

# 웹 스크래핑

출처: [나도코딩](#)

# HTML

뼈대

# CSS

예쁘게

# JavaScript

살아있게

```
<html>
  <head>
    <title>홈페이지</title>
  </head>
  <body>
    <h1>어서오세요</h1>
  </body>
</html>
```

# HTML

빠대

```
<button id="search_btn"  
  type="submit"  
  title="검색"  
  class="btn_submit">
```

# XPath

Element 의 경로

특징 (id, class, text)

```
//*[@id="search_btn"]
```

전체 경로

```
/html/body/div[2]/div[1]/div[1]/div/div[3]/form/fieldset/button
```

# HTML

뼈대

```
<button id="search_btn"  
  type="submit"  
  title="검색"  
  class="btn_submit">
```

# XPath

Element 간의 관계

```
<부모>  
  <자식/>  
  <자식/>  
  <자식/>  
</부모>
```

# Chrome

개발자 도구

The screenshot shows the Naver homepage in a Chrome browser. The Chrome DevTools 'Elements' panel is open on the right, displaying the HTML structure of the page. A red box highlights the login button's HTML element, which is a link with the text '로그인' (Login). The text 'Xpath 를 쉽게' (Easily find Xpath) is written next to the highlighted element.

네이버를 시작페이지로 > | 쥘아이네이버 | 해피빈

네이버를 더 안전하고 편리하게 이용하세요

**NAVER 로그인**

아이디 · 비밀번호찾기 | 회원가입

증시 | 다우 26,469.89 ▼ 182.44 -0.68%

Elements Console Sources Network Performance Memory >> | ⚙️ | ⋮

```
<!DOCTYPE html>
<html lang="ko" data-dark="false" data-useragent="Mozilla/5.0
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.89 Safari/537.36">
  <head>_</head>
  <body>
    <div id="u_skip">_</div>
    <div id="wrap">
      <style type="text/css">_</style>
      <div id="header" role="banner">_</div>
      <div id="container" role="main">
        <div style="position:relative;width:1130px;margin:0 auto;z-index:11">_</div>
        <div id="NM_INT_LEFT" class="column_left">_</div>
        <div id="NM_INT_RIGHT" class="column_right">
          <div class="column_fix_wrap">
            <div id="da_brand"></div>
            <div id="account" class="sc_login">
              <h2 class="blind">로그인</h2>
              <p class="login_msg">네이버를 더 안전하고 편리하게 이용하세요</p>
              ...
              <a href="https://nid.naver.com/nidlogin.login?
mode=form&url=https%3A%2F%2Fwww.naver.com" class="link_login" data-clk=
"log_off.login">_</a> == $0
            <div class="sub_area">_</div>
          </div>
        <div id="timesquare" class="sc_timesquare">_</div>
      </div>
    </div>
  </body>
</html>
```

Xpath 를 쉽게

# Chrome

개발자 도구

네이버를 시작페이지로 > | 즐겨찾아내기 | 해피빈

웹툰 더보기 ⓘ 검색어 필터 설정이 필요합니다

네이버를 더 안전하고 편리하게 이용하세요

**NAVER 로그인**

아이디 · 비밀번호 찾기 회원가입

증시 | 다우 26,469.89 ▼ 182.44 -0.68%

< >

Elements Console Sources Network Performance Memory >> ⚙ ⋮

```
<!DOCTYPE html>
<html lang="ko" data-dark="false" data-useragent="Mozilla/5.0
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.89 Safari
  <head>_</head>
  <body>
    <div id="u_skip">_</div>
    <div id="wrap">
      <style type="text/css">_</style>
      <div id="header" role="banner">_</div>
      <div id="container" role="main">
        <div style="position:relative;width:1130px;margin:0 auto;z-index:11">_</div>
        <div id="NM_INT_LEFT" class="column_left">_</div>
        <div id="NM_INT_RIGHT" class="column_right">
          <div class="column_fix_wrap">
            <div id="da_brand"></div>
            <div id="account" class="sc_login">
              <h2 class="blind">로그인</h2>
              <p class="login_msg">네이버를 더 안전하고 편리하게 이용하세요</p>
              ...
              <a href="https://nid.naver.com/nidlogin.login?
mode=form&url=https%3A%2F%2Fwww.naver.com" class="link_login" data-clk=
"log_off.login">_</a> == $0
            <div class="sub_area">_</div>
          </div>
        <div id="timesquare" class="sc_timesquare">_</div>
```

Xpath 를 쉽게

# 정규식

규칙을 가진 문자열을 표현하는 식

주민등록번호

901230-1111111

. (ca.e) 하나의 문자

^(^de) 문자열의 시작

\$(se\$) 문자열의 끝

이메일 주소

[nadocoding@gmail.com](mailto:nadocoding@gmail.com)

match() 처음부터 일치하는지

search() 일치하는 게 있는지

findall() 일치하는 것 모두 리스트로

차량 번호

11가 1234

# User-Agent

어떤 페이지를 보여줄까?

근데 진짜 사람 맞아???



