

2022 미래차 충전인프라구축운영 전문인력 양성교육
전기차 충전기 빅데이터 심화

파이썬을 활용한 웹 크롤링(Web Crawling)

2022.11



2022 미래차 충전인프라구축운영 전문인력 양성교육

키워드

• 웹 크롤러 (Web crawler)

영문-(출처: wikipedia)

" A Web crawler, sometimes called a spider or spiderbot and often shortened to crawler, is an Internet bot that systematically browses the World Wide Web, typically for the purpose of Web indexing (web spidering). "

웹 인덱싱을 위한 www를 체계적으로 탐색(수집)해나가는 것

• 파싱 (Parsing)

" Parsing, syntax analysis, or syntactic analysis is the process of analyzing a string of symbols, either in natural language, computer languages or data structures, conforming to the rules of a formal grammar. "

웹 상의 자연어, 컴퓨터 언어 등의 일련의 문자열을 분석하는 프로세스

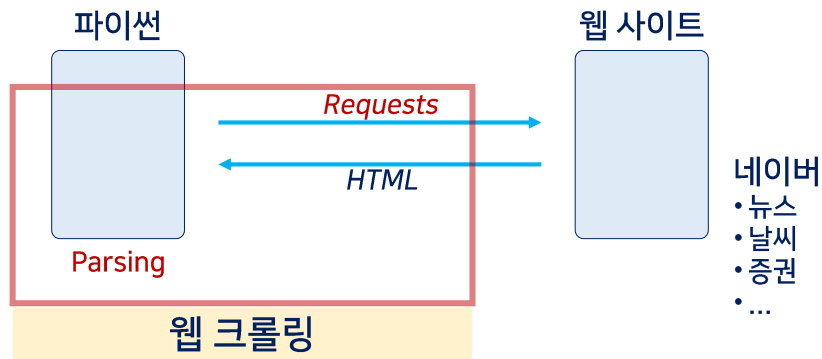
• 웹 스크래핑 (Web scraping)

" Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. "

다양한 웹사이트로부터 데이터를 추출하는 기술



웹 크롤링



웹 크롤링(Web Crawling)

- 웹 크롤러(Web Crawler)
 - 웹문서, 이미지 등을 주기적으로 수집하여 자동으로 데이터베이스화하는 프로그램
 - 웹문서의 복사본 생성
 - 검색 엔진은 생성된 데이터를 인덱싱하여 빠른 검색을 도움



BeautifulSoup4

- **BeautifulSoup**: 웹 페이지의 정보를 쉽게 스크랩할 수 있도록 기능을 제공하는 라이브러리. HTML과 XML 파일에서 데이터를 읽어내는 파이썬 라이브러리. 파서 트리를 탐색, 검색, 수정
- **Requests**: HTTP 요청을 보낼 수 있도록 기능을 제공하는 라이브러리



초 SPEED

HTML 기초 이해



영화 목록 연결 웹 문서 예시

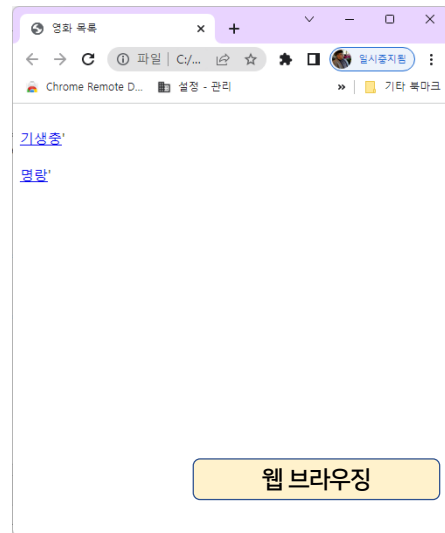


```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>영화 목록</title>
</head>
<body>

  <td id="td1" class="title">
    <br>
    <div class="tit1">
      <a href="/movie/bi/mi/basic.naver?code=161967"
        title="기생충">기생충</a>
    </div>
    <br>
    <div class="tit2">
      <a href="/movie/bi/mi/basic.naver?code=93756"
        title="명량">명량</a>
    </div>
  </td>

</body>
</html>
```

HTML 문서



영화 목록 웹 크롤링 기초

```
from bs4 import BeautifulSoup

htmlDoc = \
    """<td id="td1" class="title">
      <div class="tit3">
        <a href="/movie/bi/mi/basic.naver?code=161967"
          title="기생충">기생충</a>
      </div>
      <div class="tit4">
        <a href="/movie/bi/mi/basic.naver?code=93756"
          title="명량">명량</a>
      </div>
    </td>"""
```

