

# ⋮ 문제해결을 위한 데이터 분석 및 시각화

## 웹 크롤링(1)

한성대학교 노은희 교수

“미래로 향하는 새로운 이정표”



# 오늘의 학습

---

## 학습내용

- 웹 크롤링



## 용어

---

크롤링은 웹 인덱싱을 위해 www를 체계적으로 탐색해나가는 것을 의미

웹 크롤링은 인터넷에 있는 웹페이지를 방문하여 페이지의 자료를 자동으로 수집하는 작업 의미

웹 스크래핑은 다양한 웹사이트로부터 데이터를 추출하는 기술을 의미

웹 파싱은 웹 상의 자연어, 컴퓨터 언어 등의 일련의 문자열들을 분석하는 프로세스



## 용어 이해

크롤링이란 무수히 많은 컴퓨터에 분산 저장되어 있는 문서를 수집하여 검색 대상의 색인으로 포함시키는 기술을 의미하며, 스크래핑(Scraping) 이라고도 한다.

웹 페이지를 가져와서 그 안에서 데이터를 추출하는 기술

크롤링은 웹 인덱싱을 위해 www를 체계적으로 탐색해나가는 것을 의미

웹 크롤링은 인터넷에 있는 웹페이지를 방문하여 페이지의 자료를 자동으로 수집하는 작업 의미

웹 스크래핑은 다양한 웹사이트로부터 데이터를 추출하는 기술을 의미

웹 파싱은 웹 상의 자연어, 컴퓨터 언어 등의 일련의 문자열들을 분석하는 프로세스

### 크롤링을 할 때 주의할 점

웹사이트에서 크롤링봇 접근을 Disallow 하는 페이지는 크롤링을 해서는 안된다. 이는 처벌을 받을 수 있다.

최상위 도메인주소 뒤에 /robots.txt를 입력하면 접근 허용 여부 콘텐츠를 확인



# 크롤링할때 주의할 점

---

## 크롤링을 할 때 주의할 점

웹사이트에서 크롤링봇 접근을 Disallow 하는 페이지는 크롤링을 해서는 안된다. 이는 처벌을 받을 수 있다.  
최상위 도메인주소 뒤에 /robots.txt를 입력하면 접근 허용 여부 콘텐츠를 확인

# 크롤링

## 크롤링을 할 때 주의할 점

- 웹사이트에서 크롤링봇 접근을 Disallow 하는 페이지는 크롤링을 해서는 안된다. 이는 처벌을 받을 수 있다.
- 자동 크롤링 로봇은 사이트 방문시 로봇배제표준 설정파일(robots.txt)를 확인한 후 이를 준수하여 콘텐츠를 수집  
**로봇의 크롤링 허가 여부를 명시해 놓은 파일"**
- 최상위 도메인주소 뒤에 /robots.txt를 입력하면 접근 허용 여부 콘텐츠를 확인

<https://www.google.com/robots.txt>

- Disallow 라고 되어있는 하위 디렉토리 페이지들에서는 크롤링을 할 수 없다.
- \* 크롤링 허용과 저작권 문제는 또 다른 사안이니 주의

