A0%EB%A2%B0%ED%95%A0_%EC%88%98_%EC%9E%88%EB%8A%94_%EC%B6%9C%EC%B2%9
8
/wiki/%EC%9C%84%ED%82%A4%EB%B0%B1%EA%B3%BC:%ED%99%95%EC%9D%B8_%EA%B0%80%EB%8A%A5
/wiki/%ED%86%A0%EB%A1%A0:%EA%B8%B0%EA%B3%84_%ED%95%99%EC%8A%B5
/wiki/%EB%8D%B0%EC%9D%B4%ED%84%B0_%EB%A7%88%EC%9D%B4%EB%8B%9D
/wiki/%ED%8C%8C%EC%9D%BC:Kernel_Machine.svg
/wiki/%ED%81%B4%EB%9F%AC%EC%8A%A4%ED%84%B0_%EB%B6%84%EC%84%9D
/wiki/%ED%9A%8C%EA%B7%80_%EB%B6%84%EC%84%9D
/wiki/%EA%B0%95%ED%99%94_%ED%95%99%EC%8A%B5
/wiki/%EC%A4%80_%EC%A7%80%EB%8F%84_%ED%95%99%EC%8A%B5
/wiki/%EB%B9%84_%EC%A7%80%EB%8F%84_%ED%95%99%EC%8A%B5
/wiki/%EC%A7%80%EB%8F%84_%ED%95%99%EC%8A%B5
/wiki/%ED%9A%8C%EA%B7%80_%EB%B6%84%EC%84%9D
/wiki/%EA%B2%B0%EC%A0%95_%ED%8A%B8%EB%A6%AC_%ED%95%99%EC%8A%B5%EB%B2%95
/.../...
```



# HTML 파싱

- ❖ 위키피디아에서 기계\_학습으로 검색한 결과의 링크만 가져와서 출력하기

```
import requests, bs4
from urllib.parse import quote
import re

try:
    # wikipedia 에서 기계_학습 검색 결과 가져오기
    keyword = '기계_학습'
    keyword = quote(keyword)
    resp = requests.get('https://ko.wikipedia.org/wiki/' + keyword)
    resp.raise_for_status()

    html = resp.text
    # 텍스트를 DOM으로 만들기
    bs = bs4.BeautifulSoup(html, 'html.parser')
except Exception as e:
    print("예외 발생1:", e)
```

# HTML 파싱

- ❖ 위키피디아에서 기계\_학습으로 검색한 결과의 링크만 가져와서 출력하기

else:

#a 태그 전부 가져오기

for link in bs.findAll('a'):

#속성에 href 가 있는 경우

if 'href' in link.attrs:

#href 안의 내용 중 /wiki/가 포함된 것만 골라내기

href = link.attrs['href']

p = re.compile('^(/wiki/)')

if p.search(href) != None:

print(href)

# HTML 파싱

- ❖ select() 함수를 사용할 수 있는데 이 함수의 파라미터는 CSS 스타일의 Selector
  - ✓ 선택자
    - 태그는 태그이름만 기재
    - 클래스 이름은 .을 앞에 추가
    - 아이디는 #을 추가
    - 태그 안에 클래스 속성이 일치하는 데이터를 찾을 때는 태그 이름과 클래스 이름을 연속해서 작성
    - 선택자 안의 선택자를 나타낼 때는 공백으로 구분
    - 선택자의 첫번째 레벨의 자식만 가져올 때는 선택자와 선택자 사이에 >를 추가
  - ✓ select( ' .news li ' ) 는 news 라는 css 클래스 안에 모든 li 태그들을 가져옴
  - ✓ 가져온 결과는 태그(s4.element.Tag)들의 리스트(list)
  - ✓ 태그 요소(bs4.element.Tag)로부터 태그 내 문자열을 가져오기 위해서는 getText()를 호출
  - ✓ 특정 태그의 attribute를 얻기 위해서는 get('attribute명')를 호출

# HTML 파싱

## ❖ 네이버 주식에서 타이틀 가져오기

```
import requests, bs4
try:
    resp = requests.get('https://finance.naver.com/')
    resp.raise_for_status()

    resp.encoding='euc-kr'
    html = resp.text
    bs = bs4.BeautifulSoup(html, 'html.parser')
    tags = bs.select('span > a') # Top 뉴스
    title = tags[0].getText()
    print(title)
except Exception as e:
    print("예외 발생:", e)
```

# HTML 파싱

## ❖ 네이버 팟 캐스트에서 타이틀 가져오기