HTML 문서 예시

크롤링 기본

```
# 1. 조회
def ex1():
    # html 문자열에 대해서, html 파싱
    bs = BeautifulSoup(htmlDoc, 'html.parser')

    # a 태그 출력
    tag = bs.a
    #print(tag, type(tag))

    tags = bs.findAll('a')
    for tag in tags:
        print(tag['href'])
```

bs.find('a')

Example #1 -----

```
/movie/bi/mi/basic.naver?code=161967
/movie/bi/mi/basic.naver?code=93756
```

결과 예시

영화 목록 웹 크롤링 기초

예시2

```
# 2. Attribute 값 받아오기
def ex2():
    bs = BeautifulSoup(htmlDoc, 'html.parser')

    tag = bs.td
    print(tag['class']) # ['title'] => 리스트
    print(tag['id']) # td1
    print(tag.attrs) # {'id': 'td1', 'class': ['title']} => 딕셔너리

    tag = bs.div
    # print(tag['id']) # id가 없으므로 error
```

결과2

```
Example #2 -----
['title']
td1
{'id': 'td1', 'class': ['title']}
```

예시3

2022 미래차 중전인프라 구축 운영 전문인력 양성 교육

```
# 3. Attribute 검색
def ex3():
    bs = BeautifulSoup(htmlDoc, 'html.parser')

    # div 태그 중, class가 tit3인 태그를 찾는다.
    tag = bs.find('div', attrs={'class': 'tit3'})
    print(tag)

    tag = bs.find('div')
    print(tag)

    # 없는 태그를 조회할 경우
    tag = bs.find('td', attrs={'class': 'not_exist'})
    print(tag) # None
```

결과3

```
Example #3 -----
<div class="tit3">
  <a href="/movie/bi/mi/basic.naver?code=161967" title="기생충">기생충</a>
</div>
<div class="tit3">
  <a href="/movie/bi/mi/basic.naver?code=161967" title="기생충">기생충</a>
</div>
None
```

영화 목록 웹 크롤링 기초

예시4

```
# 4. select(), content() 메서드
def ex4():
    bs = BeautifulSoup(htmlDoc, 'html.parser')

    # CSS 처럼 셀렉터를 지정할 수 있다.
    tag = bs.select("td div a")[0]
    print(tag)

    text = tag.contents[0]
    print(text)
```

결과4

```
Example #4 -----
<a href="/movie/bi/mi/basic.naver?code=161967" title="기생충">기생충</a>
기생충
```

예시5

2022 미래차 중전인프라 구축 운영 전문인력 양성 교육

```
# 5. extract() 메서드
def ex5():
    bs = BeautifulSoup(htmlDoc, 'html.parser')
    tag = bs.select("td")[0]
    print(tag)

    # div요소를 제거
    div_elements = tag.find_all("div")
    for div in div_elements:
        div.extract()

    print(tag)
```

결과5

```
Example #5 -----
<td class="title" id="td1">
  <div class="tit3">
    <a href="/movie/bi/mi/basic.naver?code=161967" title="기생충">기생충</a>
  </div>
  <div class="tit4">
    <a href="/movie/bi/mi/basic.naver?code=93756" title="명량">명량</a>
  </div>
</td>
<td class="title" id="td1">
  '
</td>
```

음원 순위 웹 크롤링

https://music.bugs.co.kr/chart?wl_ref=M_left_02_01

F12

마우스 우클릭 → '검사'

The screenshot shows the Bugs! music chart website. The chart table lists the top 10 songs as of 2022.10.30 01:00. The first song is 'I love' by (여자)아이들 (GIRL'S GENERATION) from the album 'Nxde'. The DevTools window is open, showing the HTML structure of the chart table. The table is a

순위	곡	아티스트	앨범
1	Nxde	(여자)아이들	I love
2	ANTIFRAGILE	LE SSERAFIM (르세라핌)	ANTIFRAGILE
3	사건의 지평선	윤하 (Younha)	YOUNHA 6th
4	Hype Boy	NewJeans	NewJeans 1st

음원 순위 웹 크롤링

2022 미래사 중원인프라 구축용 전문인력 양성교육

The screenshot shows the Bugs! music chart website with DevTools open. The first song, 'I love' by (여자)아이들, is highlighted. The HTML structure for the first row is shown, with specific tags and attributes highlighted for the first song.

