



Python Web Crawling 강의 1

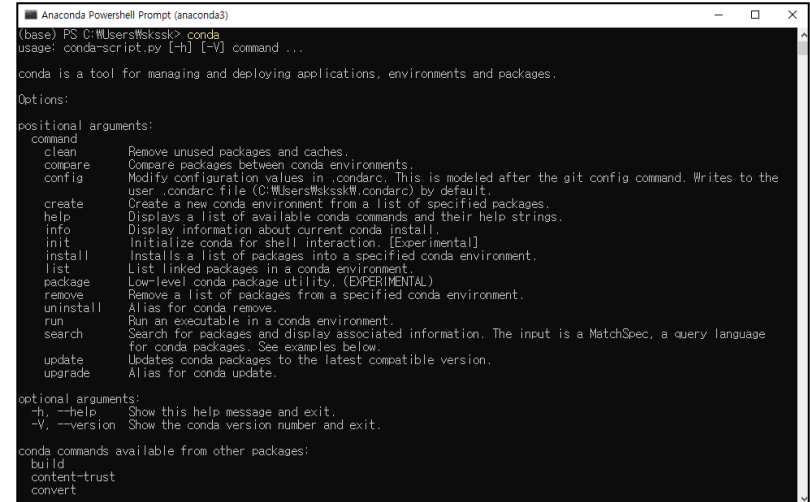
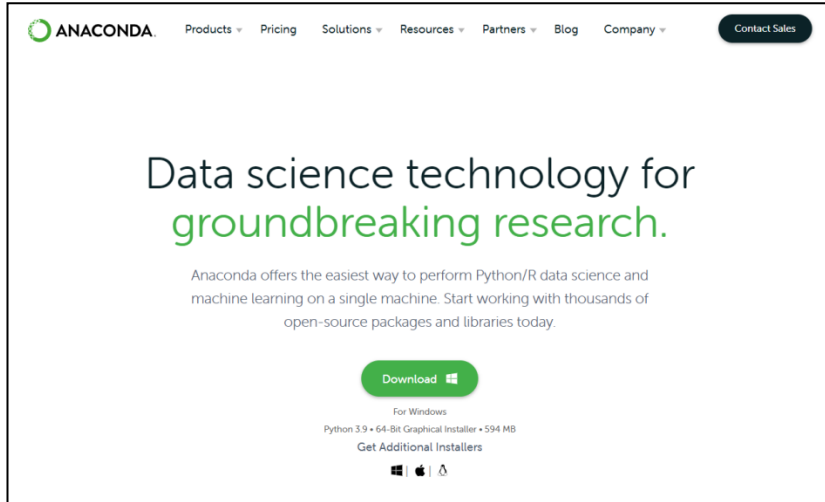
1. 환경 설정
2. WEB / SERVER / DB 역할 및 이해
3. Crawling 설명
4. 이슈

김덕호

01055704817
sksski1359@gmail.com

1. 환경 설정

- ANACONDA 다운 → <https://www.anaconda.com/>



- conda python3.6 가상환경 설정

1) 가상환경 만들기

`conda create -n 가상환경이름 python=버전`

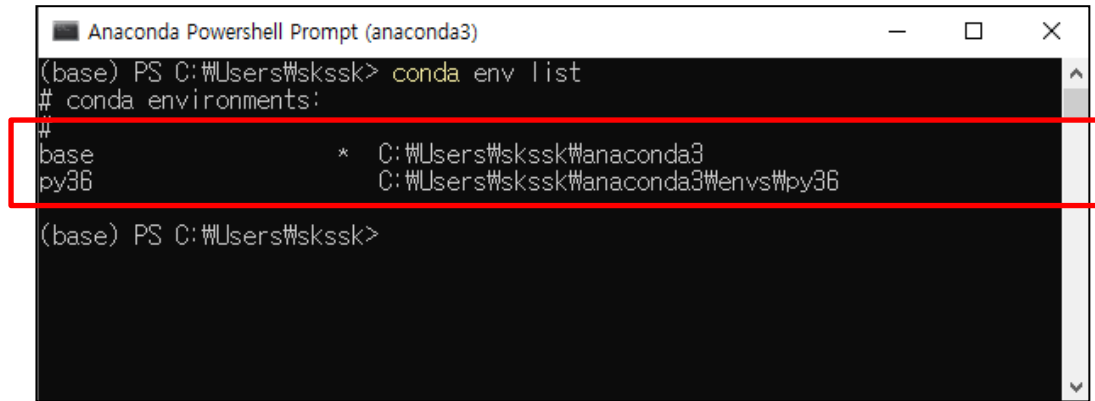
`conda create -n py36 python=3.6`

3.6 버전의 python 가상환경을 anaconda에서 만들어 버전 관리 및 모듈 관리를 용이하게 함.

1. 환경 설정

2) 가상환경 설정 확인

```
conda env list
```

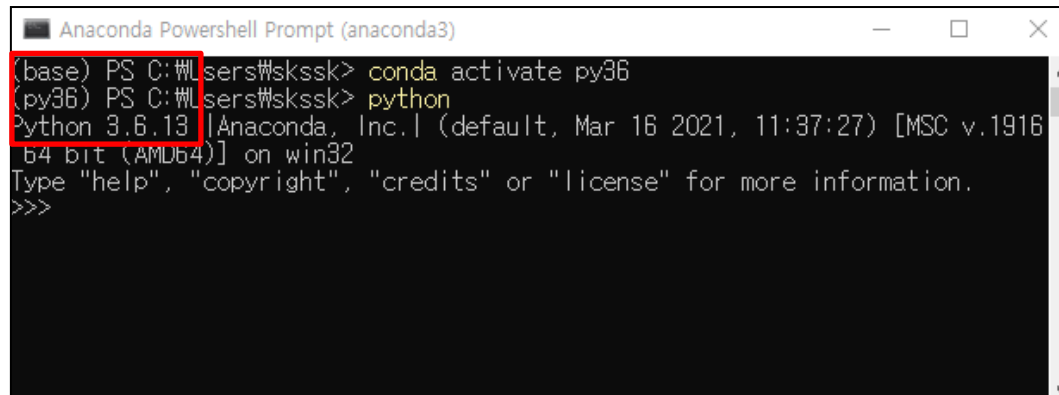


```
Anaconda PowerShell Prompt (anaconda3)
(base) PS C:\Users\Wskssk> conda env list
# conda environments:
#
base                  *  C:\Users\Wskssk\anaconda3
py36                  C:\Users\Wskssk\anaconda3\envs\py36

(base) PS C:\Users\Wskssk>
```

