
SMARCLE 2022 데이터분석 스터디

4장 셀프주유소는 정말 저렴할까

이은지, 현희섭, 신아진, 김지은

1 Selenium 사용하기

01 Selenium 사용하기

selenium: 웹사이트 테스트를 위한 도구로 브라우저 동작을 자동화할 수 있는 도구.

설치

```
(base) C:\Users\gusgm>pip install selenium
```

01 Selenium 사용하기

ChromeDriver

WebDriver is an open source tool for automated testing of webapps across many browsers. It provides capabilities for navigating to w
ChromeDriver is a standalone server that implements the [W3C WebDriver standard](#). ChromeDriver is available for Chrome on Android
ChromeOS).

You can view the current implementation status of the WebDriver standard [here](#).

All versions available in [Downloads](#)

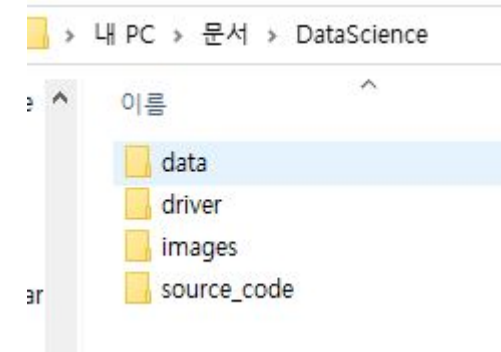
- Latest beta release: [ChromeDriver 97.0.4692.20](#)
- Latest stable release: [ChromeDriver 96.0.4664.45](#)

ChromeDriver Documentation

- [Getting started with ChromeDriver on Desktop](#) (Windows, Mac, Linux)
 - [ChromeDriver with Android](#)
 - [ChromeDriver with ChromeOS](#)
- [ChromeOptions](#), the capabilities of ChromeDriver
- [Mobile emulation](#)
- [Security Considerations](#), with recommendations on keeping ChromeDriver safe
- [Chrome Extension installation](#)
- [Verbose logging](#) and [performance data logging](#)

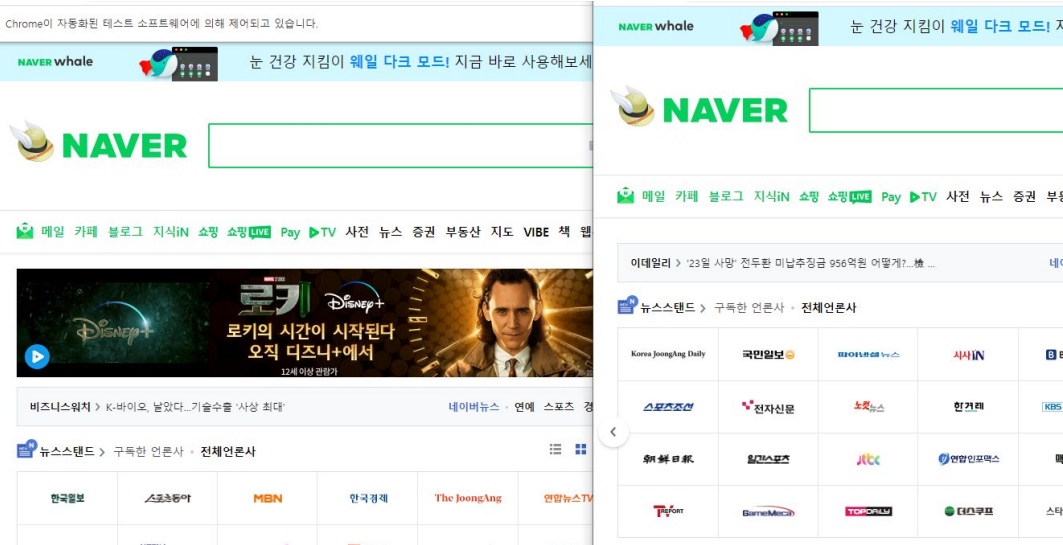
<https://chromedriver.chromium.org/>

DataScience 폴더에 driver 폴더 생성



01 Selenium 사용하기

```
In [1]: from selenium import webdriver # selenium 프레임 워크에서 webdriver 가져오기
url = 'http://naver.com' # 접속할 웹 사이트 주소 (네이버)
driver = webdriver.Chrome(executable_path="C:/Users/gusgm/Documents/DataScience/driver/chromedriver") # 크롬 드라이버로 크롬 켜기
driver.get(url)
```