tag: **td, class: check**

tag: **input, attribute: title**

tag: **p, class: artist**

tag: **a, attribute: title**

음원 순위 웹 크롤링

https://music.bugs.co.kr/chart?wl_ref=M_left_02_01

```

from urllib.request import urlopen
from bs4 import BeautifulSoup

url = "https://music.bugs.co.kr/chart?wl_ref=M_left_02_01"
html = urlopen(url)
soup = BeautifulSoup(html.read(), 'html.parser')

# td 태그에 check라는 class가 있는 td 태그를 모두 가져옴
htmlTag = 'td'
htmlClass = 'check'
musics = soup.find_all(htmlTag, htmlClass)

htmlTag = 'p'
htmlClass = 'artist'
artists = soup.find_all(htmlTag, htmlClass)

# musics의 각 태그들에 대해서
for i, music in enumerate(musics):
    # input 태그안에 title 속성값을 parsing한다.
    itemTag = 'input'
    itemAttri = 'title'
    musicTitle = music.find(itemTag)[itemAttri]
    artistTag = 'a'
    artistTitle = 'title'
    artistName = artists[i].find(artistTag)[artistTitle]
    print("{}위: 아티스트: {} 곡: {}".format(i+1, artistName, musicTitle))
    # print("{}위: {}".format(i+1, music.find('input')['title']))

```

```

C:\Users\mjkwon\AppData\Local\Programs\Python\Python39\python.exe C:/BigDATA/Example/Crawling/musicRank.py
1위: 아티스트: (여자)아이들 곡: Nxde
2위: 아티스트: LE SSERAFIM (르세라핌) 곡: ANTIFRAGILE
3위: 아티스트: 윤하(Younha/ユンナ) 곡: 사건의 지평선
4위: 아티스트: NewJeans 곡: Hype Boy
5위: 아티스트: IVE (아이브) 곡: After LIKE
6위: 아티스트: 지코 곡: 새뱅 (Prod. ZICO) (Feat. 호미들)
7위: 아티스트: NewJeans 곡: Attention
8위: 아티스트: Crush 곡: Rush Hour (Feat. j-hope of BTS)
9위: 아티스트: BLACKPINK 곡: Shut Down
10위: 아티스트: 테이(Te1) 곡: Monologue

```



2022 미래차 중전인프라 구축운영 전문인력 양성교육

웹 크롤링 결과 저장하기

CSV 저장 예시 코드

```

import csv

csvFile = open('musicRank.csv', 'wt')
try:
    writer = csv.writer(csvFile)
    writer.writerow(('순위', '아티스트', '곡'))
    for i, music in enumerate(musics):
        itemTag = 'input'
        itemAttri = 'title'
        musicTitle = music.find(itemTag)[itemAttri]
        artistTag = 'a'
        artistTitle = 'title'
        artistName = artists[i].find(artistTag)[artistTitle]
        writer.writerow((i+1, artistName, musicTitle))
finally:
    csvFile.close()

```

CSV 저장 결과 예시

순위	아티스트	곡
1	(여자)아이들	Nxde
2	LE SSERAFIM (르세라핌)	ANTIFRAGILE
3	IVE (아이브)	After LIKE
4	윤하(Younha/ユンナ)	사건의 지평선
5	NewJeans	Hype Boy
6	테이(Te1)	Monologue
7	BLACKPINK	Shut Down
8	21학번	스티커 사진
9	지코	새뱅 (Prod. ZICO) (Feat. 호미들)
10	이찬혁	파노라마



웹 크롤링 결과 저장하기

CSV 저장 예시 코드

줄 바꿈 처리 문제

```
file = 'musicRank.csv'
header = ('순위', '아티스트', '곡')
csvFile = open(file, 'wt')
try:
    with open(file, 'w', newline='') as resultFile:
        writer = csv.writer(resultFile, delimiter=',')
        writer.writerow(header)
        for i, music in enumerate(musics):
            itemTag = 'input'
            itemAttri = 'title'
            musicTitle = music.find(itemTag)[itemAttri]
            artistTag = 'a'
            artistTitle = 'title'
            artistName = artists[i].find(artistTag)[artistTitle]
            writer.writerow((i+1, artistName, musicTitle))
finally:
    csvFile.close()
```

