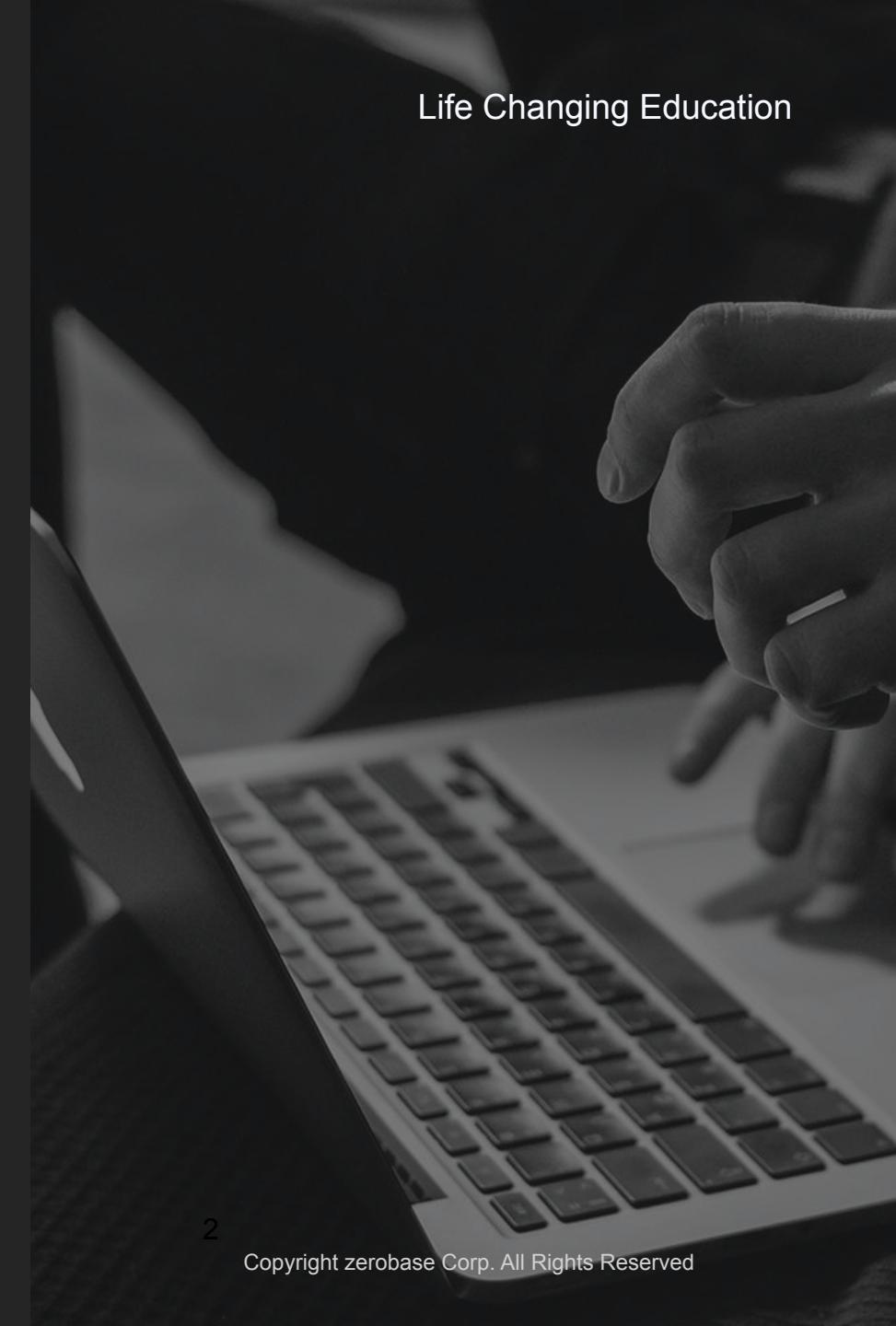
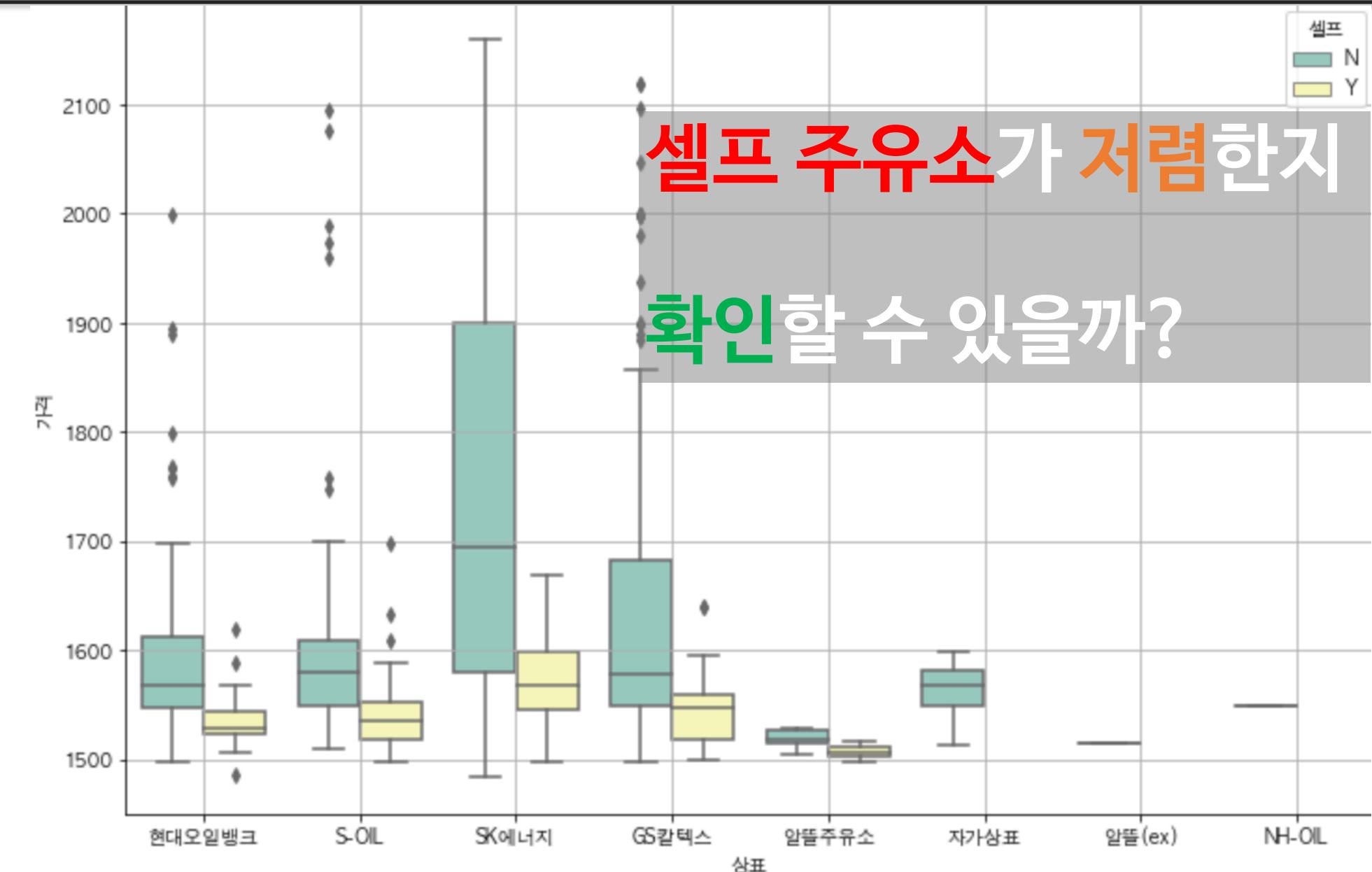


Chapter 04. Oil Price Analysis and Green construction



셀레니움 설치





Selenium 세팅

Beautiful Soup 만으로 해결할 수 없는 것

- 접근할 웹 주소를 알 수 없을 때
- 자바스크립트를 사용하는 웹페이지의 경우
- 웹 브라우저로 접근하지 않으면 안될 때

Selenium:

- 웹 브라우저를 원격 조작하는 도구
- 자동으로 URL을 열고 클릭 등이 가능
- 스크롤, 문자의 입력, 화면 캡처 등등

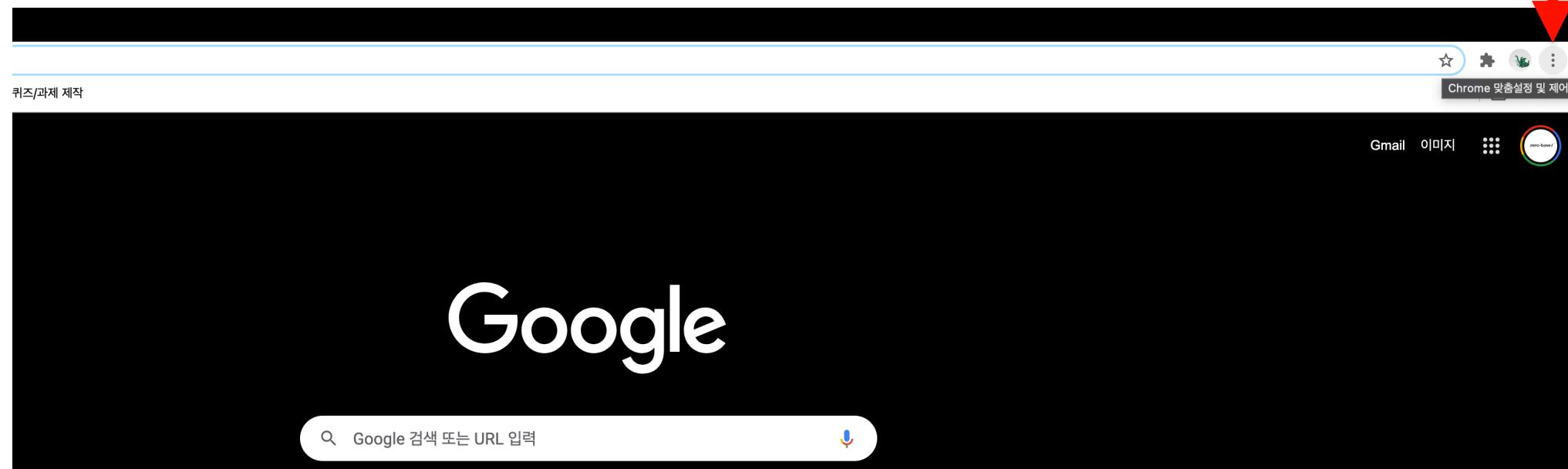
Selenium은
Python 모듈도 설치하고
크롬 드라이버도 받아야 한다.

Windows, Mac(intel)

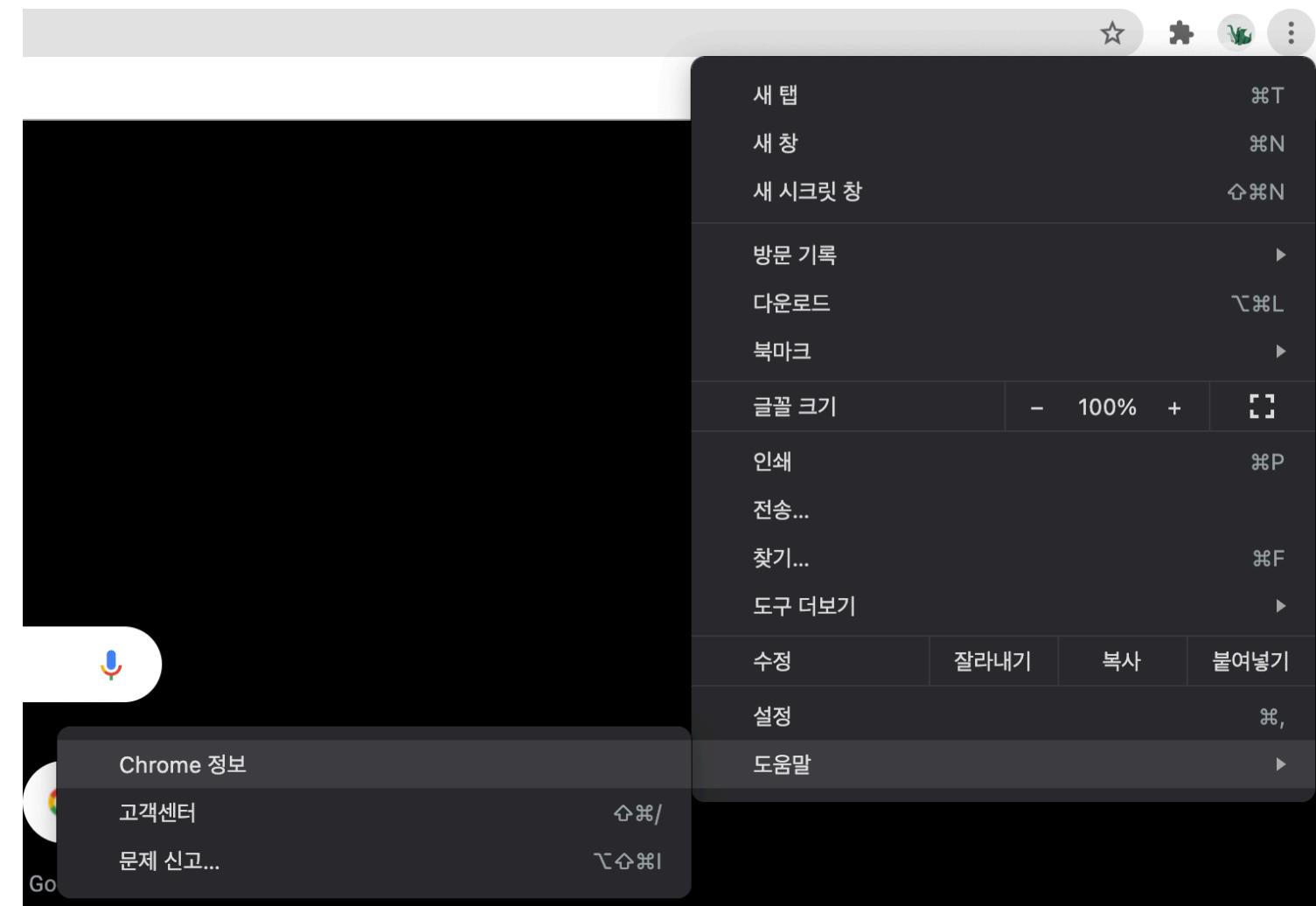
- conda install selenium

Mac(m1)

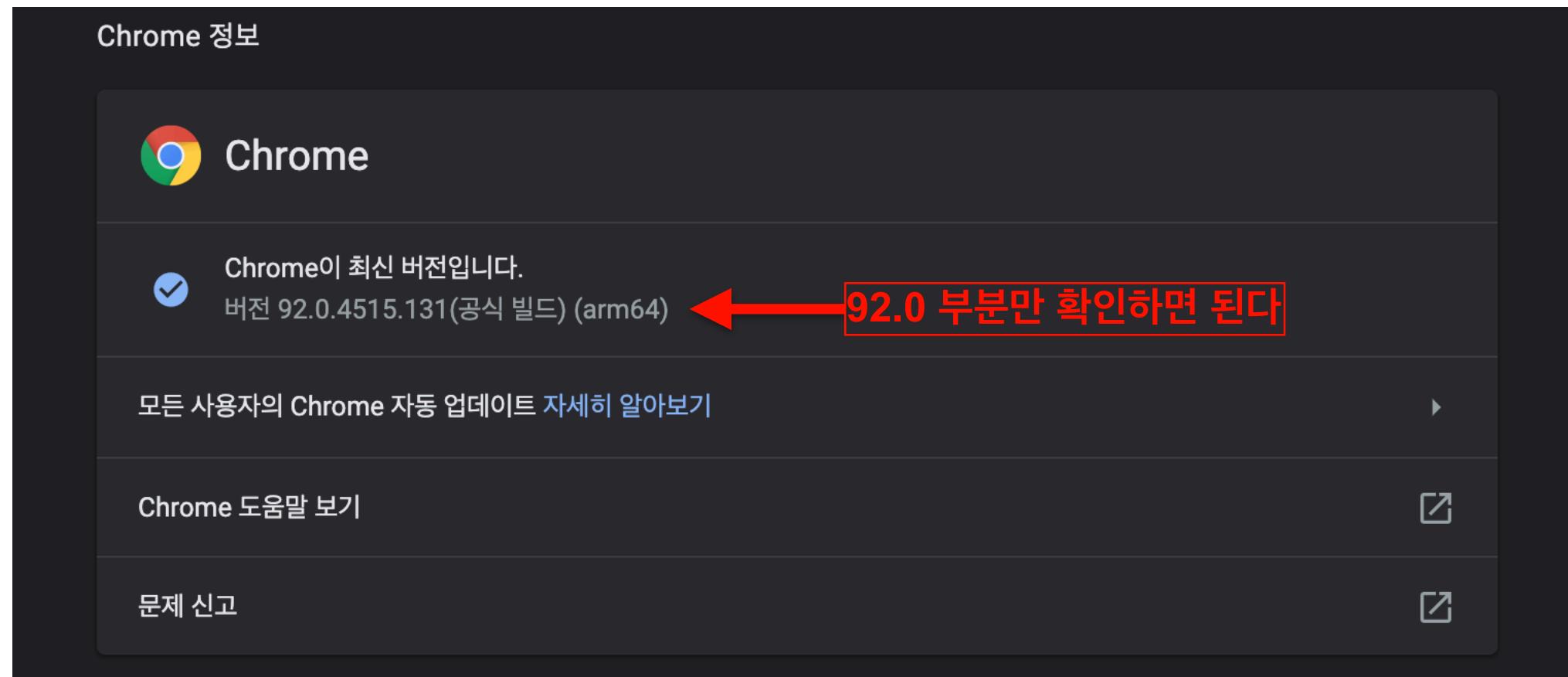
- pip install selenium

Chrome 맞춤설정 및 제어

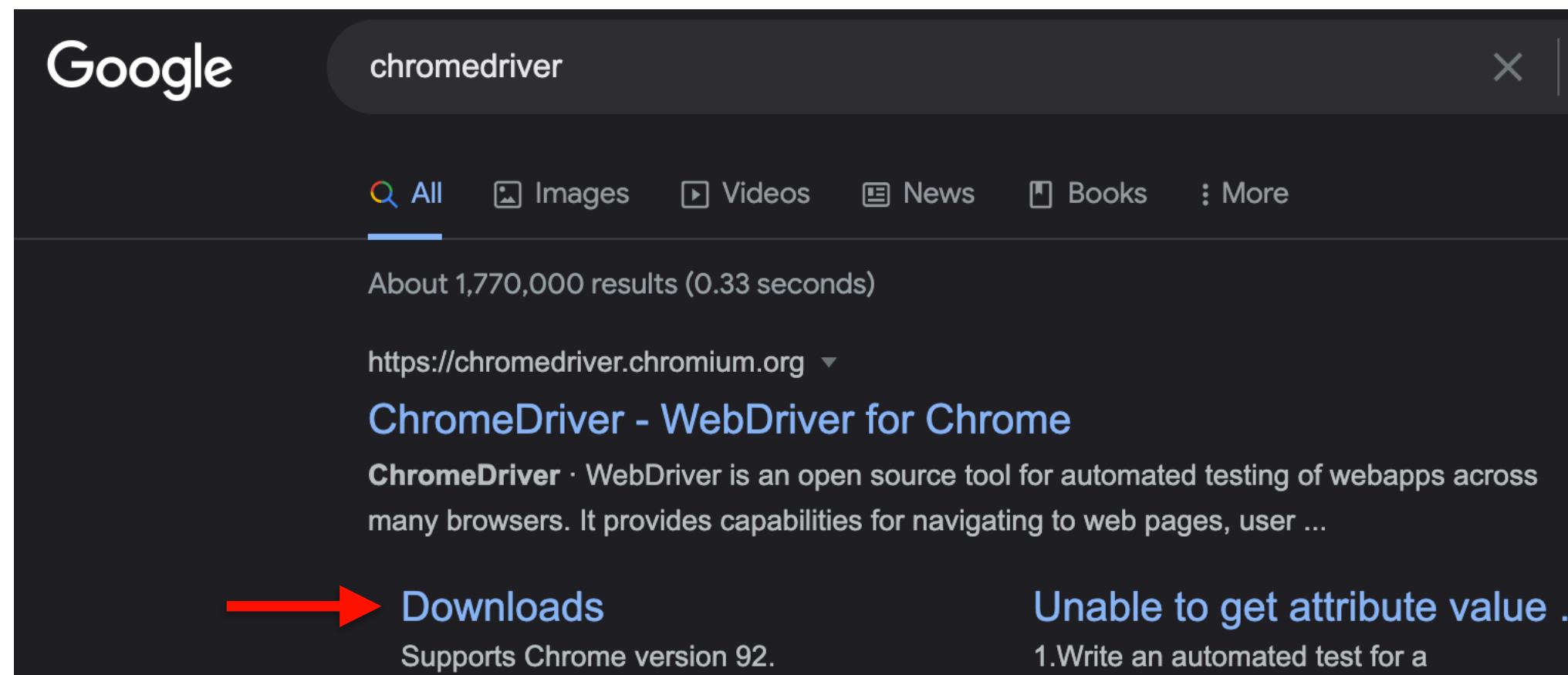
- 먼저 자신의 크롬 버전을 알아야 한다
- 크롬 오른쪽 상단 …(Chrome 맞춤설정 및 제어) 선택