← → ↻ 🔒 google.com/robots.txt

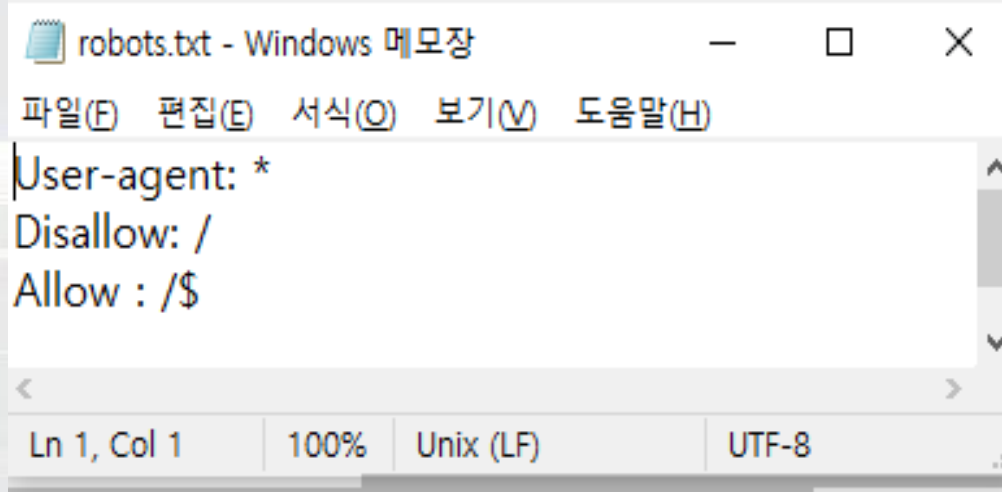
```
User-agent: *
Disallow: /search
Allow: /search/about
Allow: /search/static
Allow: /search/howsearchworks
Disallow: /sdch
Disallow: /groups
Disallow: /index.html?
Disallow: /?
Allow: /?hl=
Disallow: /?hl=*
Allow: /?hl=*&gws_rd=ssl$
Disallow: /?hl=*&gws_rd=ssl
Allow: /?gws_rd=ssl$
Allow: /?pt1=true$
Disallow: /imgres
Disallow: /u/
Disallow: /preferences
Disallow: /setprefs
Disallow: /default
Disallow: /m?
Disallow: /m/
Allow: /m/finance
```

User-agent:로봇의 이름

Allow: 허용

Disallow: 비허용 (만약 이 부분이 비어있으면 모두 허용)

# 크롤링



```
robots.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
User-agent: *
Disallow: /
Allow : /$
Ln 1, Col 1 100% Unix (LF) UTF-8
```

Disallow는 자동화 프로그램의 접근이 허용되지 않는 부분이고, Allow는 허용되는 부분  
/는 모든 페이지 /\$는 첫페이지 의미

네이버는 첫 접속 페이지를 제외하고 모든 페이지에서 웹 크롤링 접근을 제한함

User-agent : 로봇의 이름

Allow : 허용

Disallow : 비허용(이 부분이 비어 있으면 모두 허용)

## 크롤링을 위한 준비

BeautifulSoup, Requests 패키지가 설치되었는지 확인

```
Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\노은희> pip list
```