```
import requests, bs4
try:
    res = requests.get('https://tv.naver.com/r/category/drama')
    res.raise_for_status()

    html = res.text
    bs = bs4.BeautifulSoup(html, 'html.parser')
    tags = bs.select('strong span')
    for title in tags:
        print(title.getText())
except Exception as e:
    print("예외 발생:", e)
```

# HTML 파싱

- ❖ 기상청의 현재 날씨 정보를 보여주는 페이지

<http://www.weather.go.kr/weather/observation/currentweather.jsp>

| <div> <div>선택   종합</div> <div>선택 전국</div> <div>선택 2019.10.07.18:00</div> <div>검색</div> </div> <div> <div>이전자료  </div> <div>12시간 전</div> <div>3시간 전</div> <div>1시간 전</div> <div>현재</div> <div>1시간 후</div> <div>3시간 후</div> <div>12시간 후</div> </div> |       |          |            |       |          |           |          |           |         |     |            |          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|----------|------------|-------|----------|-----------|----------|-----------|---------|-----|------------|----------|
| 기상실황표 2019.10.07.18:00                                                                                                                                                                                                                             |       |          |            |       |          |           |          |           |         |     |            |          |
| 지점                                                                                                                                                                                                                                                 | 날씨    |          |            | 기온(℃) |          |           | 강수       |           | 바람      |     | 기압(hPa)    |          |
|                                                                                                                                                                                                                                                    | 현재일기  | 시정<br>km | 운량<br>1/10 | 중하운량  | 현재<br>기온 | 이슬점<br>온도 | 체감<br>온도 | 일강수<br>mm | 습도<br>% | 풍향  | 풍속<br>km/h | 해면<br>기압 |
| 강릉                                                                                                                                                                                                                                                 |       | 17.1     |            |       | 17.4     | 13.5      | 17.4     | 0.0       | 78      | 북북서 | 4.3        | 1015.0   |
| 강진군                                                                                                                                                                                                                                                |       | 11.3     |            |       | 17.9     | 15.6      | 17.9     | 10.5      | 87      | 북북서 | 7.2        | 1016.3   |
| 강화                                                                                                                                                                                                                                                 |       | 20 이상    |            |       | 14.7     | 13.2      | 14.7     | 15.5      | 91      | 북북서 | 2.9        | 1016.4   |
| 거제                                                                                                                                                                                                                                                 |       | 3.7      |            |       | 18.8     | 18.4      | 18.8     | 0.5       | 98      | 정온  | 0.7        | 1014.1   |
| 거창                                                                                                                                                                                                                                                 |       | 2.9      |            |       | 15.0     | 14.8      | 15.0     | 7.7       | 99      | 정온  | 1.1        | 1014.9   |
| 경주시                                                                                                                                                                                                                                                |       | 2.9      |            |       | 16.7     | 15.7      | 16.7     | 2.5       | 94      | 북북서 | 6.5        | 1015.6   |
| 고산                                                                                                                                                                                                                                                 |       | 20 이상    |            |       | 20.5     | 18.0      | 20.1     | 2.2       | 86      | 북동  | 35.6       | 1016.2   |
| 고창                                                                                                                                                                                                                                                 |       |          |            |       | 16.2     | 15.8      | 15.9     | 13.0      | 98      | 북서  | 11.5       | 1016.7   |
| 고창군                                                                                                                                                                                                                                                |       | 20 이상    |            |       | 15.9     | 14.9      | 15.9     | 11.5      | 94      | 북   | 3.6        | 1015.6   |
| 고흥                                                                                                                                                                                                                                                 |       | 5.1      |            |       | 16.6     | 16.4      | 16.6     | 2.0       | 99      | 정온  | 1.4        | 1014.9   |
| 광양시                                                                                                                                                                                                                                                |       | 3.9      |            |       | 16.3     | 15.3      | 16.3     | 2.0       | 94      | 동남동 | 3.2        | 1014.9   |
| 광주                                                                                                                                                                                                                                                 | 비 끝남  | 20 이상    | 8          | 6     | 16.7     | 16.3      | 16.7     | 6.9       | 98      | 서   | 1.8        | 1016.1   |
| 구미                                                                                                                                                                                                                                                 |       | 3.3      |            |       | 15.2     | 15.0      | 15.2     | 2.9       | 99      | 서북서 | 2.9        | 1016.2   |
| 군산                                                                                                                                                                                                                                                 |       |          |            |       | 15.7     | 15.3      | 15.7     | 19.4      | 98      | 정온  | 0          | 1016.3   |
| 금산                                                                                                                                                                                                                                                 |       | 7.1      |            |       | 14.4     | 13.6      | 14.4     | 14.5      | 95      | 서   | 2.5        | 1016.3   |
| 김해시                                                                                                                                                                                                                                                |       | 20 이상    |            |       | 17.6     | 15.7      | 17.6     | 0.2       | 89      | 북서  | 2.5        | 1015.2   |
| 남원                                                                                                                                                                                                                                                 |       | 3.7      |            |       | 15.8     | 15.7      | 15.8     | 7.8       | 100     | 정온  | 0          | 1015.6   |
| 남해                                                                                                                                                                                                                                                 |       | 3.4      |            |       | 16.6     | 15.6      | 16.6     | 3.0       | 94      | 북   | 4          | 1015.1   |
| 대관령                                                                                                                                                                                                                                                |       | 14.3     |            |       | 11.3     | 10.8      | 11.3     | 0.0       | 97      | 서북서 | 3.2        | 1015.5   |
| 대구                                                                                                                                                                                                                                                 | 악한비계속 | 3.8      | 10         | 8     | 15.9     | 15.7      | 15.9     | 4.5       | 99      | 정온  | 0          | 1015.5   |
| 대전                                                                                                                                                                                                                                                 | 악한비계속 | 7.5      | 10         | 6     | 15.8     | 15.1      | 15.8     | 11.8      | 96      | 정온  | 0          | 1016.4   |
| 동두천                                                                                                                                                                                                                                                |       | 8.3      |            |       | 13.8     | 13.5      | 13.6     | 20.6      | 100     | 북북동 | 3.6        | 1016.6   |
| 동해                                                                                                                                                                                                                                                 |       | 19.9     |            |       | 16.1     | 14.2      | 16.1     | 0.3       | 89      | 정온  | 1.4        | 1014.8   |
| 목포                                                                                                                                                                                                                                                 | 비 끝남  | 20 이상    | 7          | 7     | 17.6     | 16.1      | 17.2     | 9.2       | 91      | 북   | 16.9       | 1016.7   |
| 문경                                                                                                                                                                                                                                                 |       | 3.1      |            |       | 13.8     | 13.3      | 13.8     | 5.5       | 97      | 정온  | 0.4        | 1016.4   |
| 밀양                                                                                                                                                                                                                                                 |       | 1.5      |            |       | 16.7     | 16.5      | 16.7     | 2.0       | 99      | 북서  | 2.9        | 1014.9   |
| 백령도                                                                                                                                                                                                                                                | 맑음    | 20 이상    | 2          | 0     | 15.1     | 9.6       | 15.1     | 2.2       | 70      | 서   | 6.1        | 1015.5   |

# HTML 파싱

## ❖ 현재 온도와 습도 출력

