

크롤링(Web Crawler)_2

웹 크롤러(Web Crawler) 개요

- 웹 크롤러란 조직적, 자동화된 방법으로 웹(Web)에서 다양한 정보를 수집하는 프로그램으로, 웹 크롤러가 하는 작업을 '웹 크롤링(Web crawling)' 또는 '스파이더링(spidering)'이라 부른다.
- 주로 Python 언어를 이용하며, 대표적인 라이브러리는 requests, beautifulsoup4, selenium가 있다.

정적 크롤링과 동적 크롤링 비교

	정적 크롤링	동적 크롤링
연속성	주소를 통한 단발적 접근	브라우저를 사용한 연속적 접근
수집 능력	수집 데이터의 한계가 존재	수집 데이터의 한계가 없음
속도	빠름	느림
라이브러리	requests, BeautifulSoup	selenium, chromedriver

1. 동적 크롤링 vs. 정적 크롤링

정적 페이지

- 언제나 접속해도 같은 내용을 보여주는 페이지를 의미 (즉 이미 작성된 코드를 그대로 클라이언트의 브라우저로 보냄)
- 클라이언트의 요청을 받은 웹서버는 추가적인 작업을 하지 않고, 응답하는 구조임

동적 페이지

- 서버의 추가적인 처리 (데이터베이스 연동 등)를 한 후에 클라이언트에게 응답하는 구조임
- 또한, 클라이언트 사용자(페이지 방문자)와 상호작용을 하면서 시시각각 페이지의 내용이 바뀌게 되기 때문에 정적 페이지와 다른 방식으로 처리가 요구됨
- requests 라이브러리를 통한 HTML response를 받아와도 (응답내용에는 정적인 HTML 코드만 포함되어 있어) 동적인 내용은 보이지 않음

1. 동적 크롤링 – 대표 라이브러리

- 동적 크롤링에 필요한 Python 라이브러리는 대표적으로 selenium이다.
- Jupyter Notebook을 활용하여 동적 사이트를 분석하고 수행

Open API

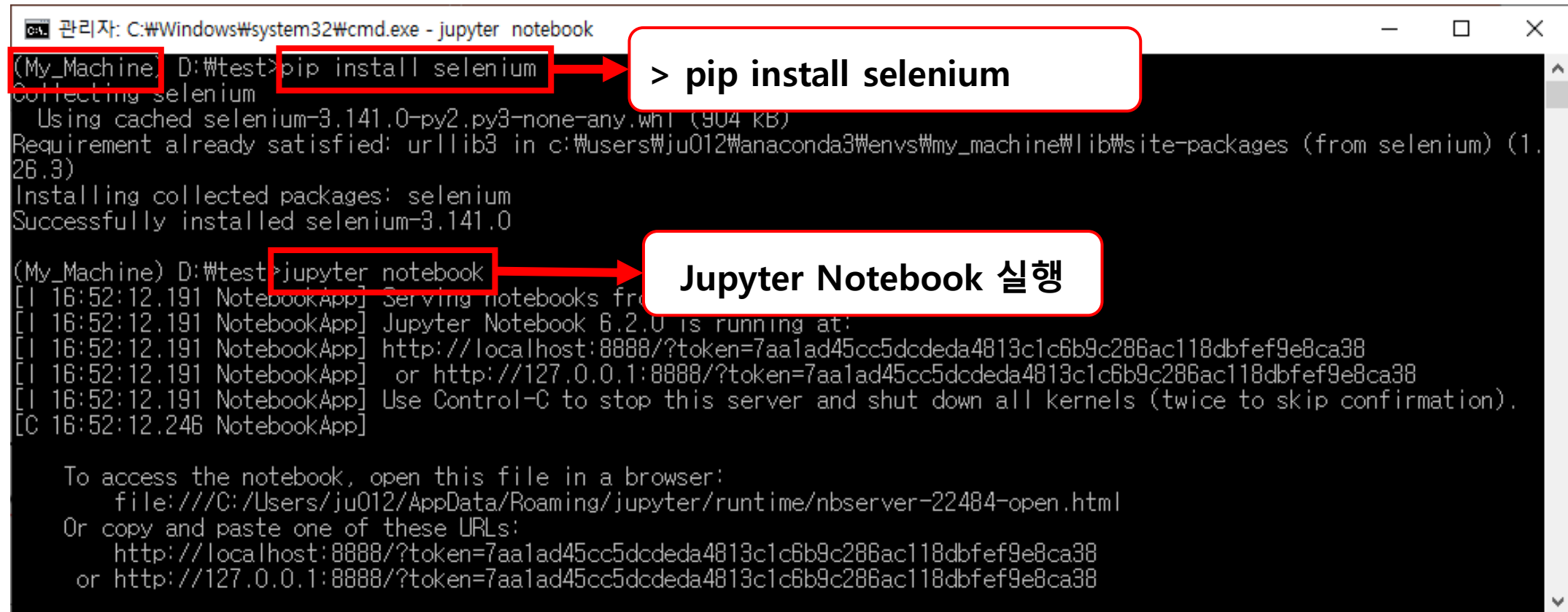
- Youtube, Facebook 등 많은 페이지들이 동적페이지로 운영되고 있으며, Open API를 제공함
- 예: 다음(www.daum.net)의 검색 결과를 크롤링 하려면, 카카오 OpenAPI를 이용

selenium

- 웹사이트 테스트를 위한 도구로 브라우저 동작을 자동화 할 수 있는 파이썬 대표적인 라이브러리로, 동적 웹 페이지 크롤링 방법으로 유용하다.
- 웹 드라이버는 크롬, 파이어폭스 등 다양한 브라우저 드라이버를 지원한다.

1-1. 동적 크롤링 - Python 라이브러리 설치(selenium)

- Jupyter Notebook을 사용하여 웹크롤링에 필요한 Python 라이브러리를 설치 (*별도 PPT 참고)
- 라이브러리 설치하는 "activate (가상머신명)"으로 활성화된 상태에서 진행



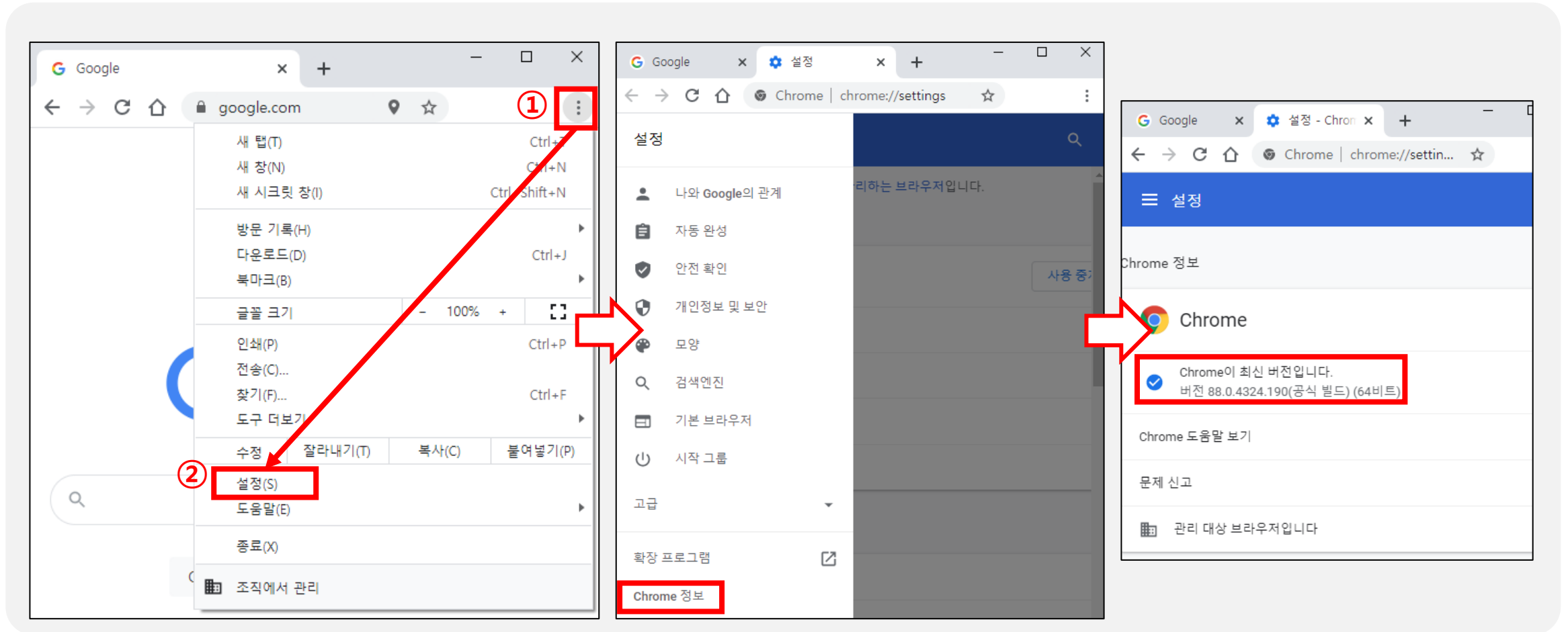
```
관리자: C:\Windows\system32\cmd.exe - jupyter notebook
(My_Machine) D:\wtest>pip install selenium
Collecting selenium
  Using cached selenium-3.141.0-py2.py3-none-any.whl (904 KB)
Requirement already satisfied: urllib3 in c:\users\ju012\anaconda3\envs\my_machine\lib\site-packages (from selenium) (1.26.3)
Installing collected packages: selenium
Successfully installed selenium-3.141.0

(My_Machine) D:\wtest>jupyter notebook
[I 16:52:12.191 NotebookApp] Serving notebooks from http://localhost:8888/
[I 16:52:12.191 NotebookApp] Jupyter Notebook 6.2.0 is running at:
[I 16:52:12.191 NotebookApp] http://localhost:8888/?token=7aa1ad45cc5dcdeda4813c1c6b9c286ac118dbfef9e8ca38
[I 16:52:12.191 NotebookApp] or http://127.0.0.1:8888/?token=7aa1ad45cc5dcdeda4813c1c6b9c286ac118dbfef9e8ca38
[I 16:52:12.191 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 16:52:12.246 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/ju012/AppData/Roaming/jupyter/runtime/nbserver-22484-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=7aa1ad45cc5dcdeda4813c1c6b9c286ac118dbfef9e8ca38
or http://127.0.0.1:8888/?token=7aa1ad45cc5dcdeda4813c1c6b9c286ac118dbfef9e8ca38
```