- 도움말에서 Chrome 정보 선택



- Chrome 버전 확인



- 구글에서 chromedriver 검색
- Downloads 페이지 접속

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

ChromeDriver 92.0.4515.107

Supports Chrome version 92

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

ChromeDriver 92.0.4515.43

Supports Chrome version 92

- Resolved issue 3389: Host validation for ChromeDriver requests

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

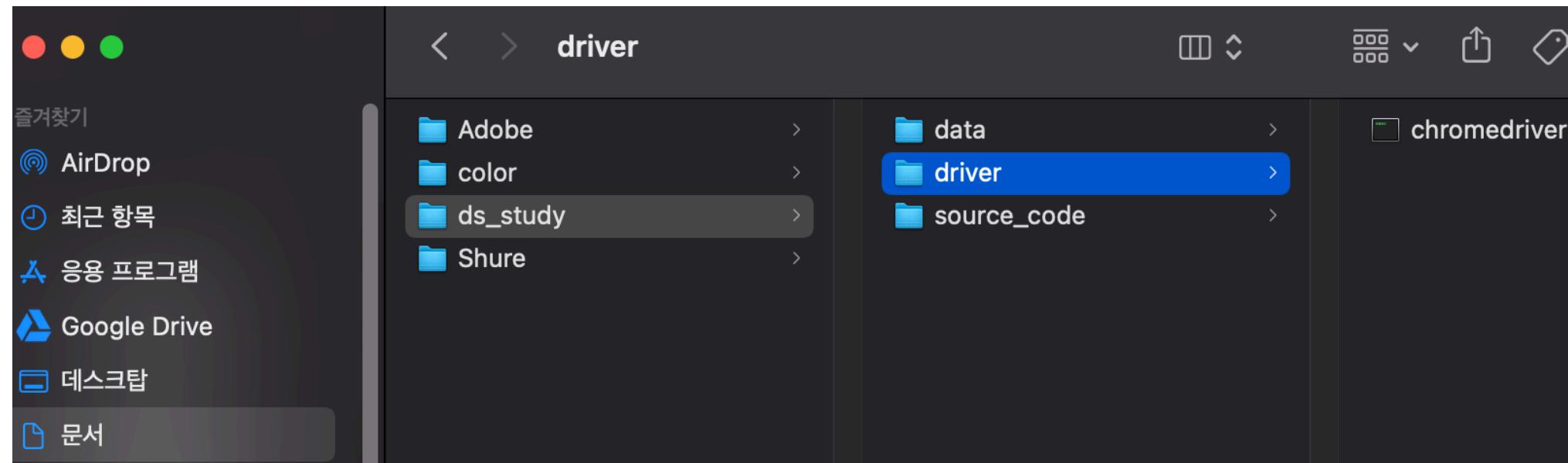
ChromeDriver 91.0.4472.101

- <https://chromedriver.chromium.org/downloads>
- 크롬 버전 앞 부분 숫자에 맞는 것 중 선택

Index of /92.0.4515.107/

Name	Last modified	Size	ETag
Parent Directory		-	
 chromedriver_linux64.zip	2021-07-29 06:57:55	5.70MB	7a2d4054ba1bc7e05b6f206c8f130b48
 chromedriver_mac64.zip	2021-07-29 06:57:58	7.74MB	ec791bc14f87c826d68f54c89bda1207
 chromedriver_mac64_m1.zip	2021-07-29 06:58:00	7.09MB	f8391dca45556f3a371e1796db46c827
 chromedriver_win32.zip	2021-07-29 06:58:02	5.66MB	c4bef54fef659a07324b9f98724430c4
 notes.txt	2021-07-29 06:58:07	0.00MB	f7b7788702eb882f811d88096a23029c

- 사용중인 OS 버전에 맞춰서 다운로드
- 다운받은 후 압축을 풀어준다



- ds_study - source_code가 있는 폴더에 driver 폴더 생성
- driver 폴더 안에 **chromedriver** 파일을 넣어준다

셀레니움 기초

- 셀레니움을 이용해서 처음으로 접근해보기

```
from selenium import webdriver

driver = webdriver.Chrome('./chromedriver')
driver.get('https://pinkwink.kr/')
```

- 크롬드라이버 실행
- webdriver.Chrome 명령으로 크롬 드라이버의 경로 지정
- get 명령으로 접근하고 싶은 주소 지정

- 새로운 크롬이 나타나면서 지정된 웹 주소에 접근함

PinkWink :: PinkWink

pinkwink.kr

Chrome이 자동화된 테스트 소프트웨어에 의해 제어되고 있습니다.

PinkWink HOME

PinkWink 프로필 아기 미바워 여행기 연재 목록 GuestBook 관리자

NOTICE

▷ - 스팸 및 댓글 자동 삭제에 대해 -
▷ - 질문하실 때 주의사항 -

MORE+

Google 지원

Posts by PinkWink (1244)

Robot (157)

- ↳ Project (41)
- ↳ Robot Program - ROS (64)
- ↳ Robot Module (16)
- ↳ Block Coding (13)
- ↳ Reference (23)

많고 많은 캠핑장. 딱 나에게 맞는 캠핑장 추천해 드립니다.

이 글은 2021년 봄부터 이른 여름까지 수업을 한 패스트캠퍼스 성수 스쿨(지금은 스노우볼)에서 수업한 데이터사이언스 스쿨 17기의 프로젝트 중 하나를 소개하는 글입니다. 부족한 강사를 만나 더 좋은 성과를 낼 수 있었지만 그러지 못한 것이 미안할 따름입니다. 이번에는 요즘 ...

Theory/Project | 2021. 8. 4. 08:00

맥 Mac M1에서 Tensorflow 2.5 설치

이번에는 애플의 정말 멋진 노트북 M1 노트북에서 Tensorflow 2.5를 설치하는 방법을 이야기하려고 합니다. 사실 매우 쉽지만 좀더 편하게 따라 오실 수 있도록 글을 만들었습니다. 이 글은 맥 m1 노트북에 아직 파이썬을 위한 어떤 개발환경이 설치되지 않았다고 가정하고 있습니다...

Software/Mac OS | 2021. 7. 26. 08:52

Copyright zerobase Corp. All Rights Reserved

스크롤 가능한 높이를 가져오자

```
In [3]: last_height = driver.execute_script("return document.body.scrollHeight")  
last_height
```

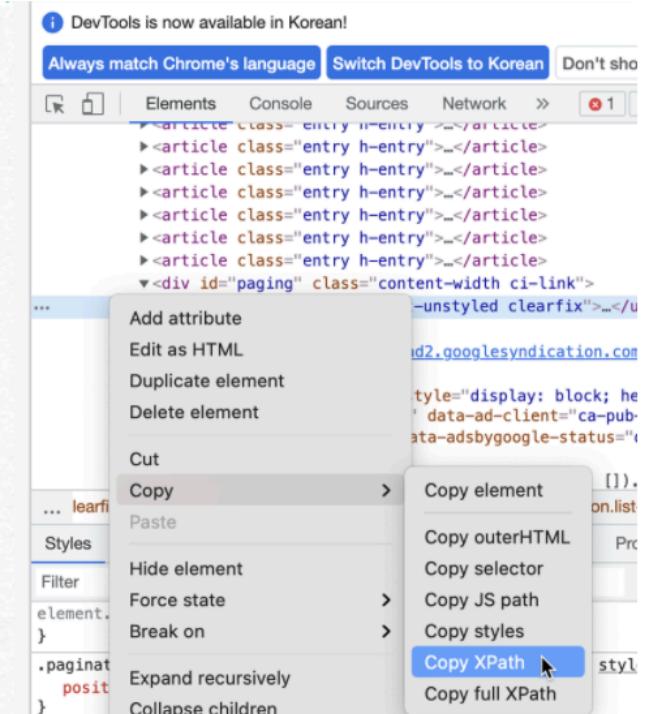
```
Out[3]: 2797
```

화면 스크롤

```
In [4]: driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
```

화면 페이지를 지정하는 지점의 xpath를 찾자

오늘은 ROS melodic에서 카메라를 calibration하는 과정을 설명하려고 합니다. 카메라 영상을 어떻게 사용하는지에 따라 캘리브레이션이 필요할 때가 있는데요. 어떻게 하는지 간...



그렇게 찾은 xpath 지점까지 스크롤하는 코드

```
In [6]: from selenium.webdriver import ActionChains

some_tag = driver.find_element_by_xpath('' '//*[@id="paging"]/ul''')

action = ActionChains(driver)
action.move_to_element(some_tag).perform()
```

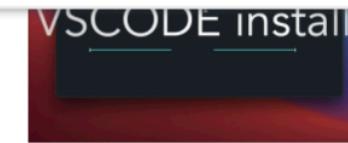
그 결과

← → C pinkwink.kr

Chrome이 자동화된 테스트 소프트웨어에 의해 제어되고 있습니다.

세팅하는 과정을 이야기했었는데요. pinkwink.kr/1354?category=600041 Mac M1 노트북 (Mac silicon) 터미널 세팅 - iTerm2 요즘 노트북 세계에 한 가지 큰 이슈라면 애플에서 나온 맥 m1 노...

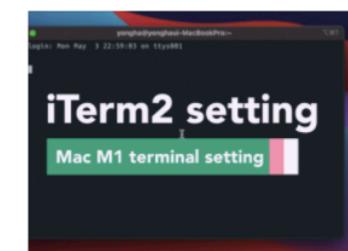
Software/Mac OS | 2021.5.10.10:25



Mac M1 노트북 (Mac silicon) 터미널 세팅 - iTerm2

요즘 노트북 세계에 한 가지 큰 이슈라면 애플에서 나온 맥 m1 노트북이 아닐까요. 정말 많은 이슈를 만들고 있는 것 같습니다. 저는 예전에 맥북 사용에서 해두면 편한 몇 가지 설정을 이야기한 적이 있습니다. pinkwink.kr/1186?category=600041 맥 설정 - 초보자를 위한 가이드 맥...

Software/Mac OS | 2021.5.9.08:00



ROS 카메라 캘리브레이션 수행하기

오늘은 ROS melodic에서 카메라를 calibration하는 과정을 설명하려고 합니다. 카메라 영상을 어떻게 사용하는지에 따라 캘리브레이션이 필요할 때가 있는데요. 어떻게 하는지 간단히 보여 드릴까 합니다. roscore를 실행하고~ 사용하시는 카메라를 기동하면 됩니다. 저는 최근에 ...

Robot/Robot Program - ROS | 2021.4.27.08:00



1

2

3

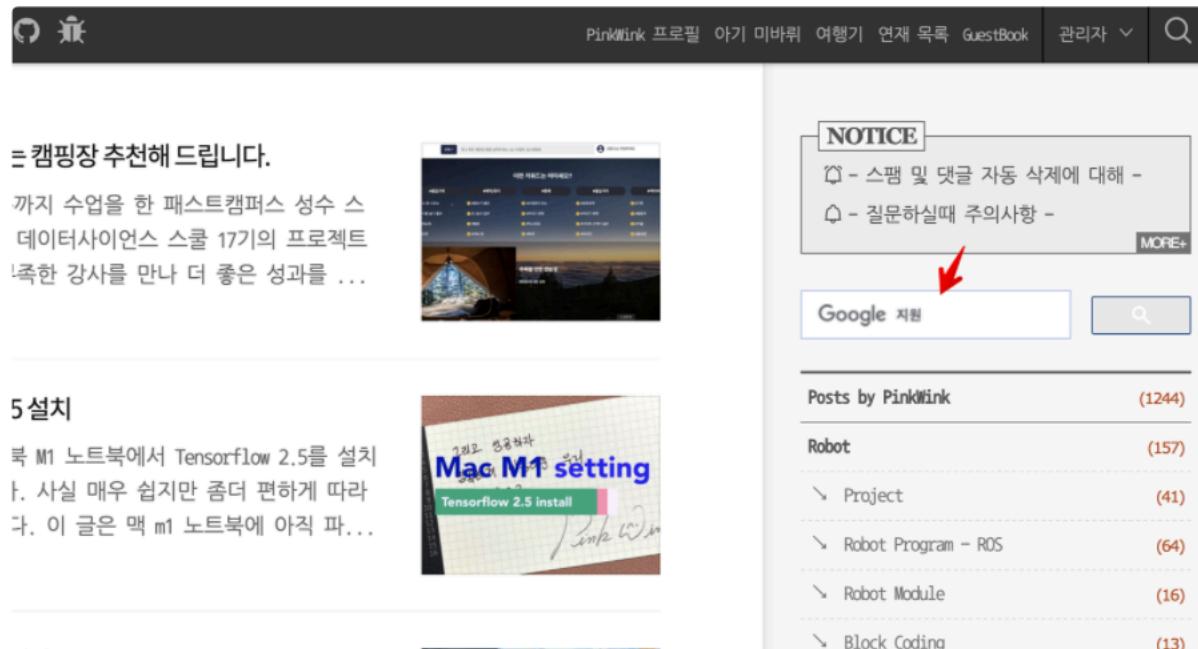
4

...

125

다음

id로 찾기



The screenshot shows a Korean blog post with a search bar at the top. The search bar has the ID 'gsc-iw-id1'. A red arrow points from the DevTools code block below to the search bar's input field.

DevTools is now available in Korean!

Always match Chrome's language Switch DevTools to Korean Don't show again

Elements Console Sources Network ...

```
<div class="gsc-input-box" id="gsc-iw-id1">
  <table cellspacing="0" cellpadding="0" id="gs_id50" class="gstl_50 gsc-input" style="width: 100%; padding: 0px;">
    <tbody>
      <tr>
        <td class="gsib_a" style="width: 100%; padding: 0px; border: none; margin: 0px; height: 1.25em; background-color: #fff; border-bottom: 1px solid #ccc; font-size: 1em; line-height: 1.25em; padding-left: 0.25em; padding-right: 0.25em; position: relative; z-index: 1; ">
          <input autocomplete="off" type="text" size="10" class="gsc-input" name="search" title="검색" id="gsc-i-id1" dir="ltr" spellcheck="false" style="width: 100%; padding: 0px; border: none; margin: 0px; height: 1.25em; background-color: #fff; border-bottom: 1px solid #ccc; font-size: 1em; line-height: 1.25em; padding-left: 0.25em; padding-right: 0.25em; position: relative; z-index: 1; ">
        </td>
        <td class="gsib_b"></td>
      </tr>
    </tbody>
  </table>
</div>
```

입력 창에 글자 넣기

```
In [2]: some_tag = driver.find_element_by_id(''gsc-i-id1'')
some_tag.send_keys('data science')
```

그러면 채워짐

Chrome이 자동화된 테스트 소프트웨어에 의해 제어되고 있습니다.

많고 많은 캠핑장. 딱 나에게 맞는 캠핑장 추천해 드립니다.

이 글은 2021년 봄부터 이른 여름까지 수업을 한 패스트캠퍼스 성수 스쿨(지금은 스노우볼)에서 수업한 데이터사이언스 스쿨 17기의 프로젝트 중 하나를 소개하는 글입니다. 부족한 강사를 만나 더 좋은 성과를 낼 수 있었지만 그러지 못한 ...

Theory/Project | 2021. 8. 4. 08:00

맥 Mac M1에서 Tensorflow 2.5 설치

이번에는 애플의 정말 멋진 노트북 M1 노트북에서 Tensorflow 2.5를 설치하는 방법을 이야기하려고 합니다. 사실 매우 쉽지만 좀더 편하게 따라 오실 수 있도록

NOTICE

- ▷ - 스팸 및 댓글 자동 삭제에 대해 -
- ▷ - 질문하실때 주의사항 -

MORE+

data science

Posts by PinkWink (1244)

Robot (157)

↳ Project (41)

다시 검색 버튼을 찾음

The screenshot shows a search results page from Google. In the search bar, there is a blue button labeled '검색' (Search). A red arrow points to this button. To the right of the search bar, the DevTools panel is open, showing the HTML code for the search button. The code is as follows:

```

<input autocomplete="off" type="text" size="10" class="gsc-input" name="search" title="검색" id="gsc-i-id1" dir="ltr" spellcheck="false" style="width: 100%; padding: 0px; border: none; margin: -0.0625em 0px 0px; height: 1.25em; background: url("https://www.google.com/cse/static/images/1x/ko/branding.png") left center no-repeat rg b(255, 255, 255); outline: none;">
</td>
<td class="gsib_b"></td>
</tr>
</tbody>
</div>
</td>
<td class="gsc-search-button">
  <button class="gsc-search-button gsc-search-button-v2"></button> == $0
</td>
<td class="gsc-clear-button"></td>
</tr>
</tbody>
...

```

xpath를 찾음

The screenshot shows a sidebar menu on the left and a context menu open over an element on the right. The sidebar menu items include:

- Posts by PinkWink (1244)
- Robot
 - Project (41)
 - Robot Program - ROS (64)
 - Robot Module (16)
 - Block Coding (13)
 - Reference (23)
- Theory
 - ControlTheory (49)
 - DataScience (50)
 - MachineLearning (22)
 - Project (9)

The context menu is open over a button element with the class "gsc-search-button". The menu options are:

- Add attribute
- Edit as HTML
- Duplicate element
- Delete element
- Cut
- Copy
- Paste
- Hide element
- Force state
- Break on
- Expand recursively
- Collapse children
- Capture node screenshot
- Scroll into view
- Focus
- Copy element
- Copy outerHTML
- Copy selector
- Copy JS path
- Copy styles
- Copy XPath**
- Copy full XPath

The "Copy XPath" option is highlighted in purple. The element's CSS styles are also visible in the developer tools panel.

```

<td class="gsc-search-button-v2">
  <button class="gsc-search-button gsc-search-button-v2">
    ...
  </button>
</td>

```

```

... -box tbody tr
Styles Computed
Filter
element.style {
}
.gsc-search-button-v2:focus {
  border-color: #000000;
  background-color: #000000;
  background-image: none;
  filter: none;
}
.gsc-search-button-

```

default.css:45

버튼을 클릭하는 코드 실행

```
In [6]: xpath = '//*[@id="__gcse_0"]/div/form/table/tbody/tr/td[2]/button'
some_tag = driver.find_element_by_xpath(xpath).click()
```

그러면 이렇게

Chrome이 자동화된 테스트 소프트웨어에 의해 제어되고 있습니다.

[Data Science] Conda 환경을 만들고 기초 모듈 설치하기 - PinkWink

PinkWink > ...



Python의 다양한 버전과 또 엄청~ 다양한 모듈들의 버전과 그들 사이의 관계들로 (모듈을 포함해서) 파이썬을 설치하는 경우의 수가 아주 많이 생길 ...

[Data Science] 서울시민들이 운동을 하지 않는 이유에 ... - PinkWink

PinkWink > ...



안녕하세요. 지난번에 Conda 환경을 세팅하는 이야기를 했구요. 이번에는 첫 번째 프로젝트(라고 부르기 창피하지만)로 서울시민들이 운동을 하지 ...

[Data Science] 서울시 흡연률 데이터에서 연령별 흡연률 ... - PinkWink

PinkWink > ...



이 글은 아주 쉬운 데이터를 실습용으로 사용해서 파이썬과 데이터 사이언스 분야를 공부하는 컨셉의 글입니다. 이번에 사용할 데이터는 서울시 ...