3) 가상환경 변경

```
conda activate py36
```



```
Anaconda PowerShell Prompt (anaconda3)
(base) PS C:\Users\Wskssk> conda activate py36
(py36) PS C:\Users\Wskssk> python
Python 3.6.13 [Anaconda, Inc.] (default, Mar 16 2021, 11:37:27) [MSC v.1916
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

1. 환경 설정

- python 크롤링 관련 모듈 설치

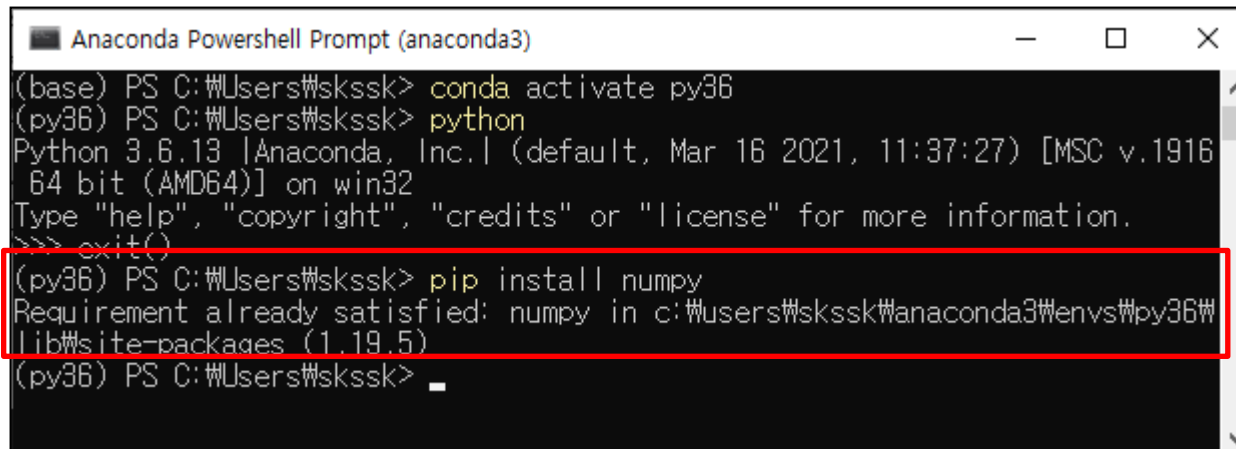
```
pip install 모듈이름
```

1) 모듈

- numpy: 수학 수식 및 계산을 할 수 있는 모듈
- pandas: 데이터를 프레임 형태로 정리할 수 있는 모듈
- requests: 서버에 호출하고 응답을 받을 수 있는 모듈
- urllib: url 을 다루는 모듈
- BeautifulSoup4: 정적 페이지의 html을 긁어오는 모듈
- selenium: 웹을 동적 제어하는 모듈
- matplotlib: 그래프 그리는 모듈

ex) pip를 이용한 설치와 설치 완료 확인 화면