> 내 PC > 문서 > DataScience > images



001.jpg

```
In [5]: driver.save_screenshot('C:/Users/gusgm/Documents/DataScience/images/001.jpg')

D:\anaconda3\lib\site-packages\selenium\webdriver\remote\webdriver.py:1309: UserWarning: Using the save_screenshot method, the file name should end with a '.png' extension
  warnings.warn("name used for saved screenshot does not match file ")

Out [5]: True
```

01 Selenium 사용하기

예전

아이디

로그인

비밀번호

일회용 로그인

IP보안 **ON**

☐ 로그인 상태 유지

[회원가입](#)

[아이디/비밀번호 찾기](#)

지금

네이버를 더 안전하고 편리하게 이용하세요

NAVER 로그인

☐ 아이디 · 비밀번호 찾기

☐ 회원가입

NAVER

PC방 등 공용PC라면 QR코드 로그인이 더 안전해요 X

ID 로그인

[1] 일회용 번호

QR코드

아이디

비밀번호

☒ 로그인 상태 유지

IP보안 ☒

로그인

[비밀번호 찾기](#) | [아이디 찾기](#) | [회원가입](#)

01 Selenium 사용하기

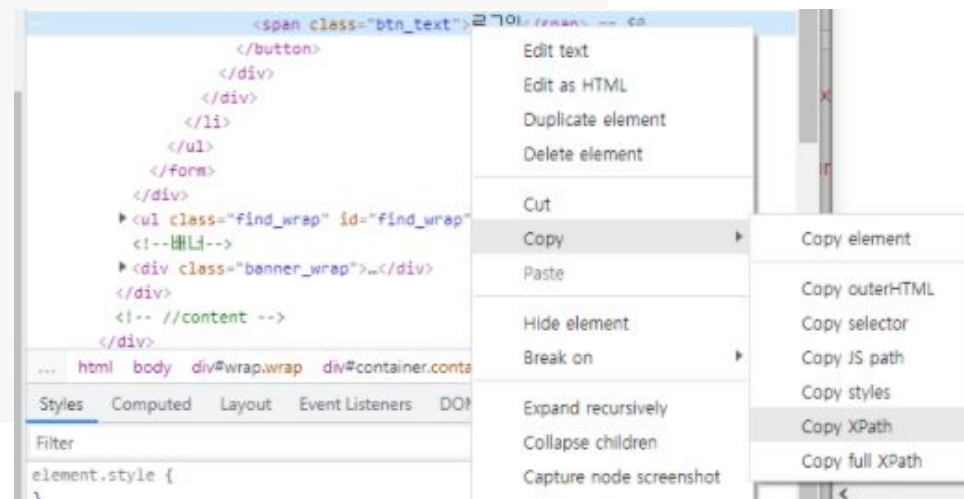
```
In [13]: import time
driver = webdriver.Chrome(executable_path="C:/Users/gusgm/Documents/DataScience/driver/chromedriver")

#로그인 링크 이동
driver.get('https://nid.naver.com/nidlogin.login?mode=form&url=https%3A%2F%2Fwww.naver.com')
time.sleep(1)

# 로그인 정보 입력
elem = driver.find_element_by_id("id")
elem.send_keys("****")
time.sleep(1)

elem = driver.find_element_by_id("pw")
elem.send_keys("****")
time.sleep(1)

#로그인 버튼 클릭
xpath = "//*[@id='log_login']/span"
driver.find_element_by_xpath(xpath).click()
```



01 Selenium 사용하기

그러나..

이름	개수	가격
명란 아보 번	1	5,400
갈릭 베이컨 치즈	1	4,700
<hr/>		
판 매 금 액 :	18.000원	
부 가 세 액 :	1.800원	
총 액 :	19.800원	

해당 영수증은 가상으로 제작된 것으로 실제 영수증 사진이 아닙니다.

영수증에서 구매한 물건은 총 몇 개 입니까?

정답을 입력해주세요.

↺

🔊

01 Selenium 사용하기

사용하기 위해서는

```
In [3]: import time
from time import sleep
driver = webdriver.Chrome(executable_path="C:/Users/gusgm/Documents/DataScience/driver/chromedriver")

#로그인 링크 이동
driver.get('https://nid.naver.com/nidlogin.login?mode=form&url=https%3A%2F%2Fwww.naver.com')
time.sleep(1)

# 로그인 정보 입력
sleep(1)
driver.execute_script("document.getElementsByName('id')[0].value=₩" + '본인 아이디' + "₩")

sleep(1)
driver.execute_script("document.getElementsByName('pw')[0].value=₩" + '본인 비밀번호' + "₩")

#로그인 버튼 클릭
xpath= ""//*[@id="log_login"]/span""
driver.find_element_by_xpath(xpath).click()
```

```
In [4]: driver.close()
```

2 서울시 구별 주유소 가격 정보 얻기

02 서울시 구별 주유소 가격 정보 얻기

```
In [11]: driver = webdriver.Chrome(executable_path="C:/Users/gusgm/Documents/DataScience/driver/chromedriver.exe")
driver.get("http://www.opinet.co.kr/searPgSelect.do")
```

Chrome이 자동화된 테스트 소프트웨어에 의해 제어되고 있습니다.