'Theory/DataScience' 카테고리의 글 목록 (2 Page) - PinkWink

PinkWink > ... > 영상 인식으로 월리를 찾아볼까? -mahotas-



TheoryDataScience 검색 결과. 해당 글 50건. [Data Science] 서울시 흡연률 데이터에서 연령별 흡연률 데이터 시각화. 이 글은 아주 쉬운 데이터를 실습용으로 사용 ...

'Theory/DataScience' 카테고리의 글 목록 - PinkWink

현재 화면의 html 코드 가져오기

```
In [8]: from bs4 import BeautifulSoup  
  
req = driver.page_source  
soup = BeautifulSoup(req, 'html.parser')
```

광고를 제외한 검색결과가 나타나는 태그를 찾거나

IBM Data Analyst - Learn Online on Coursera®
<https://www.coursera.org/prof-cert/data-analyst> ▾
 Unlock your potential in **data** analytics. Build job-ready skills for a **data** analyst career.
[▶ Visit Website](#)

Become a Data Scientist. - The #1 Place to Learn Online
<http://www.datacamp.com/> ▾
 Learn **data science** step by step though quick exercises and short videos.

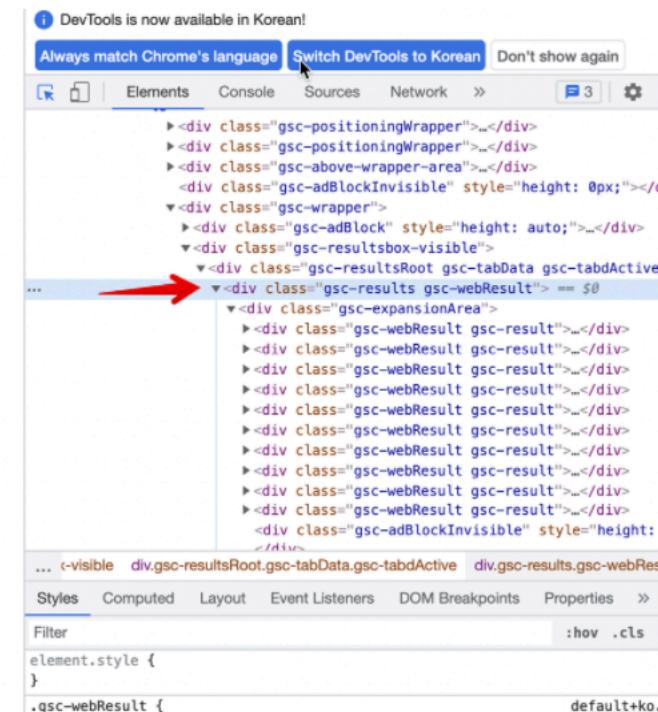
div.gsc-results.gsc-webResul 1070.19 x 1139

[Data Science] Conda 환경을 만들고 기초 모듈 설치하기 - PinkWink
 PinkWink > ...
 Python의 다양한 버전과 또 업첨~ 다양한 모듈들의 버전과 그들 사이의 관계들로 (모듈을 포함해서) 파이썬을 설치하는 경우의 수가 아주 많이 생길 ...

[Data Science] 서울시민들이 운동을 하지 않는 이유에 ... - PinkWink
 PinkWink > ...
 안녕하세요. 지난번에 Conda 환경을 세팅하는 이야기를 했구요. 이번에는 첫 번째 프로젝트(라고 부르기 창피하지만)로 서울시민들이 운동을 하지 ...

[Data Science] 서울시 흡연률 데이터에서 연령별 흡연률 ... - PinkWink
 PinkWink > ...
 이 글은 아주 쉬운 데이터를 실습용으로 사용해서 파이썬과 데이터 사이언스 분야를 공부하는 컨셉의 글입니다. 이번에 사용할 데이터는 서울시 ...

'Theory/DataScience' 카테고리의 글 목록 (2 Page) - PinkWink

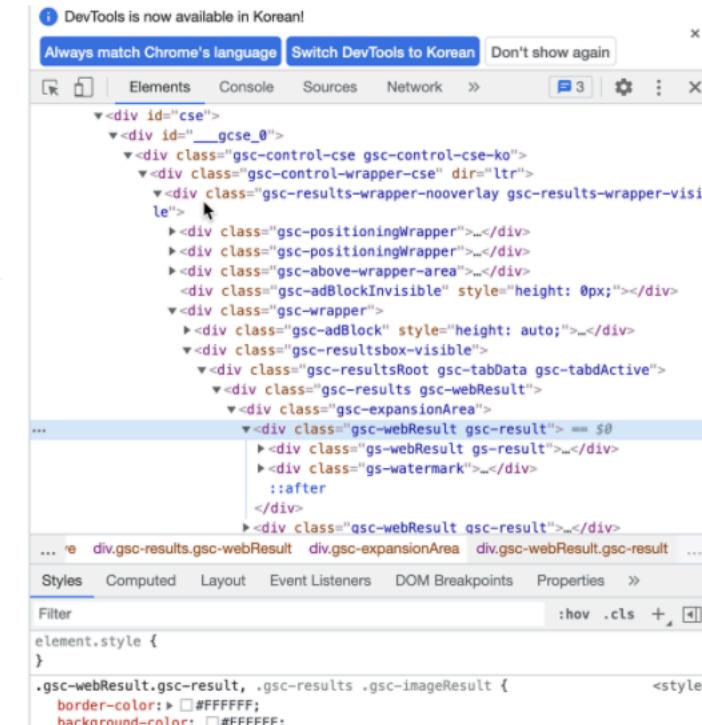


Selenium 기초

검색결과 하나하나를 의미하는 태그를 찾을 수 있음

The screenshot shows a search results page from Coursera. The results are as follows:

- [BM Data Analyst - Learn Online on Coursera®]**
Unlock your potential in **data** analytics. Build job-ready skills for a **data** analyst career.
[Visit Website](https://www.coursera.org/prof-cert/data-analyst)
- Become a Data Scientist. - The #1 Place to Learn Online**
[Visit](http://www.datacamp.com/)
Learn **data science** step by step through quick exercises and short videos.
[Visit Website](#)
- [Data Science] Conda 환경을 만들고 기초 모듈 설치하기 - PinkWink**
PinkWink > ...
Python의 다양한 버전과 또 엄청~ 다양한 모듈들의 버전과 그들 사이의 관계들로 (모듈을 포함해서) 파이썬을 설치하는 경우의 수가 아주 많이 생길 ...
- [Data Science] 서울시민들이 운동을 하지 않는 이유에 ... - PinkWink**
PinkWink > ...
안녕하세요. 지난번에 Conda 환경을 세팅하는 이야기를 했구요. 이번에는 첫 번째 프로젝트(라고 부르기 창피하지만)로 서울시민들이 운동을 하지 ...
- [Data Science] 서울시 흡연률 데이터에서 연령별 흡연률 ... - PinkWink**
PinkWink > ...
이 글은 아주 쉬운 데이터를 실습용으로 사용해서 파이썬과 데이터 사이언스 분야를 공부하는 컨셉의 글입니다. 이번에 사용할 데이터는 서울시 ...
- Theory/DataScience' 카테고리의 글 목록 (2 Page) - PinkWink**
PinkWink > ... > 영상 인식으로 월리를 찾아볼까? -mahotas-
Theory/DataScience' 카테고리에 채다 글 더하기 「Data Science」 서울시 흡연률

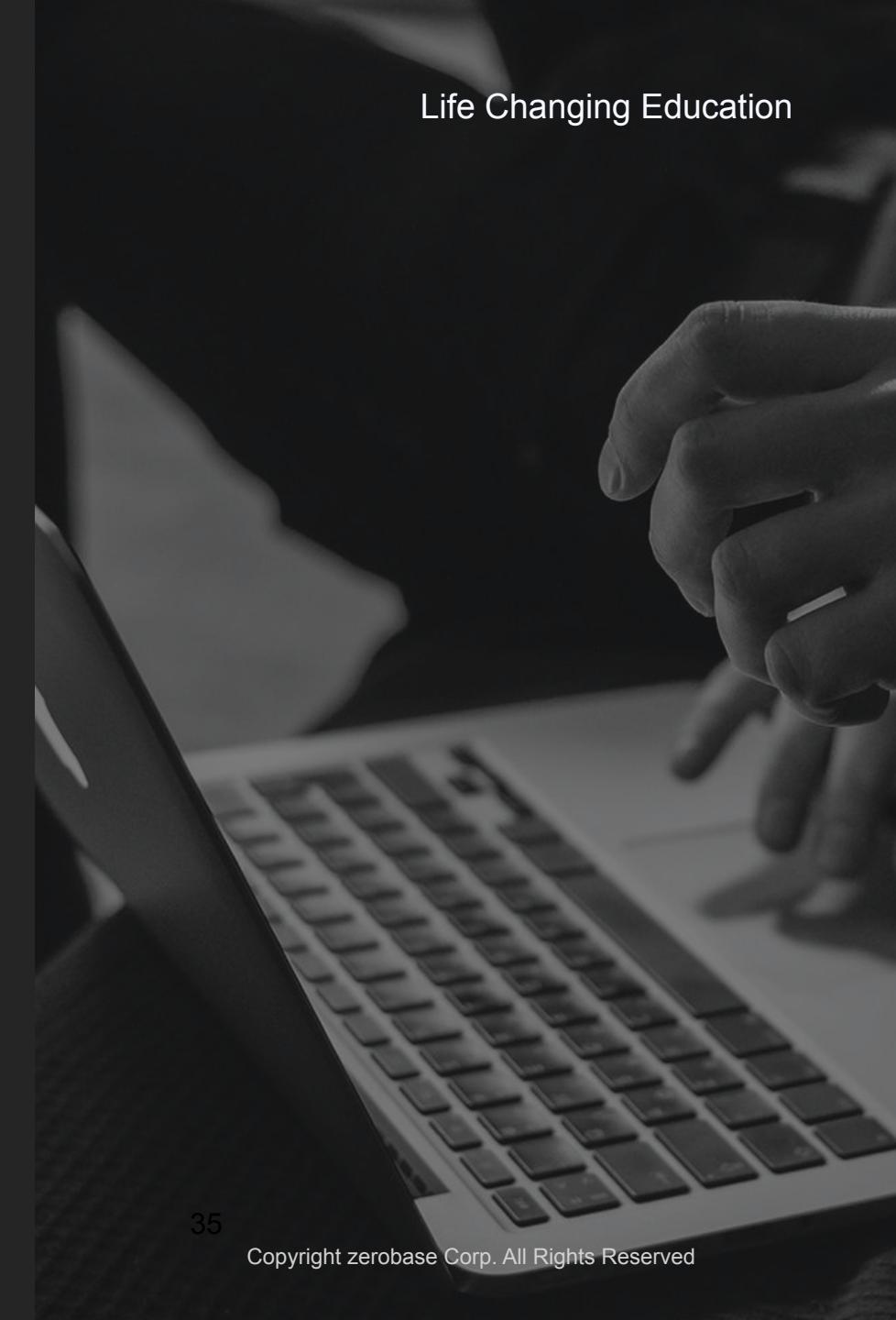


그럼 우리는 이렇게 접근 가능함

```
In [11]: result = soup.find_all('div', 'gsc-webResult gsc-result')
result[0]
```

```
Out[11]: <div class="gsc-webResult gsc-result"><div class="gs-webResult gs-result"><div class="gsc-thu
mbnail-inside"><div class="gs-title"><a class="gs-title" data-ctorig="https://pinkwink.kr/128
6" data-cturl="https://www.google.com/url?client=internal-element-cse&cx=partner-pub-2982
459700139266:1067827452&q=https://pinkwink.kr/1286&sa=U&ved=2ahUKEwjOjY2H6ZPzAhUP
E6YKHSnIDj0QFnoECAAQAO&usg=A0vVaw3xmROhq5xpVUgi4v5aQFD3" dir="ltr" href="https://pinkwin
k.kr/1286" target="_blank">[Data Science] Conda 환경을 만들고 기초 모듈 설치하기 - PinkWink</a></div>
</div><div class="gsc-url-top"><div class="gs-bidi-start-align gs-visibleUrl gs-visibleUrl-sh
ort" dir="ltr">pinkwink.kr</div><div class="gs-bidi-start-align gs-visibleUrl gs-visibleUrl-l
ong" dir="ltr" style="word-break:break-all;">https://pinkwink.kr/1286</div><div class="gs-bid
i-start-align gs-visibleUrl gs-visibleUrl-breadcrumb"><span>PinkWink</span><span> > ...</span
></div></div><div class="gsc-table-result"><div class="gsc-table-cell-thumbnail gsc-thumbnai
l"><div class="gs-image-box gs-web-image-box gs-web-image-box-landscape"><a class="gs-image"
data-ctorig="https://pinkwink.kr/1286" data-cturl="https://www.google.com/url?client=internal
-element-cse&cx=partner-pub-2982459700139266:1067827452&q=https://pinkwink.kr/1286&am
p;sa=U&ved=2ahUKEwjOjY2H6ZPzAhUPE6YKHSnIDj0QFnoECAAQAO&usg=A0vVaw3xmROhq5xpVUgi4v5aQF
D3" href="https://pinkwink.kr/1286" target="_blank"><img alt="미리보기 이미지" class="gs-image" s
rc="https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GctozrtmnTf6zuix2++Rkuuv-uaZXTDwvTXDg
"
```

셀프 주유소 정말 저렴하나요? - 데이터 확보하기 위한 작업





싼 주유소찾기

국내유가통계

유가관련정보

불법행위공표

이용안내

로그인

회원가입

기름값 아끼는 지름길
한국석유공사 오피넷에 있습니다.

국내유가 안정과 국민경제 향상에 앞장서겠습니다.

오늘의 유가 (2021.08.10)	휘발유 경유 LPG	전국평균 (원/리터) 1647.29 ▼0.03 최저가 1544 최고가 2493	서울평균 (원/리터) 1731.69 ▼0.21 최저가 1595 최고가 2493
-------------------------------	------------------	--	--

- <https://www.opinet.co.kr/user/main/mainView.do>
- 대한민국 주유 가격을 알아보는 사이트

The screenshot shows the Opinet website's search interface for fuel stations. At the top, there is a navigation bar with the Opinet logo, a search bar labeled "싼 주유소 찾기" (Find Cheap Fuel Station), and several menu items: 국내유가통계 (Domestic Oil Price Statistics), 유가관련정보 (Oil Price Related Information), 불법행위공표 (Violation Disclosure), 이용안내 (Usage Instructions), and two buttons for 로그인 (Login) and 회원가입 (Sign Up). Below the search bar, there is a sidebar with options: 지역별 (Regional), 경로별 (Route-based), 도로별 (Road-based), and 면세유 (Duty-free). The main content area is divided into two columns. The left column contains sections for 주유소/충전소 (Fuel Stations/Charging Stations) and 정유사/대리점/LPG (Refineries/Agents/LPG). The right column contains sections for 국내유가동향 (Domestic Oil Price Trends), 국제유가 (International Oil Prices), 전자상거래 (E-commerce), 유류세 (Fuel Tax), 단위환산표 (Conversion Table), and 유가정보 API (Oil Price Information API). Two red arrows point to the "싼 주유소 찾기" search bar and the "지역별" option in the sidebar.

- 싼 주유소 찾기에서 지역별 선택



지역별	경로별	도로별	면세유
주유소			
총전소			
지역: 서울 ▾ 성동구 ▾ 읍/면/동 ▾ 검색방법: <input checked="" type="radio"/> 지역 <input type="radio"/> 지도 <input type="radio"/> 상호 <input type="radio"/> 반경 형태: <input checked="" type="checkbox"/> 일반 <input checked="" type="checkbox"/> 셀프 <input checked="" type="checkbox"/> 불법 <input checked="" type="checkbox"/> 인증 <input checked="" type="checkbox"/> 전산 상표: 전체 <input checked="" type="checkbox"/> SK <input checked="" type="checkbox"/> GS <input checked="" type="checkbox"/> GS칼텍스 <input checked="" type="checkbox"/> Hyundai Oilbank <input checked="" type="checkbox"/> GS OIL <input checked="" type="checkbox"/> 암동 <input checked="" type="checkbox"/> PB 부가정보: <input type="checkbox"/> 세차장 <input type="checkbox"/> 경정비 <input type="checkbox"/> 편의점 <input type="checkbox"/> 24시간			
조회			



싼 주유소찾기

국내유가통계

유가관련정보

불법행위공표

- <https://www.opinet.co.kr/searRgSelect.do>
- 사이트의 구조 확인
- 목표 데이터는 브랜드, 가격, 셀프 주유 여부, 위치

지역별 경로별 도로별 면세유

주유소 충전소

지역 서울 ▼ 성동구 ▼ 읍/면/동 ▼

검색방법 지역 지도 상호 반경

형태 일반 셀프 불법 인증 전산

상표 전체 SK GS Hyundai Oilbank S-OIL 알뜰 PB

부가정보 세차장 경정비 편의점 24시간

조회

- 지역에서 광역시도와 시구 정보를 입력
- 지도가 바뀌고, 또 잠시 후 주유소 정보가 로딩

사이트 구조 분석

고급휘발유	보통휘발유	경유	실내등유
• 검색결과 (총 17개)    			
주유소명	휘발유	경유	
평균가격	1763	1556	
 (주)서울에너지 ... 	1639	1434	
 정호주유소	1648	1438	
 KLP제2주유소  	1667	1447	
 현대오일뱅크(주)...  	1668	1468	
 아이콘주유소	1698	1498	
 현대오일뱅크(주)직영... 	1727	1527	
 경진주유소 	1728	1518	
 삼우주유소 	1728	1518	
 SK에너지(주) ...  	1729	1519	
 지에스칼텍스(주) 도...  	1734	1524	



- 셀프여부를 알려주고 있고, 휘발유 가격도 있지만...
- 상표와 주소 정보를 다루기가 약간 까다로울 것 같다

- 엑셀파일에는 원하는 정보가 모두 존재한다

주유소 충전소

• 지역 서울 성동구 읍/면/동

• 검색방법 지역 지도 상호 반경

• 형태 일반 셀프 불법 인증 전산

• 상표 전체 SK 에너지 GS 카드 Hyundai Oilbank S-OIL 알뜰 PB

엑셀저장

• 서울 => 각 구를 번갈아 선택

• 엑셀파일을 다운로드 받음
• 다운로드 받은 엑셀을 분석



지역별 **경로별** **도로별** **면세유**

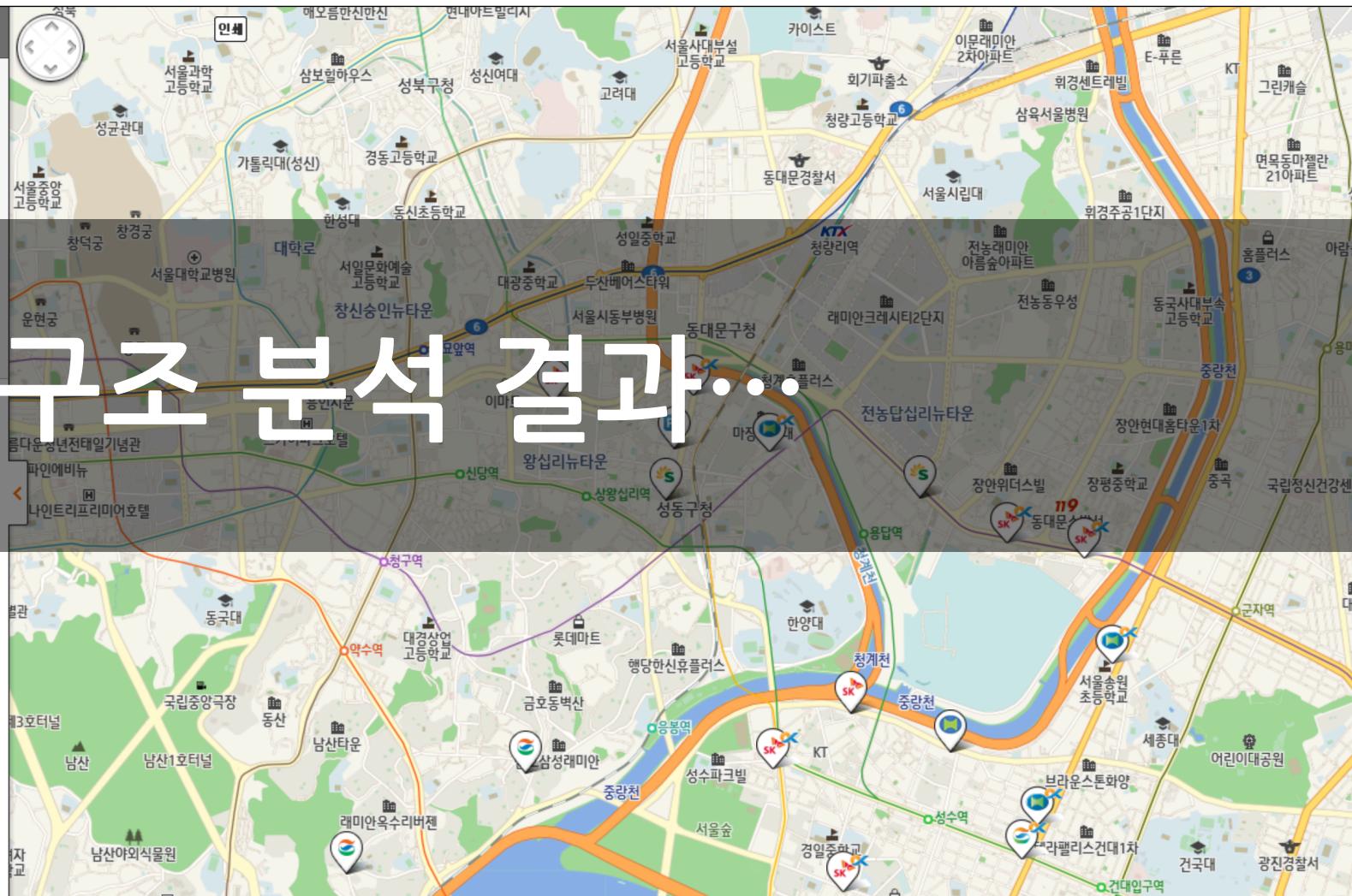
주유소 **충전소**

- 지역: 서울 ▾ 성동구 ▾ 읍/면/동 ▾
- 검색방법: 지역 지도 상호 반경
- 형태: 일반 셀프 불법 인증 전산
- 상표: 전체 SK GS칼텍스 Hyundai Oilbank S-OIL 양우 PB
- 부가정보: 세차장 경경비 편의점 24시간