1-2. 동적 크롤링 – Web Driver 설치(ChromeDriver)

- ChromeDriver설치를 위해선 본인이 사용하는 크롬 버전을 확인하고 동일한 버전을 설치해야함
- Web Driver(ChromeDriver) 다운로드: <https://chromedriver.chromium.org/downloads>



1-3. 동적 크롤링 – Web Driver 설치(ChromeDriver)

- 본인의 버전, 운영체제에 맞는 chromedriver.exe를 다운로드하여 사용하는 폴더에 이동

ChromeDriver - WebDriver for Chrome

Search this site

- CHROMEDRIVER
- CAPABILITIES & CHROME OPTIONS
- CHROME EXTENSIONS
- CHROMEDRIVER CANARY
- CONTRIBUTING
- ▼ DOWNLOADS**
 - VERSION SELECTION
- ▼ GETTING STARTED**
 - ANDROID
 - CHROME OS
- ▼ LOGGING**
 - PERFORMANCE LOG
- MOBILE EMULATION

Downloads

Current Releases

- If you are using Chrome version 89, please download [ChromeDriver 89.0.4389.23](#)
- If you are using Chrome version 88, please download [ChromeDriver 88.0.4324.96](#)**
- If you are using Chrome version 87, please download [ChromeDriver 87.0.4280.88](#)
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

Index of /88.0.4324.96/

Name	Last modified	Size	ETag
Parent Directory	-	-	-
chromedriver linux64.zip	2021-01-20 19:13:52	5.42MB	40537b052b77c418f05abc1428ecc3c3
chromedriver mac64.zip	2021-01-20 19:13:53	7.76MB	b91266f2468907e6c3e58220182cf19f
chromedriver mac64_m1.zip	2021-01-20 19:13:55	6.99MB	dd6f6ae34fa210b1993fb159d24ce330
chromedriver win32.zip	2021-01-20 19:13:57	5.36MB	9f5e7741994b46b1acca15d779cfe7ad
notes.txt	2021-01-20 19:01:19	0.00MB	cbd16414ef0a8fc16a461d9d9dfa6b51

← → ↕ ↑ > 내 PC > 로컬 디스크 (D:) > test

이름	수정한 날짜	유형	크기
.ipynb_checkpoints	2021-02-28 오후 4:59	파일 폴더	
chromedriver.exe	2021-01-15 오후 6:44	응용 프로그램	9,958KB
Cralwer.ipynb	2021-02-26 오후 11:55	IPYNB 파일	3KB
Cralwer2.ipynb	2021-02-27 오후 10:25	IPYNB 파일	2KB
Cralwer-pandas.ipynb	2021-02-26 오후 11:39	IPYNB 파일	3KB
crawl_data.csv	2021-02-26 오후 11:13	한컴오피스 2018 ...	3KB
crawl_data_df.csv	2021-02-26 오후 11:27	한컴오피스 2018 ...	3KB

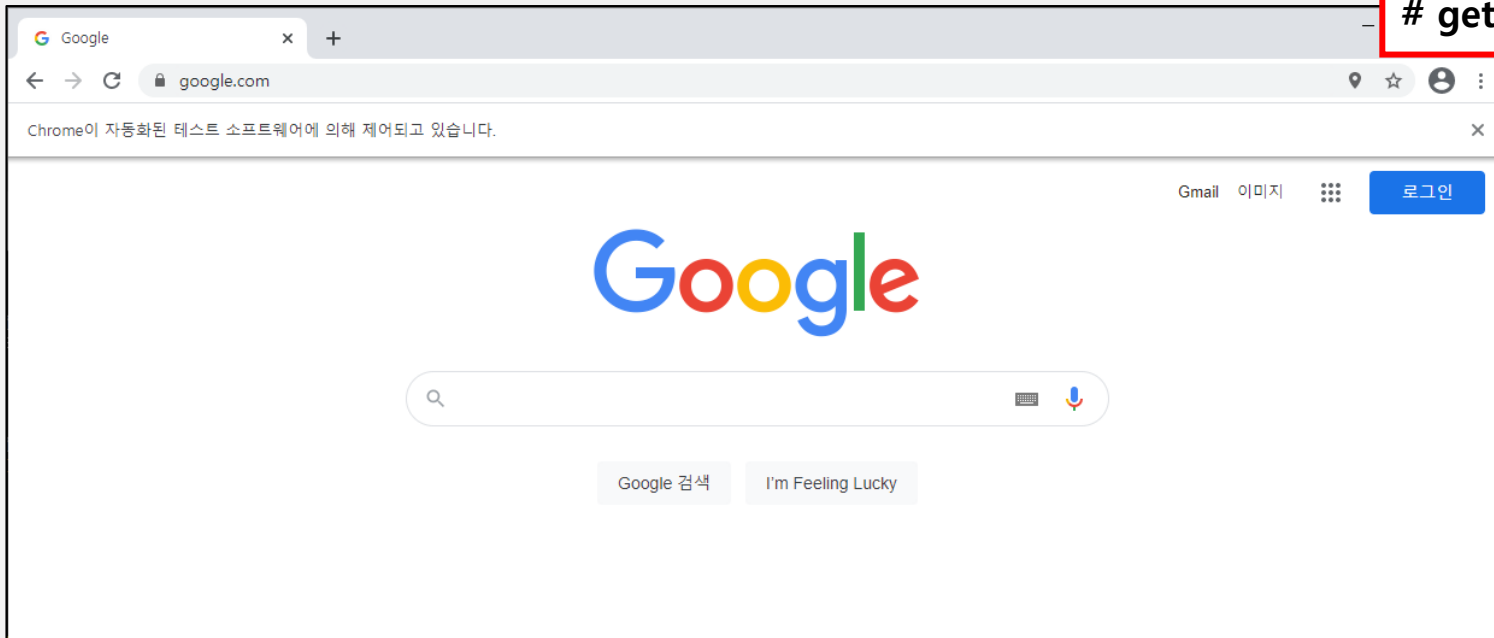
1-4. 동적 크롤링 – Web Driver 테스트

- Google에 접속하는 테스트 코드로 정상적으로 동작하는지 확인
- Web Driver가 다른 경로에 있을 경우 webdriver.Chrome('경로')를 추가해야함

```
In [1]: from selenium import webdriver  
driver = webdriver.Chrome() # chromedriver.exe가 같은경로에 있을 경우  
url = 'https://www.google.com/'  
driver.get(url)
```

driver 경로를 다른 곳에 둘 경우
driver = webdriver.Chrome('driver의 경로')

get(url) : url에서 지정한 웹 페이지를 읽어들이м



코드 실행결과 정상적으로
페이지가 뜨는 것을 볼 수 있음

1-4. 동적 크롤링 – “find_element_by_{ id | class_name | xpath }”

The image shows a screenshot of the Naver.com homepage with the DevTools browser extension open. The main content area of the page includes a search bar, a navigation bar with links like '쇼핑', 'Pay', 'TV', '사전', '뉴스', '증권', '부동산', '지도', '영화', 'VIBE', '책', and '웹툰', and a grid of partner logos at the bottom such as '한국일보', 'NewDaily', 'sportalkorea', '한글경제', '스팸', '정신의학신문', 'ELLE', 'SBS연예뉴스', 'TBS', '스포츠', '쿠리뉴스', 'YONHAPNEWS AGENCY', 'Da 디지털대일리', and 'MediaUS'.