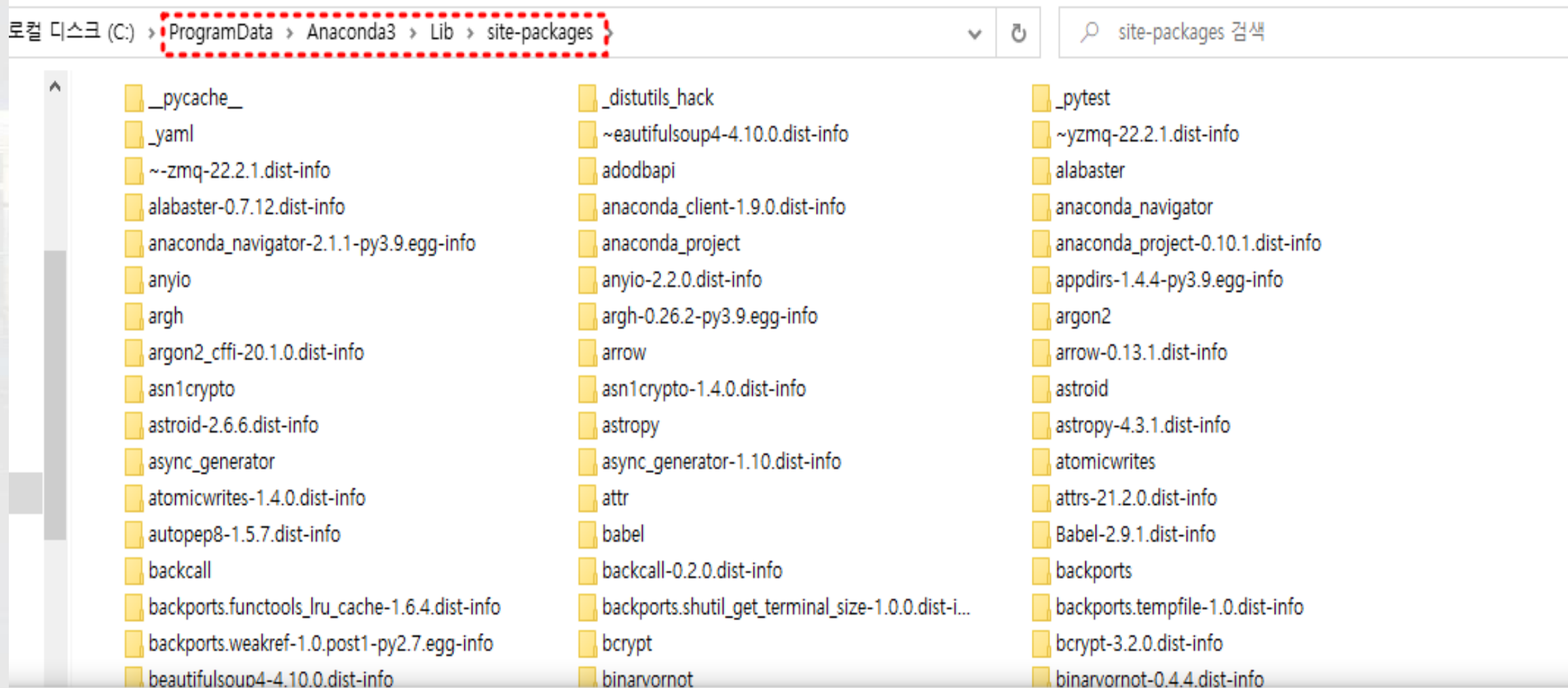
(base) PS C:\Users\사용자명> pip list

```
backcall 0.2.0
backports.functools-lru-cache 1.6.4
backports.shutil-get-terminal-size 1.0.0
backports.tempfile 1.0
backports.weakref 1.0.post1
bcrypt 3.2.0
beautifulsoup4 4.10.0
```

```
Anaconda Powershell Prompt (Anaconda3)
QtPy 1.10.0
regex 2021.8.3
requests 2.26.0
rope 0.19.0
Rtree 0.9.7
ruamel-yaml-conda 0.15.100
scikit-image 0.18.3
scikit-learn 0.24.2
scikit-learn-intelex 2021.20210714.120553
scipy 1.7.1
seaborn 0.11.2
selenium 4.1.3
```



## 설치된 라이브러리



## 크롤링을 위한 준비

BeautifulSoup, Requests 패키지 설치

```
Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\노은희> pip install requests

Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\노은희> pip install beautifulsoup4
```

**BeautifulSoup:** 웹 페이지의 정보를 쉽게 스크랩할 수 있도록 기능을 제공하는 라이브러리  
웹사이트 내의 html코드를 긁어오고 본격적인 데이터 추출을 하기 위함

**Beautiful Soup**은 HTML 및 XML 문서를 구문 분석하기 위한 Python 패키지  
HTML에서 데이터를 추출하는 데 사용할 수 있는 구문 분석된 페이지에 대한 구문 분석 트리를 만들며, 웹 스크래핑에 유용

**Requests:** HTTP 요청을 보낼 수 있도록 기능을 제공하는 라이브러리



# XML이란

---

## XML이란?

XML은 EXtensible Markup Language의 약자이며, 1998년에 W3C 표준 권고안에 포함

XML은 HTML과 매우 비슷한 문자 기반의 마크업 언어(text-based markup language)

XML은 HTML처럼 데이터를 보여주는 목적이 아닌, 데이터를 저장하고 전달할 목적으로만 만들어짐.