주유소명 **휘발유** **경유**

주유소명	휘발유	경유
평균가격	1763	1556
(주)서울에너지 ...	1639	1434
PB 정호주유소	1648	1438
KLP제2주유소	1667	1447
현대오일뱅크(주)	1668	1468
아이콘주유소	1698	1498
현대오일뱅크(주)직영	1727	1527
SK 삼우주유소	1728	1518
경진주유소	1728	1518
SK에너지(주)	1729	1519
지에스칼텍스(주)	1734	1524

엑셀저장



사이트 구조 분석 결과 어려운 점

zero-base /

지역별 경로별 도로별 면세유

주유소 충전소

• 지역 서울 ▼ 성동구 ▼ 읍/면/동 ▼

• 검색방법 지역 지도 상호 반경

• 형태 일반 셀프 불법 인증 전산

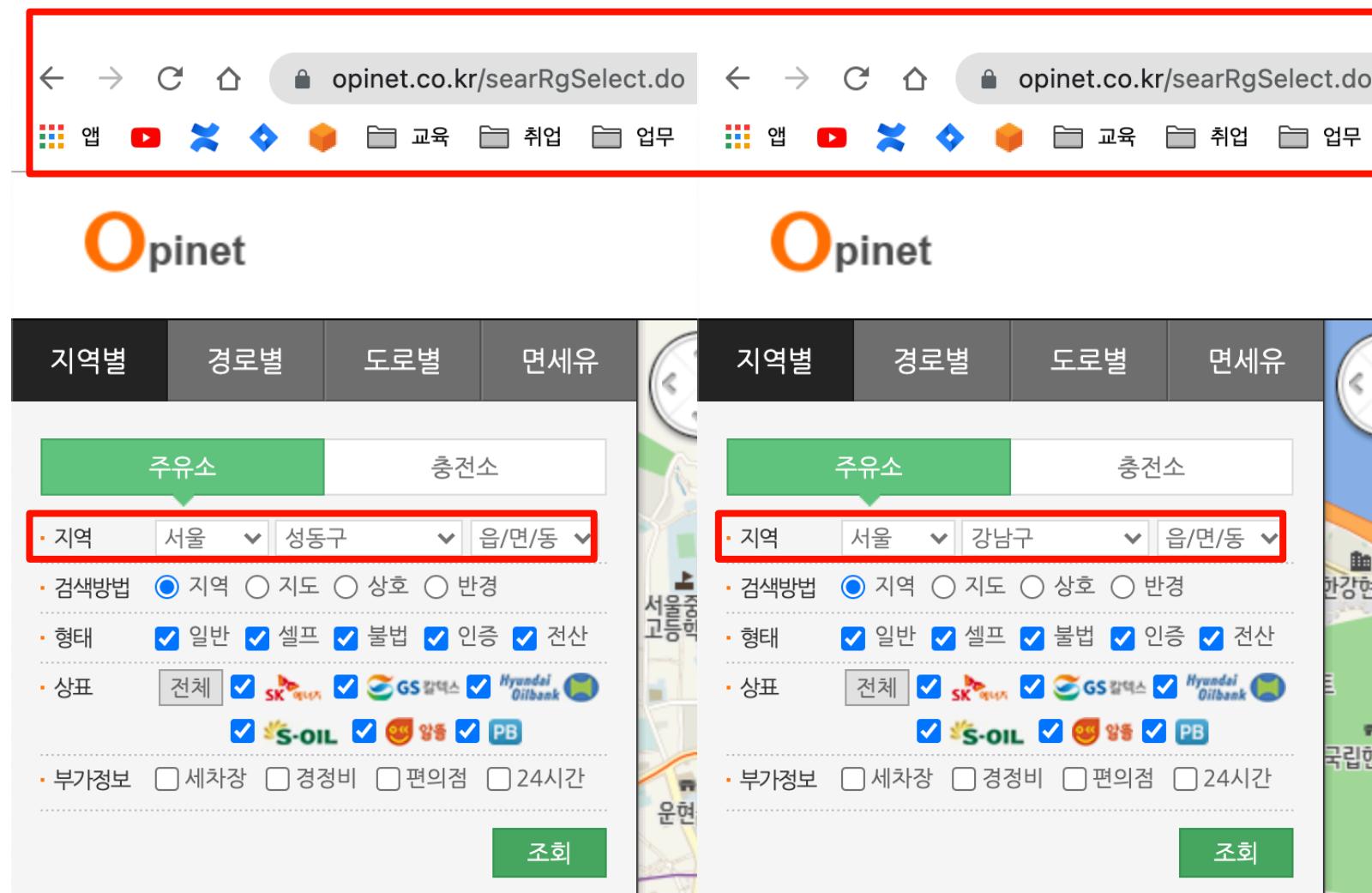
• 상표 전체 SK 에너지 GS 카드 Hyundai Oilbank 
 S-OIL 알뜰 PB

• 부가정보 세차장 경정비 편의점 24시간

조회

- 마우스를 클릭하여 지역을 선택해야 한다

사이트 구조 분석 결과 어려운 점



- 지역을 선택해도 드러난 웹 주소가 변경되지 않는다

고급휘발유	보통휘발유	경유	실내등유
검색결과 (총 36개)			
평균가격	1862	1652	
오일프러스 셀프 (3)	1653	1444	
현대오일뱅크 도곡...	1663	1454	
(주)보성 세곡주...	1675	1459	
현대오일뱅크(주)...	1678	1469	
대교주유소	1685	1490	
방죽주유소 (3)	1687	1489	
혁대우 일백크(주)	1689	1459	

```

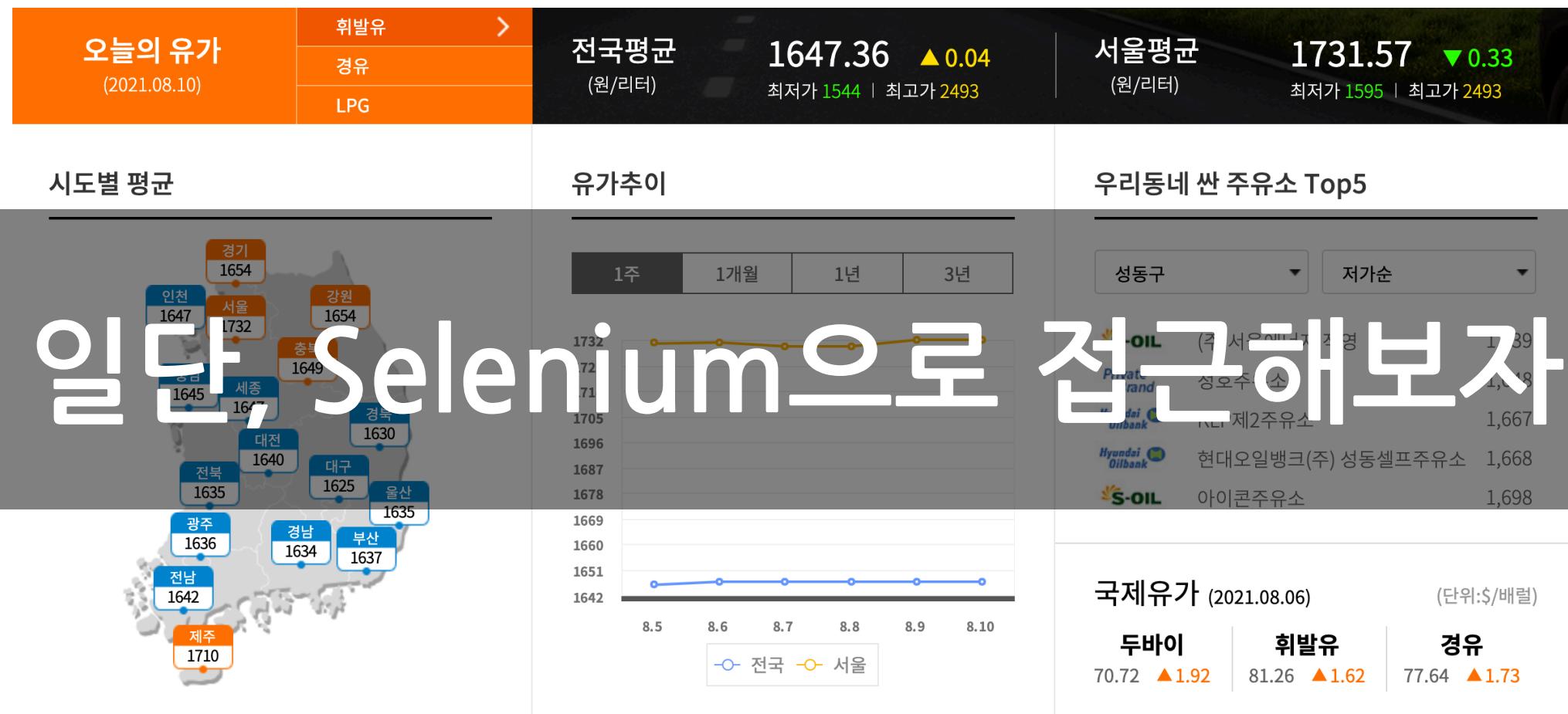
<!-- 일반휘발유 -->
▼<tr>
  ...
    <td class="rlist"> == $0
      
      ▶<a href="javascript:fn_osPop('', '1821', '1653', '1444', '99999', '', '2021-08-10 07:05:...환로2651 (도곡동)', 'Y', 'Y', 'N', ' ', 'N', 'A0009835', '2130694673', 'N', 'N', 'N');">...</a>
      ▶<span class="ic ico_self">...</span>
    </td>
    <td class="price"> 1653 </td>
    <td class="price">...</td>
  ...

```

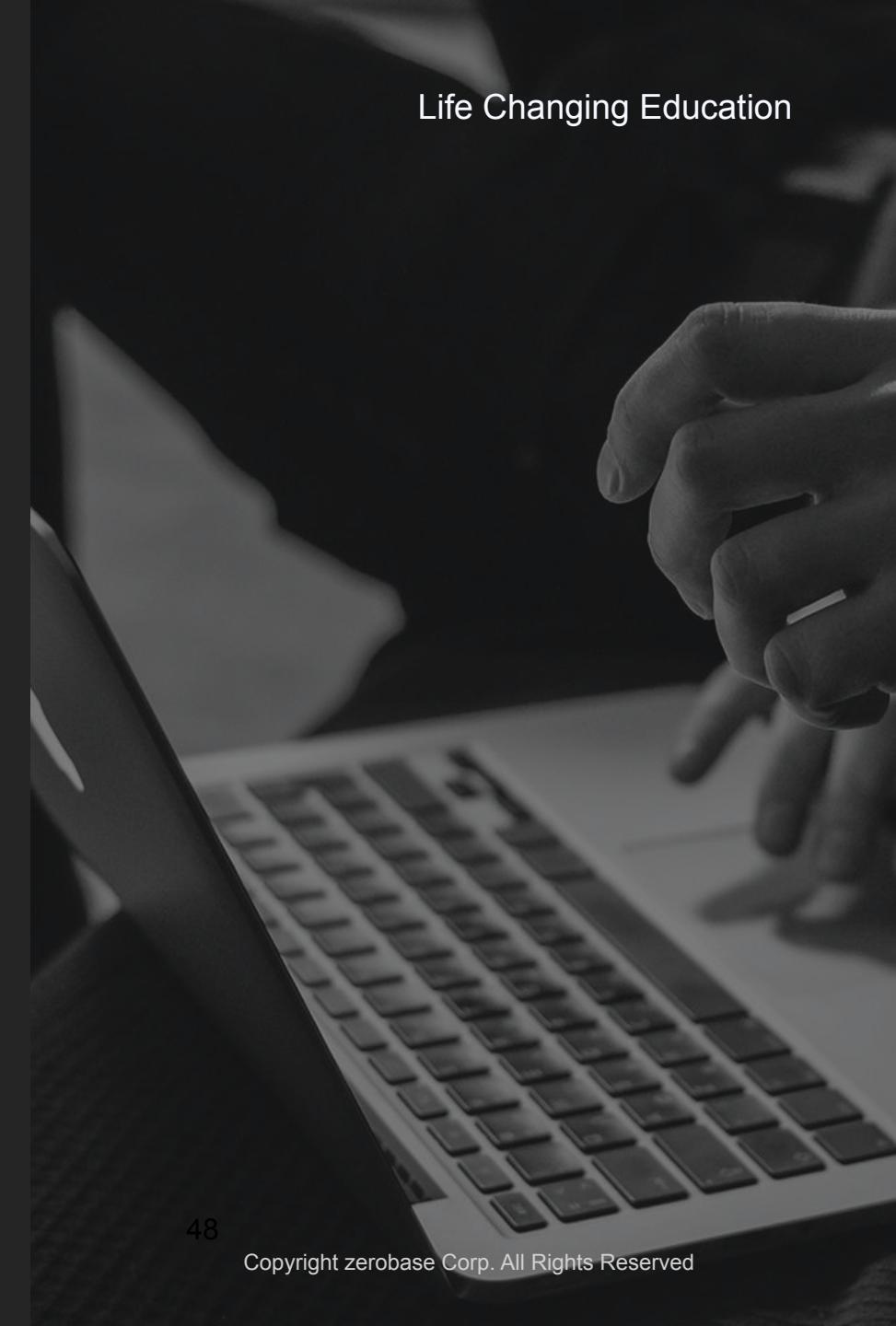
- 생각보다는 HTML 소스에서 원하는 정보를 얻는 것도 쉬워보이지 않는다

지금까지 내용을 정리 해보면?

- 서울시 주유소를 대상으로 **셀프 주유소가 저렴한지 확인하는 목표**
- 서울시 등록된 주유소 전체의 주유 가격을 확인할 수 있는 사이트 발견
- 그러나… 험난한 과정이 예상됨
- 그 중에서도 제일 문제는…
Beautiful Soup으로는 바로 접근하기 어려워 보인다는 것



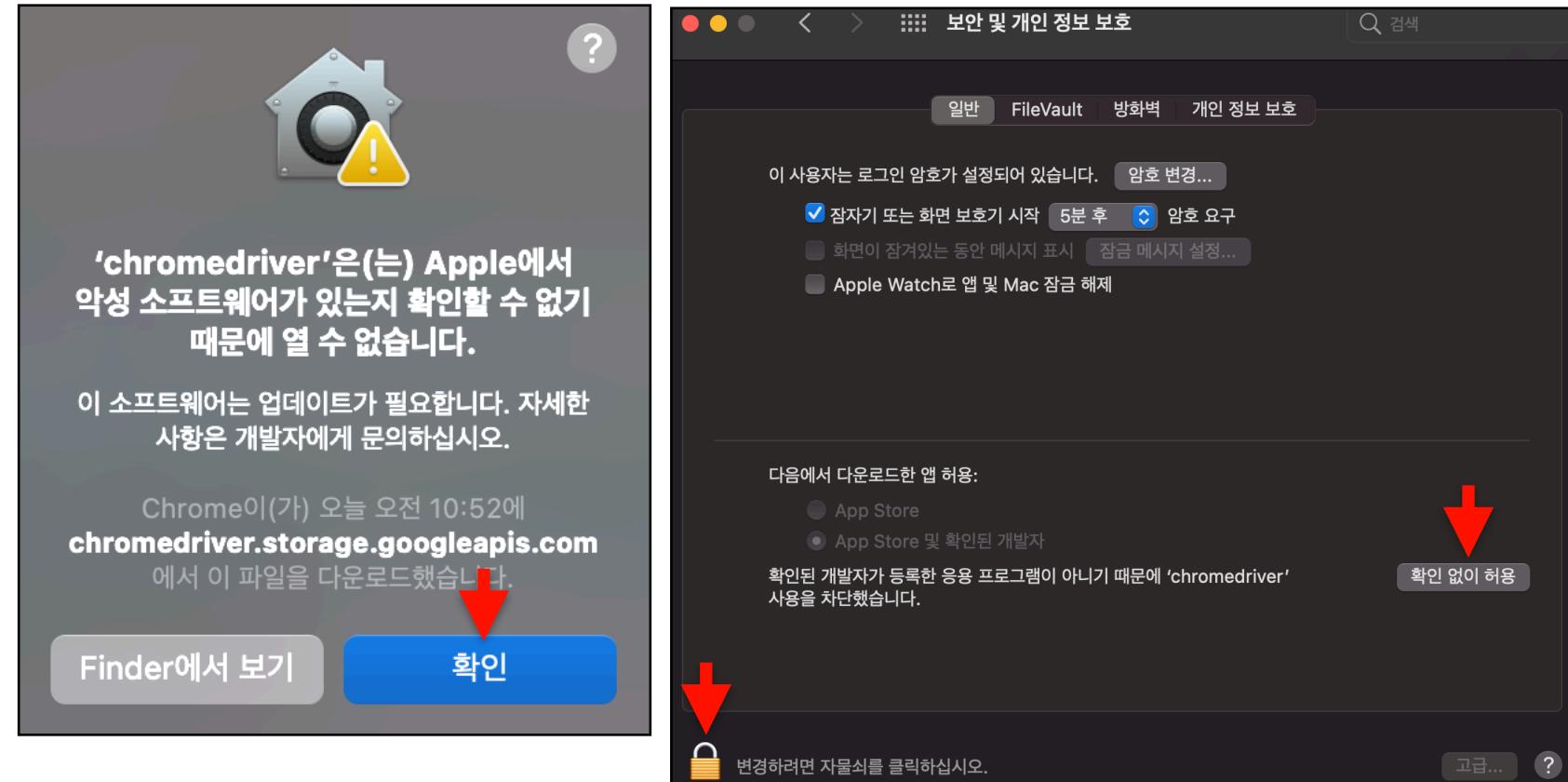
셀레니움으로 접근해보자



주유 가격을 확인해보자

```
In [2]: from selenium import webdriver  
  
driver = webdriver.Chrome("../driver/chromedriver")  
driver.get("http://www.opinet.co.kr")
```

- Selenium에서, Chrome Driver를 이용해서,
- 원하는 웹 주소로 접근합니다.