The screenshot shows the Opinet website with a header for 'Opinet' and '국내유가통계'. The main content area features a large image of a city skyline and a table for '오늘의 유가 (2021.11.23)'. The table lists prices for 휘발유 (Gasoline), 경유 (Diesel), and LPG. Below this, there is a map of South Korea showing average gas prices by province: 인천 (1676), 경기 (1691), 서울 (1743), 강원 (1692), 충청 (1694), and 대구 (1707). The right sidebar shows '전국평균 (원/리터)' and '유가추이' (Price Trend) for the last week (1주).

오늘의 유가 (2021.11.23)	휘발유	경유	LPG
전국평균 (원/리터)			

시도별 평균	유가추이
인천 1676	1주
경기 1691	1707
서울 1743	1704
강원 1692	
충청 1694	
대구 1707	

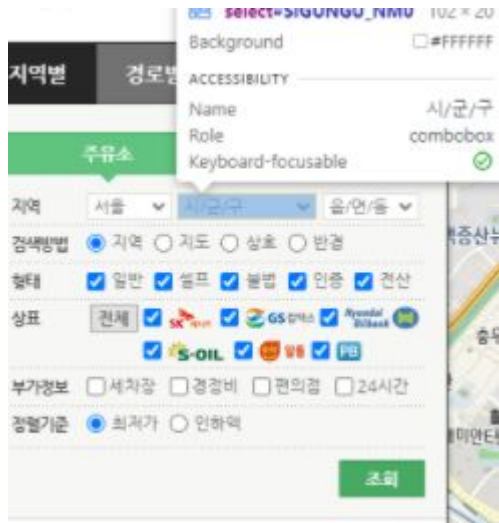
싼 주유소찾기 -> 지역별

The screenshot shows the search results for gas stations in Seoul. The filters are set to '지역별' (By Region) and '서울' (Seoul). The search results table shows the average price for 휘발유 (Gasoline) as 1693 and for (인하액) (Discounted Price) as 1600. The table also lists the search results for '서창산업(주) 용...' (Seochang Industry Co., Ltd. Yong...). The right sidebar shows a map of Seoul with gas station locations marked.

지역별	경로별	도로별	면세유		
주유소	충전소				
지역	서울	시/군/구	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 checked="" type="checkbox"/> 전체	<input checked="" type="checkbox"/> SK	<input checked="" type="checkbox"/> GS	<input checked="" type="checkbox"/> GS	<input checked="" type="checkbox"/> Hyundai
부가정보	<input type="checkbox"/> 세차장	<input type="checkbox"/> 경정비	<input type="checkbox"/> 편의점	<input type="checkbox"/> 24시간	
정렬기준	<input checked="" type="radio"/> 최저가	<input type="radio"/> 인하액			

고급휘발유	보통휘발유	경유	실내등유	
검색결과 (총 100개)	<input checked="" type="radio"/> 불법	<input checked="" type="radio"/> 셀프	<input checked="" type="radio"/> 인증	<input checked="" type="radio"/> 전산
주유소명	휘발유	(인하액)		
평균가격	1693			
서창산업(주) 용...	1600	(↓164)		

02 서울시 구별 주유소 가격 정보 얻기



```
In [12]: gu_list_raw = driver.find_element_by_xpath("//*[id='SIGUNGU_NMO']")  
gu_list = gu_list_raw.find_elements_by_tag_name("option")
```

```
In [12]: gu_names = [option.get_attribute("value") for option in gu_list]  
gu_names.remove('')  
gu_names
```

```
Out [12]: ['강남구',  
'강동구',  
'강북구',  
'강서구',  
'관악구',  
'광진구',  
'구로구',  
'금천구',  
'노원구',  
'도봉구',  
'동대문구',  
'동작구',  
'마포구',  
'서대문구',  
'서초구',  
'성동구',  
'성북구',  
'송파구',  
'양천구',  
'영등포구',  
'용산구',  
'은평구',  
'종로구',  
'중구',  
'중랑구']
```

02 서울시 구별 주유소 가격 정보 얻기

```
In [29]: element = driver.find_element_by_id("SIGUNGU_NMD")
         element.send_keys(gu_names[1])
```

```
In [25]: xpath = "//*[@id='glopopd_excel']"
         element_get_excel = driver.find_element_by_xpath(xpath).click()
```