```
pip install numpy
```

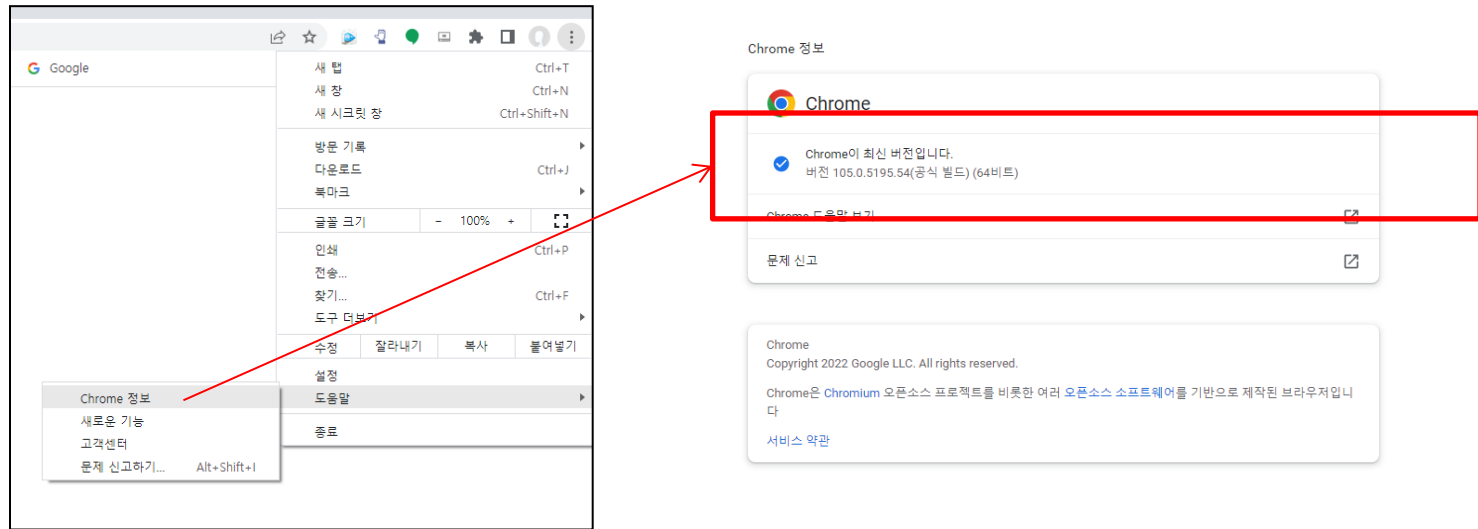


```
Anaconda Powershell Prompt (anaconda3)
(base) PS C:\Users\Wskssk> conda activate py36
(py36) PS C:\Users\Wskssk> python
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
(py36) PS C:\Users\Wskssk> pip install numpy
Requirement already satisfied: numpy in c:\Users\Wskssk\anaconda3\envs\py36\l
ib\site-packages (1.19.5)
(py36) PS C:\Users\Wskssk> _
```

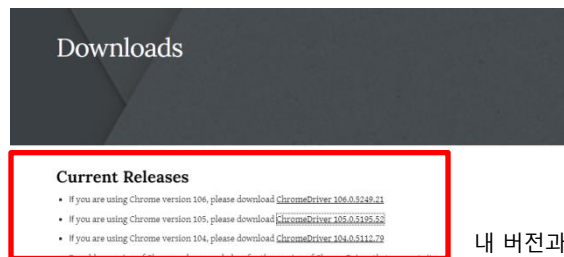
1. 환경 설정

2) selenium 크롬 브라우저 설정

- Chrome 정보, 버전 확인: 105.0.5195.54



3) selenium 크롬 브라우저 다운: <https://chromedriver.chromium.org/downloads>



내 버전과 같은 버전 + 같은 OS의 브라우저 다운

If you are using Chrome from Dev or Canary channel, please following instructions on the [ChromeDriver Canary](#) page.

For more information on selecting the right version of ChromeDriver, please see the [Version Selection](#) page.

ChromeDriver 106.0.5249.21

Supports Chrome version 106

- Resolved issue 4016: Add basic BIDI support to ChromeDriver (Mapper based) [Pri-1]

For more details, please see the [release notes](#).

ChromeDriver 105.0.5195.52

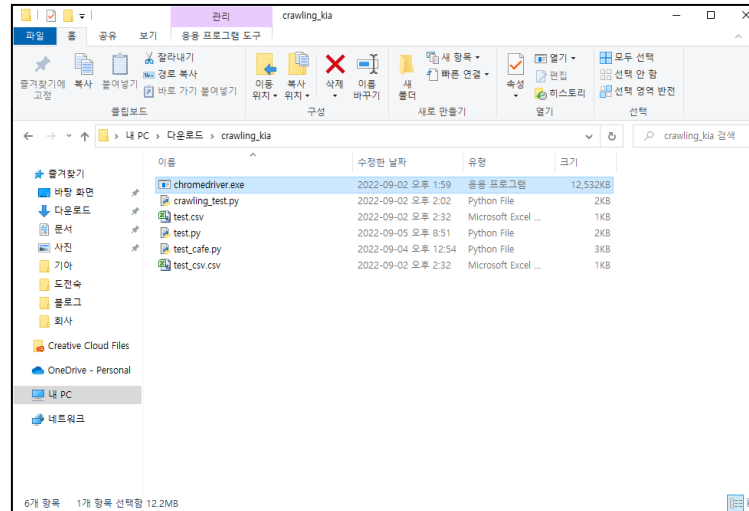
Supports Chrome version 105

For more details, please see the [release notes](#).

ChromeDriver 105.0.5195.19

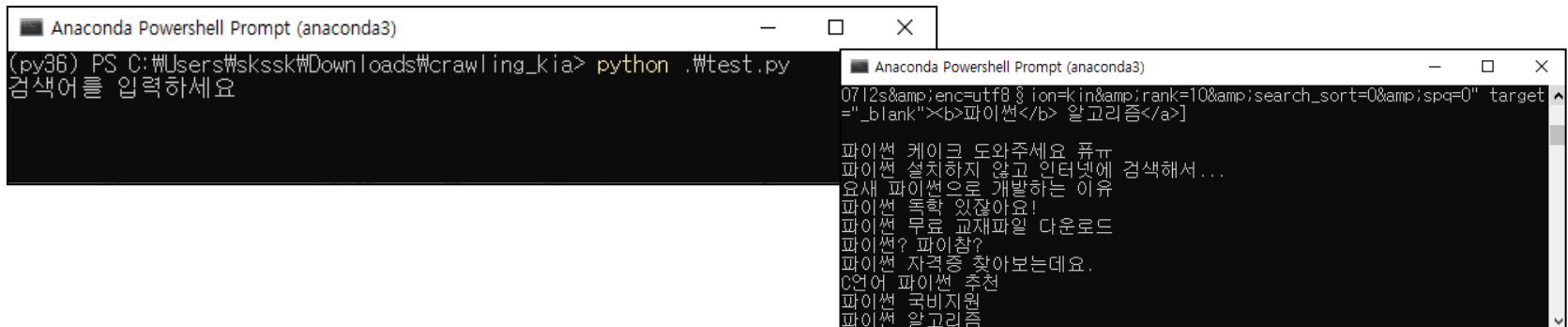
1. 환경 설정

4) chromedriver.exe 파일을 내가 작업할 폴더 내에 위치



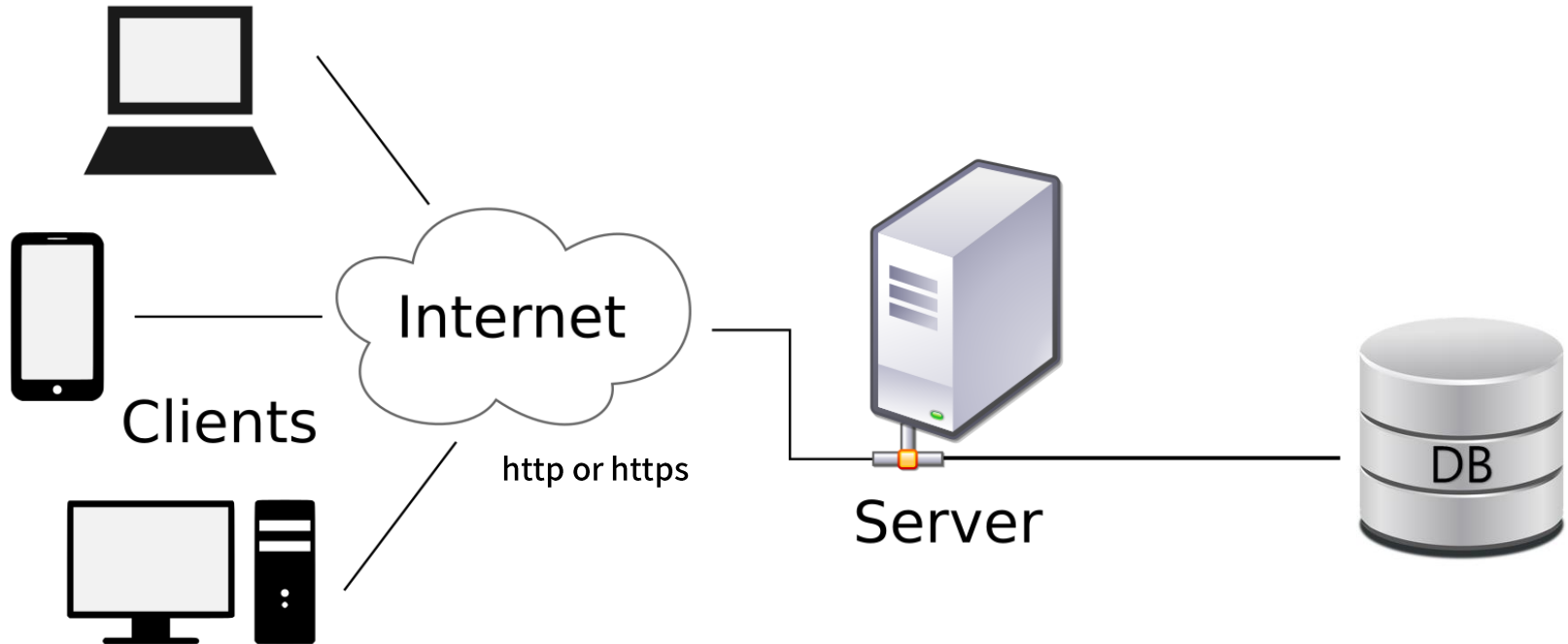
- 나눠준 test.py 실행해서 동작하면 설정 완료

(py36) PS C:\user\경로> python test.py



2. WEB / SERVER / DB 역할 및 이해

- 유저의 요청에 따른 데이터의 흐름



Front
Server Side

- Html
- CSS
- ANDROID
- Object C
- JavaScript
- PHP

Back
Server

- JAVA
- PYTHON
- Node.js
- C++
- ...

DB SQL

- MYSQL
- Oracle SQL
- Maria DB
- ...

2. WEB / SERVER / DB 역할 및 이해

- 요청과 응답