```
import requests          # 웹 페이지의 HTML을 가져오는 모듈
from bs4 import BeautifulSoup  # HTML을 파싱하는 모듈

# 웹 페이지를 가져온 뒤 BeautifulSoup 객체로 만들
response =
requests.get('http://www.weather.go.kr/weather/observation/currentweather.jsp')

soup = BeautifulSoup(response.content, 'html.parser')

table = soup.select('#weather_table > tbody')
loc = []
temp = []
hum = []
```

# HTML 파싱

## ❖ 현재 온도와 습도 출력

```
# 데이터를 저장할 리스트 생성
for tr in table[0].find_all('tr'):    # 모든 <tr> 태그를 찾아서 반복(각 지점의 데이터를 가져옴)
    tds = list(tr.find_all('td'))    # 모든 <td> 태그를 찾아서 리스트로 만들
    for td in tds:                  # <td> 태그 리스트 반복(각 날씨 값을 가져옴)
        if td.find('a'):            # <td> 안에 <a> 태그가 있으면(지점인지 확인)
            loc.append(td.find('a').text)    # <a> 태그 안에서 지점을 가져옴
            temp.append(tds[5].text)    # <td> 태그 리스트의 여섯 번째(인덱스 5)에서 기온을
가져옴
            hum.append(tds[9].text)    # <td> 태그 리스트의 열 번째(인덱스 9)에서 습도를
가져옴
print('도시 이름:', loc[0:5])
print('현재 온도:', temp[0:5])
print('습도:', hum[0:5])
print()
print()
```



# HTML 파싱

## ❖ 현재 온도와 습도 출력