CSV 저장 결과 예시

순위	아티스트	곡
1	(여자)아이들	Nixde
2	LE SSERAFIM (르세라핌)	ANTIFRAGILE
3	IVE (아이브)	After LIKE
4	윤하(Younha/ユンナ)	사건의 지평선
5	NewJeans	Hype Boy
6	태이(Tel)	Monologue
7	BLACKPINK	Shut Down
8	21학번	스티커 사진
9	9지크	새벽 (Prod. ZICO) (Feat. 호미들)
10	이장혁	화노라만
11	NewJeans	Attention
12	10CM	약 10CM만
13	Crush	Rush Hour (Feat. J-hope of BTS)
14	Charlie Puth(찰리 푸스)	I Don't Think That I Like Her
15	LE SSERAFIM (르세라핌)	FEARLESS
16	BEO (비오)	자각지심 (Feat. ZICO)
17	멜로망스(MeloMance)	사랑인가 봐
18	권진아	진심이었던 사람만 배보가 돼
19	IVE (아이브)	LOVE DIVE
20	BLACKPINK	Pink Venom
21	Wilson's Get Old	Wilson's Get Old

이미지 웹 크롤링

찾기: 월봉

이미지 웹 크롤링

```
from urllib.request import urlretrieve
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen("https://finance.naver.com/item/main.naver?code=005930").read().decode('cp949')
bsObj = BeautifulSoup(html)

imageSrc = bsObj.find("img", {"id": "img_chart_area"})

imageLocation = imageSrc["src"]
urlretrieve(imageLocation, "삼성전자.jpg")
```



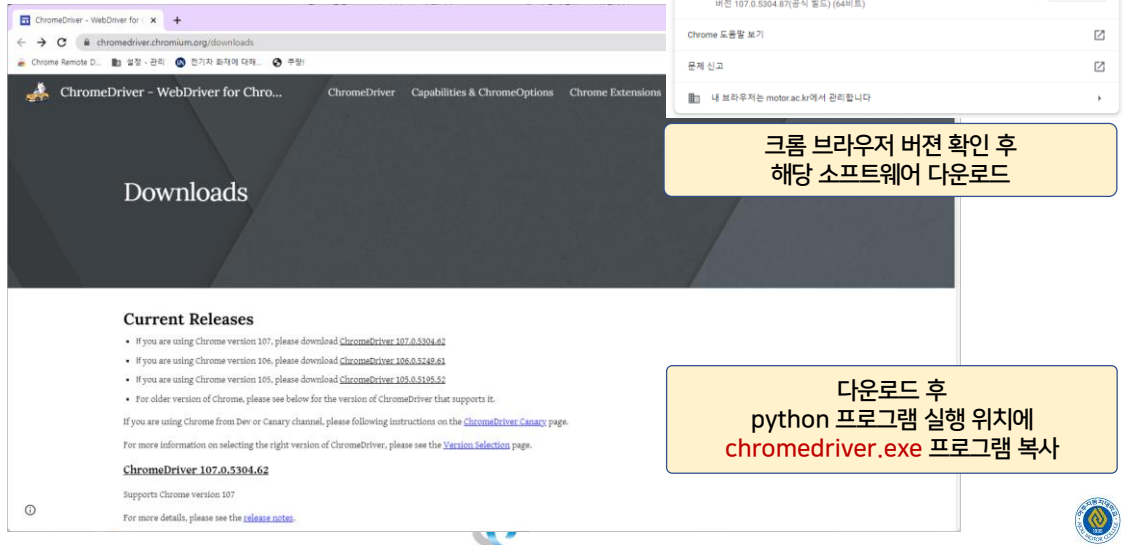
Selenium ?

- 웹 브라우저를 컨트롤하여 **웹UI를 Automation**하는 도구 중 하나
- **동적**으로 변하는 웹 페이지의 데이터들까지 설정하여 **크롤링** 가능
- 사용하는 이유
 - 자바스크립트가 **동적**으로 만든 데이터를 크롤링하기 위해
 - 사이트의 다양한 HTML 요소에 클릭, 키보드 입력 등 이벤트를 주기 위해
- 사용 용도
 - 자동으로 로그인하기
 - 메일보내기 자동화
 - 블로그 이웃새글 자동좋아요 누르기
 - 인스타그램 자동
 - ...

pip install selenium

Selenium 사용하기

• ChromeDriver 설치하기



The screenshot shows the ChromeDriver download page. A yellow box on the right says "크롬 브라우저 버전 확인 후 해당 소프트웨어 다운로드" (Check Chrome browser version and download the corresponding software). Another yellow box below it says "다운로드 후 python 프로그램 실행 위치에 chromedriver.exe 프로그램 복사" (After download, copy the chromedriver.exe program to the location where the python program is executed). The page lists current releases for Chrome versions 107, 106, and 105.