The DevTools window is open to the 'Elements' tab, showing the DOM tree. The selected element is the login button, with the following HTML structure:

```
<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" style="position: absolute; width: 350px; height: 200px; right: 0px; top: 208px; z-index: 1;">...</div>
      <div id="account" class="sc_login">...</div>
      <div id="timesquare" class="sc_timesquare">...</div>
      <div id="veta_branding">...</div>
      <div id="shopcast" class="sc_shopcast">...</div>
    </div>
  </div>
  <a id="NM_scroll_top_btn" href="#wrap" class="content_top">...</a>
  <button id="NM_darkmode_btn" type="button" role="button" class="btn_theme" aria-pressed="false">...</button>
</div>
```

The 'Styles' pane shows the default styles for the selected element:

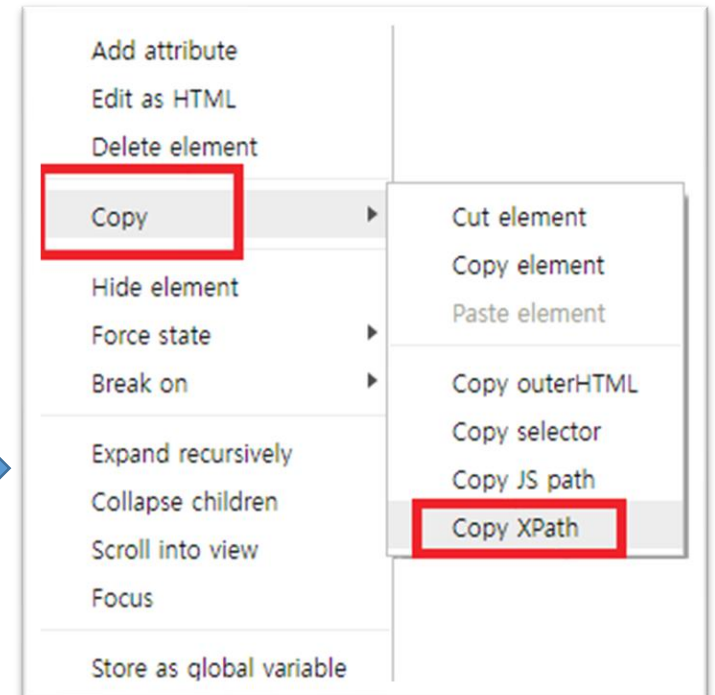
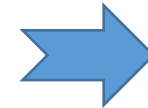
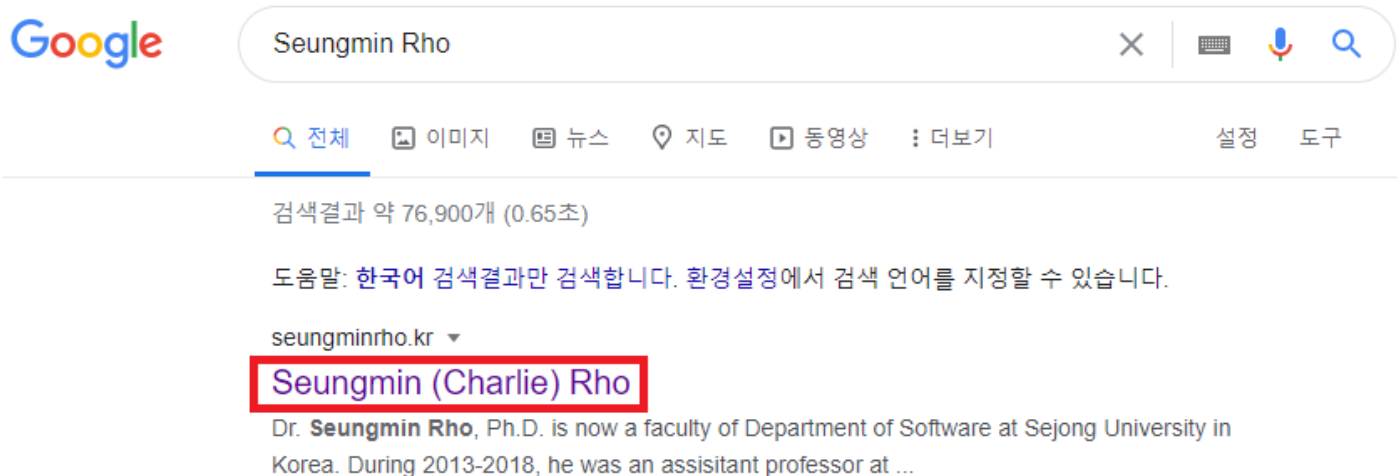
```
.sc_login {
  padding: 16px 16px 12px 17px;
  background-color: #f7f9fa;
  border: 1px solid #dae1e6;
}
```

The 'Console' tab shows two error messages related to loading SourceMaps:

```
DevTools failed to load SourceMap: Could not load content for https://pm.pstatic.net/dist/js/nmain.c42f61b5.js.map: HTTP error: status code 404, net::ERR_HTTP_RESPONSE_CODE_FAILURE
DevTools failed to load SourceMap: Could not load content for https://pm.pstatic.net/dist/js/polyfill_async.c6f27998.js.map: HTTP error: status code 404, net::ERR_HTTP_RESPONSE_CODE_FAILURE
```

1-4. 동적 크롤링 - XPath

- 크롤링 시 class나 id값이 계속 변하거나 아예 없어서 특정 태그를 지정하여 가져오기 힘든 경우
- XPath
 - HTML과 같은 마크업 언어에서 특정 요소를 찾기 위한 경로(path)를 나타냄
 - 기본 구문: `//tag_name[@attribute='value']`
 - 절대 경로: `/HTML/BODY/DIV/DIV/ ... /FORM/DIV/INPUT`
 - 상대 경로: `//DIV[@class='test']`
- Copy XPath
 - `//*[@id="rso"]/div[1]/div[1]/div/div[1]/a/h3/span`



1-4. 동적 크롤링 – XPath

<https://jsonformatter.org/xml-viewer>

← → ↺ jsonformatter.org/xml-viewer ☆ Se ⚙️ 👤

앱 💡 💡 DB 📄 블랙보드 - 과제채점 🔄 교무팀 🔄 교무팀

{JSON formatter}

JSON PARSER

JSON PRETTY PRINT

JSBEAUTIFIER

SAVE

RECENT LINKS

LOGIN

XML Viewer

Input XML

Sample



```
1 <address-book>
2 <entry>
3   <name>Seungmin Rho</name>
4   <photo>rho.jpg</photo>
5   <address>#1131, 310 Gwan, 84 Heukseok-Ro, Dongjak-Gu, Seoul
6     </address>
7   <tel>02-820-1234</tel>
8   <email>smrho@cau.ac.kr</email>
9 </entry>
10 <entry>
11   <name>Hangbae Chang</name>
12   <photo>chang.jpg</photo>
13   <address>#1131, 310 Gwan, 84 Heukseok-Ro, Dongjak-Gu, Seoul
14     </address>
15   <tel>02-820-5678</tel>
16   <email>hbchang@cau.ac.kr</email>
17 </entry>
18 </address-book>
```

Load Data

Validate

XML Viewer

Format / Beautify

3호선 남부터미널역
초역세권 주거형 오피스텔

해링턴 타워 서초

4월 오픈예정 1600-6990

Minify / Compact

Download

XML Tree



object ▶ address-book ▶ entry ▶ 1 ▶

▼ object {1}

▼ address-book {1}

▼ entry [2]

▼ 0 {5}

name : Seungmin Rho

photo : rho.jpg

address : #1131, 310 Gwan, 84 Heukseok-Ro, Dongjak-Gu, Seoul

tel : 02-820-1234

email : smrho@cau.ac.kr

▼ 1 {5}

name : Hangbae Chang

photo : chang.jpg

address : #1131, 310 Gwan, 84 Heukseok-Ro, Dongjak-Gu, Seoul

tel : 02-820-5678

email : hbchang@cau.ac.kr

1-4. 동적 크롤링 - XPath

- XPath

- 기본 구문: `//tag_name[@attribute='value']`
- / : 루트 노드(node)로부터 선택
- // : 현재 노드(node)로부터 문서상의 모든 노드를 조회
- . : 현재 노드 선택
- .. : 부모 노드 선택
- @ : 현재 노드의 속성 선택
- `//div[@name]` : name 속성 값을 가지는 div 태그들을 가져옴
- `//div[@*]` : 속성 값을 가지는 모든 div 태그들을 가져옴

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19041.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\User>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 0BFA-9360