XML 태그는 HTML 태그처럼 미리 정의되어 있지 않고, 사용자가 직접 정의하여 사용 가능

# XML이란

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <shop city="서울" type="마트">
4
5      <food>
6
7          <name>사과</name>
8
9          <sort>과일</sort>
10
11          <cost>10000</cost>
12
13      </food>
14
15      <food>
16
17          <name>배추</name>
18
19          <sort>야채</sort>
20
21          <cost>3000</cost>
22
23      </food>
24
25  </shop>
```

```
xml.xml - Windows 메모장
파일(F)  편집(E)  서식(O)  보기(V)  도움말(H)

<?xml version="1.0" encoding="UTF-8"?>

<shop city="서울" type="마트">

    <food>

        <name>사과</name>

        <sort>과일</sort>

        <cost>10000</cost>

    </food>

    <food>

        <name>배추</name>

        <sort>야채</sort>

        <cost>3000</cost>

    </food>

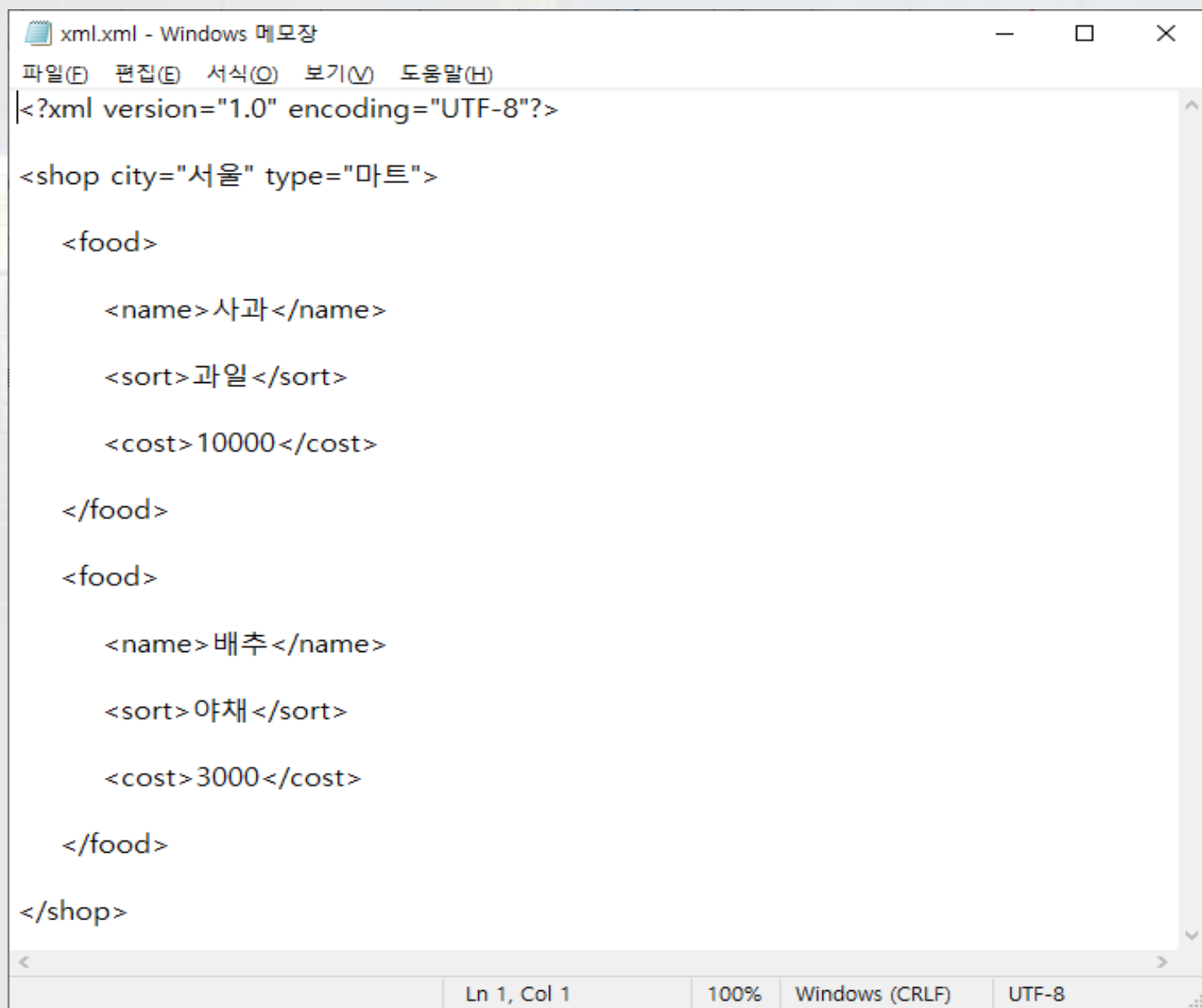
</shop>
```

Ln 1, Col 1    100%    Windows (CRLF)    UTF-8

-이픈(-), 언더스코어  
작성  
문자를 구분  
작해야 하며, 공백을  
의 이름으로 사용할 수  
름은 반드시 대소문자  
구조 및 기타 기능을



# XML이란 -> 메모장으로 파일 열기