Current Releases

- If you are using Chrome version 107, please download [ChromeDriver 107.0.5304.62](#)
- If you are using Chrome version 106, please download [ChromeDriver 106.0.5249.61](#)
- If you are using Chrome version 105, please download [ChromeDriver 105.0.5195.52](#)
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

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 107.0.5304.62
Supports Chrome versions 107
For more details, please see the [release notes](#).

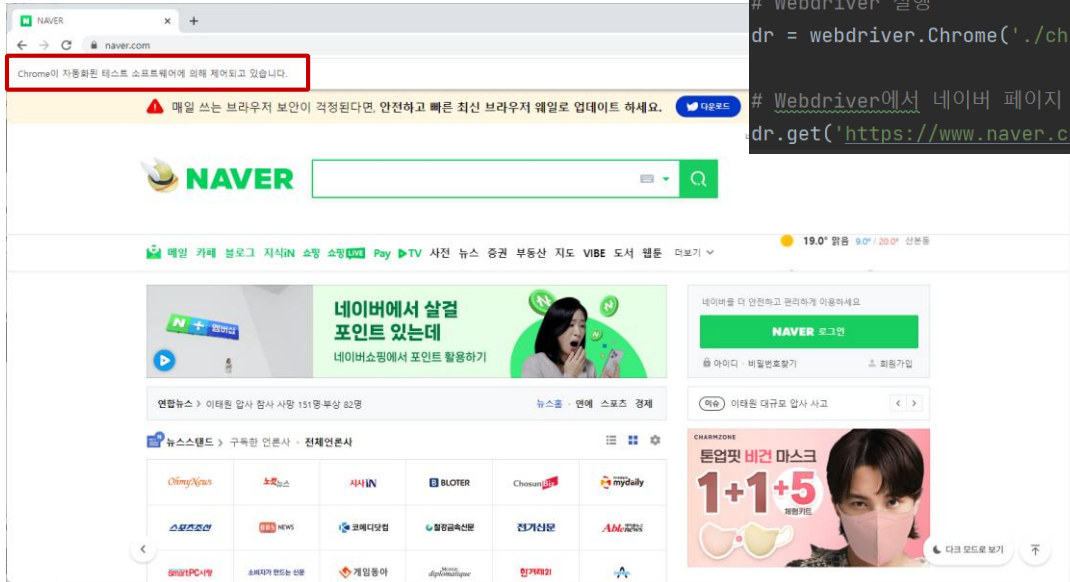
Selenium 사용하기

• 기본 구성

```
from selenium import webdriver

path = "크롬 웹드라이버 경로" # chromedriver.exe 경로
driver=webdriver.Chrome(path) #chromedriver.exe가 같은 위치에 있을 때 비어 있어도 된다.
driver.get("크롤링하려는 웹페이지 URL주소")
# 파싱할 웹 페이지 주소 입력
print(driver.title)
```

Selenium 사용하기



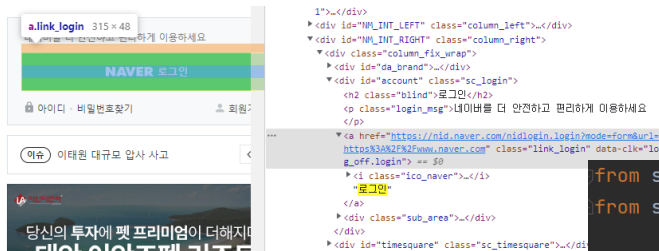
```
from selenium import webdriver

# Webdriver 실행
dr = webdriver.Chrome('./chromedriver.exe')

# Webdriver에서 네이버 페이지 접속
dr.get('https://www.naver.com/')
```



Selenium 사용하기



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

# selenium에서 사용할 웹 드라이버 절대 경로 정보
chromedriver = 'C:\BigDATA\Webdriver\chromedriver.exe'

# selenium의 webdriver에 앞서 설치한 chromedriver를 연동
driver = webdriver.Chrome(chromedriver)

# driver로 특정 페이지를 크롤링한다.
driver.get('http://www.naver.com')

element = driver.find_element(By.CLASS_NAME, 'link_login')
element.click()
```

웹 드라이브 연결
요소 찾기
동작 실행



Selenium 사용하기



네이버를 시

네이버를 더 안전하고 보