```
cities= ['서울', '인천', '대전', '대구', '광주', '부산', '울산', '창원']
temperatures = []
humidities =[]
for city in cities:
    j = 0
    for c in loc:
        if city == c:
            temperatures.append(temp[j])
            humidities.append(hum[j])
            break
    j = j + 1

print('도시 이름:', cities)
print('현재 온도:', temperatures)
print('습도:', humidities)
```

# XML 파싱

## ❖ XML 파싱

- ✓ BeautifulSoup 패키지로 가능
- ✓ xml 패키지 이용

- xml 파일 읽기 - local file

```
import xml.etree.ElementTree as ET
doc = ET.parse(file_name)
root = doc.getroot()
```

- url 읽기

```
url = '
request = urllib.request.Request(url)
response = urllib.request.urlopen(request)
tree = ET.parse(response)
root = tree.getroot()
```

- DOM을 찾아오는 함수

- findall('태그'): 태그의 list로 리턴
- find('태그'): 태그 1개 리턴
- 찾은 객체의 text 속성을 이용하면 태그 안의 값을 선택

# XML 파싱

## ❖ Kakao API 읽기

팜엑스약국  
도심공항약국  
삼성수약국  
이도건강약국  
미소약국  
해성약국  
장생당약국  
삼성하나약국  
참좋은온누리약국  
사랑온누리약국  
소망약국

서울 강남구 삼성동 159  
서울 강남구 삼성동 159  
서울 강남구 삼성1동 155-11  
서울 강남구 삼성동 154-12  
서울 강남구 대치2동 944  
서울 강남구 대치2동 942  
서울 강남구 대치동 943-16  
서울 강남구 삼성동 158-16  
서울 강남구 삼성동 158-10  
서울 강남구 삼성동 158-14  
서울 강남구 삼성동 158-12

# XML 파싱

## ❖ Kakao API 읽기

```
import requests
import bs4
url =
'https://dapi.kakao.com/v2/local/search/category.xml?category_group_code=PM9&re
ct=127.0561466,37.5058277,127.0602340,37.5142554'
headers = {'Authorization': 'KakaoAK
{}'.format(c454c0e64688ce2bde2dfff9cceced87')}]
response = requests.post(url, headers=headers)
#print(response.text)
```

#파싱 객체 생성

```
xml = bs4.BeautifulSoup(response.text, "html.parser")
place_names = xml.find_all('place_name')
address_names = xml.find_all('address_name')
```

i = 0

```
for place_name in place_names:
    address_name = address_names[i]
    print(format(place_name.getText(), '30s'), end='Wt')
    print(format(address_name.getText(), '30s'), end='Wt')
    i = i + 1
```

# selenium

## ❖ selenium

### ✓ 개요

- selenium은 주로 웹 앱을 테스트하는데 이용하는 프레임워크
- Webdriver라는 API를 통해 운영체제에 설치된 Chrome 등의 브라우저를 제어
- 브라우저를 동작시켜서 JavaScript를 이용해 비동기적으로 혹은 뒤늦게 불러와지는 콘텐츠 들을 가져올 수 있음
- requests에서 사용했던 .text의 경우 브라우저에서 소스 보기를 한 것과 같이 동작하여 JS 등을 통해 동적으로 변화되는 HTML을 가져올 수 없음
- Selenium은 실제 웹 브라우저가 동작하기 때문에 JS로 렌더링이 완료된 후의 DOM 결과물에 접근 가능
- Selenium Manual
  - ◆ <http://selenium-python.readthedocs.io/index.html>
  - ◆ <http://docs.seleniumhq.org/docs/>
- 설치
  - ◆ `pip install selenium`

# selenium

## ❖ 브라우저 제어

### ✓ 크롬 제어

- 브라우저를 제어하기 위해서는 먼저 브라우저 드라이버가 필요
- 아래 주소에서 크롬 드라이버 다운로드 - 자신의 크롬 버전 확인

<https://chromedriver.chromium.org/downloads>

- 다운로드 받아서 압축을 해제