C:\Users\User 디렉터리

2021-03-14 오후 09:53 <DIR> .
2021-03-14 오후 09:53 <DIR> ..
2021-03-09 오후 03:42 <DIR> .conda
2021-03-09 오전 09:37 43 .condarc
2021-02-19 오후 05:13 <DIR> .docker
2021-03-14 오후 09:27 <DIR> .ipynb_checkpoints
2021-02-23 오후 02:46 <DIR> .ipython
2021-02-23 오후 03:58 <DIR> .jupyter
2021-02-23 오후 03:08 <DIR> .matplotlib
2020-10-15 오후 09:30 <DIR> 3D Objects
2020-05-30 오후 07:26 2,091 Adobe Reader 9.lnk
2021-03-09 오전 09:15 <DIR> anaconda3
2021-03-14 오후 09:52 10,695,680 chromedriver.exe
2020-10-15 오후 09:30 <DIR> Contacts
2021-03-14 오후 12:32 6,141 crawl_api.txt
2021-03-09 오후 05:01 813 crawl_dict.csv
2021-03-11 오후 02:01 <DIR> Desktop
2021-03-10 오전 10:18 <DIR> Documents
2021-03-14 오후 09:40 <DIR> Downloads
```

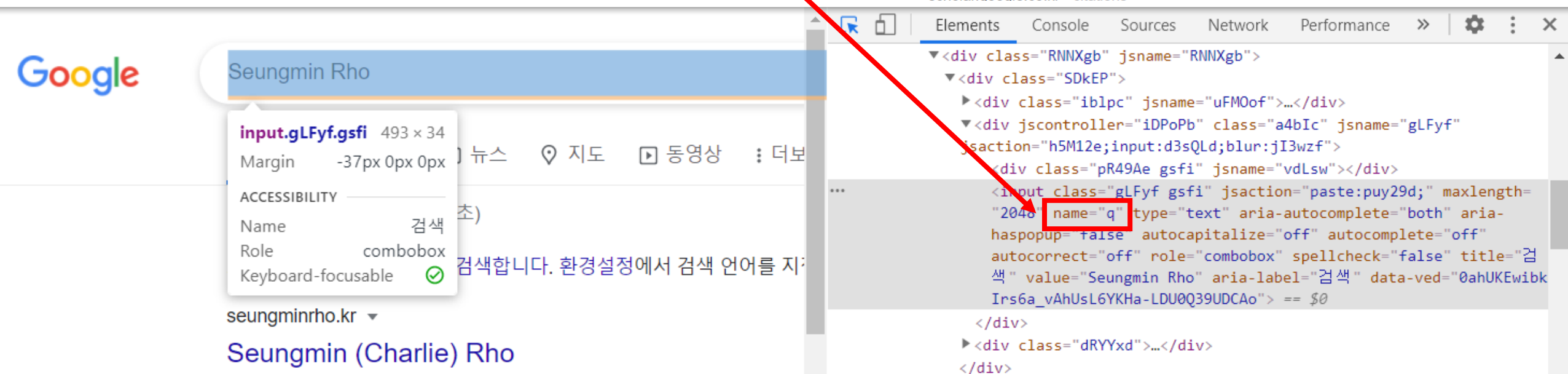
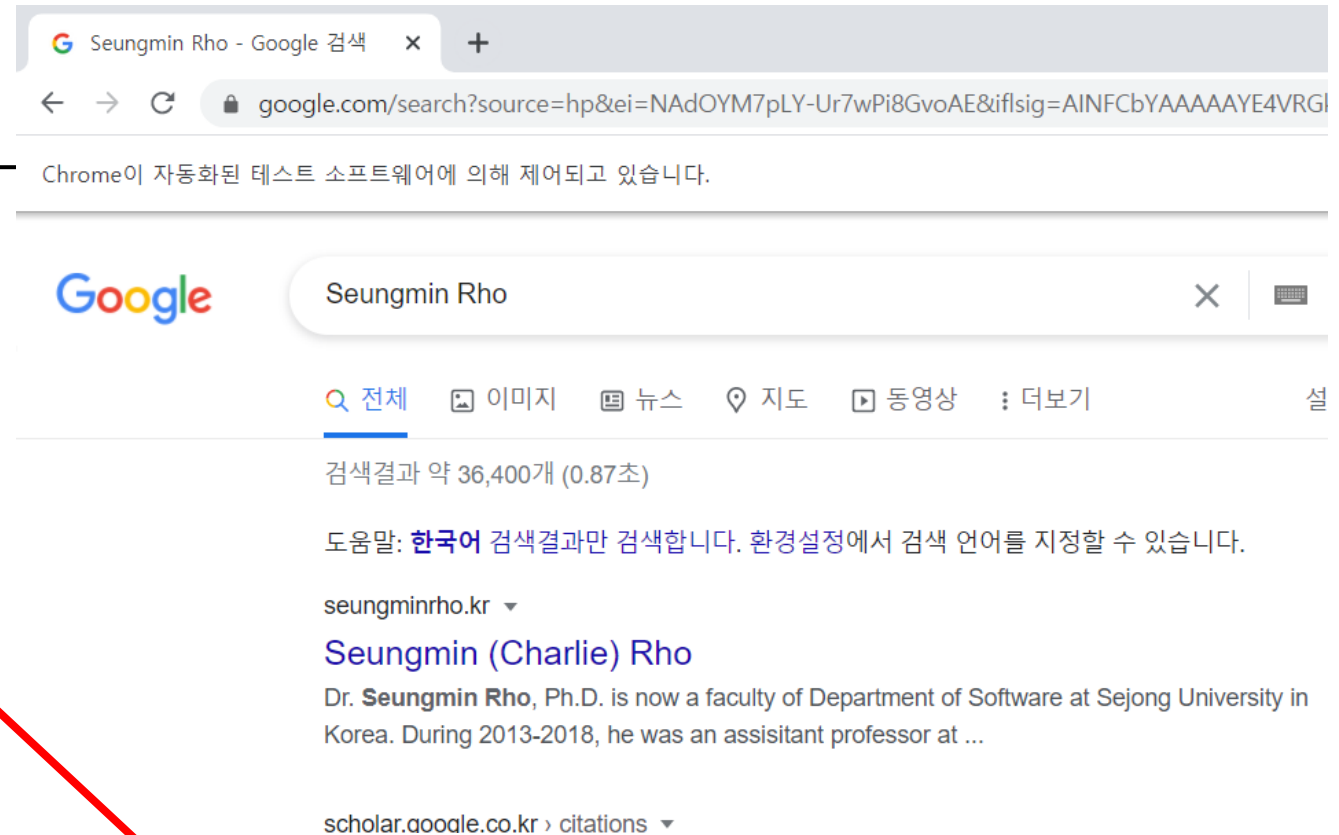
- XPath에 대한 추가적인 정보는 W3C 표준(<https://www.w3.org/TR/xpath/>)에서 정의를 참고하도록 하며, XPath에 대한 예제들은 다음(https://www.w3schools.com/xml/xpath_syntax.asp)에서 확인하기 바람

1-4. 동적 크롤링 – “find_element_by_ ...”

- find_element_by_name

```
In [7]: from selenium import webdriver

driver = webdriver.Chrome()
driver.get("http://google.com/")
search_box = driver.find_element_by_name("q")
search_box.send_keys("Seungmin Rho")
search_box.submit()
```



2. 동적 크롤링 - 크롤링 대상 데이터 선정

<https://threatmap.checkpoint.com/>



2. 동적 크롤링 - 크롤링 대상 데이터 선정

<https://threatmap.checkpoint.com/>

The screenshot displays the Check Point ThreatCloud Live Cyber Threat Map. The main area features a world map with attack paths and a sidebar with statistics. The browser's developer tool is open on the right, showing the HTML structure of the page.

Check Point THREATCLOUD
LIVE CYBER THREAT MAP
6,434,260 ATTACKS ON THIS DAY

RECENT DAILY ATTACKS

Time	Source	Destination	Type
21:33:53	Turkey	Russia	Web Server Enforcement Violation
21:33:53	US, United States	India	Content Protection Violation
21:33:53	US, United States	India	Content Protection Violation
21:33:53	US, United States	India	Content Protection Violation
21:33:52	US, United States	India	Content Protection Violation
21:33:52	US, United States	India	Content Protection Violation
21:33:52	US, United States	Dominican R...	Web Client Enforcement Violation
21:33:52	US, United States	Dominican R...	Web Client Enforcement Violation

TOP TARGETED COUNTRIES
Highest rate of attacks per organization in the last day.

- Nepal
- Bolivia
- Indonesia
- Angola
- Mongolia

TOP TARGETED INDUSTRIES
Highest rate of attacks per organization in the last day.

- Education
- Government
- Communications

TOP MALWARE TYPES
Malware types with the highest global impact in the last day.

- Botnet
- Phishing
- Mobile

Developer Tool (Chrome DevTools):

- Elements:** Shows the HTML structure, including a `div.section` with a tooltip showing dimensions `235.39 x 328.92`.
- Console:** Displays a message: "You need to enable JavaScript to run this page."
- Network:** Shows a request to `https://threatmap.checkpoint.com/`.
- Inspector:** Shows the CSS for the `div.section` element, including `display: block;` and `border-top: 1px solid #e44986;`.

3. 동적 크롤링 - 수집 방법(1)

- 아래는 파이썬 라이브러리 selenium을 통해 동적사이트를 크롤링한 결과이다.
- 반복문(while, for) 및 데이터 전처리 기법을 적용하여 실시간 수집이 가능함

ATTACKS 🌐 Current rate - 4 +



Conficker_B.TC.baqnp	20:04:34 China → China
Apple QuickTime traf Atom Out-Of-B...	20:04:34 China → China
Conficker_B.TC.baqnp	20:04:34 China → China
cnc server.TC.jjzg	20:04:33 China → China
Conficker_B.TC.baqnp	20:04:33 China → China
Content Protection Violation	20:04:33 Portugal → Portugal
Conficker_B.TC.baqnp	20:04:33 China → China