```
test.py 4 ●
C: > Users > skssk > Downloads > crawling_kia > test.py > ...

4  import numpy as np
5  import os
6  import time
7  import urllib.parse
8  from bs4 import BeautifulSoup
9
10
11
12  if __name__ == "__main__" :
13
14
15      url = 'https://kin.naver.com/search/list.nhn?query='
16
17      search = input("검색어를 입력하세요")
18
19      url = url+urllib.parse.quote_plus(search)
20
21      response = requests.get(url)
22      data = {}
23      f = open('test_csv.csv', 'w', encoding='utf-8', newline='')
24      wr = csv.writer(f)
25      i=1
26
27      if response.status_code == 200:
28          html = response.text
29          soup = BeautifulSoup(html, 'html.parser')
30          ul = soup.select_one('ul.basic1')
31
32
33          titles = ul.select('li > dl > dt > a')
34          div = soup.select_one('div.option_search')
35          test_sl = soup.select_one('#s_content > div.section > ul > li:nth-child(1) > dl > dt > a')
36
37          #print(test_sl)
38          #print(test_sl.get_text())
39          print(ul.get_text())
40          print(titles)
41
```

[정보]

100 : 처리 중

[성공]

200 : 요청에 따른 응답 성공

[리다이렉션]

300 : 응답을 받고 난 다음 수행함.

[클라이언트 오류]

400 : 잘못된 요청

401 : 접근 권한 X

403 : 접근 금지

404 : 요청을 잘못함(ex. url 틀림)

[서버 오류]

500 : 내부 서버 오류(ex. 비정상 종료)

503 : 서버 과부화(일시적)

504 : 시간 초과

→ requests를 이용해 서버를 호출하고 그 결과를 받아 처리한다.

2. WEB / SERVER / DB 역할 및 이해

- 정적 화면 / 동적 화면

카테고리 탭

The screenshot shows the Naver homepage with the following elements:

- Header:** Naver logo, search bar, and user account links (네이버를 시작페이지로, 중나어데이터, 해피민).
- Category Bar:** A horizontal bar with icons and labels for different content categories: 메일, 카페, 블로그, 지식iN, 쇼핑, 쇼핑 LIVE, Pay, TV, 사진, 뉴스, 증권, 부동산, 지도, VIBE, 도서, 웹툰. This bar is highlighted with a red box.
- Main Banner:** A large banner for "전통무협의 귀환 GRAND OPEN" featuring a character from a martial arts drama.
- News Section:** A section titled "뉴스스탠드" with a grid of news sources including 연합뉴스, 한국경제, 세계일보, JJJI.COM, OlymNews, 데일리안, NEWSIS, 한국일보, 한국경제TV, ZDNET Korea, Korea JoongAng Daily, The Korea Herald, 프리시안, sportalkorea, KBS WORLD, 일요신문, SBS, MBC, and others.
- Right Sidebar:** Contains a login section (NAVER 로그인), a COVID-19 update (이슈) about the 19th case, and a car insurance advertisement (9~10월 자동차보험 만기라면?).
- Footer:** A section titled "오늘 읽을만한 글" (Articles to read today) with a grid of article thumbnails. This section is also highlighted with a red box.

블로그 캐리셀

3. Crawling 설명

- robots.txt

크롤링이 가능한 부분과 불가능한 부분을 명시해 놓은 경고장.
물리적으로 막을 수는 없으나 대항력을 가짐.

<https://www.longblack.co/robots.txt>