```
from selenium import webdriver
import os
os.environ['webdriver.chrome.driver'] = '크롬드라이버경로'
driver = webdriver.Chrome()
while(True):
    pass
```

- 다른 브라우저를 이용하고자 하면 브라우저에 해당하는 드라이버를 다운로드 받아야 하고 pantom.js를 이용하면 브라우저를 구동하지 않고도 가능

# selenium

## ❖ 크롬 실행

#크롬 실행

```
from selenium import webdriver
```

```
import os
```

```
os.environ['webdriver.chrome.driver'] =
```

```
'/Users/adam/Documents/lecture/python/1.Python/source/data/chromedriver'
```

```
driver = webdriver.Chrome()
```

```
while(True):
```

```
    pass
```

# selenium

## ❖ 함수

- ✓ `implicitly_wait(시간)`: 시간 동안 대기
- ✓ `get(url)`: url에 접속
- ✓ `page_source`: html 코드
- ✓ 페이지의 단일 element에 접근하는 api
  - `find_element_by_name('HTML_name')`
  - `find_element_by_id('HTML_id')`
  - `find_element_by_xpath('/html/body/some/xpath')`
- ✓ 페이지의 여러 elements에 접근하는 api
  - `find_element_by_css_selector('#css > div.selector')`
  - `find_element_by_class_name('some_class_name')`
  - `find_element_by_tag_name('h1')`
- ✓ `quit()`: 브라우저 종료
- ✓ `execute_script(스크립트코드)`



# selenium

## ❖ html 가져오기

```
#html 가져오기
from selenium import webdriver
import os
os.environ['webdriver.chrome.driver'] =
    '/Users/adam/Documents/lecture/python/1.Python/source/data/chromedriver'
driver = webdriver.Chrome()
driver.get("https://pt.masanggames.co.kr/")
print(driver.page_source)
while(True):
    pass
```

# selenium

## ❖ 다음 카페 페이지로 이동

#다음 카페 로그인 페이지로 이동

```
from selenium import webdriver
```

```
import os
```

```
os.environ['webdriver.chrome.driver'] =
```

```
'/Users/adam/Documents/lecture/python/1.Python/source/data/chromedriver'
```

```
driver = webdriver.Chrome()
```

```
driver.implicitly_wait(5)
```

#다음 카페 페이지로 이동 - 로그인 되어 있지 않으므로 로그인 페이지로 이동

```
driver.get('https://accounts.kakao.com/login/?continue=https%3A%2F%2Flogins.daum.net
```

```
%2Faccounts%2Fksso.do%3Frescue%3Dtrue%26url%3Dhttps%253A%252F%252Fwww.dau
```

```
m.net%252F&login_type=simple#login')
```

```
driver.implicitly_wait(5)
```

```
while(True):
```

```
    pass
```

# selenium

## ❖ 함수

- ✓ Element를 찾아서 `send_keys('값')`을 호출하면 선택한 Element에 값을 입력
- ✓ Element를 찾아서 `click()` 이나 `submit()`을 호출하면 버튼을 누른 것과 동일한 효과 수행
- ✓ BeautifulSoup 객체를 이용하려면 `driver.page_source` 속성을 이용해 현재 렌더링 된 페이지의 Elements를 모두 가져올 수 있음



# selenium

## ❖ 다음 로그인

#다음 카페 로그인

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import os
import time
```

```
os.environ['webdriver.chrome.driver'] =
'/Users/adam/Documents/lecture/python/1.Python/source/data/chromedriver'
driver = webdriver.Chrome()
driver.implicitly_wait(5)
```

```
#다음 카페 페이지로 이동 - 로그인 되어 있지 않으므로 로그인 페이지로 이동
driver.get('https://accounts.kakao.com/login/?continue=https%3A%2F%2Flogins.daum.net%2Faccounts%2Fksso.do%3Frescue%3Dtrue%26url%3Dhttps%253A%252F%252Fwww.daum.net%252F&login_type=simple#login')
driver.implicitly_wait(5)
```

# selenium

## ❖ 다음 로그인

#아이디 와 비밀번호 입력받기

```
userid = input("아이디:")
```

```
userpw = input("비밀번호:")
```

```
time.sleep(3)
```