지역: 서울 강남구 읍/면/동

검색방법: ☒ 지역 ☐ 지도 ☐ 상호 ☐ 반경

형태: ☒ 일반 ☒ 셀프 ☒ 불법 ☒ 인증 ☒ 전산

상표: 전체 ☒ SK ☒ GS칼텍스 ☒ Hyundai Oilbank ☒ S-OIL ☒ 알뜰 ☒ PB

부가정보: ☐ 세차장 ☐ 경정비 ☐ 편의점 ☐ 24시간

```
In [24]: xpath = "//*[@id='searAgSelect']"
         element_sel_gu = driver.find_element_by_xpath(xpath).click()
```

부가정보: ☐ 세차장 ☐ 경정비 ☐ 편의점 ☐ 24시간

정렬기준: ☒ 최저가 ☐ 인하액

조회

고급휘발유	보통휘발유	경유	실내등유
검색결과 (총 35개) 불법 셀프 인증 전산			
주유소명		휘발유	(인하액)
평균가격		1871	
 (주)보성 세곡주...	5	1671	(↓164)
 오일플러스 셀프	5	1679	(↓161)
 현대오일뱅크 도곡...	5	1679	(↓164)
 현대오일뱅크(주)...	5	1681	(↓164)
 현대오일뱅크(주)...	5	1694	(↓164)
 SK서광주유소	5 5	1725	(↓150)
 방죽주유소	5	1727	(↓130)
 (주)중앙에너지...	5	1728	(↓190)
 자곡셀프주유소	5	1730	(↓156)
엑셀저장			

02 서울시 구별 주유소 가격 정보 얻기

```
In [26]: import time
from tqdm import tqdm_notebook

for gu in tqdm_notebook(gu_names):
    element = driver.find_element_by_id("SIGUNGU_NMO")
    element.send_keys(gu)

    time.sleep(2)

    xpath = "//*[@id='searRgSelect']"
    element_sel_gu = driver.find_element_by_xpath(xpath).click()

    time.sleep(1)

    xpath = "//*[@id='glopopd_excel']"
    element_get_excel = driver.find_element_by_xpath(xpath).click()

    time.sleep(1)
```

```
In [27]: driver.close()
```

지역_위치별(주유소) (1).xls
지역_위치별(주유소) (2).xls
지역_위치별(주유소) (3).xls
지역_위치별(주유소) (4).xls
지역_위치별(주유소) (5).xls
지역_위치별(주유소) (6).xls
지역_위치별(주유소) (7).xls
지역_위치별(주유소) (8).xls
지역_위치별(주유소) (9).xls
지역_위치별(주유소) (10).xls
지역_위치별(주유소) (11).xls
지역_위치별(주유소) (12).xls
지역_위치별(주유소) (13).xls
지역_위치별(주유소) (14).xls
지역_위치별(주유소) (15).xls
지역_위치별(주유소) (16).xls
지역_위치별(주유소) (17).xls
지역_위치별(주유소) (18).xls
지역_위치별(주유소) (19).xls
지역_위치별(주유소) (20).xls
지역_위치별(주유소) (21).xls
지역_위치별(주유소) (22).xls
지역_위치별(주유소) (23).xls

3 구별 주유 가격에 대한 데이터의 정리

```
In [1]: import pandas as pd
        from glob import glob

        station_files = glob('../data/지역*.xls')
        station_files
```

```
Out[1]: ['../data/지역_위치별(주유소) (1).xls',
        '../data/지역_위치별(주유소) (10).xls',
        '../data/지역_위치별(주유소) (11).xls',
        '../data/지역_위치별(주유소) (12).xls',
        '../data/지역_위치별(주유소) (13).xls',
        '../data/지역_위치별(주유소) (14).xls',
        '../data/지역_위치별(주유소) (15).xls',
        '../data/지역_위치별(주유소) (16).xls',
        '../data/지역_위치별(주유소) (17).xls',
        '../data/지역_위치별(주유소) (18).xls',
        '../data/지역_위치별(주유소) (19).xls',
        '../data/지역_위치별(주유소) (2).xls',
        '../data/지역_위치별(주유소) (20).xls',
        '../data/지역_위치별(주유소) (21).xls',
        '../data/지역_위치별(주유소) (22).xls',
        '../data/지역_위치별(주유소) (23).xls',
        '../data/지역_위치별(주유소) (24).xls',
        '../data/지역_위치별(주유소) (3).xls',
        '../data/지역_위치별(주유소) (4).xls',
        '../data/지역_위치별(주유소) (5).xls',
        '../data/지역_위치별(주유소) (6).xls',
        '../data/지역_위치별(주유소) (7).xls',
        '../data/지역_위치별(주유소) (8).xls',
        '../data/지역_위치별(주유소) (9).xls']
```