- Mac에서는 크롬드라이버 권한을 줘야한다

한 번 접근해 보기

zero-base /



The screenshot shows a Jupyter Notebook interface with two code cells. The first cell contains the following Python code:

```
driver.execute_script(input_js) time.sleep(3) # 자동화 탐지를 우회하기 위한 delay
driver.find_element_by_id("log.login").click() time.sleep(3)
driver.close()
```

The second cell is titled "주유 가격을 확인해보자" (Check fuel prices) and contains the following code:

```
In [3]: from selenium import webdriver
driver = webdriver.Chrome("../driver/chromedriver")
driver.get("http://www.opinet.co.kr")
```

```
In [3]: driver.get("http://www.opinet.co.kr/searRgSelect.do")
```

- 주유 가격 정보 사이트인 Opinet은 지도 정보를 가지고 있어서 충분한 시간적 여유를 주어야 한다.

```
In [4]: sido_list_raw = driver.find_element_by_id("SIDO_NM0")
sido_list_raw
```

```
Out[4]: <selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="ad6e44d4-2659-4b2f-926d-dc381211c040")>
```

```
In [5]: sido_list = sido_list_raw.find_elements_by_tag_name("option")
sido_list
```

```
Out[5]: [<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="52711562-c76c-4846-ad6f-7149ddaf9ffef2">,
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="5e094668-d82d-4117-b379-6d3f2961c5e5">),
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="a937324b-88cf-4d6b-8d52-f793c84edf70">),
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="56adaaf2-2466-4388-a068-67134f26c86b">),
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="89481cf9-5ee7-4d4c-950d-bc80074c0f90">),
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="a0553352-1000-1-15">,
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="a0553352-1000-1-15")>]
```

- 이렇게 창이 뜨죠(이 창은 작업을 종료할 때 까지 닫으면 안됩니다)

```
In [3]: driver.get("http://www.opinet.co.kr/searRgSelect.do")
```

- 주유 가격 정보 사이트인 Opinet은 지도 정보를 가지고 있어서 충분한 시간적 여유를 주어야 한다.
- 그런데 간혹 페이지가 전환되지 않는 경우가 있습니다
- 이 주소를 한 번 더 실행합니다

한 번 접근해 보기

크롬 드라이버

Opinet 쌈 주유소찾기 국내유가통계 유가관련정보 불법행위공표 이용안내

지역별 경로별 도로별 면세유

주유소 총전소

지역 서울 성동구 읍면동

검색방법 지역 지도 상호 반경

형태 일반 셀프 불법 인증 전산

상표 전체 SK GS GS 편스 편스 편스 편스

부가정보 세차장 경정비 편의점 24시간

조회

고급취약유 보통취약유 경유 실내등유

검색결과 (총 17개) 불법 셀프 인증 전산

주유소명	휘발유	경유
평균가격	1761	1555
(S) (주)서울에너지 ...	1639	1434
PB 경호주유소	1648	1438
KLP제2주유소	1667	1447
현대오일뱅크(주)...	1668	1468
아이콘주유소	1698	1498
SK 에너지(주)...	1699	1499
현대오일뱅크(주) ...	1727	1527
SK 삼우주유소	1728	1518
SK 경진주유소	1728	1518
지에스칼텍스(주) ...	1734	1524

엑셀저장

* 본 가격정보는 특정시점에 수집된 가격이며 실제 판매가격과 다를 수 있습니다.

로컬 PC

jupyter 04. Oil Price Analysis and Green construction - ... (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
driver.execute_script(input_js) time.sleep(3) # 자동화팀지를 우회하기 위한 delay
```

```
driver.find_element_by_id("log.login").click() time.sleep(3)
```

```
driver.close()
```

주유 가격을 확인해보자

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

```
driver = webdriver.Chrome("../driver/chromedriver")
```

```
driver.get("http://www.opinet.co.kr")
```

```
In [4]: driver.get("http://www.opinet.co.kr/searRgSelect.do")
```

- 주유 가격 정보 사이트인 Opinet은 지도 정보를 가지고 있어서 충분한 시간적 여유를 주어야 한다.

```
In [4]: sido_list_raw = driver.find_element_by_id("SIDO_NM0")
```

```
sido_list_raw
```

```
Out[4]: <selenium.webdriver.remote.webelement.WebElement (session="95838124bfd14 0c5b8db52c1988acf1", element="ad6e44d4-2659-4b2f-926d-dc381211c040")>
```

```
In [5]: sido_list = sido_list_raw.find_elements_by_tag_name("option")
```

```
sido_list
```

```
Out[5]: [<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd14 30c5b8db52c1988acf1", element="52711562-c76c-4846-ad6f-7149da9ffee2")>, <selenium.webdriver.remote.webelement.WebElement (session="95838124bfd14 30c5b8db52c1988acf1", element="5e094668-d82d-4117-b379-6d3f2961c5e5")>, <selenium.webdriver.remote.webelement.WebElement (session="95838124bfd14 30c5b8db52c1988acf1", element="a937324b-88cf-4d6b-8d52-f793c84edf70")>, <selenium.webdriver.remote.webelement.WebElement (session="95838124bfd14 30c5b8db52c1988acf1", element="56adaaf2-2466-4388-a068-67134f26c86b")>, <selenium.webdriver.remote.webelement.WebElement (session="95838124bfd14 30c5b8db52c1988acf1", element="89481cf9-5ee7-4d4c-950d-bc80074cf90")>, <selenium.webdriver.remote.webelement.WebElement (session="95838124bfd14
```

- 크롬드라이버로 왼쪽 페이지까지 접근하면 일단 성공

지역별 경로별 도로별 면세유

주유소 충전소

• 지역 서울 ▼ 성동구 ▼ 읍/면/동 ▼

• 검색방법 지역 지도 상호 반경

• 형태 일반 셀프 불법 인증 전산

• 상표 전체 SK 에너지 GS 카드 Hyundai Oilbank 
 S-OIL 알뜰 PB

• 부가정보 세차장 경정비 편의점 24시간

조회

• 이 부분을 바꿔줘야 합니다

한 번 접근해 보기

The screenshot shows a user interface for a search function. On the left, a sidebar displays various filters and a table of results. On the right, the browser's developer tools are open, showing the DOM structure. A red box highlights the '서울' option in a dropdown menu, and a red arrow points from this box to the corresponding HTML code in the Elements tab, specifically the line `<select style="width:62px;" id="SIDO_NM0" name="SIDO_NM0">...</select>`.

```

<input type="hidden" id="MAP_FIRST_X" name="MAP_FIRST_X" value>
<input type="hidden" id="MAP_FIRST_Y" name="MAP_FIRST_Y" value>
<input type="hidden" id="LPG_YN" name="LPG_YN" value>
<input type="hidden" id="SESSION_USER_ID" name="SESSION_USER_ID" value>
▼<div class="search_gis">
  ▼<div class="inner">
    ▶<div class="tab_type1">...</div>
    ▼<div class="search_form">
      ▼<dl>
        ▶<dt class="fir">...</dt>
        ▼<dd class="fir">
          ▶<span class="hidden">...</span>
          ▶<select style="width:62px;" id="SIDO_NM0" name="SIDO_NM0">...</select> == $0
          ▶<span class="hidden">...</span>
          ▶<select style="width:102px;" id="SIGUNGU_NM0" name="SIGUNGU_NM0">...</select>
          ▶<span class="hidden">...</span>
          ▶<select style="width:70px;" id="DONG_NM" name="DONG_NM">...</select>
          <input id="GIS_X_COOR" name="GIS_X_COOR" type="hidden" value>
...
  ...

```

- 먼저 광역시도를 선택하는 곳부터 도전...

```
► <div class="tab_type1">...</div>
▼ <div class="search_form">
  ▼ <dl>
    ► <dt class="fir">...</dt>
    ▼ <dd class="fir">
      ► <span class="hidden">...</span>
      ► <select style="width:62px;" id="SIDO_NM0" name="SIDO_NM0">...</select> == $0
      ► <span class="hidden">...</span>
      ► <select style="width:102px;" id="SIGUNGU_NM0" name="SIGUNGU_NM0">...</select>
      ► <span class="hidden">...</span>
      ► <select style="width:70px;" id="DONG_NM" name="DONG_NM">...</select>
        <input id="GIS_X_COOR" name="GIS_X_COOR" type="hidden" value>
        <input id="GIS_Y_COOR" name="GIS_Y_COOR" type="hidden" value>
        <input id="GIS_X_COOR_S" name="GIS_X_COOR_S" type="hidden" value>
        <input id="GIS_X_COOR_E" name="GIS_X_COOR_E" type="hidden" value>
        <input id="GIS_Y_COOR_S" name="GIS_Y_COOR_S" type="hidden" value>
        <input id="GIS_Y_COOR_E" name="GIS_Y_COOR_E" type="hidden" value>
    </dd>
```

- id가 있다면 아주 고마운 일이다
- id="SIDO_NM0"

```
In [4]: sido_list_raw = driver.find_element_by_id("SIDO_NM0")
sido_list_raw
```

```
Out[4]: <selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988acf", element="ad6e44d4-2659-4b2f-926d-dc381211c040")>
```

- 그래서 `find_element_by_id("SIDO_NM0")` 로 찾았는데…
- 기대한 결과가 아니다

```
▼<select style="width:62px;" id="SIDO_NM0" name="SIDO_NM0"> == $0
  <option value>시/도</option>
  <option value="서울특별시" selected="selected">서울</option>
  <option value="부산광역시">부산</option>
  <option value="대구광역시">대구</option>
  <option value="인천광역시">인천</option>
  <option value="광주광역시">광주</option>
  <option value="대전광역시">대전</option>
  <option value="울산광역시">울산</option>
  <option value="세종특별자치시">세종</option>
  <option value="경기도">경기</option>
  <option value="강원도">강원</option>
  <option value="충청북도">충북</option>
  <option value="충청남도">충남</option>
  <option value="전라북도">전북</option>
  <option value="전라남도">전남</option>
  <option value="경상북도">경북</option>
  <option value="경상남도">경남</option>
  <option value="제주특별자치도">제주</option>
</select>
```

- 그래서 id가 있던 태그를 좀더 살펴봤더니…
- option이라는 태그 안에 광역시도의 값을 받아와야 할 것 같다
- 그건 sido_list_raw 안에 있다

```
In [5]: sido_list = sido_list_raw.find_elements_by_tag_name("option")
sido_list

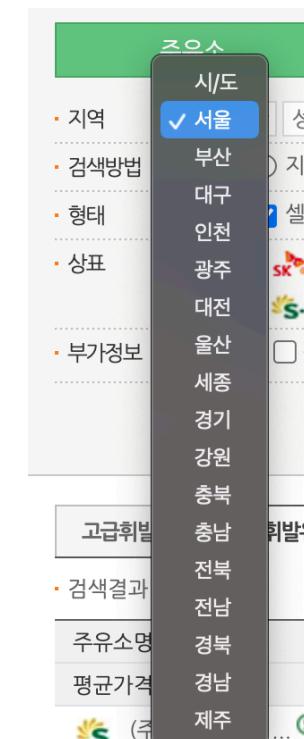
Out[5]: [<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="52711562-c76c-4846-ad6f-7149dda9ffe2")>,
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="5e094668-d82d-4117-b379-6d3f2961c5e5")>,
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="a937324b-88cf-4d6b-8d52-f793c84edf70")>,
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="56adaaf2-2466-4388-a068-67134f26c86b")>,
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="89481cf9-5ee7-4d4c-950d-bc80074c0f90")>,
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="0d2ac949-4fa2-45c6-8147-5b201668f5be")>,
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="e9dcfd8a-80a0-4dfa-b0b0-790affe19c95")>,
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="b7facc07-b9dd-41b7-af60-c5ee4e733597")>,
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="1efddc50-27fb-47d5-bc19-04f6533a7152")>,
<selenium.webdriver.remote.webelement.WebElement (session="95838124bfd1430c5b8db52c1988ac1f", element="9c46a4eb-9aab-469f-99f6-86eaa3ac1e5c")>,
```

- 뭔가 있긴 한데..

```
In [6]: sido_names = [option.get_attribute("value") for option in sido_list]
sido_names.remove("")
sido_names
```

```
Out[6]: ['서울특별시',
'부산광역시',
'대구광역시',
'인천광역시',
'광주광역시',
'대전광역시',
'울산광역시',
'세종특별자치시',
'경기도',
'강원도',
'충청북도',
'충청남도',
'전라북도',
'전라남도',
'경상북도',
'경상남도',
'제주특별자치도']
```

- 지역 리스트를 가져올 수 있다



```
<select style="width:62px;" id="SIDO_NM0">
<option value>시/도</option>
<option value="서울특별시" selected="selected">서울</option>
<option value="부산광역시">부산</option>
<option value="대구광역시">대구</option>
<option value="인천광역시">인천</option>
<option value="광주광역시">광주</option>
<option value="대전광역시">대전</option>
<option value="울산광역시">울산</option>
<option value="세종특별자치시">세종</option>
<option value="경기도">경기</option>
<option value="강원도">강원</option>
<option value="충청북도">충북</option>
<option value="충청남도">충남</option>
<option value="전라북도">전북</option>
<option value="전라남도">전남</option>
<option value="경상북도">경북</option>
<option value="경상남도">경남</option>
<option value="제주특별자치도">제주</option>
</select>
```

한 번 접근해 보기

zero-base /

지역별	경로별	도로별	면세유
주유소	충전소		
지역 <input type="button" value="서울"/> 성동구 <input type="button" value="읍/면/동"/>			
검색방법 <input checked="" type="radio"/> 지역 <input type="radio"/> 지도 <input type="radio"/> 상호 <input type="radio"/> 반경			
형태 <input checked="" type="checkbox"/> 일반 <input checked="" type="checkbox"/> 셀프 <input checked="" type="checkbox"/> 불법 <input checked="" type="checkbox"/> 인증 <input checked="" type="checkbox"/> 전산			
상표 <input type="button" value="전체"/> <input checked="" type="checkbox"/> SK 에너지 <input checked="" type="checkbox"/> GS 카드스 <input checked="" type="checkbox"/> Hyundai Gilbank <input checked="" type="checkbox"/> S-OIL <input checked="" type="checkbox"/> 양풀 <input checked="" type="checkbox"/> PB			
부가정보 <input type="checkbox"/> 세차장 <input type="checkbox"/> 경정비 <input type="checkbox"/> 편의점 <input type="checkbox"/> 24시간			
<input type="button" value="조회"/>			



- 값은 얻어 왔는데, 정작 저 박스의 내용은 어떻게 바꾸지?

한 번 접근해 보기

The screenshot shows a search interface for oil stations in Seoul, specifically in Seongdong-gu. The search results list 17 stations, each with a logo, name, address, and coordinates. A red arrow points from the highlighted 'SIDO_NM0' field in the code to the '성동구청' station entry in the list.

주유소명	휘발유	경유
평균가격	1761	1555
(주)서울에너지 ...	1639	1434
PB 정호주유소	1648	1438
KLP제2주유소	1667	1447
현대오일뱅크(주)...	1668	1468
아이콘주유소	1698	1498
SK에너지(주) ...	1699	1499
최고급 이비코/모지어	1777	1527

```

<!-- //header -->
<!-- //header -->
<div id="container">
  <div class="content" id="contents" style="height: 832px;">
    <div class="p_title">...</div>
    <!-- fly tab -->
    <div class="fly_box" style="height: 832px;">
      <div class="fly_inner" style="height: 797px;">
        <div class="header_path_gsi">...</div>
        <form id="searrgVO" name="form1" action="/searRgSelect.do" method="post">
          <input id="BTN_DIV" name="BTN_DIV" type="hidden" value="os_btn">
          <input id="BTN_DIV_STR" name="BTN_DIV_STR" type="hidden" value="..."/>
          <input id="POLL_ALL" name="POLL_ALL" type="hidden" value="all">
          <input type="hidden" id="SIDO_NM" name="SIDO_NM" value="서울특별시">
          <input type="hidden" id="SIGUNGU_NM" name="SIGUNGU_NM" value="성동구">
          <input type="hidden" id="SIDO_CD" name="SIDO_CD" value="01">
          <input type="hidden" id="SIGUN_CD" name="SIGUN_CD" value="0104">
          <input type="hidden" id="MAP_CENTER_X" name="MAP_CENTER_X" value="..."/>
          <input type="hidden" id="MAP_CENTER_Y" name="MAP_CENTER_Y" value="..."/>
          <input type="hidden" id="MAP_ZOOM" name="MAP_ZOOM" value="..."/>
          <input type="hidden" id="MAP_FIRST_X" name="MAP_FIRST_X" value="..."/>
          <input type="hidden" id="MAP_FIRST_Y" name="MAP_FIRST_Y" value="..."/>
          <input type="hidden" id="LPG_YN" name="LPG_YN" value="..."/>
          <input type="hidden" id="SESSION_USER_ID" name="SESSION_USER_ID" value="..."/>
        </form>
        <div class="search_gis">
          <div class="inner">
            <div class="tab_type1">...</div>
            <div class="search_form">
              <dl>
                <dt class="fir">...</dt>
                <dd class="fir">
                  <span class="hidden">...</span>
                  <select style="width:62px;" id="SIDO_NM0" name="SIDO_NM0">...</select> == $0
                  <span class="hidden">...</span>
                  <select style="width:102px;" id="SIGUNGU_NM0" name="SIGUNGU_NM0">...</select>
                  ...
                </dd>
              </dl>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

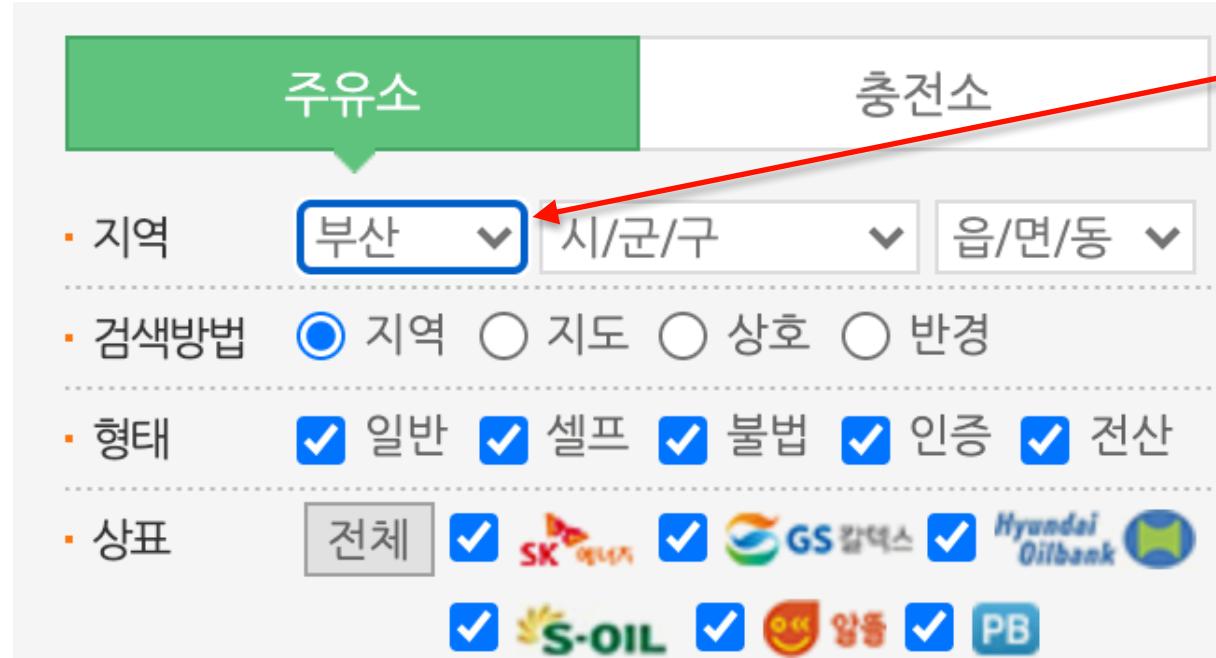
```

- 애초 저 박스의 id는 SIDO_NM0 였고,
- driver에서 해당 id를 sido_list_raw 변수로 저장했었다