#입력받은 내용을 삽입하기

```
driver.find_element(By.XPATH, '//*[@id="loginId--1"]').send_keys(userid)
```

```
driver.find_element(By.XPATH, '//*[@id="password--2"]').send_keys(userpw)
```

#버튼

```
btn = driver.find_element(By.XPATH,
```

```
'//*[@id="mainContent"]/div/div/form/div[4]/button[1]')
```

```
btn.click()
```

```
while(True):
```

```
    pass
```

# selenium

## ❖ 프레임 접근

- ✓ Selenium을 활용하여 크롤링 하다보면 눈에는 원하는 정보가 보이고 크롬의 개발자 도구에서도 원하는 부분이 다 보이는데 파이썬으로 접근이 안되는 경우는 대부분 프레임 구조로 만들어져 있는 경우 나 ajax로 데이터를 불러오는 경우

- ✓ 소스 코드로 frame 확인

# 프레임 살펴보기

```
iframes = driver.find_elements_by_css_selector('iframe')
```

```
for iframe in iframes:
```

```
    print(iframe.get_attribute('name'))
```

- ✓ 프레임으로의 전환

```
driver.switch_to_default_content #상위 프레임으로 전환
```

```
driver.switch_to_frame('프레임이름') # 특정 프레임으로 전환
```

# selenium

## ❖ 카페 목록 페이지로 이동한 후 카페로 이동

#다음 카페 목록 페이지로 이동

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import os
import time
```

```
os.environ['webdriver.chrome.driver'] =
'/Users/adam/Documents/lecture/python/1.Python/source/data/chromedriver'
driver = webdriver.Chrome()
driver.implicitly_wait(5)
```

```
#다음 카페 페이지로 이동 - 로그인 되어 있지 않으므로 로그인 페이지로 이동
driver.get('https://accounts.kakao.com/login/?continue=https%3A%2F%2Flogins.daum.net%2Faccounts%2Fksso.do%3Frescue%3Dtrue%26url%3Dhttps%253A%252F%252Fwww.daum.net%252F&login_type=simple#login')
driver.implicitly_wait(5)
```

# selenium

❖ 카페 목록 페이지로 이동한 후 카페로 이동

#아이디 와 비밀번호 입력받기

```
userid = input("아이디:")
```

```
userpw = input("비밀번호:")
```

```
time.sleep(3)
```

#입력받은 내용을 삽입하기

```
driver.find_element(By.XPATH, '//*[@id="loginId--1"]').send_keys(userid)
```

```
driver.find_element(By.XPATH, '//*[@id="password--2"]').send_keys(userpw)
```

#버튼

```
btn = driver.find_element(By.XPATH,
```

```
'//*[@id="mainContent"]/div/div/form/div[4]/button[1]')
```

```
btn.click()
```

```
time.sleep(3)
```

```
driver.get('http://top.cafe.daum.net/_c21_/my_cafe')
```

```
time.sleep(3)
```

```
driver.find_element(By.XPATH, '//*[@id="mArticle"]/div/div[1]/div/div[2]/ul/li/a').click()
```

```
while(True):
```

```
    pass
```



# selenium

## ❖ 메모 작성

```
#다음 카페 목록 페이지로 이동
from selenium import webdriver
from selenium.webdriver.common.by import By
import os
import time
```

```
os.environ['webdriver.chrome.driver'] =
'/Users/adam/Documents/lecture/python/1.Python/source/data/chromedriver'
driver = webdriver.Chrome()
driver.implicitly_wait(5)
```

```
#다음 카페 페이지로 이동 - 로그인 되어 있지 않으므로 로그인 페이지로 이동
driver.get('https://accounts.kakao.com/login/?continue=https%3A%2F%2Flogins.daum.net%2Faccounts%2Fksso.do%3Frescue%3Dtrue%26url%3Dhttps%253A%252F%252Fwww.daum.net%252F&login_type=simple#login')
driver.implicitly_wait(5)
```

# selenium

## ❖ 메모 작성

#아이디 와 비밀번호 입력받기