모듈 import
지역 파일 읽기

```
stations_files = glob('../data/지역_위치별*.xls')
stations_files
```

```
['../data/지역_위치별(주유소) (1).xls',
 '../data/지역_위치별(주유소) (10).xls',
 '../data/지역_위치별(주유소) (11).xls',
 '../data/지역_위치별(주유소) (12).xls',
 '../data/지역_위치별(주유소) (13).xls',
 '../data/지역_위치별(주유소) (14).xls',
 '../data/지역_위치별(주유소) (15).xls',
 '../data/지역_위치별(주유소) (16).xls',
 '../data/지역_위치별(주유소) (17).xls',
 '../data/지역_위치별(주유소) (18).xls',
 '../data/지역_위치별(주유소) (19).xls',
 '../data/지역_위치별(주유소) (2).xls',
 '../data/지역_위치별(주유소) (20).xls',
 '../data/지역_위치별(주유소) (21).xls',
 '../data/지역_위치별(주유소) (22).xls',
 '../data/지역_위치별(주유소) (23).xls',
 '../data/지역_위치별(주유소) (24).xls',
 '../data/지역_위치별(주유소) (3).xls',
 '../data/지역_위치별(주유소) (4).xls',
 '../data/지역_위치별(주유소) (5).xls',
 '../data/지역_위치별(주유소) (6).xls',
 '../data/지역_위치별(주유소) (7).xls',
 '../data/지역_위치별(주유소) (8).xls',
 '../data/지역_위치별(주유소) (9).xls',
 '../data/지역_위치별(주유소).xls']
```

변수에 각 엑셀 파일의 경로와 이름을
리스트로 저장

03 구별 주유 가격에 대한 데이터의 정리

```
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)
```

```
station_raw.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 537 entries, 0 to 45
Data columns (total 10 columns):
지역          537 non-null object
상호          537 non-null object
주소          537 non-null object
상표          537 non-null object
전화번호      537 non-null object
셀프여부      537 non-null object
고급휘발유    537 non-null object
휘발유        537 non-null object
경유          537 non-null object
실내등유      537 non-null object
dtypes: object(10)
memory usage: 46.1+ KB
```

```
station_raw.head()
```

```
station_raw.head()
```

	지역	상호	주소	상표	전화번호	셀프여부	고급휘발유	휘발유	경유	실내등유
0	서울특별시	오렌지주유소	서울 강동구 성안로 102 (성내동)	SK에너지	02-484-6165	N	-	1554	1354	997
1	서울특별시	구천면주유소	서울 강동구 구천면로 357 (암사동)	현대오일뱅크	02-441-0536	N	-	1556	1355	-
2	서울특별시	GS칼텍스(주)직영 신월주유소	서울 강동구 양재대로 1323 (성내동)	GS칼텍스	02-475-2600	N	1855	1559	1349	1000
3	서울특별시	광성주유소	서울 강동구 올림픽로 673 (천호동)	S-OIL	02-470-5133	N	-	1578	1388	1100
4	서울특별시	(주)소모에너지엔테크놀로지성내주유소	서울 강동구 올림픽로 578 (성내동)	GS칼텍스	02-479-3838	Y	-	1588	1388	-

```
stations = pd.DataFrame({'Oil_store':station_raw['상호'],
                        '주소':station_raw['주소'],
                        '가격':station_raw['휘발유'],
                        '셀프':station_raw['셀프여부'],
                        '상표':station_raw['상표'] })

stations.head()
```

03 구별 주유 가격에 대한 데이터의 정리

	Oil_store	가격	상표	셀프	주소
0	오렌지주유소	1554	SK에너지	N	서울 강동구 성안로 102 (성내동)
1	구천면주유소	1556	현대오일뱅크	N	서울 강동구 구천면로 357 (암사동)
2	GS칼텍스(주)직영 신월주유소	1559	GS칼텍스	N	서울 강동구 양재대로 1323 (성내동)
3	광성주유소	1578	S-OIL	N	서울 강동구 올림픽로 673 (천호동)
4	(주)소모에너지엔테크놀로지성내주유소	1588	GS칼텍스	Y	서울 강동구 올림픽로 578 (성내동)

```
stations['구'] = [eachAddress.split()[1] for eachAddress in stations['주소']]
stations.head()
```