# Requests

웹 페이지(html) 읽어오기

빠르다

동적 웹 페이지 X

# Requests

웹 페이지(html) 읽어오기

빠르다

동적 웹 페이지 X

# Selenium

웹 페이지 자동화

느리다

동적 웹 페이지 O

# Requests

웹 페이지(html) 읽어오기

빠르다

동적 웹 페이지 X



# Selenium

웹 페이지 자동화

느리다

동적 웹 페이지 O



# BeautifulSoup

원하는 데이터 추출 (웹 스크래핑)

# Requests

주어진 url 을 통해 받아온 html 에 원하는 정보가 있을 때

문제 없겠죠?

`res.raise_for_status()`

# Selenium

로그인, 어떤 결과에 대한 필터링 등 어떤 동작을 해야 하는 경우

★ 크롬 버전에 맞는 chromedriver.exe 가 반드시 있어야 해요

# Selenium

로그인, 어떤 결과에 대한 필터링 등 어떤 동작을 해야 하는 경우

★ 크롬 버전에 맞는 `chromedriver.exe` 가 반드시 있어야 해요

`find_element(s)_by_id`

id 로 찾기

`find_element(s)_by_class_name`

class name 으로 찾기

`find_element(s)_by_link_text`

링크 text 로 찾기

`find_element(s)_by_xpath`

xpath 로 찾기

`click()`

클릭

`send_keys()` `clear()`

글자 입력

# Selenium

때로는 기다려주세요

```
try:
    elem = WebDriverWait(browser, 10)
        .until(EC.presence_of_element_located((By.XPATH, "//*[@id='content']")))
    # 성공했을 때 동작 수행
    pass
finally:
    browser.quit()
```

# Selenium

## 스크롤을 내려주세요

```
import time
interval = 2 # 2초에 한번씩 스크롤 내림

# 현재 문서 높이를 가져와서 저장
prev_height = browser.execute_script("return document.body.scrollHeight")

# 반복 수행
while True:
    # 스크롤을 가장 아래로 내림
    browser.execute_script("window.scrollTo(0, document.body.scrollHeight)")

    # 페이지 로딩 대기
    time.sleep(interval)

    # 현재 문서 높이를 가져와서 저장
    curr_height = browser.execute_script("return document.body.scrollHeight")
    if curr_height == prev_height:
        break

    prev_height = curr_height
```



# BeautifulSoup

`find`

조건에 맞는 첫 번째 element

`find_all`

조건에 맞는 모든 element 리스트로

`find_next_sibling(s)`

다음 형제 찾기

`find_previous_sibling(s)`

이전 형제 찾기

`soup["href"]`

속성

`soup.get_text()`

텍스트

# 이미지 다운로드

```
with open("파일명", "wb") as f:  
    f.write(res.content)
```

영화순위 ①

[다른 사이트 보기](#)

영화 순위

역대 관객순위

1 / 6

역대

2019

2018

2017

2016

2015

2014

2013

2012

2011



극한직업

★★★★★ 7.4  
5,763명 참여

개봉 2019.01.23.

연간 16,265,618명

누적 16,266,338명



어벤져스: 엔드...

★★★★★ 7.8  
4,596명 참여

개봉 2019.04.24.

연간 13,934,592명

누적 13,977,602명



겨울왕국 2

★★★★★ 7.4  
2,581명 참여

개봉 2019.11.21.

연간 13,369,070명

누적 13,747,792명



알라딘

★★★★★ 8.4  
1,965명 참여

개봉 2019.05.23.

연간 12,552,283명

누적 12,573,514명



기생충

★★★★★ 7.9  
8,552명 참여

개봉 2019.05.30.

연간 10,085,275명

누적 10,313,087명

# CSV

import csv

```
f = open(filename, "w", encoding="utf-8-sig", newline="")
```

	A	B	C	D	E	F	G	H	I	J	K	L
1	N	종목명	현재가	전일비	등락률	액면가	시가총액	상장주식수	외국인비율	거래량	PER	ROE
2		1 삼성전자	54,000	100	-0.18%	100	3,223,683	5,969,783	55.46	5,741,247	17.23	8.69
3		2 SK하이닉스	82,800	400	0.49%	5,000	602,786	728,002	48.16	1,873,166	38.67	4.25
4		3 삼성바이오로직스	767,000	8,000	-1.03%	2,500	507,486	66,165	10.26	66,669	182.53	4.77
5		4 NAVER	282,000	2,000	-0.70%	100	463,223	164,263	55.21	773,978	72.81	10.56
6		5 셀트리온	321,500	500	-0.16%	1,000	433,830	134,939	21.25	483,344	127.88	11.19
7		6 삼성전자우	47,000	50	-0.11%	100	386,757	822,887	88.23	545,204	15	N/A
8		7 LG화학	519,000	13,000	-2.44%	5,000	366,374	70,592	36.68	175,159	290.27	1.84

# Headless Chrome

브라우저를 띄우지 않고 동작

때로는 User-Agent 정의 필요

59 버전부터 (최신 버전이면 모두 가능)