```
In [3]: from selenium import webdriver
import time

driver = webdriver.Chrome() # chromedriver.exe가 같은경로에 있을 경우
driver.get('https://threatmap.checkpoint.com/')

time.sleep(20)
table = driver.find_element_by_xpath('//*[@id="root"]/div/div/div/div/div/div/div/div[1]/div[4]')

print(table.text)

Conficker_B.TC.baqnp
20:04:34 China China
cnc server.TC.jjzg
20:04:33 China China
Conficker_B.TC.baqnp
20:04:33 China China
Content Protection Violation
20:04:33 Portugal Portugal
Conficker_B.TC.baqnp
20:04:33 China China
Conficker_B.TC.baqnp
20:04:32 China China
Conficker_B.TC.baqnp
20:04:32 China China
```


3. 동적 크롤링 - 수집 방법(1)

- 페이지가 반응형인 경우 브라우저 크기가 작게 열릴 경우 동작 안할 수 있음

```
In [9]: from selenium import webdriver
import time

options = webdriver.ChromeOptions()
options.add_argument('window-size=1920,1080')

driver = webdriver.Chrome('chromedriver', options=options) # chromedriver.exe가 같은경로에 있을 경우
driver.get('https://threatmap.checkpoint.com/')

time.sleep(20) #20초간 대기(PAUSE)
table = driver.find_element_by_xpath('//*[@id="root"]/div/div/div/div/div/div/div[1]/div[4]')

print(table.text)
```

```
XMLRig.TC.epv
16:08:58 US, United States  Philippines
XMLRig.TC.epv
16:08:58 US, United States  Philippines
XMLRig.TC.epv
16:08:58 US, United States  Philippines
XMLRig.TC.epv
16:08:57 US, United States  Philippines
XMLRig.TC.epv
16:08:57 US, United States  Philippines
Content Protection Violation
16:08:57 US, United States  Germany
NTP Enforcement Violation
16:08:57 Kuwait  Kuwait
```

3. 동적 크롤링 - 수집 방법(2)

브라우저 개발자 도구(F12)

- 개발자 도구(F12)의 Network를 통해 브라우저와 서버 사이에 주고 받는 동적 데이터를 탐색
- Network는 서버와의 통신 내용을 보여주는 도구로 보이지 않는 리소스를 탐색하는데 유용

The image shows a web browser window displaying a 'Check Point THREATLIVE LIVE CYBER THREAT MAP' with 15,196,290 attacks on this day. The map highlights regions like the US, China, and Indonesia. Below the map, there are statistics for malware, phishing, and exploit attacks. A red box highlights the 'Network' tab in the developer tools. A red arrow points from the 'Network' tab to the 'General' tab of a selected request. The 'General' tab shows the request URL: <https://threatmap-api.checkpoint.com/ThreatMap/api/feed>. The 'Request Headers' tab shows the request method: GET, status code: 200, and various headers like 'Access-Control-Allow-Origin' and 'Content-Type'.

<https://threatmap-api.checkpoint.com/ThreatMap/api/feed>

3. 동적 크롤링 - 수집 방법(2)

브라우저 개발자 도구(F12)

- 수집할 동적페이지를 Python을 활용하여 크롤링 코드를 구현하여 분석할 데이터를 수집



```
data: {"recentPeriod":
[33977967, 35218000, 33374629, 33373585, 31998341, 31849556, 35649398, 42446171, 45793219, 40374785, 38959189, 31311678, 31777946, 34486566, 33502970, 3
8113677, 34459567, 33704579, 31559034, 33611155, 34952501, 38134497, 37194920, 37467061, 35827864, 33281561, 34368495, 39481632, 38358700, 37916040, 403
12339, 39422249, 33990297, 36482136, 38985722, 41636484, 43901274, 43031120, 39899755, 36209274, 36732941, 42352852, 37231436, 36748056, 42233462, 42010
016, 39802116, 40027098, 44113803, 45630385, 44655896, 42731571, 39447897, 41856372, 38578967, 44980486, 46204521, 63563222, 48101127, 45339282, 3770091
7, 36926262, 41435435, 41130337, 44391083, 43909671, 45590125, 41762482, 41339161, 46705407, 51633854, 52269967, 47203189, 44138176, 36262463, 40255166,
54714630, 62082510, 55326589, 49633790, 46652105, 43919303, 45386853, 42595445, 30513529, 38047053, 39594412, 27107611, 43354489, 16656408], "today": 15
217926}
event: counter
retry: 10000

event: counter
data: {"recentPeriod":
[33977967, 35218000, 33374629, 33373585, 31998341, 31849556, 35649398, 42446171, 45793219, 40374785, 38959189, 31311678, 31777946, 34486566, 33502970, 3
8113677, 34459567, 33704579, 31559034, 33611155, 34952501, 38134497, 37194920, 37467061, 35827864, 33281561, 34368495, 39481632, 38358700, 37916040, 403
12339, 39422249, 33990297, 36482136, 38985722, 41636484, 43901274, 43031120, 39899755, 36209274, 36732941, 42352852, 37231436, 36748056, 42233462, 42010
016, 39802116, 40027098, 44113803, 45630385, 44655896, 42731571, 39447897, 41856372, 38578967, 44980486, 46204521, 63563222, 48101127, 45339282, 3770091
7, 36926262, 41435435, 41130337, 44391083, 43909671, 45590125, 41762482, 41339161, 46705407, 51633854, 52269967, 47203189, 44138176, 36262463, 40255166,
54714630, 62082510, 55326589, 49633790, 46652105, 43919303, 45386853, 42595445, 30513529, 38047053, 39594412, 27107611, 43354489, 16656408], "today": 15
220323}
retry: 10000

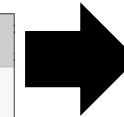
event: attack
data: {"a_c": 50, "a_n": "Application Servers Protection
Violation", "a_t": "exploit", "d_co": "IL", "d_la": 31.5, "d_lo": 34.75, "d_s": null, "s_co": "NL", "s_lo": 4.8975, "s_la": 52.3759, "s_s": "NH", "t": null}
retry: 10000

event: attack
data: {"a_c": 6, "a_n": "Content Protection
Violation", "a_t": "exploit", "d_co": "IL", "d_la": 31.5, "d_lo": 34.75, "d_s": null, "s_co": "IL", "s_lo": 34.75, "s_la": 31.5, "s_s": null, "t": null}
retry: 10000

event: attack
data: {"a_c": 8, "a_n": "Content Protection
Violation", "a_t": "exploit", "d_co": "IL", "d_la": 31.5, "d_lo": 34.75, "d_s": null, "s_co": "IL", "s_lo": 34.75, "s_la": 31.5, "s_s": null, "t": null}
retry: 10000

event: attack
data: {"a_c": 3, "a_n": "Content Protection
Violation", "a_t": "exploit", "d_co": "IL", "d_la": 31.5, "d_lo": 34.75, "d_s": null, "s_co": "US", "s_lo": -122.0461, "s_la": 37.589, "s_s": "CA", "t": null}
retry: 10000

event: attack
data: {"a_c": 9, "a_n": "GitList Remote Code Execution (CVE-2018-1000533) -
Ver2", "a_t": "exploit", "d_co": "FR", "d_la": 43.551, "d_lo": 5.1943, "d_s": "13", "s_co": "US", "s_lo": -97.822, "s_la": 37.751, "s_s": null, "t": null}
retry: 10000
```



크롤링 코드 구현 및 개발

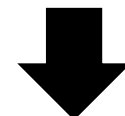
```
event: attack
data: {"a_c": 50, "a_n": "Application Servers Protection
Violation", "a_t": "exploit", "d_co": "IL", "d_la": 31.5, "d_lo": 34.75, "d_s": null, "s_co": "NL", "s_lo": 4.8975, "s_la": 52.3759, "s_s": "NH", "t": null}
retry: 10000

event: attack
data: {"a_c": 6, "a_n": "Content Protection
Violation", "a_t": "exploit", "d_co": "IL", "d_la": 31.5, "d_lo": 34.75, "d_s": null, "s_co": "IL", "s_lo": 34.75, "s_la": 31.5, "s_s": null, "t": null}
retry: 10000

event: attack
data: {"a_c": 8, "a_n": "Content Protection
Violation", "a_t": "exploit", "d_co": "IL", "d_la": 31.5, "d_lo": 34.75, "d_s": null, "s_co": "IL", "s_lo": 34.75, "s_la": 31.5, "s_s": null, "t": null}
retry: 10000

event: attack
data: {"a_c": 3, "a_n": "Content Protection
Violation", "a_t": "exploit", "d_co": "IL", "d_la": 31.5, "d_lo": 34.75, "d_s": null, "s_co": "US", "s_lo": -122.0461, "s_la": 37.589, "s_s": "CA", "t": null}
retry: 10000