```
userid = input("아이디:")
```

```
userpw = input("비밀번호:")
```

```
time.sleep(3)
```

#입력받은 내용을 삽입하기

```
driver.find_element(By.XPATH, '//*[@id="loginId--1"]').send_keys(userid)
```

```
driver.find_element(By.XPATH, '//*[@id="password--2"]').send_keys(userpw)
```

#버튼

```
btn = driver.find_element(By.XPATH,
```

```
'//*[@id="mainContent"]/div/div/form/div[4]/button[1]')
```

```
btn.click()
```

```
time.sleep(3)
```

```
driver.get('http://top.cafe.daum.net/_c21_/my_cafe')
```

```
time.sleep(3)
```

```
driver.find_element(By.XPATH, '//*[@id="mArticle"]/div/div[1]/div/div[2]/ul/li/a').click()
```

# selenium

## ❖ 메모 작성

# 메모 작성

```
time.sleep(3)
```

```
driver.get('http://cafe.daum.net/samhak7/_memo')
```

```
time.sleep(3)
```

```
driver.switch_to.frame('down')
```

```
driver.find_element(By.XPATH,
```

```
'//*[@id="primaryContent"]/div/div[1]/div[2]/div/div/div[1]/textarea').send_keys(  
    '모두들 안녕 매크로 연습이야')
```

```
time.sleep(3)
```

```
driver.find_element(By.XPATH,
```

```
'//*[@id="primaryContent"]/div/div[1]/div[2]/div/div/div[2]/div[2]/div/button').click()
```

```
time.sleep(3)
```

```
driver.get('http://top.cafe.daum.net/_c21_/my_cafe')
```

```
time.sleep(3)
```

```
driver.find_element(By.XPATH,
```

```
'//*[@id="mArticle"]/div/div[2]/div[2]/ul/li[1]/div[1]/a/div[2]/div/div/div[1]/strong'  
)click()
```

# selenium

## ❖ 크롬 브라우저를 화면에 출력하지 않고 데이터 가져오기

#크롬 브라우저를 구동하지 않고 데이터 가져오기

```
from selenium import webdriver
import os
```

```
options = webdriver.ChromeOptions()
options.add_argument('headless')
options.add_argument('window-size=1920x1080')
options.add_argument("disable-gpu")
# 혹은 options.add_argument("--disable-gpu")
```

# Chrome의 경우 | 아까 받은 chromedriver의 위치를 지정

```
os.environ['webdriver.chrome.driver'] =
'/Users/adam/Documents/lecture/python/1.Python/source/data/chromedriver'
driver = webdriver.Chrome(options=options)
driver.get('https://www.naver.com')
html = driver.page_source
print(html)
```

# selenium

## ❖ 네이버 자동 로그인

#네이버 자동 로그인

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import os
import time
```

# Chrome의 경우 | 아까 받은 chromedriver의 위치를 지정

```
os.environ['webdriver.chrome.driver'] =
'/Users/adam/Documents/lecture/python/1.Python/source/data/chromedriver'
driver = webdriver.Chrome()
driver.get('https://nid.naver.com/nidlogin.login')
time.sleep(3)
```

```
userid = input("아이디:")
userpw = input("비밀번호:")
```

```
driver.execute_script("document.getElementsByName('id')[0].value=W'" + userid + "W'")
driver.execute_script("document.getElementsByName('pw')[0].value=W'" + userpw + "W'")
driver.find_element(By.XPATH, '//*[@id="log.login"]').click()
while(True):
    pass
```

# selenium

## ❖ 유튜브 검색 결과 스크롤 해서 가져오기

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import os
import time
from selenium.webdriver.common.keys import Keys
import bs4

os.environ['webdriver.chrome.driver'] =
'/Users/adam/Documents/lecture/python/1.Python/source/data/chromedriver'
driver = webdriver.Chrome()
```

# selenium

## ❖ 유튜브 검색 결과 스크롤 해서 가져오기

```
driver.get('https://www.youtube.com/results?search_query=%EB%A7%88%EB%A7%88%EB%AC%B4')
time.sleep(5)
body = driver.find_element(By.TAG_NAME, 'body') # 스크롤하기 위해 소스 추출
num_of_pagedowns = 10
# 10번 밑으로 내리는 것
while num_of_pagedowns:
    body.send_keys(Keys.PAGE_DOWN)
    time.sleep(2)
    num_of_pagedowns -= 1

html = bs4.BeautifulSoup(driver.page_source, 'html.parser')
print(html)
```