주소에서 구 이름 추출

	Oil_store	가격	상표	셀프	주소	구
0	오렌지주유소	1554	SK에너지	N	서울 강동구 성안로 102 (성내동)	강동구
1	구천면주유소	1556	현대오일뱅크	N	서울 강동구 구천면로 357 (암사동)	강동구
2	GS칼텍스(주)직영 신월주유소	1559	GS칼텍스	N	서울 강동구 양재대로 1323 (성내동)	강동구
3	광성주유소	1578	S-OIL	N	서울 강동구 올림픽로 673 (천호동)	강동구
4	(주)소모에너지엔테크놀로지성내주유소	1588	GS칼텍스	Y	서울 강동구 올림픽로 578 (성내동)	강동구

03 구별 주유 가격에 대한 데이터의 정리

```
stations['구'].unique()
```

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

- unique() 검사를 수행하니, '서울특별시'와 '특별시' 항목이 구 이름이 아닌데 들어있는 것 발견

```
stations[stations['구']=='서울특별시']
```

	Oil_store	가격	상표	셀프	주소	구
12	SK네트웍스(주)효진주유소	1654	SK에너지	N	1 서울특별시 성동구 동일로 129 (성수동2가)	서울특별시

```
stations.loc[stations['구']=='서울특별시', '구'] = '성동구'  
stations['구'].unique()
```

03 구별 주유 가격에 대한 데이터의 정리

```
stations[stations['가격']=='-']
```

	Oil_store	가격	상표	셀프	주소	구
18	명진석유(주)등서울주유소	-	GS칼텍스	Y	서울특별시 강동구 천호대로 1456 (상일동)	강동구
33	하나주유소	-	S-OIL	N	서울특별시 영등포구 도림로 236 (신길동)	영등포구
12	(주)에이앤이청담주유소	-	SK에너지	Y	서울특별시 강북구 도봉로 155 (미아동)	강북구
13	송정주유소	-	자가상표	N	서울특별시 강북구 인수봉로 185 (수유동)	강북구

- 가격정보 입력 X 주유소는 제외

```
stations = stations[stations['가격'] != '-']  
stations.head()
```

	Oil_store	가격	상표	셀프	주소	구
0	오렌지주유소	1554	SK에너지	N	서울 강동구 성안로 102 (성내동)	강동구
1	구천면주유소	1556	현대오일뱅크	N	서울 강동구 구천면로 357 (암사동)	강동구
2	GS칼텍스(주)직영 신월주유소	1559	GS칼텍스	N	서울 강동구 양재대로 1323 (성내동)	강동구
3	광성주유소	1578	S-OIL	N	서울 강동구 올림픽로 673 (천호동)	강동구
4	(주)소모에너지엔테크놀로지성내주유소	1588	GS칼텍스	Y	서울 강동구 올림픽로 578 (성내동)	강동구

03 구별 주유 가격에 대한 데이터의 정리

```
stations['가격'] = [float(value) for value in stations['가격']]
```

- 가격 정보 숫자형으로 변환

```
stations.reset_index(inplace=True)  
del stations['index']
```

- 인덱스 재기록

```
stations.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 533 entries, 0 to 532  
Data columns (total 6 columns):  
Oil_store    533 non-null object  
가격         533 non-null float64  
상표         533 non-null object  
셀프         533 non-null object  
주소         533 non-null object  
구           533 non-null object  
dtypes: float64(1), object(5)  
memory usage: 25.1+ KB
```

```
stations.head()
```

	Oil_store	가격	상표	셀프	주소	구
0	오렌지주유소	1554.0	SK에너지	N	서울 강동구 성안로 102 (성내동)	강동구
1	구천면주유소	1556.0	현대오일뱅크	N	서울 강동구 구천면로 357 (암사동)	강동구
2	GS칼텍스췌직영 신월주유소	1559.0	GS칼텍스	N	서울 강동구 양재대로 1323 (성내동)	강동구
3	광성주유소	1578.0	S-OIL	N	서울 강동구 올림픽로 673 (천호동)	강동구
4	(주)소모에너지엔테크놀로지성내주유소	1588.0	GS칼텍스	Y	서울 강동구 올림픽로 578 (성내동)	강동구