event: attack
data: {"a_c": 9, "a_n": "GitList Remote Code Execution (CVE-2018-1000533) -
Ver2", "a_t": "exploit", "d_co": "FR", "d_la": 43.551, "d_lo": 5.1943, "d_s": "13", "s_co": "US", "s_lo": -97.822, "s_la": 37.751, "s_s": null, "t": null}
retry: 10000
```



데이터 정제 및 분석

3. 동적 크롤링 - 수집 방법(2)

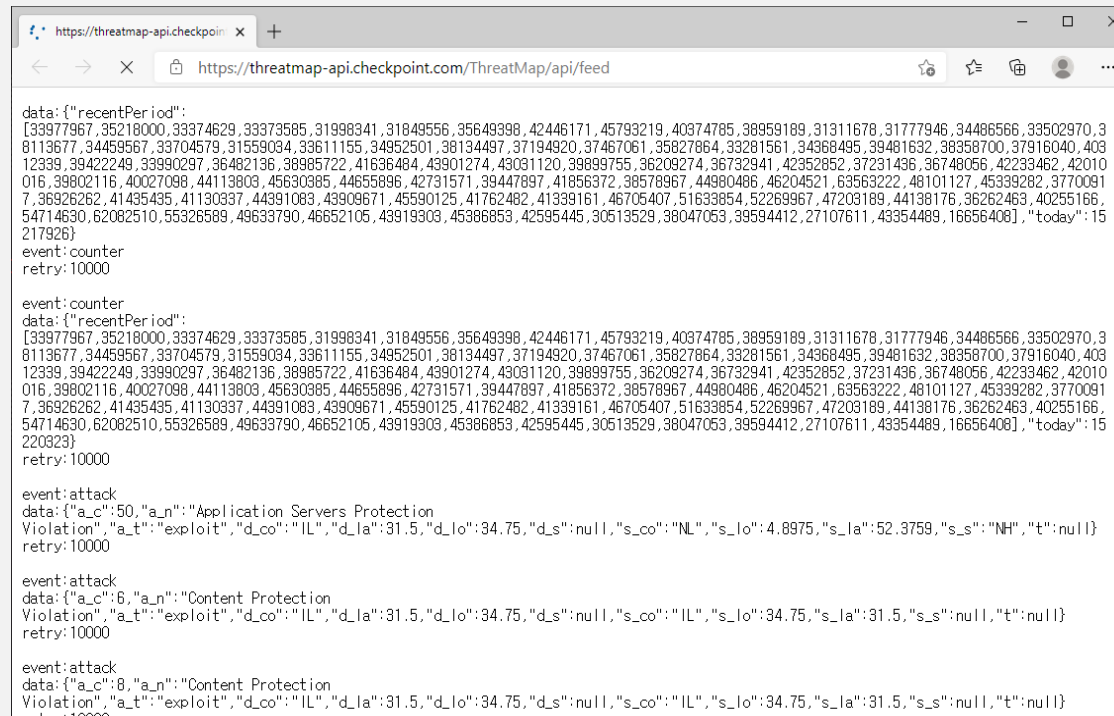
브라우저 개발자 도구(F12)

- 브라우저 개발자 도구를 통해 동적 데이터를 파악하여 해당 데이터를 수집 가능함
- 동적 크롤링 (1), (2)번 중 보다 효율적인 데이터를 수집 및 분석하여 다양한 방법으로 활용

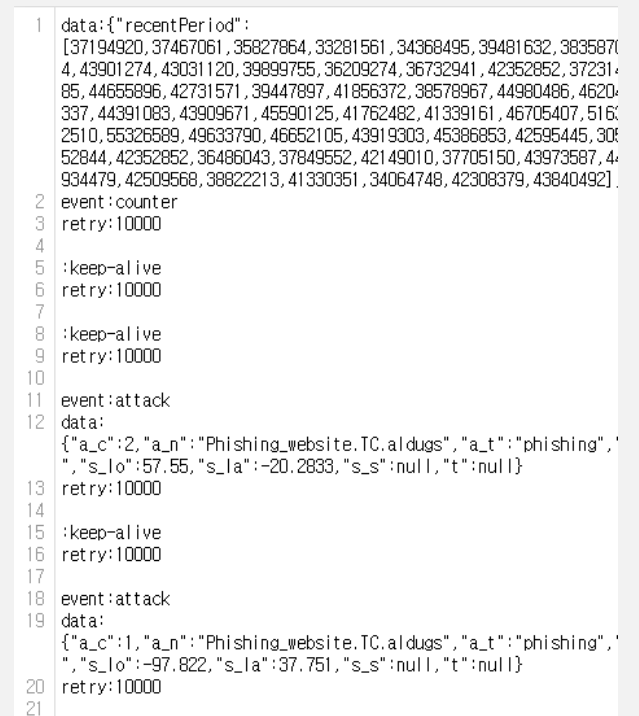
웹 사이트(동적)



브라우저 개발자 도구를 통해 분석한 Web



크롤링을 통해 도출한 데이터(TXT)



3. 동적 크롤링 - 수집 방법(2)

브라우저 개발자 도구(F12)

소스코드

```
from selenium import webdriver

def save_txt(data):
    f = open('crawl_api.txt', 'w', newline='', encoding='UTF-8')
    f.write(data)
    f.close()

driver = webdriver.Chrome() # chromedriver.exe가 같은경로에 있을 경우
url = 'https://threatmap-api.checkpoint.com/ThreatMap/api/feed'
driver.set_page_load_timeout(10)
try:
    driver.get(url)
except:
    print("Page Out")

table = driver.find_element_by_xpath("/html/body/pre")
save_txt(table.text)
driver.quit()
print("Crawling Finish")
```

Page Out
Crawling Finish

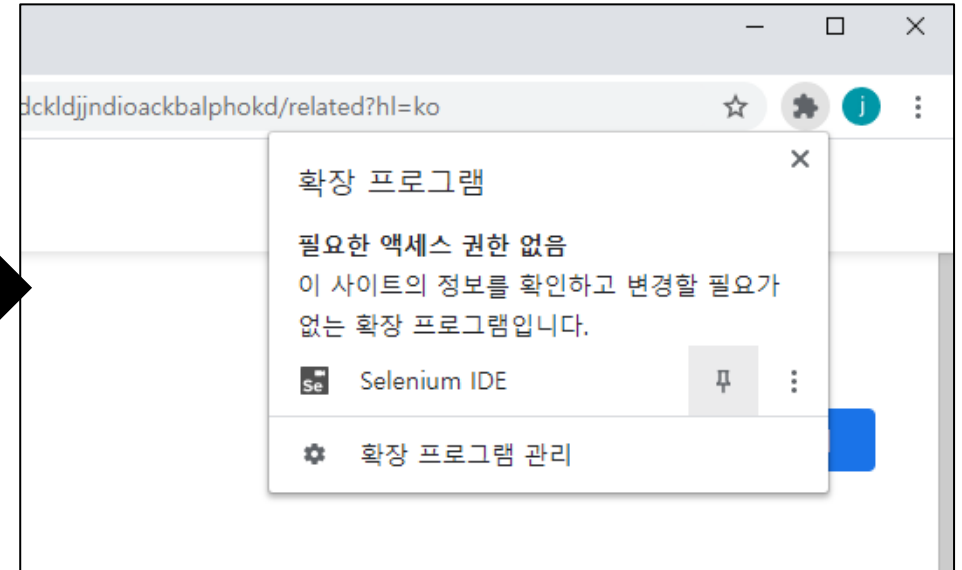
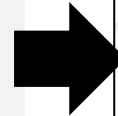
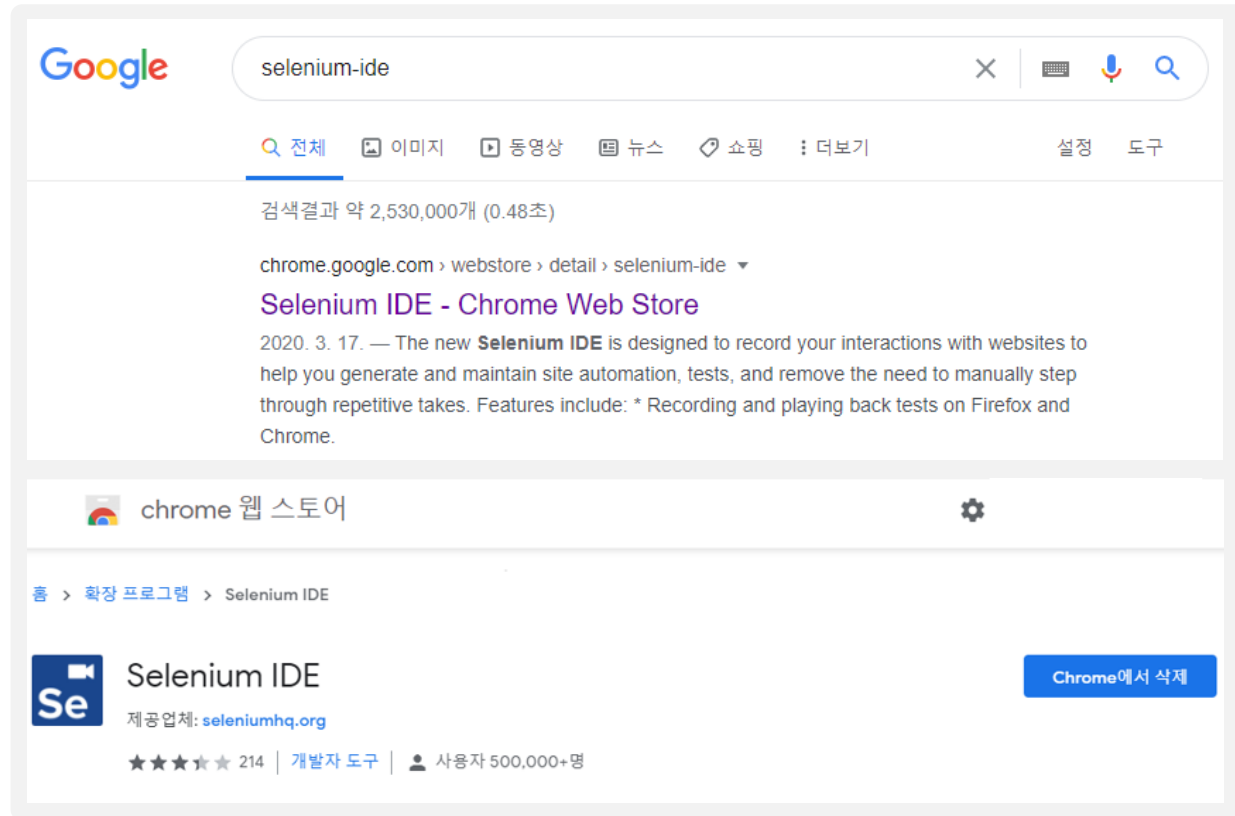
데이터 저장