# 막 쓰면 안돼요

무분별한 웹 크롤링 / 웹 스크래핑은 대상 서버에 부하

→ 계정 / IP 차단

데이터 사용 주의

→ 이미지, 텍스트 등 데이터 무단 활용 시 저작권 등 침해 요소, 법적 제재

robots.txt

→ 법적 효력 X, 대상 사이트의 권고



# 막 쓰면 안돼요

무분별한 웹 크롤링 / 웹 스크래핑은 대

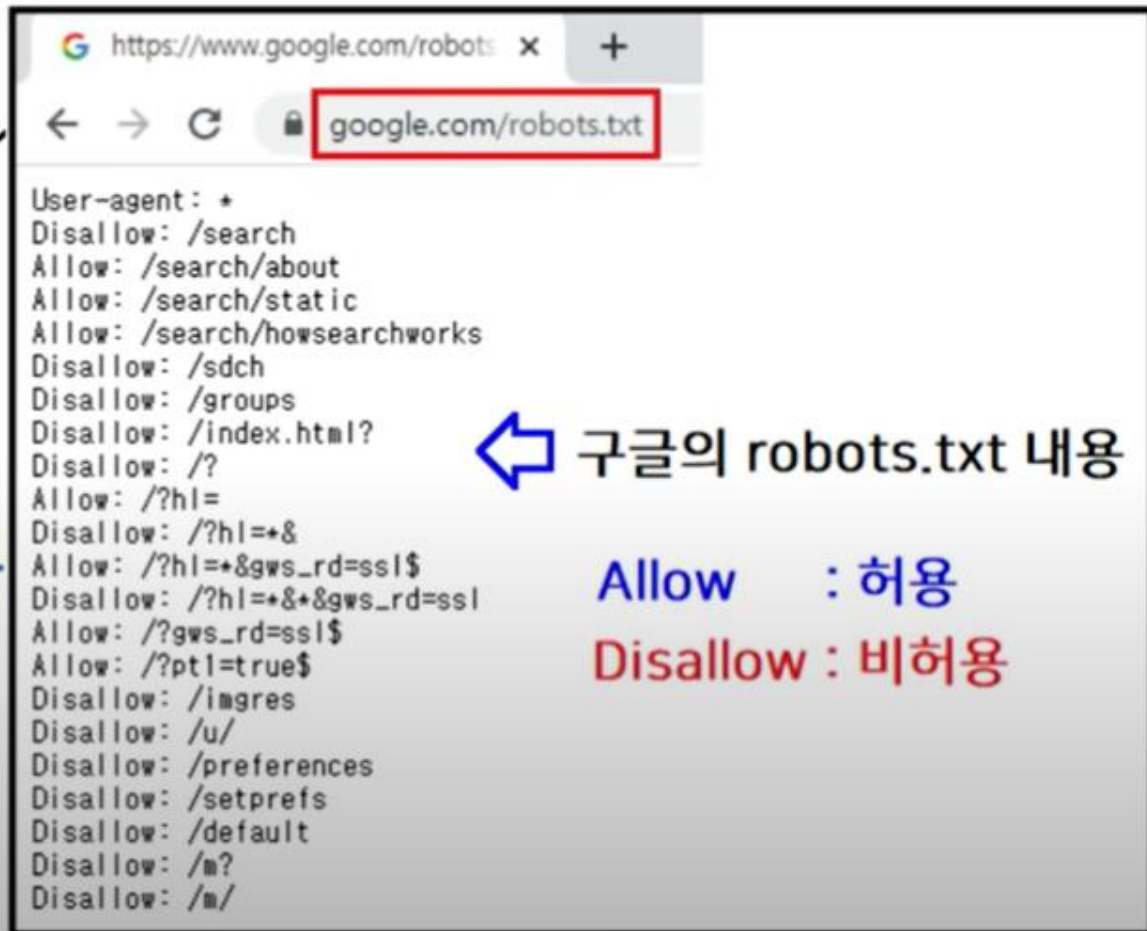
→ 계정 / IP 차단

데이터 사용 주의

→ 이미지, 텍스트 등 데이터 무단 활용

robots.txt

→ 법적 효력 X, 대상 사이트의 권고



The screenshot shows a web browser window with the address bar displaying 'https://www.google.com/robots.txt'. The page content lists various 'Allow' and 'Disallow' rules for web crawlers. A blue arrow points from the text '구글의 robots.txt 내용' to the screenshot. To the right of the screenshot, there is a legend: 'Allow : 허용' (Allow : Allow) in blue and 'Disallow : 비허용' (Disallow : Disallow) in red.

```
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: /?ptl=true$
Disallow: /imgres
Disallow: /u/
Disallow: /preferences
Disallow: /setprefs
Disallow: /default
Disallow: /m?
Disallow: /m/
```

구글의 robots.txt 내용

Allow : 허용  
Disallow : 비허용

# 숙제

대상 웹페이지를 살펴보고, 어떤 정보(데이터)가 있는지 정리해 본다.

- 주제: 해당 인물이 출연한 영상에서의 댓글들 및 조회수, 공감수를 분석한다.
- 대상 : 유튜브, 카카오tv, 네이버tv, VLIVE