```
xml.xml - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<?xml version="1.0" encoding="UTF-8"?>

<shop city="서울" type="마트">

  <food>

    <name>사과</name>

    <sort>과일</sort>

    <cost>10000</cost>

  </food>

  <food>

    <name>배추</name>

    <sort>야채</sort>

    <cost>3000</cost>

  </food>

</shop>
```

Ln 1, Col 1    100%    Windows (CRLF)    UTF-8

## 크롤링을 위한 준비

### find() 함수

find() 함수는 조건을 만족하는 태그를 하나만 가져오는 함수

### find\_all() 함수

find\_all() 함수는 원하는 태그가 여러 개 있을 경우 해당하는 태그를 한꺼번에 가져오는 함수

### 여러 가지의 태그를 찾아야 하는 상황

찾고 싶은 태그를 리스트로 find\_all() 함수에 넣어주면 된다.

*# find\_all 함수로 여러 가지 태그를 조회하고 싶다면?*

```
soup.find_all(['p', 'img'])
```

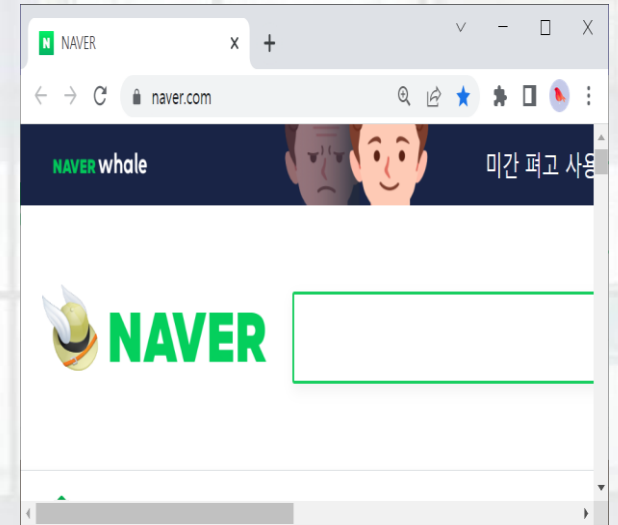
```
[<p align="center">text 1</p>,  
<p align="center">text 2</p>,  
<p align="center">text 3</p>,  
]
```

# 웹 브라우저로 웹사이트 접속하기

## 웹 브라우저로 웹사이트에 접속하기

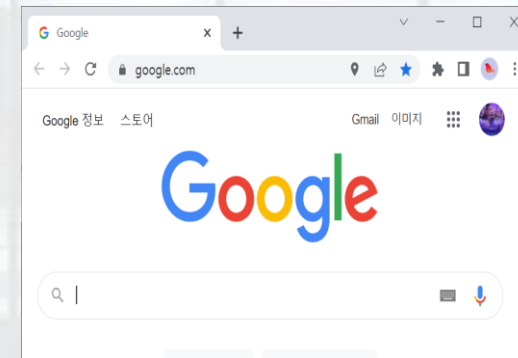
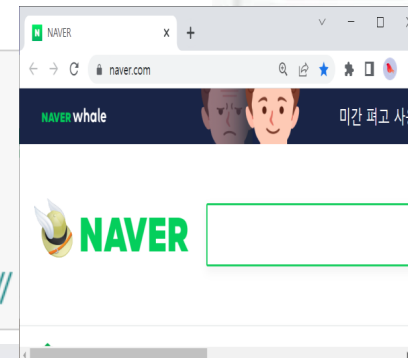
```
1 import webbrowser # webbrowser 모듈 불러오기
2 url = 'www.naver.com'
3 webbrowser.open(url) # 변수 url에 지정된 웹사이트(www.naver.com)에 접속
```

True



## 여러개의 웹 사이트에 접속하기