```
1 data:{"recentPeriod":
  [37194920,37467061,35827864,33281561,34368495,39481632,3835870
  4,43901274,43031120,39899755,36209274,36732941,42352852,37231
  85,44655896,42731571,39447897,41856372,38578967,44980486,4620
  337,44391083,43909671,45590125,41762482,41339161,46705407,516
  2510,55326589,49633790,46652105,43919303,45386853,42595445,30
  52844,42352852,36486043,37849552,42149010,37705150,43973587,4
  934479,42509568,38822213,41330351,34064748,42308379,43840492]
2 event:counter
3 retry:10000
4
5 :keep-alive
6 retry:10000
7
8 :keep-alive
9 retry:10000
10
11 event:attack
12 data:
  {"a_c":2,"a_n":"Phishing_website.TC.aldugs","a_t":"phishing",
  ",","s_lo":57.55,"s_la":-20.2833,"s_s":null,"t":null}
13 retry:10000
14
15 :keep-alive
16 retry:10000
17
18 event:attack
19 data:
  {"a_c":1,"a_n":"Phishing_website.TC.aldugs","a_t":"phishing",
  ",","s_lo":-97.822,"s_la":37.751,"s_s":null,"t":null}
20 retry:10000
21
```

3. 동적 크롤링 - 수집 방법(3)

Selenium IDE (Chrome Extensions)

- 셀레늄 IDE(Selenium IDE)는 사용자가 웹 브라우저에서 수행한 동작을 기록, 재현
- 크롬 또는 파이어폭스를 통해 셀레늄 IDE를 사용할 수 있으며, 본 강의는 크롬을 활용함

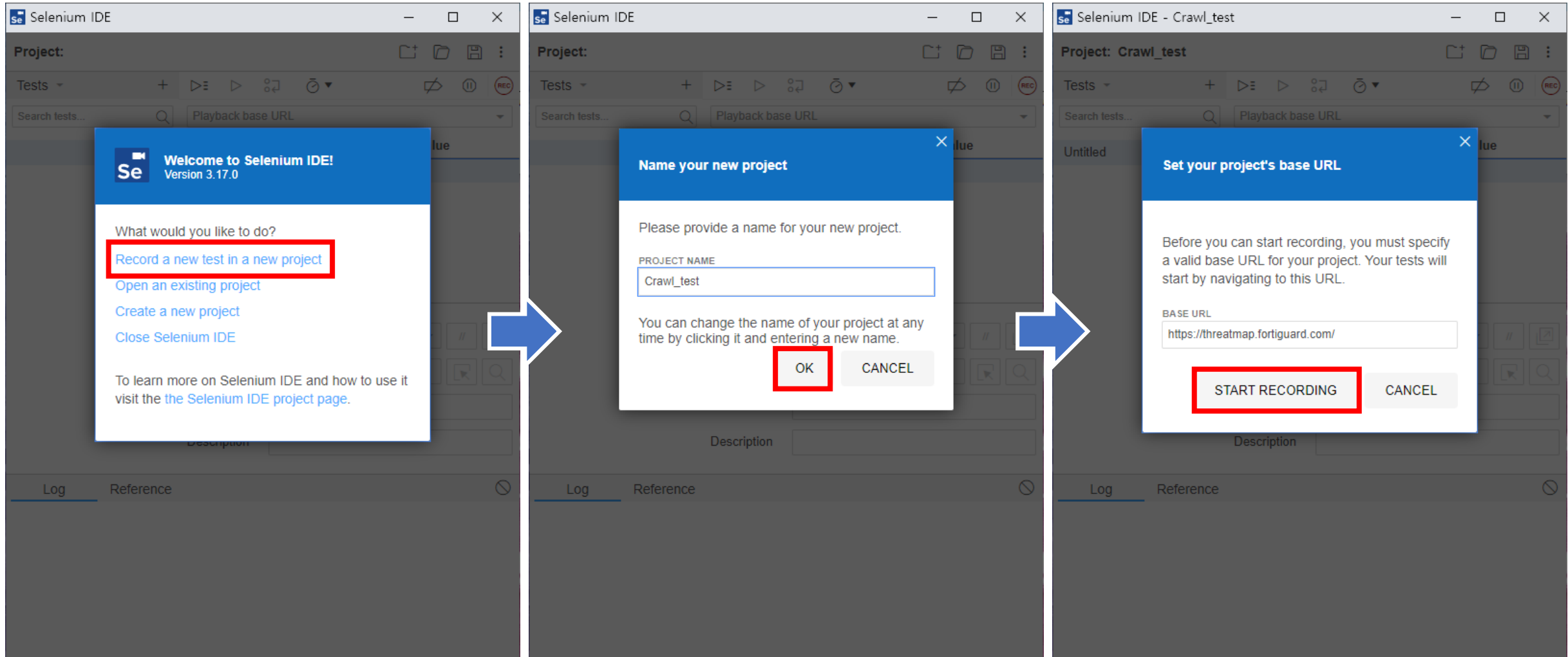


<https://www.selenium.dev/selenium-ide/>

3. 동적 크롤링 - 수집 방법(3)

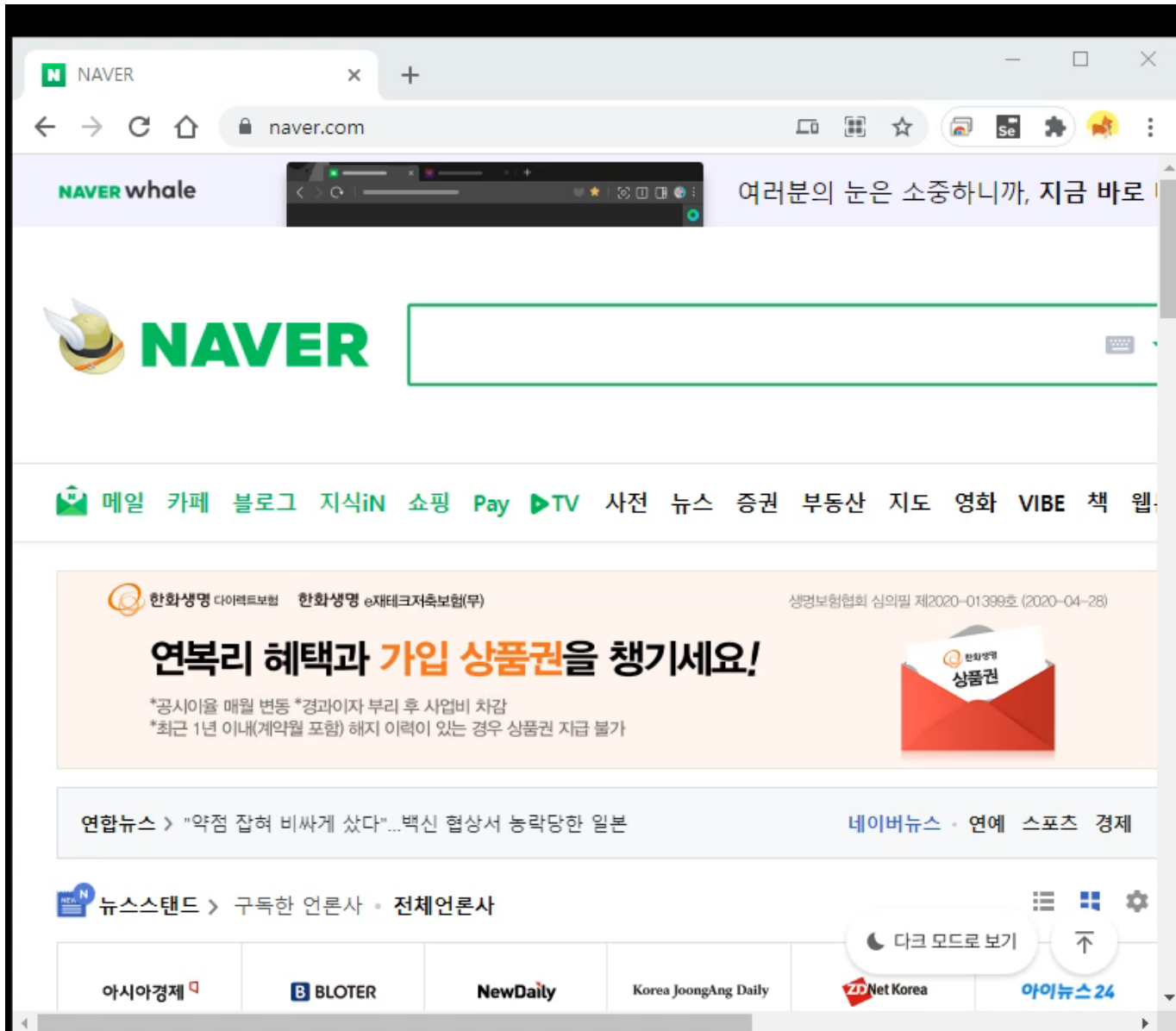
Selenium IDE (Chrome Extensions)

- 설치한 다음 새로운 프로젝트를 만들어서 TARGET URL을 지정



3. 동적 크롤링 - 수집 방법(3)

Selenium IDE (Chrome Extentions)



4. 실습 예제 - 네이버 금융

<https://finance.naver.com/main/main.nhn>

- 네이버 금융 TOP 종목 크롤링을 위해 메인 페이지를 대상으로 크롤러 구현



크롤링 대상 데이터 - 동적 데이터

거래상위	상한가	하한가	시가총액	상위
KODEX 200선물인버스2X	2,130	▼ 45	-2.07%	
이화전기	261	▲ 60	+29.85%	
서울식품	474	▲ 102	+27.42%	
이트론	668	▲ 154	+29.96%	
이아이디	423	▲ 97	+29.75%	
골든센츄리	496	▲ 62	+14.29%	