```
<input type="hidden" id="os" name="os" value disabled="disabled">
<input type="hidden" id="bid" name="bid" value disabled="disabled">
<input type="hidden" id="pkid" name="pkid" value disabled="disabled">
<input type="hidden" id="eid" name="eid" value disabled="disabled">
<input type="hidden" id="mra" name="mra" value disabled="disabled">
<div class="green_window" style="
<!-- [AU] data-atcmp-element 에 해당하는 attribute를 추가해주세요. -->
<input id="query" name="query" type="text" title="검색어 입력" maxlength="255" class="input_text" tabindex="1" accesskey="s" style="line-mode:active;" autocomplete="off" placeholder="검색어를 입력해 주세요." onlick="document.getElementById('fbm').value=1;" value data-atcmp-element>
```

검색창에 "빅데이터" 입력 후
엔터키 입력

```
from selenium.webdriver.common.keys import Keys

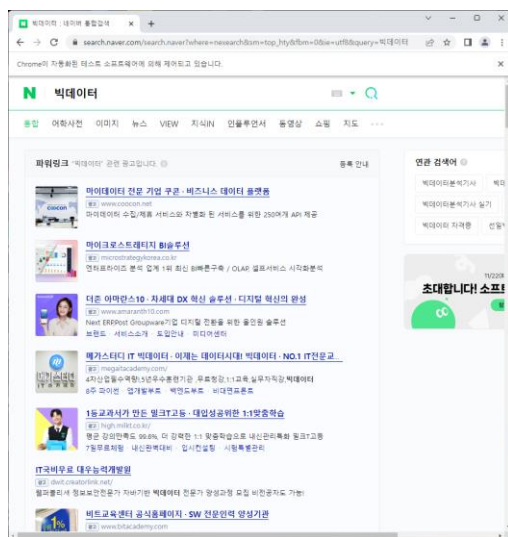
url = "http://naver.com"
browser = webdriver.Chrome()
browser.get(url)

element = browser.find_element(By.ID, 'query')

element.send_keys("빅데이터")
element.send_keys(Keys.ENTER)
```

Selenium 사용하기

페이지와 관련된 함수



```
# 뒤로가기
browser.back()

# 앞으로 가기
browser.forward()

# 새로고침
browser.refresh()

# 탭 닫기
browser.close()

# 창 닫기
browser.quit()

# 창 최대화
browser.maximize_window()

# 창 최소화
browser.minimize_window()

# 브라우저 HTML 정보 출력
print(browser.page_source)
```

Selenium 사용하기

옵션 사용 방법

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options

options = Options()

#옵션 설정하기
options.add_argument('--start-fullscreen')
driver = webdriver.Chrome("http://www.naver.com", chrome_options=options)
```

1) selenium 실행 시 웹드라이브가 뜨지 않도록 하기
options.add_argument('headless')

2) 창 크기를 조절
options.add_argument('window-size=1920x1080')

3) 전체 화면(F11)으로 조절
options.add_argument('--start-fullscreen')

4) user-agent 설정하기
user_agent = "Mozilla/5.0 (Linux; Android 9; SM-G975F)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.83
Mobile Safari/537.36"

#지정한 user-agent로 설정합니다.
options.add_argument('user-agent=' + user_agent)

5) 이미지를 로딩하지 않도록 설정
options.add_argument('--blink-settings=imagesEnabled=false')

6) 음소거
options.add_argument('--mute-audio')

브라우저 제어 등
<https://jennana.tistory.com/162>

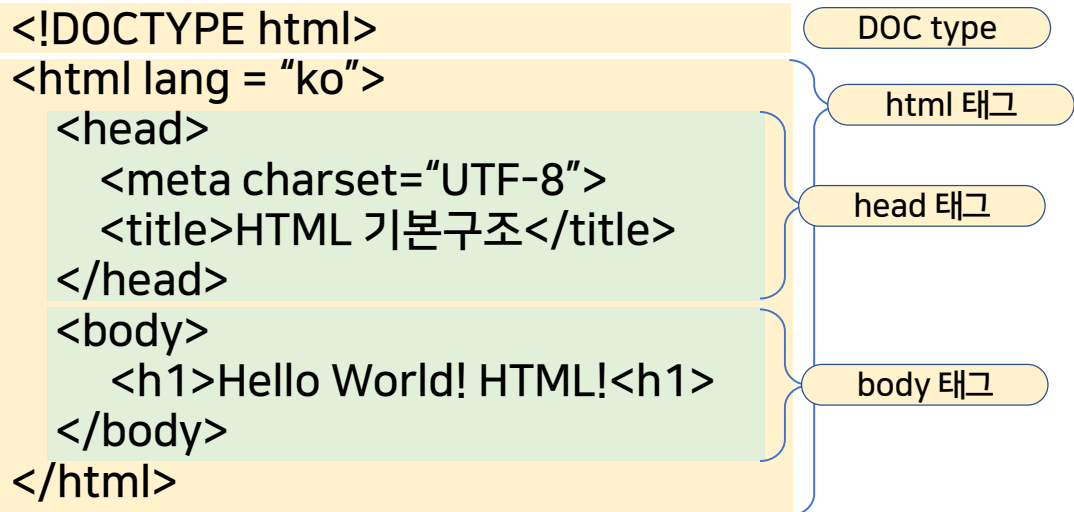


7) 시크릿 모드로 실행하게 설정
options.add_argument('incognito')



HTML 기본 이해

HTML 문서 구조



HTML의 태그란?