- 데이터 준비 완료

4 셀프 주유소는 정말 저렴한지 boxplot으로 확인하기

04 셀프 주유소는 정말 저렴한지 boxplot으로 확인하기

필요한 모듈 import

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

import platform
```

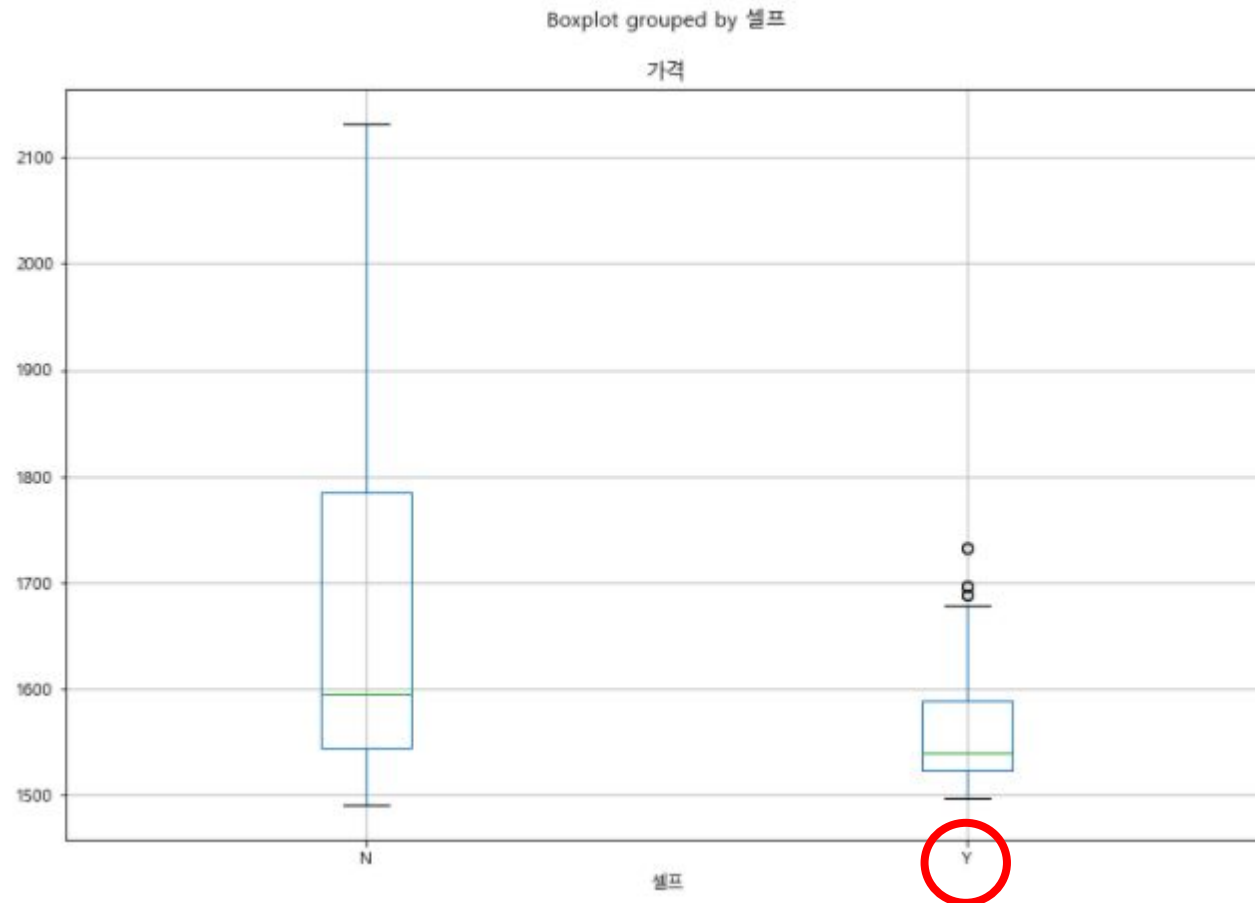
한글 문제를 해결하는 코드 준비

```
path = "c:/Windows/Fonts/malgun.ttf"          # 한글 문제 해결하는 코드 준비
from matplotlib import font_manager, rc