```
from selenium import webdriver

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

driver = webdriver.Chrome('chromedriver', chrome_options=options)
driver.get('https://finance.naver.com/main/main.nhn')
table = driver.find_element_by_id('_topItems1')
print(table.text)
```

<ipython-input-8-d79a18904b91>:8: DeprecationWarning: use options instead of chrome_options

```
driver = webdriver.Chrome('chromedriver', chrome_options=options)
```

KODEX 200선물인버스2X 2,130
하락
45 -2.07%
이화전기 261
상승
60 +29.85%
서울식품 474
상승
102 +27.42%
이트론 668
상승
154 +29.96%
이아이디 423
상승
97 +29.75%
골든센츄리 496
상승
62 +14.29%

5. Assignments

1. 예제 사이트(threatmap.checkpoint.com)와 같은 threatMap 중 사이트를 선택하여 데이터를 수집하는 크롤링 코드를 작성하시오. (<https://www.lesliesikos.com/cyberattack-maps/>)

조건1) 원하는 데이터를 정제하여 데이터 형태(.csv, .txt 등)으로 저장할 것

조건2) 되도록 동적 크롤링 (1), (2) 방법 모두 활용하여 이용할 것

2. 네이버 금융 동적 크롤링 (2) 방법을 활용하여 수집하는 크롤링 코드를 작성하시오.

조건1) 브라우저 개발자 도구(F12)를 이용하여 동적 데이터 URL을 이용할 것

조건2) 원하는 데이터를 정제하여 데이터 형태(.csv, .txt 등)으로 저장할 것

5. Assignments

https://data.gg.go.kr/portal/data/dataset/searchDatasetPage.do

3. 경기데이터드림 사이트를 selenium을 사용하여 다양한 페이지에 대한 크롤링 코드를 작성하시오.

- 크롤링 조건(필수)
- 데이터 목록(제목, 내용)을 수집가능하도록 코드 작성
 - 데이터 저장은 csv, txt 등 형식으로 저장할 것
 - 여러 페이지를 수집가능하도록 코드 작성 (예시) 1page~10page의 목록만 수집

- 도전 과제
- 정적크롤링에 활용되는 라이브러리(requests 등)를 통해 데이터 목록을 수집하는 코드를 작성하시오.

경기데이터드림

데이터 맞춤형 데이터 활용 참여 소통 소개

데이터셋 분류 체계

데이터

데이터셋

민간공익 데이터

멀티미디어 데이터

경기도 통계데이터

민간데이터 카탈로그

보유기관별 검색

경기도

행정안전부

경기도장애인복지종합...

교육부

국토교통부

+ 더보기

인기데이터

1 지역화폐 가맹점 현황

2 경기도 코로나19 신천...

3 CCTV 현황(개방표준)

4 공장등록 현황

5 지식산업센터 현황

인기태그별 검색

데이터셋

데이터의 특성을 고려하여 Sheet, Chart, Map, File, Link 서비스와 개발자를 위한 Open API 서비스 등 편리하게 데이터를 활용할 수 있도록 다양한 방식으로 서비스를 제공합니다.

· 유형선택

SHEET CHART MAP FILE API LINK

복수 선택 가능

· 정렬선택











등록일자순 데이터명순 높은 조회순

다양한 정렬방식으로 조회 가능

· 전체 1,453건. (1/146 page)

10개씩보기

확인

서비스	서비스 유형	조회수	등록일자 최종 수정일자	분류
 경기도 명소 데이터 현황 경기도 명소 데이터 현황입니다.	<div>SHEET MAP API</div>	40	2021-03-05 2021-03-05	 관광문화체육
 경기도 테마관광지 현황 경기도 테마관광지 현황입니다.	<div>SHEET MAP API</div>	36	2021-03-05 2021-03-05	 관광문화체육
 경기도 체험관광지 현황 경기도 체험관광지 현황입니다.	<div>SHEET MAP API</div>	28	2021-03-05 2021-03-05	 관광문화체육
 경기도 자연관광지 현황 경기도 자연관광지 현황입니다.	<div>SHEET MAP API</div>	30	2021-03-05 2021-03-05	 관광문화체육
 경기도 역사관광지 현황 경기도 역사관광지 현황입니다.	<div>SHEET MAP API</div>	25	2021-03-05 2021-03-05	 관광문화체육

5. Assignments

<https://www.open.go.kr/com/tema/temaSrhList.do>

4. 정보공개포털 데이터를 원하는 카테고리의 데이터를 아래 조건에 따라 순차적으로 크롤링하는 코드를 작성하시오.

크롤링 조건(필수)

- ①번 카테고리 별로 크롤링이 가능하도록 코드를 작성할 것
- 카테고리 별 크롤링 대상 데이터는 ② 데이터 목록이며, 데이터 저장
은 csv, txt 등 형식으로 저장할 것

크롤링 조건(선택)

- 카테고리 별 여러 페이지를 수집가능하도록 코드 작성
(예시) 건강 카테고리의 데이터를 1page~10page의 목록만 수집

도전 과제

- 정적크롤링에 활용되는 라이브러리(requests 등)를 통해 데이터 목록
을 수집하는 코드를 작성하시오.



검색할 단어를 입력해주세요.



로그인

전체

홈 > 주제별정보 > 전체

전체	건강	경제	교육	규제개혁	복지	안전
	여가	보육	일자리	주택	행정재정	환경

검색어	검색할 단어를 입력해주세요.		<input type="checkbox"/> 초·중·고등학교포함
기관선택	<input type="text"/>	기관찾기	<input type="checkbox"/> 중앙행정기관 <input type="checkbox"/> 지방자치단체 <input type="checkbox"/> 교육청 <input type="checkbox"/> 공공기관
기간검색	기간설정	2021-02-06	2021-03-07

* 설정 한 기간들의 자료입니다. 원활한 검색서비스를 위해 검색기간은 1년으로 제한합니다.

원문정보(69,269)건	사진정보(734)건	10개씩보기	조회	날짜순	조
노인일자리 경로당사업 참여자 교육(03.05) 충청북도>옥천군>기획감사실		군정사진기록 2021.03.06			
2021년 서산시 청년통계 개발계획 충청남도>서산시>자치행정국>정보통신과		통계일반 2021.03.06			
구리한강시민공원 야구장 조명탑 및 전광판 준공식 계획보고 경기도>구리시>환경관리사업소>공원녹지과		공원일반관리 2021.03.06			
제76회 식목일 기념행사 추진계획 전라남도>해남시>도시계획과		조림자가꾸기사업 2021.03.06			

③

<< < 1 2 3 4 5 6 7 8 9 10 >