```
← → ↻ 🏠 🔒 longblack.co/robots.txt

User-agent: *
Allow: /
Allow: /note
Allow: /note/*
Allow: /article
Allow: /article/*

User-agent: AhrefsBot
Disallow: /

User-agent: Cloudfind
Disallow: /

User-agent: dotbot
Disallow: /

User-agent: BLEXBot
Disallow: /

User-agent: SemrushBot
Disallow: /

User-Agent: MJ12bot
Disallow: /
```

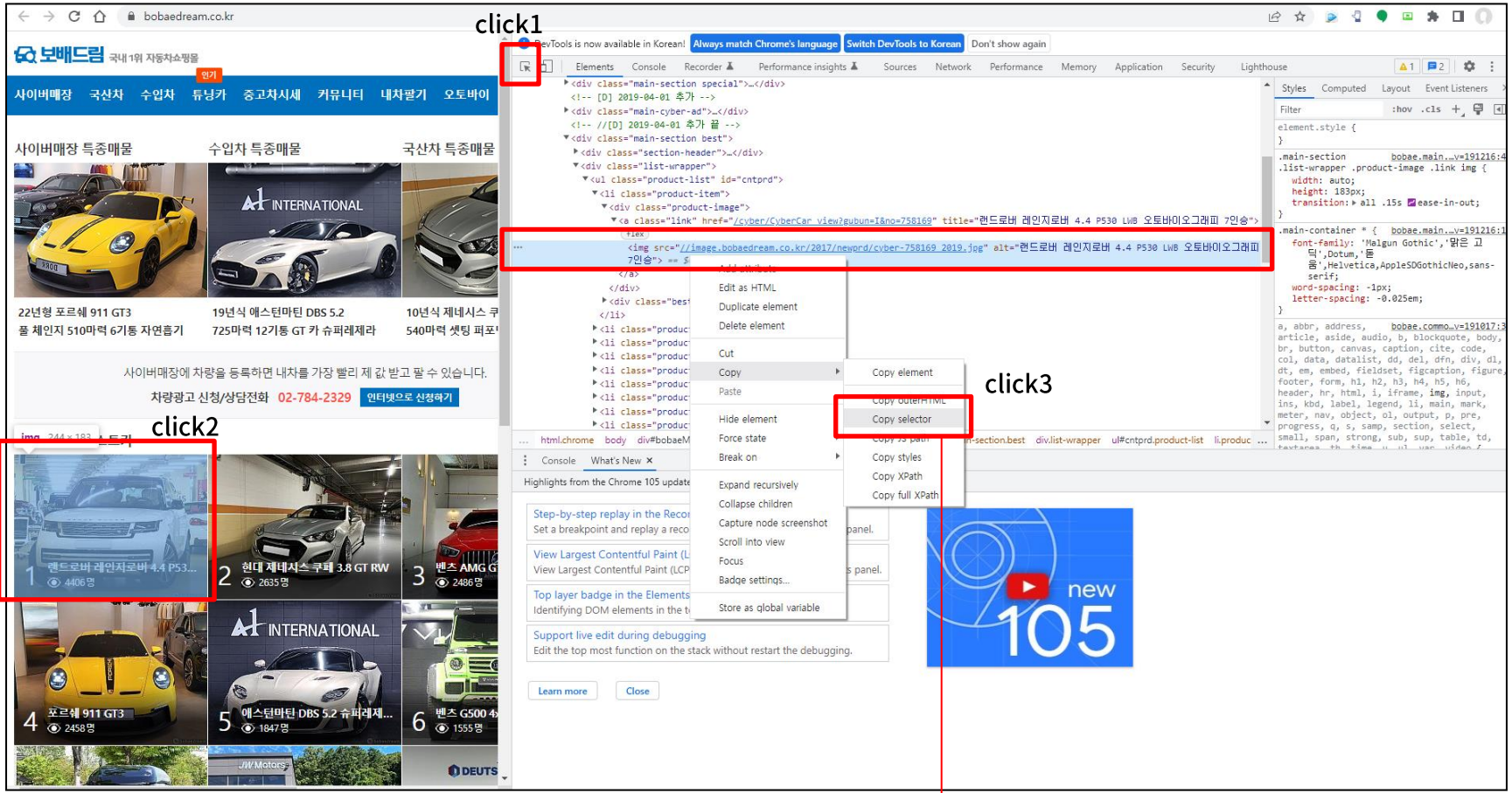
- user-agent: 규칙 적용 가능한 로봇
- allow: Crawling 가능한 경로
- disallow: Crawling 제한되는 경로
- sitemap: 사이트맵의 전체 url, 사이트 내에 정보를 제공하는 파일. (<https://tiredoctor.tistory.com/sitemap>)