```
1 import webbrowser
2
3 urls = ['www.naver.com', 'www.google.com']
4 for url in urls:
5     webbrowser.open(url) # 변수 url에 지정된 웹사이트(www.naver.com)에
```





## 응답데이터 Response Content

속성	설명
<b>status_code</b>	응답 상태를 확인
<b>headers</b>	headers정보를 확인
<b>cookies</b>	cookies정보를 확인
<b>encoding</b>	데이터 인코딩을 확인
<b>text</b>	'str 타입의 데이터
<b>content</b>	bytes 타입의 데이터
<b>.json()</b>	dict 타입의 데이터 일 경우 사용



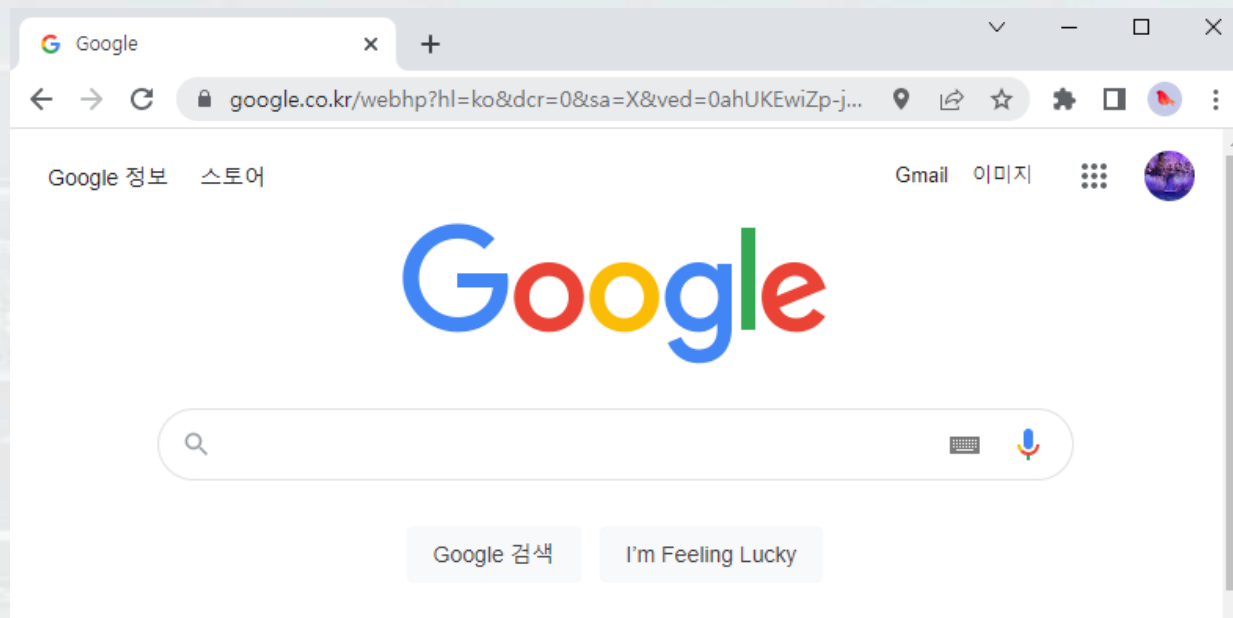
# request 라이브러리

## request 라이브러리 사용