```
In [10]: sido_list_raw.send_keys(sido_names[1])
```

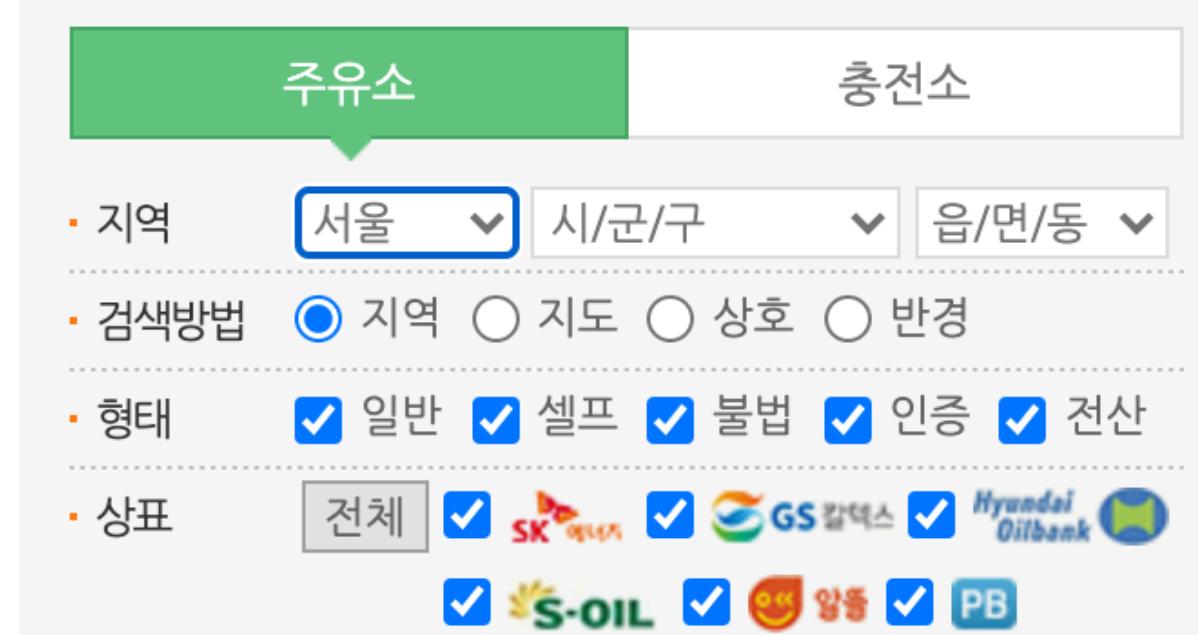
out[8]:

```
sido_names
['서울특별시',
'부산광역시',
'대구광역시',
'인천광역시',
'광주광역시',
'대전광역시',
'울산광역시',
'세종특별자치시',
'경기도',
'강원도',
'충청북도',
'충청남도',
'전라북도',
'전라남도',
'경상북도',
'경상남도',
'제주특별자치도']
```

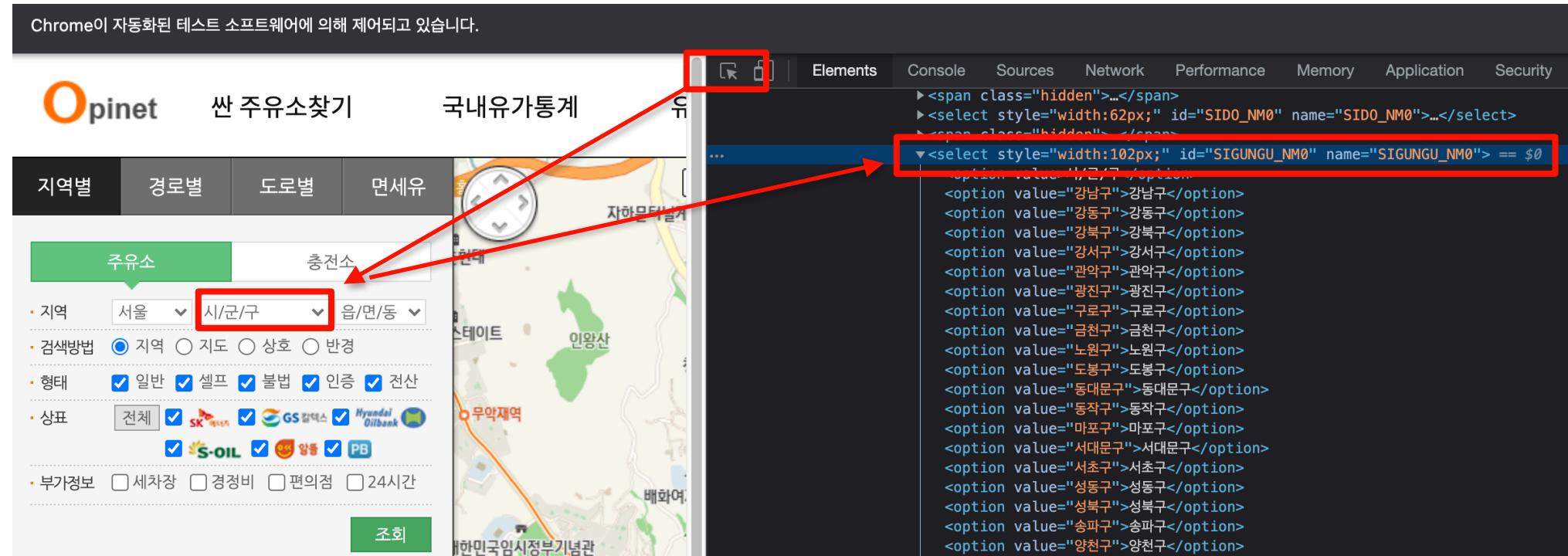


- `sido_list_raw` 변수에 리스트 결과값을 저장한 `sido_names`
- `sido_names`의 값을 `send_keys`를 이용해서 보내면 된다
- 0은 서울, 1은 부산, 2는 대구 … 16은 제주

```
In [11]: sido_list_raw.send_keys(sido_names[0])
```



- 우리는 서울이 관심사이니, 서울로 고정하자



- 두 번째 리스트인 시/군/구 항목도 동일하게, id와 option을 확인하고…
- **id="SIGUNGU_NM"**

```
In [8]: gu_list_raw = driver.find_element_by_id("SIGUNGU_NM0")
gu_list = gu_list_raw.find_elements_by_tag_name("option")

gu_names = [option.get_attribute("value") for option in gu_list]
gu_names.remove("")
gu_names
```

```
Out[8]: ['강남구',
'강동구',
'강북구',
'강서구',
'관악구',
'광진구',
'구로구',
'금천구',
```

- option에서 리스트안의 변수값들을 저장하자

한 번 접근해 보기

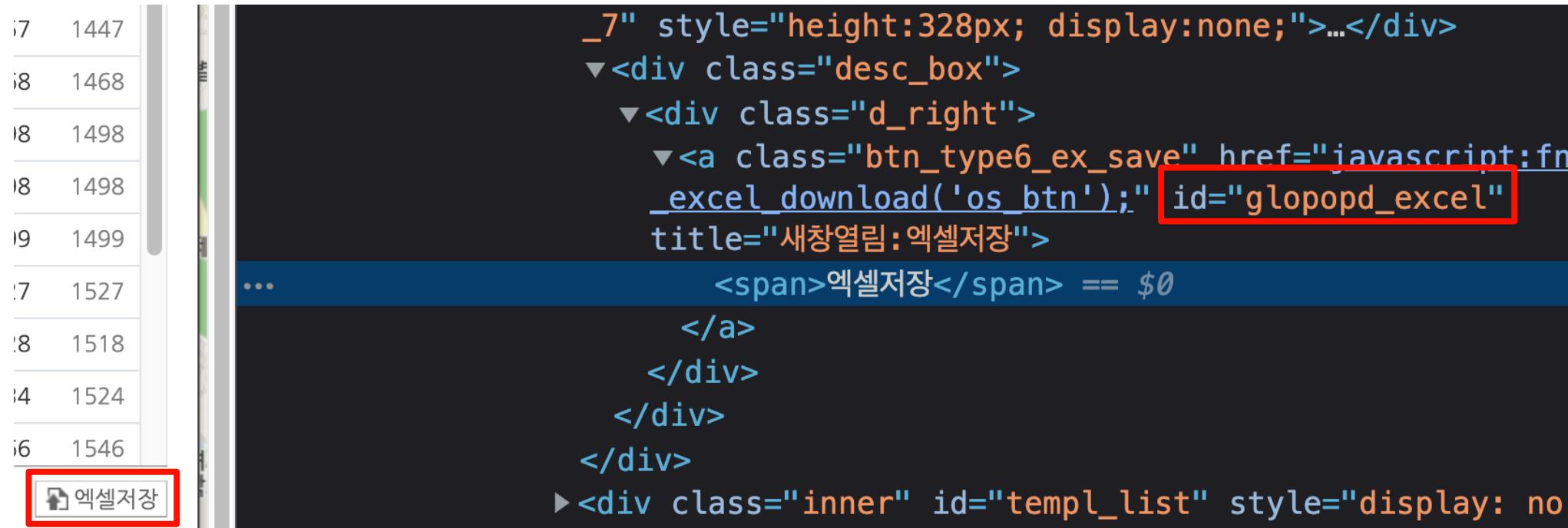
In [9]: `gu_list_raw.send_keys(gu_names[0])`

The screenshot shows a search interface for gas stations. At the top, there are two tabs: '주유소' (Gas Station) and '충전소' (Charging Station). The '주유소' tab is active. Below the tabs are dropdown menus for '지역' (Region) set to '서울' (Seoul), '강남구' (Gangnam-gu), and '읍/면/동' (Eul/Myeon/Dong). A section for '검색방법' (Search Method) has a radio button for '지역' (Region) selected. Under '형태' (Type), all checkboxes for '일반' (General), '셀프' (Self-service), '불법' (Illegal), '인증' (Certified), and '전산' (Computerized) are checked. In the '상표' (Trademark) section, '전체' (All) is selected, and logos for SK 에너지, GS 카드, Hyundai Oilbank, S-OIL, 알뜰, and PB are shown. The '부가정보' (Additional Information) section contains checkboxes for '세차장' (Car Wash), '경정비' (Toll), '편의점' (Convenience Store), and '24시간' (24 hours), none of which are checked. At the bottom is a green '조회' (Search) button.

In [56]: `gu_list_raw.send_keys(gu_names[1])`

This screenshot is identical to the one above, except for the '강남구' (Gangnam-gu) entry in the '지역' (Region) dropdown is now replaced by '강동구' (Gangdong-gu).

- `send_keys(gu_names[num])`로 해당 값을 변경해보자
- 0: 강남구, 1: 강동구 ...

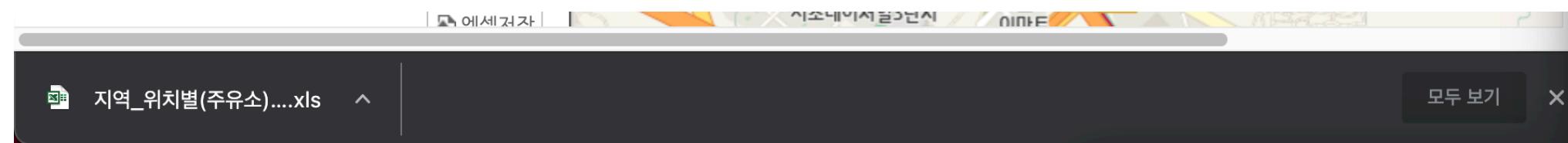


The screenshot shows a list of numbers on the left and a detailed view of an 'Excel 저장' button on the right. The button's ID is highlighted with a red box.

```
_7" style="height:328px; display:none;">...</div>
▼<div class="desc_box">
  ▼<div class="d_right">
    ▼<a class="btn_type6_ex_save" href="javascript:fn_excel_download('os_btn');" id="glopopd_excel" title="새창열림: 엑셀저장">
      <span>엑셀저장</span> == $0
    </a>
  </div>
</div>
</div>
▶<div class="inner" id="templ_list" style="display: none;">
```

- 엑셀저장 태그 id 값 확인
- id="glopopd_excel"

```
In [24]: element_get_excel = driver.find_element_by_id("glopod_excel").click()
```



- 하나의 시와 구를 선택해서 엑셀 파일 저장까지 되는지 테스트 했다
- 테스트용으로 받은 엑셀 파일은 삭제 ^^

In [26]:

```
import time
from tqdm import tqdm_notebook

for gu in tqdm_notebook(gu_names):
    element = driver.find_element_by_id("SIGUNGU_NM0")
    element.send_keys(gu)
    time.sleep(2)

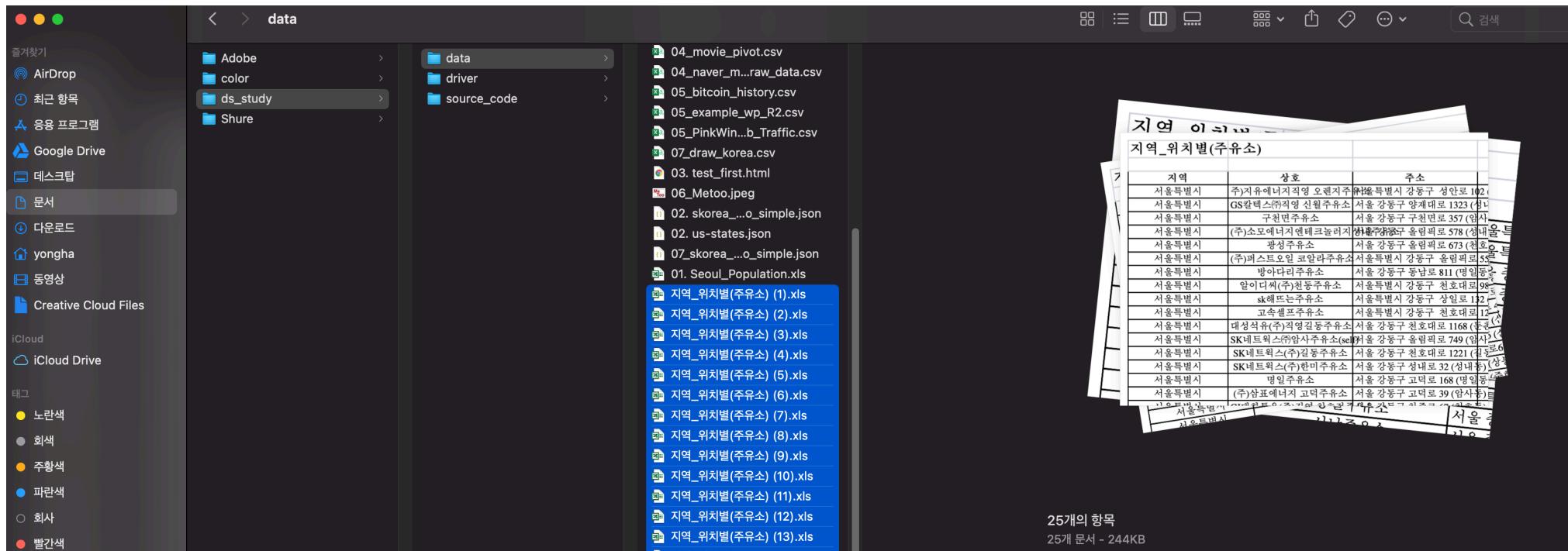
    element_get_excel = driver.find_element_by_id("glopopd_excel").click()
    time.sleep(1)
```

100%

25/25 [01:50<00:00, 4.28s/it]

- 반복문으로 모든 구의 데이터를 엑셀 파일로 저장

데이터 모두 수집



- Download 폴더에 25개의 엑셀 파일이 있어야 한다
- 25개가 정상적으로 있다면 ds_study - data 폴더에 넣어주자

	A	B	C	D	E	F	G	H	I	J
1	지역_위치별(주유소)									
2	(단위 : 원/리터)									
3	지역	상호	주소	상표	전화번호	셀프여부	고급휘발유	휘발유	경유	실내등유
4	서울특별시	수유동주유소	울 강북구 도봉로 395 (수유동)	GS칼텍스	02-902-9470	Y	-	1618	1418	-
5	서울특별시	(주)에스이엔알 미아주유소	울 강북구 도봉로 122 (미아동)	GS칼텍스	02-987-5152	Y	1863	1618	1418	-
6	서울특별시	북서울고속주유소	울 강북구 삼양로 410 (수유동)	S-OIL	02-907-5182	Y	-	1622	1422	-
7	서울특별시	(주)석산에너지	울 강북구 삼양로 316 (수유동)	현대오일뱅크	02-980-1448	Y	-	1622	1422	900
8	서울특별시	KH에너지(주)직영 송정주유소	울 강북구 인수봉로 185 (수유동)	SK에너지	02-993-5189	N	-	1635	1429	-
9	서울특별시	SK에너지(주) 매일주유소	울 강북구 삼양로 217 (미아동)	SK에너지	02-982-8855	Y	-	1644	1444	-
10	서울특별시	덕릉로주유소	서울 강북구 덕릉로 158 (번동)	S-OIL	02-989-9806	N	-	1644	1435	-
11	서울특별시	오일뱅크(주)직영 새한셀프주	서울 강북구 도봉로 75	현대오일뱅크	02-985-4382	Y	1889	1648	1448	-
12	서울특별시	세월주유소	서울 강북구 한천로 918 (번동)	SK에너지	02-907-6890	Y	-	1648	1448	1100
13	서울특별시	오일뱅크(주)직영 번동셀프주	서울 강북구 한천로 963 (번동)	현대오일뱅크	02-992-3331	Y	1889	1648	1448	-
14	서울특별시	오일뱅크(주)직영 미아셀프주	울 강북구 도봉로 200 (미아동)	현대오일뱅크	02-945-7999	Y	1889	1648	1448	-
15	서울특별시	에덴주유소	서울 강북구 월계로 195 (번동)	SK에너지	02-987-8182	Y	-	1649	1437	-
16	서울특별시	대성석유(주)신광주유소	서울 강북구 삼양로 484	GS칼텍스	02-908-3320	N	-	1665	1455	1150
17										

- 엑셀 파일을 열어보면 이렇게 정리되어 있다
- 지역, 상호, 주소, 상표, 전화번호, 셀프여부, 종류

```
In [12]: driver.close()
```

- 모든 작업이 끝났다면, 반드시 **driver.close()** 를 해주자
- 안그럼.. 크롬이 여러 개 켜져 있을 수 있다 😱

데이터 정리하기



```
In [13]: import pandas as pd  
from glob import glob
```

```
In [14]: glob("../data/지역_*.xls")
```

```
Out[14]: ['../data/지역_위치별(주유소) (9).xls',  
          '../data/지역_위치별(주유소) (12).xls',  
          '../data/지역_위치별(주유소) (5).xls',  
          '../data/지역_위치별(주유소) (24).xls',  
          '../data/지역_위치별(주유소) (4).xls',  
          '../data/지역_위치별(주유소) (13).xls',  
          '../data/지역_위치별(주유소) (8).xls',  
          '../data/지역_위치별(주유소) (3).xls',  
          '../data/지역_위치별(주유소) (18).xls',  
          '../data/지역_위치별(주유소) (22).xls',  
          '../data/지역_위치별(주유소) (14).xls',  
          '../data/지역_위치별(주유소) (15).xls',
```

- 파일 목록을 한 번에 가져올 수 있다

```
In [15]: stations_files = glob("../data/지역_*.xls")
stations_files
```

```
Out[15]: ['../data/지역_위치별(주유소) (9).xls',
          '../data/지역_위치별(주유소) (12).xls',
          '../data/지역_위치별(주유소) (5).xls',
          '../data/지역_위치별(주유소) (24).xls',
          '../data/지역_위치별(주유소) (4).xls',
          '../data/지역_위치별(주유소) (13).xls',
          '../data/지역_위치별(주유소) (8).xls',
          '../data/지역_위치별(주유소) (3).xls',
          '../data/지역_위치별(주유소) (18).xls',
          '../data/지역_위치별(주유소) (22).xls',
          '../data/지역_위치별(주유소) (14).xls',
          '../data/지역_위치별(주유소) (15).xls',
          '../data/지역_위치별(주유소) (23).xls',
          '../data/지역_위치별(주유소) (19).xls',
          '../data/지역_위치별(주유소) (2).xls']
```

- 파일명을 저장하고 ~

```
In [16]: tmp = pd.read_excel(stations_files[0], header=2)
tmp.head()
```

Out[16]:

	지역	상호	주소	상표	전화번호	셀프여부	고급휘발유	휘발유	경유	실내등유
0	서울특별시	한이에너지(주)KLP제1주유소	서울특별시 도봉구 도봉로 596 (창동)	현대오일뱅크	02-902-5189	N	-	1508	1298	-
1	서울특별시	도봉제일주유소(주)송만에너지	서울특별시 도봉구 도봉로 783 (도봉동)	현대오일뱅크	02-955-5185	N	-	1508	1298	940
2	서울특별시	쌍문주유소	서울특별시 도봉구 도봉로 547 (쌍문동)	S-OIL	02-992-0220	Y	-	1509	1299	1000
3	서울특별시	삼미상사(주)북부주유소	서울 도봉구 도봉로 437 (쌍문동)	SK에너지	02-991-5561	Y	-	1509	1299	1200
4	서울특별시	정다운셀프주유소	서울특별시 도봉구 도봉로 635 (쌍문동)	SK에너지	02-996-5151	Y	-	1511	1301	-

- 먼저 하나만 읽어보자 ~~