- HTML
 - Hypertext Markup Language
- HTML 태그
 - (사전적 의미) 어떤 표시를 하기 위해 붙인 꼬리표
 - (웹문서) 어떤 표시를 해주는 것. 대상 - 글씨 크기, 글자색, 글자 모양 등...



HTML의 태그

• 형식

왼쪽 꺾쇠 오른쪽 꺾쇠

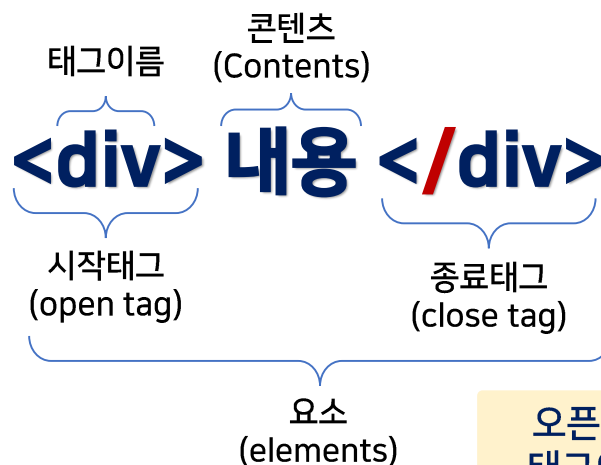
< 태그이름 >

• 종류

- 콘텐츠(Contents, 내용)가 있는 태그
 - <div>, <p>, ...
- 콘텐츠가 없는 태그
 -
, , <meta>...



HTML의 태그 - 콘텐츠가 있는 태그 형식



오픈태그와 종료태그의
태그이름은 동일해야 함



HTML의 태그 - 콘텐츠가 있는 태그 형식

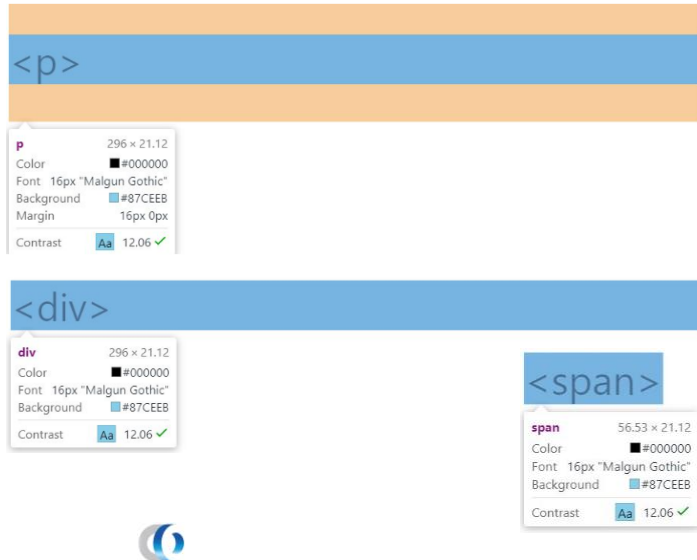
- <p>: 문단 분량의 content
+ 상하 margin이 있음

paragraph

division

- <div>: block level element(블록 라벨 엘리먼트) - 화면 전체에 영향. content에 해당하는 <div>라는 텍스트가 행 자체를 차지
- : inline element(인라인 엘리먼트)

span



HTML의 태그 - 콘텐츠가 없는 태그 형식

- br 태그

빈 태그
(empty tag)

• br : break - 줄 바꿈

HTML의 태그 - 콘텐츠가 없는 태그 형식

• meta 태그

1) 검색 엔진을 위한 키워드(keyword)를 정의하는 예제

```
<meta name="keyword" content="HTML, meta, tag, element, reference">
```

2) 웹 페이지에 대한 설명(description)을 정의하는 예제

```
<meta name="description" content="HTML meta tag page">
```