```
1 import requests
2 r = requests.get('http://www.google.com')
3 r
```

<Response [200]>

접속이 잘 되면 Response [200]반환



# request 라이브러리

```
1 import requests
2 r = requests.get('http://www.google.com').text
3 r[0:100]
```

```
'<!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="a6VQ8Ahr i03WUvJh6F1B5g==">(function(){window.google={kEl:₩'p6NNYrH0AufQ2roPhOuDkA8₩',kEXPl:₩'0,1302536,56873,1709,4349,207,2414,2390,925,1391,383,246,5,1354,4013,1237,1122516,1197774,627,380090,16114,28684,17572,4858,1362,9291,3026,2817,14765,4020,978,13227,3848,4192,6431,7431,15309,5081,885,709,1278,2742,149,562,541,840,6297,3514,606,2025,1775,520,14670,3227,2845,7,17450,15768,552,1851,15756,3,346,230,6459,149,13975,4,1528,2304,7039,25073,2658,7355,32,13628,4437,9358,7428,5815,2542,4094,4052,3,3541,1,14263,2544,25347,2,14022,1931,4317,1272,743,5853,10463,1160,5679,1020,2378,2721,18234,9,2,6,7773,4567,6259,9497,13921,1249,4591,2,6,1239,11862,3106,1538,2794,19,4658,1412,1395,445,2,2,1,6394,565,3831,513,13476,14'
```

# BeautifulSoup 라이브러리로 파싱(Parsing)하기

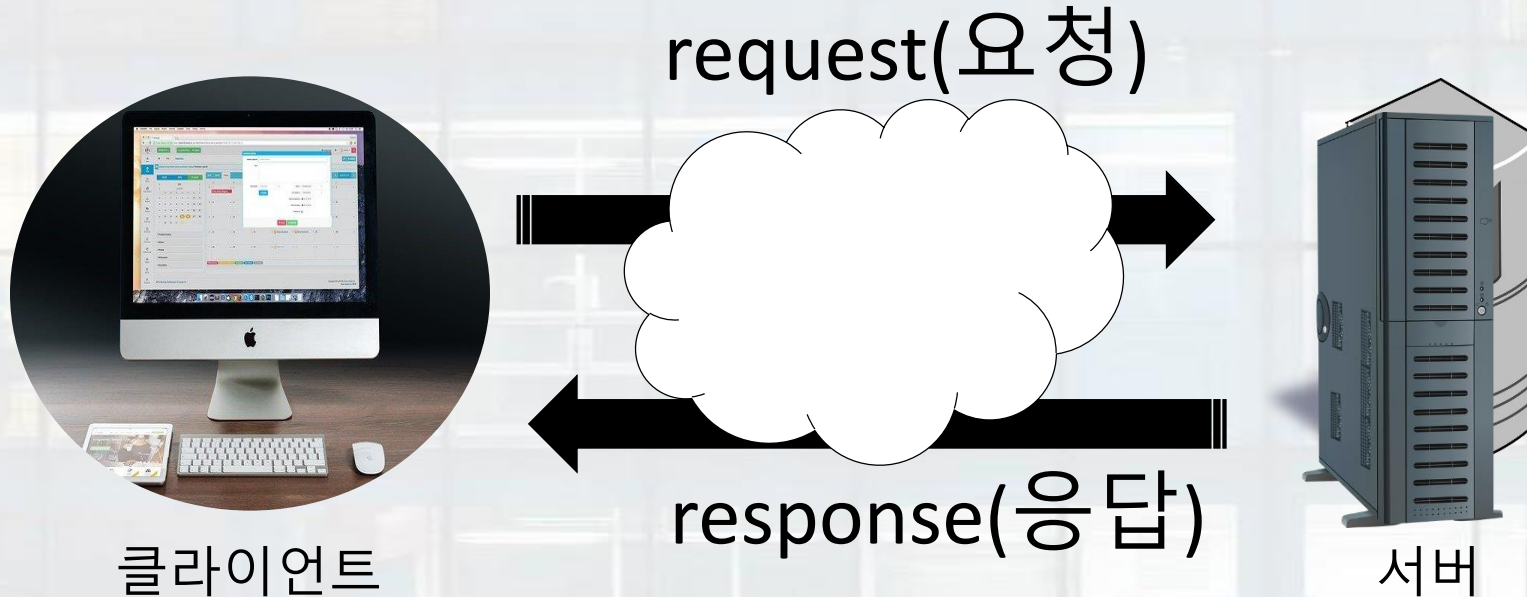
---