```
/note
= /note.html
= /note/test.html
= /node.php?id=anything
```

```
/note/
= /note/
= /note/test.html
```

3. Crawling 설명

- Crawling 외부에서 접근하기
- 개발자 도구(키보드: Fn+F12)

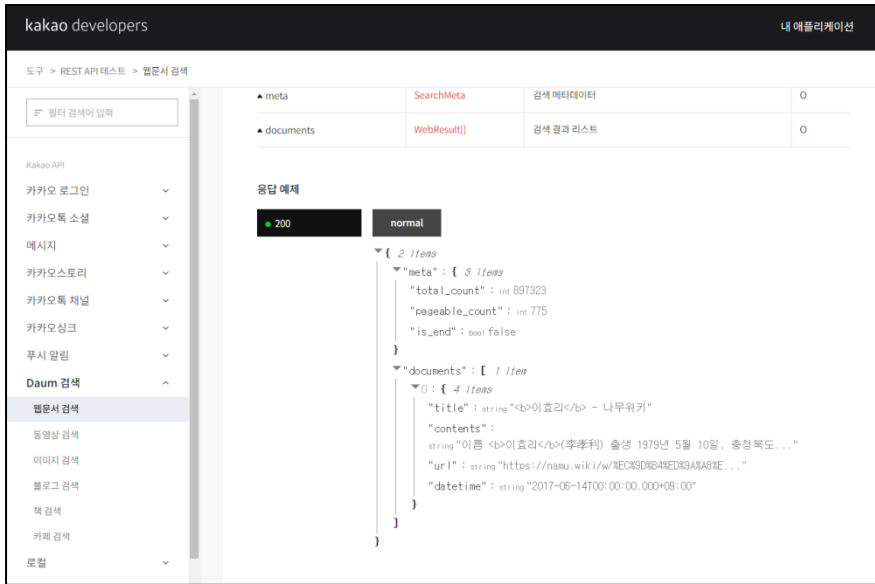


경로: #cntprd > li:nth-child(1) > div.product-image > a > img

→ BeautifulSoup, Selenium 2개를 조합해서 html 데이터를 정리정돈, 다양한 웹사이트에 적용 가능

3. Crawling 설명

- Crawling 내부 API이용(권장)



→ 카카오 개발자
<https://developers.kakao.com/>

네이버 개발자 ←
<https://developers.naver.com>

블로그
뉴스
책
성인 검색어 판별
백과사전
영화
카페글
지식iN
지역
오터변환
웹문서
이미지
쇼핑

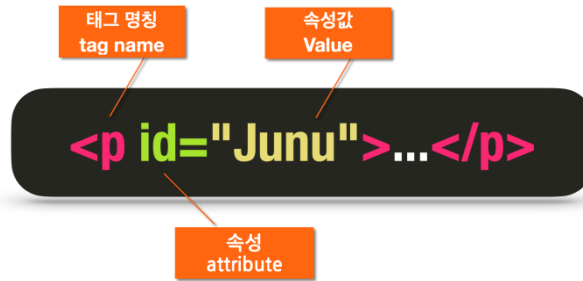
Python

```
# 네이버 검색 API에는 블로그를 비롯 전자자료까지 호출방법이 동일하므로 blog검색만 대표로 예제를 올렸습니다.
# 네이버 검색 Open API 예제 - 블로그 검색
import os
import sys
import urllib.request
client_id = "YOUR_CLIENT_ID"
client_secret = "YOUR_CLIENT_SECRET"
encText = urllib.parse.quote("검색할 단어")
url = "https://openapi.naver.com/v1/search/blog?query=" + encText # json 결과
# url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # xml 결과
request = urllib.request.Request(url)
request.add_header("X-Naver-Client-Id",client_id)
request.add_header("X-Naver-Client-Secret",client_secret)
response = urllib.request.urlopen(request)
rescode = response.getcode()
if(rescode==200):
    response_body = response.read()
    print(response_body.decode('utf-8'))
else:
    print("Error Code:" + rescode)
```

→ 각 회사에서 제공하는 API를 통해 합법적으로 이용. 해당 플랫폼만 이용 가능

3. Crawling 설명

- BeautifulSoup의 HTML 태그/클래스/아이디 선택자 이용 [find / find_all / select / select_all]



```
tag = "<p class='cl' id='id'>test</p>"
soup = BeautifulSoup(tag)
```

find_all / find → 모든 태그 / 가장 첫 번째 태그

```
soup.find('p')
soup.find(class='cl')
soup.find(attr={'class':'cl'})
soup.find('p', class_='cl')
```

```
soup.find('p').name
>> p
soup.find('p').text
>> test
soup.find_all('p').text
>> [test]
```

select / select_one → 모든 태그 / 가장 첫 번째 태그

```
soup.select_one('p')
soup.select_one('.cl')
soup.select_one('p.cl')
soup.select_one('#id')
soup.select_one('p#id')
soup.select_one('p.cl#id')
```

```
soup.select_one('p').text
>> test
soup.select('p').text
>> [test]
```

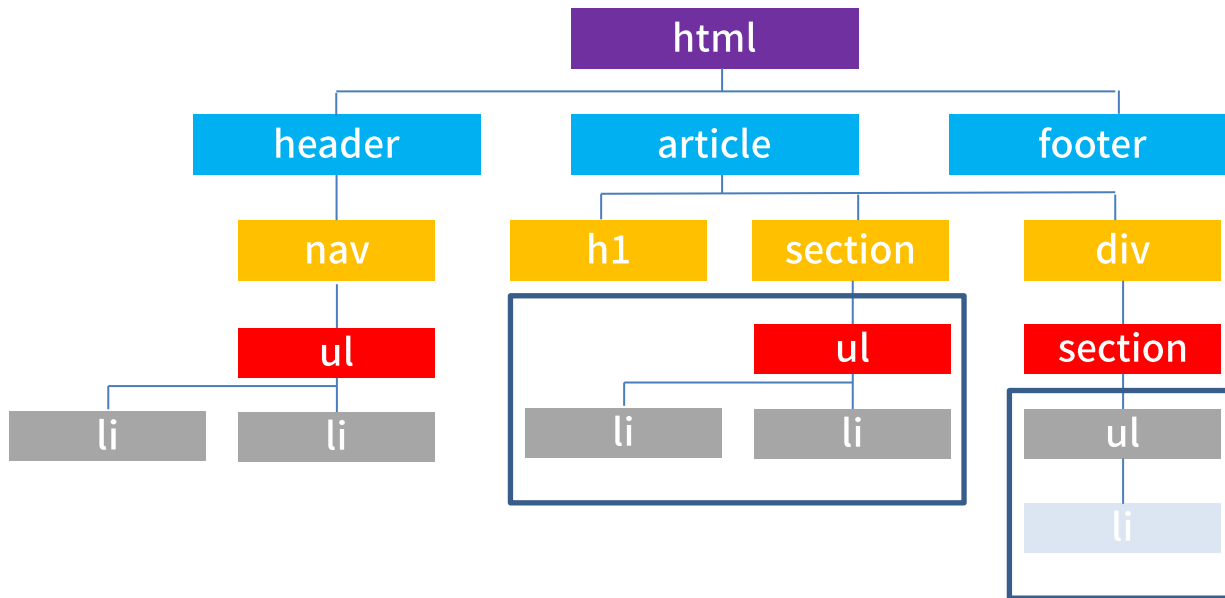
3. Crawling 설명

- BeautifulSoup의 find와 select 차이: 하위 태그 접근 형태가 다름