3) 문서의 저자(author)를 정의하는 예제

```
<meta name="author" content="TCPSchool">
```

4) 5초 뒤에 다른 페이지로 리다이렉트(redirect)시키는 예제

```
<meta http-equiv="refresh" content="5;url=http://www.tcpschool.com">
```

5) 모든 장치에서 웹 사이트가 잘 보이도록 뷰포트(viewport)를 설정하는 예제

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- 해당 문서에 대한 정보인 메타 데이터(metadata)를 정의
- <meta>요소는 <base>, <link>, <script>, <style>, <title> 요소와 같은 다른 메타데이터 관련 요소들이 나 타낼 수 없는 다양한 종류의 메타데이터를 제공할 때 사용되며, 브라우저나 검색엔진, 다른 웹 서비스에서 사용
- <meta>요소는 언제나 <head> 요소 내부에 위치



HTML의 주요태그

태그명	설명
<HTML></HTML>	해당 문서를 HTML 라는 선언을 하는 태그
<HEAD></HEAD>	해당 문서의 다양한 정보를 알려주는 태그이다.
<TITLE>타이틀 명</TITLE>	브라우저에서의 해당 문서의 제목을 알려주는 태그
<P>내용</P>	단락을 정하는 태그
링크명	다른 문서로 이동을 하기 위한 태그로 HTML 태그 중 가장 중요한 태그
<TABLE></TABLE>	Table을 정의하는 태그
<TR></TR>	Table 태그 내행을 정하는 태그
<TD></TD>	Table 태그 내열을 정하는 태그

- H1 ~ H6: 제목 태그, h1 ~ h6 순서대로 크기가 작아짐
- ul: 순서가 없는 목록 표시 (unordered list)
- li: 각 항목 넣은 태그 (list item)
- ol: 순서가 있는 태그 (ordered list)



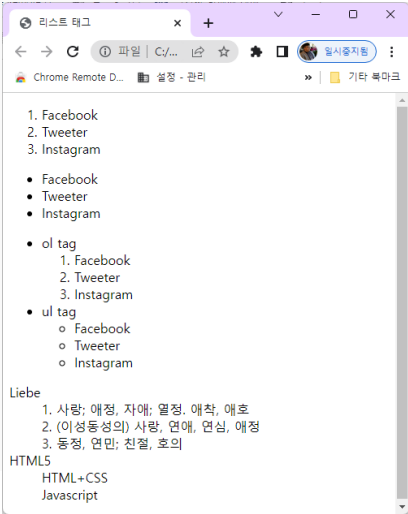
exHtml.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<title>리스트 태그</title>
<body>

  <!-- 순서가 있는 목록 태그 -->
  <ol>
    <li>Facebook</li>
    <!-- 목록 요소 -->
    <li>Tweeter</li>
    <li>Instagram</li>
  </ol>

  <!-- 순서가 없는 목록 태그 -->
  <ul>
    <li>Facebook</li>
    <li>Tweeter</li>
    <li>Instagram</li>
  </ul>

  <!-- 중첩 목록 -->
  <ul>
    <li>ol tag
      <ol>
        <li>Facebook</li>
        <li>Tweeter</li>
        <li>Instagram</li>
      </ol>
    <li>ul tag
      <ul>
        <li>Facebook</li>
        <li>Tweeter</li>
        <li>Instagram</li>
      </ul>
    <li>Liebe
      <ol>
        <li>사랑; 애정, 자애; 열정. 애착, 애호
        <li>(이성동성의) 사랑, 연애, 연심, 애정
        <li>동정, 연민; 친절, 호의
      </ol>
    <li>HTML5
      <ul>
        <li>HTML+CSS
        <li>Javascript
      </ul>
    </li>
  </ul>
  </body>
</html>
```



```

    <li>Facebook</li>
    <li>Tweeter</li>
    <li>Instagram</li>
  </ol>
</li>
<li>ul tag
  <ul>
    <li>Facebook</li>
    <li>Tweeter</li>
    <li>Instagram</li>
  </ul>
</li>
<li>Liebe
  <ol>
    <li>사랑; 애정, 자애; 열정. 애착, 애호
    <li>(이성동성의) 사랑, 연애, 연심, 애정
    <li>동정, 연민; 친절, 호의
  </ol>
  <li>HTML5
    <ul>
      <li>HTML+CSS
      <li>Javascript
    </ul>
  </li>
</ul>
</body>
</html>
```

참고

- <https://victorydntmd.tistory.com/245>
- <https://library.gabia.com/contents/9239/>
- <https://wikidocs.net/6660>
- <https://webnautes.tistory.com/779>
- <https://coding-kindergarten.tistory.com/24>