```
from bs4 import BeautifulSoup
```

Python에서 XML parser로서 주로 이용되는 패키지는 lxml

Python의 두 가지 주요 HTML 구문 분석 라이브러리는 lxml과 BeautifulSoup

## 데이터의 요청과 응답 과정



컴퓨터에서 웹 브라우저로 인터넷을 통해 웹서버에 HTTP형식으로 원하는 정보를 요청(request) 웹서버가 HTTP형식으로 응답(response)해 HTML파일을 보내준다.

- HTTP(HyperText Transfer Protocol)의 약자로 인터넷 상에서 HTML 문서의 정보를 주고받을 수 있도록 만든 프로토콜(전송규약)
- HTML : HyperText Markup Language로 웹 페이지의 구조적 구성을 위한 언어
- 웹페이지 : 웹상에 있는 HTML로 구성된 개별 문서



# HTTP 응답 코드

## 2xx 성공

200: 클라이언트의 요청을 정상적으로 수행함.

201: 클라이언트에게 생성 작업을 요청 받았고, 생성 작업을 성공함.

204: 요청은 성공 했지만 응답할 콘텐츠가 없음.

## 3xx 리다이렉션

301: 클라이언트가 요청한 리소스에 대한 URI가 영구적으로 변경되었을 때 사용함.

302: 301과 같으나 임시적으로 주소가 바뀌었을 경우 사용함.

304: 이전에 방문했을 때의 요청 결과와 다르지 않을 경우 사용함. 캐시된 페이지를 그대로 사용.

307: 임시 페이지로 리다이렉트.



# HTTP 응답 코드

---

## 4xx 클라이언트 오류

- 400: 클라이언트가 올바르지 못한 요청을 보냄.
- 401: 로그인을 하지 않아 페이지를 열 권한이 없음.
- 403: 금지된 페이지, 로그인을 하든 안하든 접근할 수 없음. (관리자 페이지)
- 404: 찾을 수 없는 페이지, 주소를 잘 못 입력했을 때 사용함.
- 403 대신에 사용할 수도 있음.(해커들의 공격을 방지하고자 페이지가 없는 것처럼 위장함)
- 408: 요청 시간이 초과됨.
- 409: 서버가 요청을 처리하는 과정에서 충돌이 발생한 경우. (회원가입 중 중복된 아이디인 경우)
- 410: 영구적으로 사용할 수 없는 페이지.

## 5xx 서버 오류

- 501: 해당 요청을 처리하는 기능이 만들어지지 않음.
- 502: 서버로 가는 요청이 중간에서 유실된 경우.
- 503: 서버가 터졌거나 유지 보수 중  
(유지 보수 중일때는 유지 보수중이라는 것을 알려주는 페이지로 전송해주는 것이 좋음)
- 504: 서버 게이트웨이에 문제가 생겨 시간 초과가 된 경우.
- 505: HTTP 버전이 달라 요청이 처리할 수 없음.