```
In [17]: tmp_raw = []

for file_name in stations_files:
    tmp = pd.read_excel(file_name, header=2)
    tmp_raw.append(tmp)

station_raw = pd.concat(tmp_raw)
```

- 형식이 동일하고 연달아 붙이기만 하면 될 때는 concat 명령을 사용한다

```
In [18]: station_raw.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 528 entries, 0 to 11
Data columns (total 10 columns):
 #   Column   Non-Null Count   Dtype  
--- 
 0   지역      528 non-null    object 
 1   상호      528 non-null    object 
 2   주소      528 non-null    object 
 3   상표      528 non-null    object 
 4   전화번호   528 non-null    object 
 5   셀프여부   528 non-null    object 
 6   고급휘발유 528 non-null    object 
 7   휘발유    528 non-null    object 
 8   경유      528 non-null    object 
 9   실내등유   528 non-null    object 
dtypes: object(10)
memory usage: 45.4+ KB
```

데이터 읽어 오기

```
In [19]: station_raw.head()
```

```
Out[19]:
```

	지역	상호	주소	상표	전화번호	셀프여부	고급휘발유	휘발유	경유	실내등유
0	서울특별시	한이에너지(주)KLP제1주유소	서울특별시 도봉구 도봉로 596 (창동)	현대오일뱅크	02-902-5189	N	-	1508	1298	-
1	서울특별시	도봉제일주유소(주)송만에너지	서울특별시 도봉구 도봉로 783 (도봉동)	현대오일뱅크	02-955-5185	N	-	1508	1298	940
2	서울특별시	쌍문주유소	서울특별시 도봉구 도봉로 547 (쌍문동)	S-OIL	02-992-0220	Y	-	1509	1299	1000
3	서울특별시	삼미상사(주)북부주유소	서울 도봉구 도봉로 437 (쌍문동)	SK에너지	02-991-5561	Y	-	1509	1299	1200
4	서울특별시	정다운셀프주유소	서울특별시 도봉구 도봉로 635 (쌍문동)	SK에너지	02-996-5151	Y	-	1511	1301	-

```
In [20]: stations = pd.DataFrame(
    {
        "상호": station_raw[ "상호" ],
        "주소": station_raw[ "주소" ],
        "가격": station_raw[ "휘발유" ],
        "셀프": station_raw[ "셀프여부" ],
        "상표": station_raw[ "상표" ],
    }
)
stations.head()
```

Out [20]:

	상호	주소	가격	셀프	상표
0	한이에너지(주)KLP제1주유소	서울특별시 도봉구 도봉로 596 (창동)	1508	N	현대오일뱅크
1	도봉제일주유소(주)송만에너지	서울특별시 도봉구 도봉로 783 (도봉동)	1508	N	현대오일뱅크
2	쌍문주유소	서울특별시 도봉구 도봉로 547 (쌍문동)	1509	Y	S-OIL
3	삼미상사(주)북부주유소	서울 도봉구 도봉로 437 (쌍문동)	1509	Y	SK에너지
4	정다운셀프주유소	서울특별시 도봉구 도봉로 635 (쌍문동)	1511	Y	SK에너지

- 이쁘게 원하는 정보만 다듬었다 ~

```
In [21]: stations["구"] = [eachAddress.split()[1] for eachAddress in stations["주소"]]
stations.head()
```

Out[21]:

	상호	주소	가격	셀프	상표	구
0	한이에너지(주)KLP제1주유소	서울특별시 도봉구 도봉로 596 (창동)	1508	N	현대오일뱅크	도봉구
1	도봉제일주유소(주)송만에너지	서울특별시 도봉구 도봉로 783 (도봉동)	1508	N	현대오일뱅크	도봉구
2	쌍문주유소	서울특별시 도봉구 도봉로 547 (쌍문동)	1509	Y	S-OIL	도봉구
3	삼미상사(주)북부주유소	서울 도봉구 도봉로 437 (쌍문동)	1509	Y	SK에너지	도봉구
4	정다운셀프주유소	서울특별시 도봉구 도봉로 635 (쌍문동)	1511	Y	SK에너지	도봉구

- 평균적으로 주유가격이 비싼 구를 확인하기 위해 주소에서 구 정보를 가져오자

```
In [22]: stations["구"].unique()
```

```
Out[22]: array(['도봉구', '특별시', '마포구', '광진구', '중랑구', '관악구', '서대문구', '노원구', '강남구',
       '양천구', '종로구', '서초구', '성동구', '서울특별시', '중구', '영등포구', '강서구', '용산구',
       '강북구', '성북구', '송파구', '은평구', '동대문구', '강동구', '금천구', '구로구', '동작구'],
      dtype=object)
```

```
In [23]: len(stations["구"].unique())
```

```
Out[23]: 27
```

- 뭔가 잘못 되었다

```
In [24]: stations[stations["구"] == "서울특별시"]
```

Out[24]:

	상호	주소	가격	셀프	상표	구
11	SK네트웍스(주)효진주유소	1 서울특별시 성동구 동일로 129 (성수동2가)	1669	N	SK에너지	서울특별시

```
In [25]: stations.loc[stations["구"] == "서울특별시", "구"] = "성동구"
stations["구"].unique()
```

Out[25]: array(['도봉구', '특별시', '마포구', '광진구', '중랑구', '관악구', '서대문구', '노원구', '강남구',
'양천구', '종로구', '서초구', '성동구', '중구', '영등포구', '강서구', '용산구', '강북구',
'성북구', '송파구', '은평구', '동대문구', '강동구', '금천구', '구로구', '동작구'],
dtype=object)

- 오류를 바로 잡고 ~

```
In [26]: stations[stations["구"] == "특별시"]
```

Out[26]:

	상호	주소	가격	셀프	상표	구
9	서현주유소	서울 특별시 도봉구 방학로 142 (방학동)	1514	Y	S-OIL	특별시

```
In [27]: stations.loc[stations["구"] == "특별시", "구"] = "도봉구"
stations["구"].unique()
```

```
Out[27]: array(['도봉구', '마포구', '광진구', '종랑구', '관악구', '서대문구', '노원구', '강남구', '양천구',
       '종로구', '서초구', '성동구', '중구', '영등포구', '강서구', '용산구', '강북구', '성북구',
       '송파구', '은평구', '동대문구', '강동구', '금천구', '구로구', '동작구'], dtype=object)
```

- 오류를 바로 잡고 ~

```
In [28]: stations["가격"] = stations["가격"].astype("float")
```

```
-----  
ValueError                                     Traceback (most recent call last)  
<ipython-input-28-8e8a5bea5f9e> in <module>  
----> 1 stations["가격"] = stations["가격"].astype("float")  
  
~/miniconda3/envs/ds_study/lib/python3.8/site-packages/pandas/core/generic.py in astype  
    5875         else:  
    5876             # else, only a single dtype is given  
-> 5877             new_data = self._mgr.astype(dtype=dtype, copy=copy, errors=errors)  
    5878             return self._constructor(new_data).__finalize__(self, method="as  
    5879  
      
```

- 가격에 대해 수치적 연산을 하기 위해 float 형으로 변경하려 했지만 에러가…😢

```
~/miniconda3/envs/ds_study/lib/python3.8/site-packages/pandas/core/dtypes/cast.py in
skipna)
1095     if copy or is_object_dtype(arr) or is_object_dtype(dtype):
1096         # Explicit copy, or required since NumPy can't view from / to object
-> 1097         return arr.astype(dtype, copy=True)
1098
1099     return arr.view(dtype)

ValueError: could not convert string to float: '-'
```

- 응??? ‘ - ’ 이 기호는 float으로 못 바꾸겠다는 당연한 소리를..ㅠㅠ.

```
In [29]: stations[stations["가격"] == "-"]
```

Out[29]:

	상호	주소	가격	셀프	상표	구
13	서강주유소	서울 마포구 독막로 134 (창전동)	-	N	SK에너지	마포구
25	목동주유소	서울 양천구 목동중앙로 125 (목동)	-	N	S-OIL	양천구
34	하나주유소	서울특별시 영등포구 도림로 236 (신길동)	-	N	S-OIL	영등포구
12	(주)에이앤이청담주유소	서울특별시 강북구 도봉로 155 (미아동)	-	Y	SK에너지	강북구
18	명진석유(주)동서울주유소	서울특별시 강동구 천호대로 1456 (상일동)	-	Y	GS칼텍스	강동구
11	(주)서우유통에너지	서울특별시 동작구 노량진로 78 (노량진동)	-	N	S-OIL	동작구

- 가격 정보 없는 주유소가 있다

```
In [30]: stations = stations[stations["가격"] != "-"]
stations.head()
```

Out [30]:

	상호	주소	가격	셀프	상표	구
0	한이에너지(주)KLP제1주유소	서울특별시 도봉구 도봉로 596 (창동)	1508	N	현대오일뱅크	도봉구
1	도봉제일주유소(주)송만에너지	서울특별시 도봉구 도봉로 783 (도봉동)	1508	N	현대오일뱅크	도봉구
2	쌍문주유소	서울특별시 도봉구 도봉로 547 (쌍문동)	1509	Y	S-OIL	도봉구
3	삼미상사(주)북부주유소	서울 도봉구 도봉로 437 (쌍문동)	1509	Y	SK에너지	도봉구
4	정다운셀프주유소	서울특별시 도봉구 도봉로 635 (쌍문동)	1511	Y	SK에너지	도봉구

- 그럼 뭐 가격 정보가 있는 아이들만 ~~

```
In [31]: stations["가격"] = stations["가격"].astype("float")
```

```
In [32]: stations.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 522 entries, 0 to 10
Data columns (total 6 columns):
 #   Column   Non-Null Count   Dtype  
--- 
 0   상호      522 non-null    object 
 1   주소      522 non-null    object 
 2   가격      522 non-null    float64
 3   셀프      522 non-null    object 
 4   상표      522 non-null    object 
 5   구        522 non-null    object 
dtypes: float64(1), object(5)
memory usage: 28.5+ KB
```

In [33]: stations

Out[33]:

		상호	주소	가격	셀프	상표	구
0	한이에너지(주)KLP제1주유소	서울특별시 도봉구 도봉로 596 (창동)	1508.0	N	현대오일뱅크	도봉구	
1	도봉제일주유소(주)송만에너지	서울특별시 도봉구 도봉로 783 (도봉동)	1508.0	N	현대오일뱅크	도봉구	
2	쌍문주유소	서울특별시 도봉구 도봉로 547 (쌍문동)	1509.0	Y	S-OIL	도봉구	
3	삼미상사(주)북부주유소	서울 도봉구 도봉로 437 (쌍문동)	1509.0	Y	SK에너지	도봉구	
4	정다운셀프주유소	서울특별시 도봉구 도봉로 635 (쌍문동)	1511.0	Y	SK에너지	도봉구	
...
6	현대오일뱅크 동작주유소	서울특별시 동작구 상도로 403 (상도동)	1587.0	N	현대오일뱅크	동작구	
7	대성산업(주) 노량진주유소	서울 동작구 노량진로 172 (노량진동)	1587.0	N	GS칼텍스	동작구	
8	21세기주유소	서울 동작구 시흥대로 616 (신대방동)	1598.0	Y	SK에너지	동작구	
9	흑석동	서울 동작구 현충로 101 (흑석동)	1627.0	Y	SK에너지	동작구	
10	살피재주유소	서울특별시 동작구 상도로 334 (상도동)	1635.0	N	SK에너지	동작구	

522 rows × 6 columns

In [34]:

```
stations.reset_index(inplace=True)
stations.head()
```

Out[34]:

	index	상호	주소	가격	셀프	상표	구
0	0	한이에너지(주)KLP제1주유소	서울특별시 도봉구 도봉로 596 (창동)	1508.0	N	현대오일뱅크	도봉구
1	1	도봉제일주유소(주)송만에너지	서울특별시 도봉구 도봉로 783 (도봉동)	1508.0	N	현대오일뱅크	도봉구
2	2	쌍문주유소	서울특별시 도봉구 도봉로 547 (쌍문동)	1509.0	Y	S-OIL	도봉구
3	3	삼미상사(주)북부주유소	서울 도봉구 도봉로 437 (쌍문동)	1509.0	Y	SK에너지	도봉구
4	4	정다운셀프주유소	서울특별시 도봉구 도봉로 635 (쌍문동)	1511.0	Y	SK에너지	도봉구

- index 번호를 다시 매기고 ~~

```
In [35]: del stations["index"]
stations.head()
```

Out[35]:

	상호	주소	가격	셀프	상표	구
0	한이에너지(주)KLP제1주유소	서울특별시 도봉구 도봉로 596 (창동)	1508.0	N	현대오일뱅크	도봉구
1	도봉제일주유소(주)송만에너지	서울특별시 도봉구 도봉로 783 (도봉동)	1508.0	N	현대오일뱅크	도봉구
2	쌍문주유소	서울특별시 도봉구 도봉로 547 (쌍문동)	1509.0	Y	S-OIL	도봉구
3	삼미상사(주)북부주유소	서울 도봉구 도봉로 437 (쌍문동)	1509.0	Y	SK에너지	도봉구
4	정다운셀프주유소	서울특별시 도봉구 도봉로 635 (쌍문동)	1511.0	Y	SK에너지	도봉구

- OK ~~~

주유 가격 정보 시각화



```
In [36]: import matplotlib.pyplot as plt
import seaborn as sns
import platform
from matplotlib import font_manager, rc

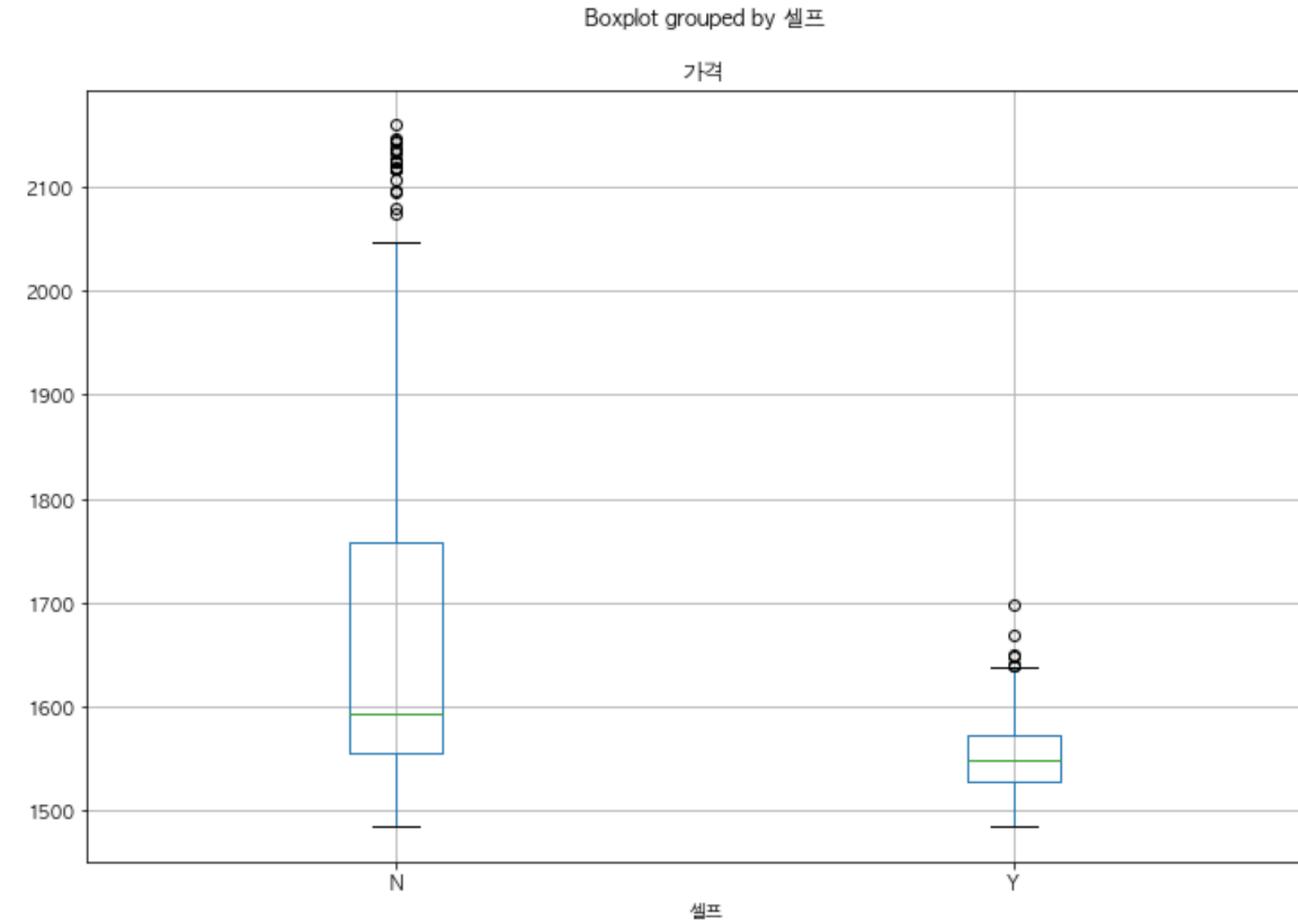
get_ipython().run_line_magic("matplotlib", "inline")
path = "c:/Windows/Fonts/malgun.ttf"

if platform.system() == "Darwin":
    rc("font", family="AppleGothic")
elif platform.system() == "Windows":
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc("font", family=font_name)
else:
    print("Unknown system... sorry~~~~")
```

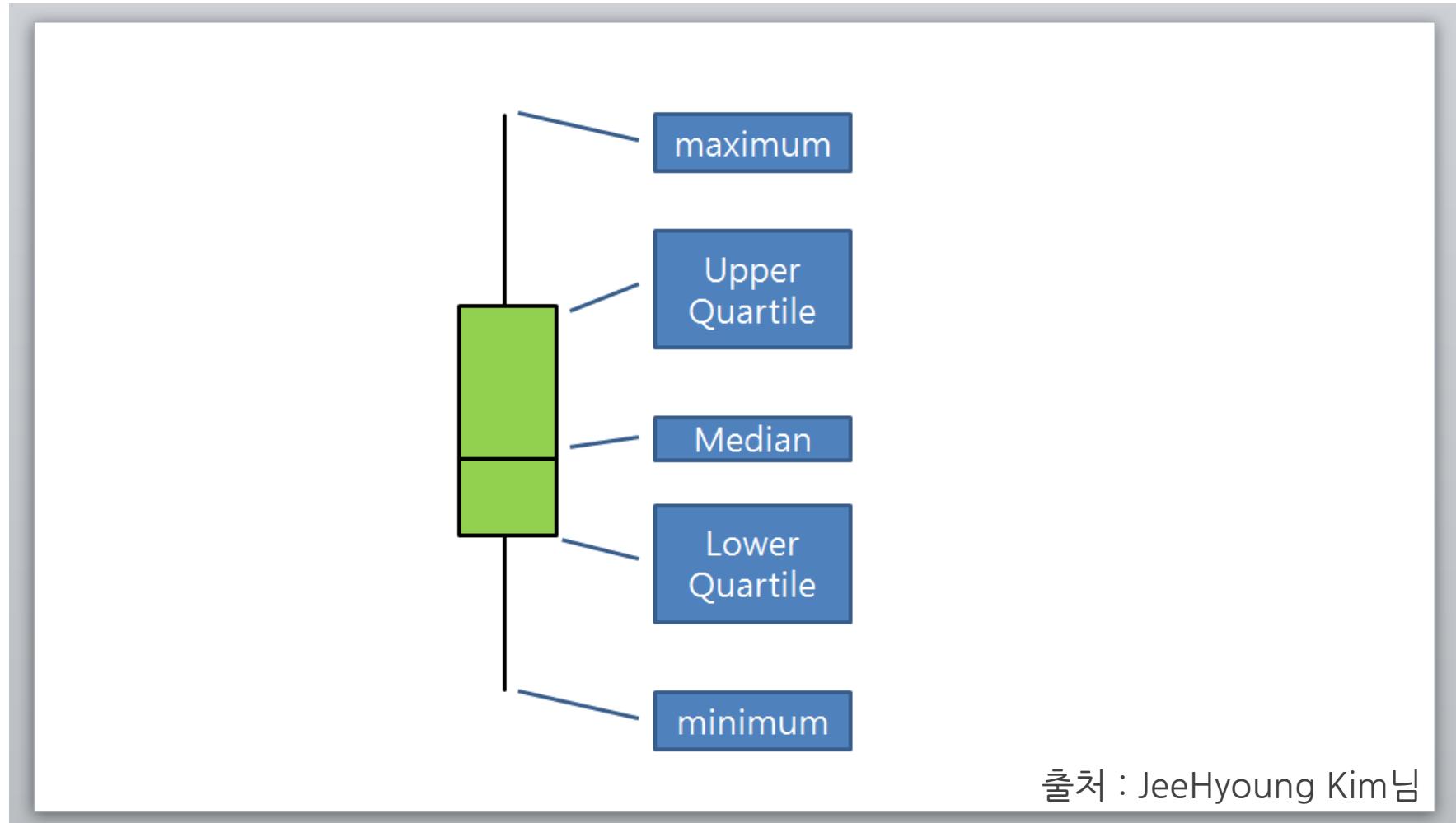
- matplotlib 한글 대응

데이터 시각화