```
soup.find('div').find('ul')  
soup.select_one('div > ul')
```

- 패밀리트리 선택자 접근(하위 선택자)

```
article section ul
```

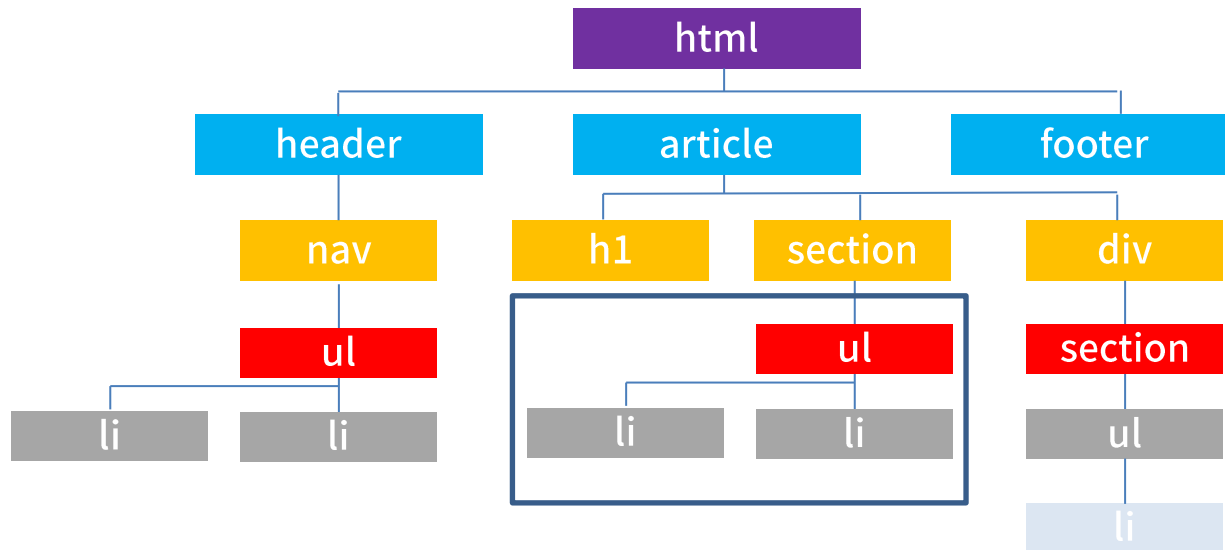


3. Crawling 설명

- 패밀리트리 선택자 접근(자식 선택자)

article > section ul

1개의 section아래 자식 ul을 찾는다.

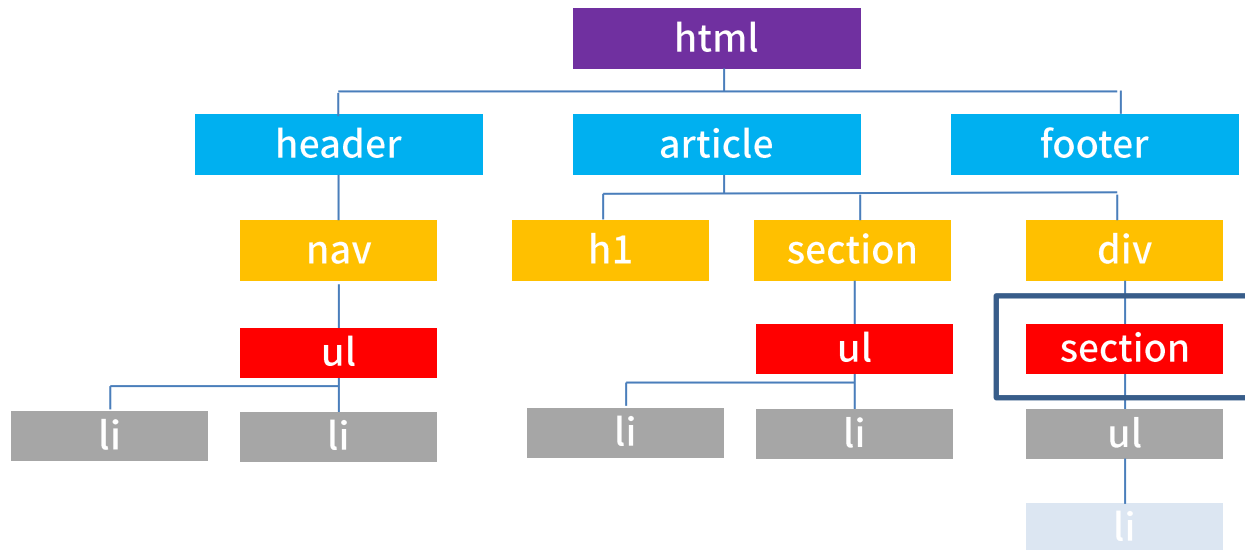


3. Crawling 설명

- 패밀리트리 선택자 접근(인접 선택자)

`h1+section+div section`

같은 층의 형제/자매 관계를 이용해 section 에 접근함



3. Crawling 설명

- Selenium의 HTML 접근 및 동적 제어 함수

```
webdriver.Chrome("경로")
```

다운 받은 크롬 드라이버를 오픈 한다.

```
driver.implicitly_wait(10)  
time.sleep(10)
```

로딩 되는 동안은 크롤링 불가, 로딩이 완료될 때까지 기다리는 함수 (사람으로 인지하기 위해서도 사용함)
implicitly: 10초 안에 로딩이 완료되면 넘어감
time.sleep: 10초를 기다림

```
driver.get("url")
```

크롬 드라이버가 해당하는 url로 이동하게 함.

```
driver.find_element_by_xpath('//*[@@="id"]').send_keys("입력하고자 하는 키워드")
```

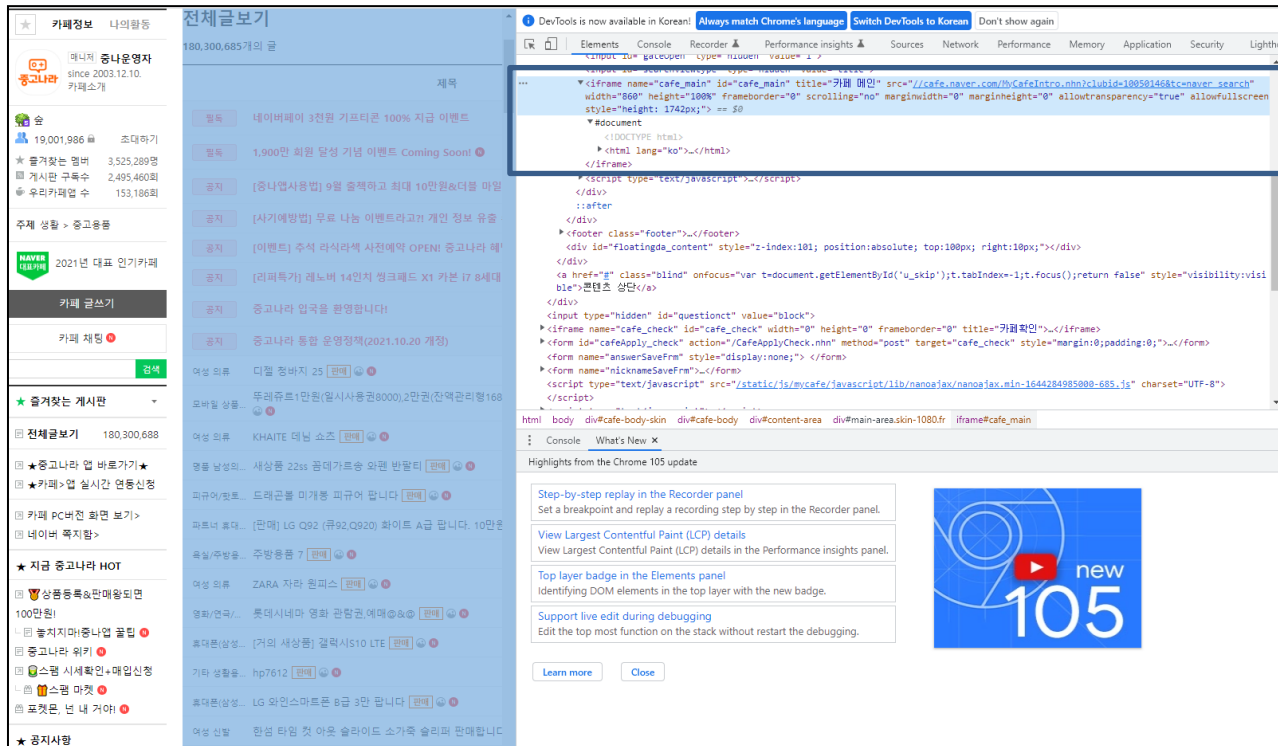
id로 접근한 입력창에 원하는 키워드를 넣음

```
driver.find_element_by_xpath('//*[@@="id"]').click()
```

Id로 접근한 버튼을 클릭

4. 이슈

- **iframe**: iframe은 내부에 페이지를 담을 수 있다. 그래서 BeautifulSoup로 접근이 불가



```
driver.switch_to.frame("cafe_main")
```

프레임을 변경해 주어야 함.

4. 이슈

- 로봇으로 인지되어 접근이 금지되었을 때(일반적으로 urlopen())이 실행되지 않음.)

```
url = f"url"
req = requests.get(url, header={'User-agent': 'mozilla/5.0'})
```

url 앞에 f 붙여 주기: f-string 문자열로 연결하는 기능

→ <https://docs.python.org/ko/3/tutorial/inputoutput.html#tut-f-strings>

header 추가: 위의 내용만 추가하면 되나 구체적으로 링크에 접속해 그림의 내용을 적어주면 된다.

→ <https://www.useragentstring.com/>

*User-Agent는 [사용자를 대표하는 컴퓨터와 브라우저 등을 나타내는 정보]

The screenshot shows the 'User Agent String explained' section of the User Agent String.Com website. It displays the user agent string: `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36`. Below this, a table breaks down the string into its components:

Component	Description
Chrome 105.0.0.0	
Mozilla	MozillaProductSlice. Claims to be a Mozilla based user agent, which is only true for Gecko browsers like Firefox and Netscape. For all other user agents it means 'Mozilla-compatible'. In modern browsers, this is only used for historical reasons. It has no real meaning anymore.
5.0	Mozilla version
Windows NT 10.0	Operating System: Windows 10
Win64	(Win32 for 64-bit-Windows) API implemented on 64-bit platforms of the Windows architecture - currently AMD64 and IA64
x64	64-bit windows version
AppleWebKit	The Web Kit provides a set of core classes to display web content in windows
537.36	Web Kit build
KHTML	Open Source HTML layout engine developed by the KDE project
like Gecko	like Gecko...
Chrome	Name: Chrome
105.0.0.0	Chrome version
Safari	Based on Safari
537.36	Safari build
Description:	Free open-source web browser developed by Google. Chromium is the name of the open source project behind Google Chrome, released under the BSD license.

At the bottom, there is a link: [All Chrome user agent strings](#).