```
In [37]: stations.boxplot(column="가격", by="셀프", figsize=(12, 8));
```

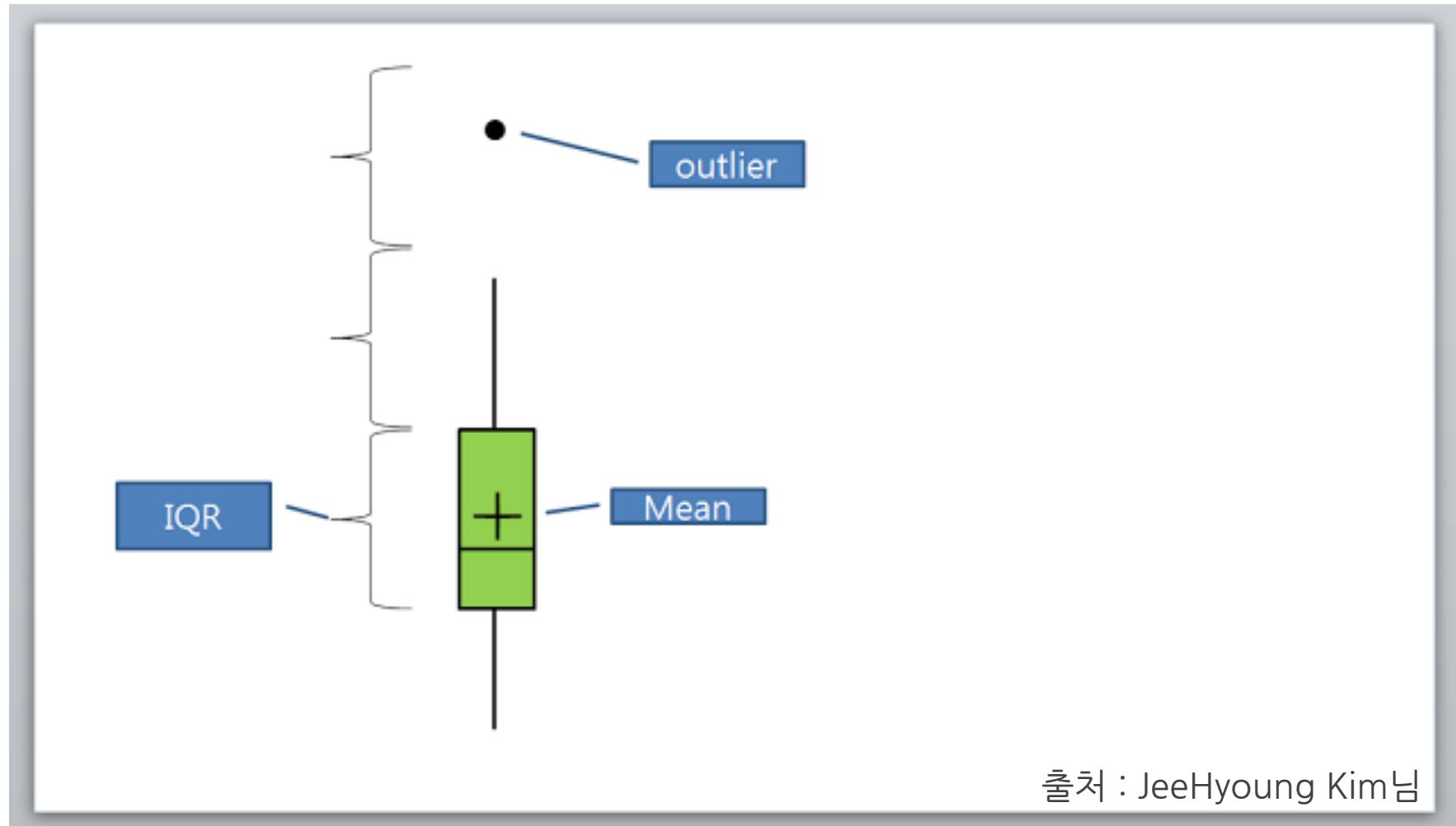


Box Plot



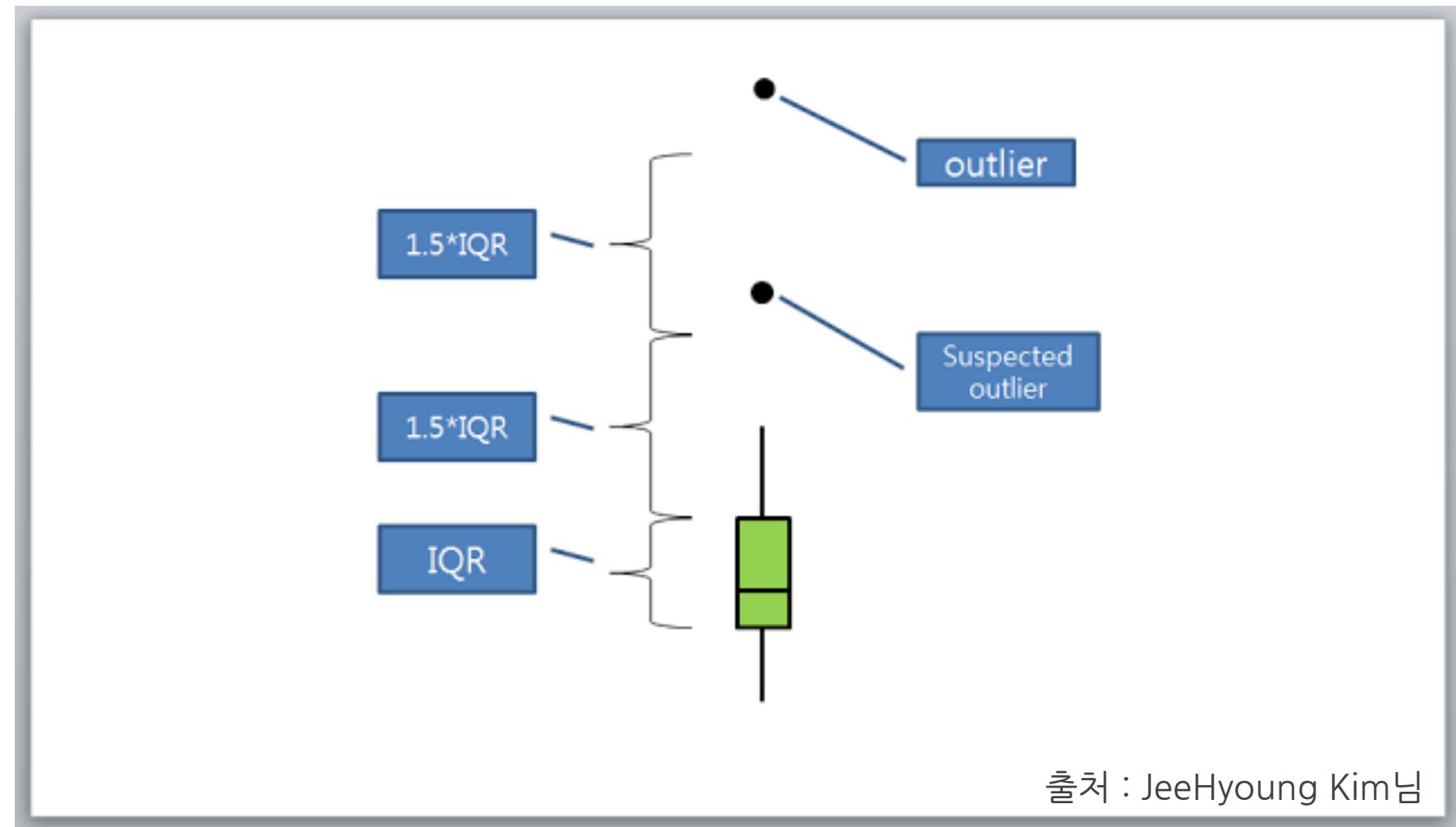
- Median 중심으로 아래 위 상자가 각각 전체 데이터의 25%씩 총 50%를 가진다

Box Plot



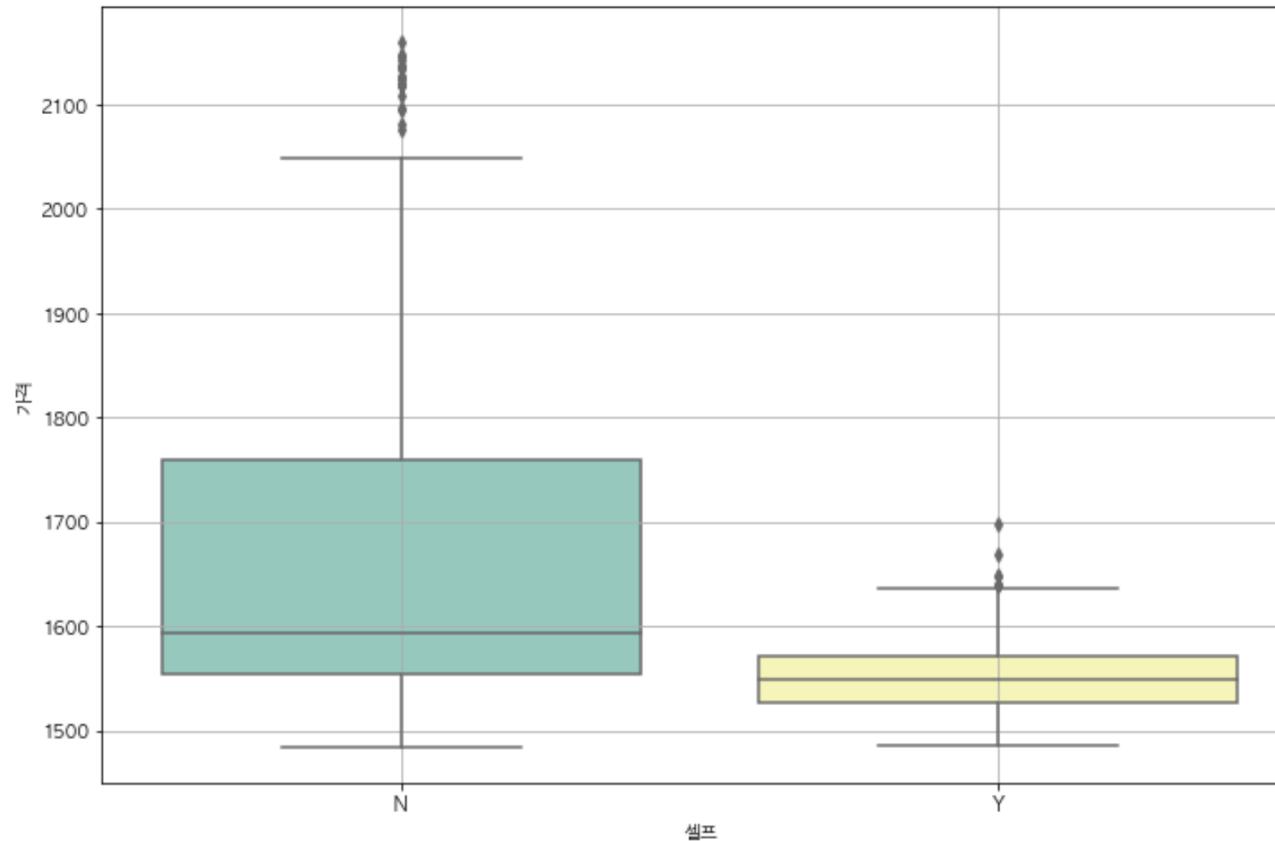
- 상자 전체 길이를 IQR(Inter Quatile Range)라고 한다
- IQR의 1.5배 이상 벗어나면 점으로 별도 표기를 한다

Box Plot

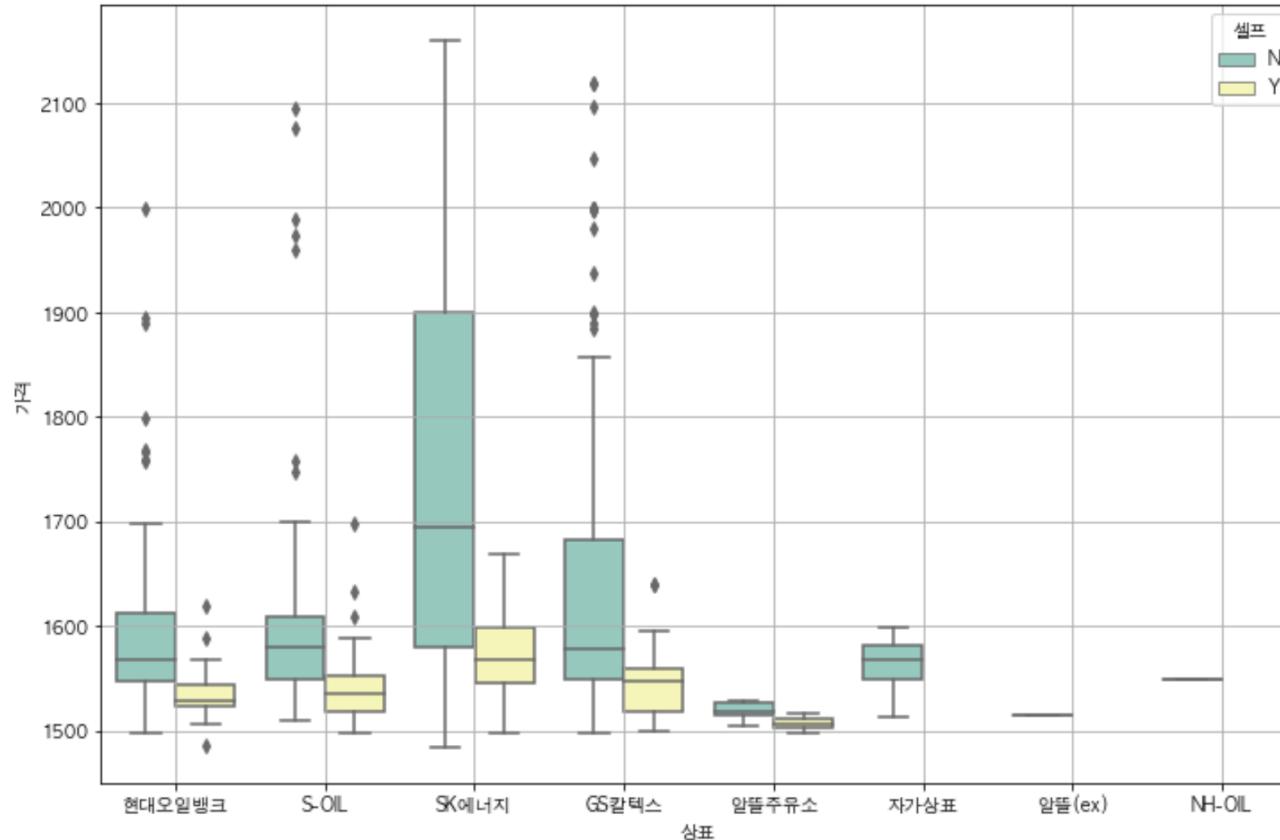


출처 : JeeHyoung Kim님

```
In [38]: plt.figure(figsize=(12, 8))
sns.boxplot(x="셀프", y="가격", data=stations, palette="Set3")
plt.grid()
plt.show()
```



```
In [39]: plt.figure(figsize=(12, 8))
sns.boxplot(x="상표", y="가격", hue="셀프", data=stations, palette="Set3")
plt.grid()
plt.show()
```



- 각 메이커 별 셀프 주유 여부를 포함해서 가격 분포를 보자 ~

```
In [40]: import json
import folium
import warnings

warnings.simplefilter(action="ignore", category=FutureWarning)
```

```
In [41]: stations.sort_values(by="가격", ascending=False).head(10)
```

Out[41]:

	상호	주소	가격	셀프	상표	구
160	뉴서울(강남)	서울 강남구 언주로 716 (논현동)	2160.0	N	SK에너지	강남구
265	장충주유소	서울 중구 장충단로 202 (장충동1가)	2147.0	N	SK에너지	중구
235	양재주유소	서울 서초구 바우뫼로 178 (양재동)	2145.0	N	SK에너지	서초구
264	서남주유소	서울 중구 통일로 30 (봉래동1가)	2143.0	N	SK에너지	중구
159	오천주유소	서울 강남구 봉은사로 503 (삼성동)	2137.0	N	SK에너지	강남구
348	강변주유소	서울특별시 용산구 원효로 9 (청암동)	2136.0	N	SK에너지	용산구
347	갈월동주유소	서울특별시 용산구 한강대로 322 (갈월동)	2136.0	N	SK에너지	용산구
346	흥국주유소	서울 용산구 한남대로 204 (한남동)	2133.0	N	SK에너지	용산구
158	삼성주유소	서울 강남구 삼성로 521 (삼성동)	2127.0	N	SK에너지	강남구
263	통일주유소	서울 중구 동호로 296 (장충동2가)	2126.0	N	SK에너지	중구

```
In [42]: stations.sort_values(by="가격", ascending=True).head(10)
```

Out[42]:

	상호	주소	가격	셀프	상표	구
416	타이거주유소	서울 은평구 수색로 188 (증산동)	1484.0	N	SK에너지	은평구
417	(주)명연에너지 수색훼미리주유소	서울 은평구 수색로 236 (수색동)	1485.0	Y	현대오일뱅크	은평구
268	성락주유소	서울 영등포구 가마산로 414 (신길동)	1498.0	Y	S-OIL	영등포구
269	(주)MS주유소	서울특별시 영등포구 대림로 230 (대림동)	1498.0	N	현대오일뱅크	영등포구
270	(주)강서오일	서울 영등포구 도신로 151 (도림동)	1498.0	N	현대오일뱅크	영등포구
271	(주)대청에너지 대청주유소	서울 영등포구 가마산로 328 (대림동)	1498.0	N	GS칼텍스	영등포구
266	도림주유소	서울 영등포구 도림로 343 (도림동)	1498.0	Y	알뜰주유소	영등포구
267	(주)블루에너지	서울특별시 영등포구 가마산로 367 (대림동)	1498.0	Y	SK에너지	영등포구
272	신길주유소	서울특별시 영등포구 신길로 74 (신길동)	1499.0	Y	GS칼텍스	영등포구
489	풀페이주유소	서울특별시 구로구 경인로 41 (온수동)	1504.0	N	SK에너지	구로구

```
In [43]: import numpy as np
```

```
gu_data = pd.pivot_table(stations, index=["구"], values=["가격"], aggfunc=np.mean)  
gu_data.head()
```

Out[43]: 가격

구

강남구 1804.325581

강동구 1633.166667

강북구 1528.583333

강서구 1580.205882

관악구 1599.411765

```
In [126]: geo_path = "../data/02. skorea_municipalities_geo_simple.json"
geo_str = json.load(open(geo_path, encoding="utf-8"))

my_map = folium.Map(location=[37.5502, 126.982], zoom_start=10.5, tiles="Stamen Toner")

my_map.choropleth(
    geo_data=geo_str,
    data=gu_data,
    columns=[gu_data.index, "가격"],
    fill_color="PuRd",
    key_on="feature.id",
)
my_map
```

데이터 시각화

zero-base /