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~~~~')
```

04 셀프 주유소는 정말 저렴한지 boxplot으로 확인하기

boxplot 이용

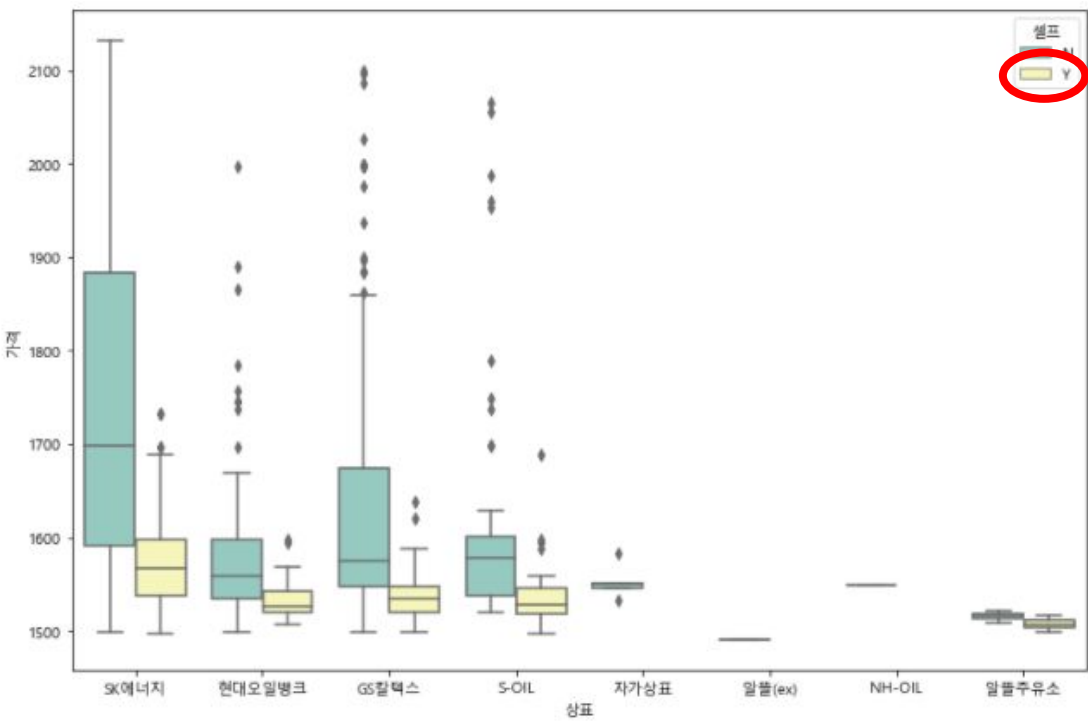
```
stations.boxplot(column='가격', by='셀프', figsize=(12,8));
```



04 셀프 주유소는 정말 저렴한지 boxplot으로 확인하기

주유소의 상표별로 셀프 주유소가 얼마나 저렴한지 확인해보기

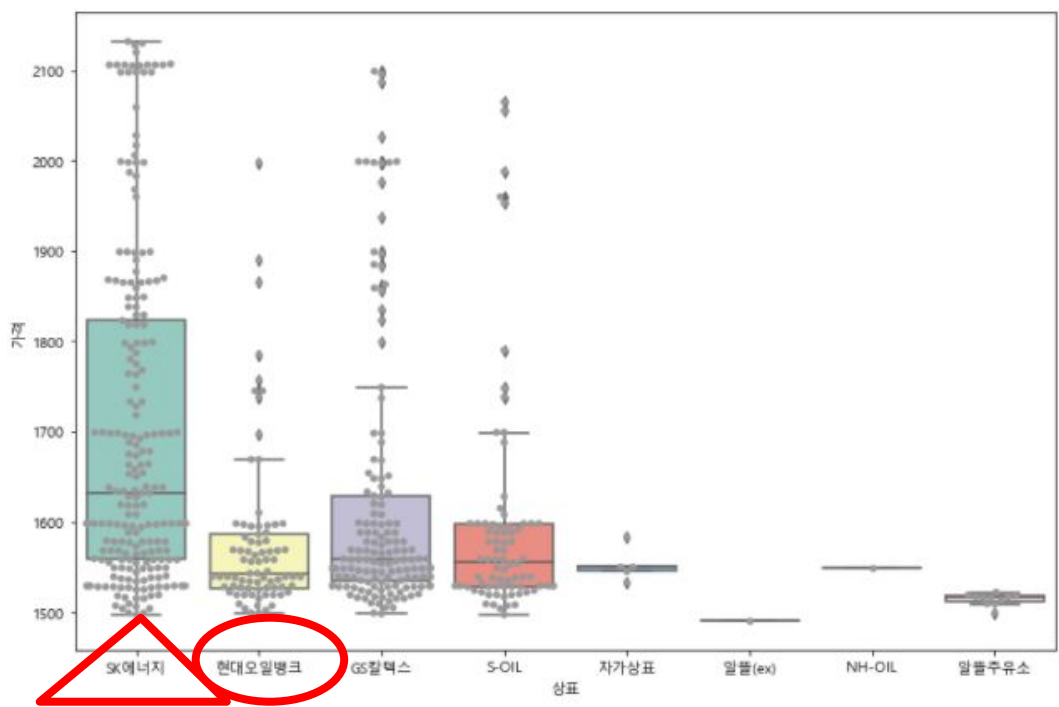
```
# 주유소의 상표별로 셀프 주유소가 얼마나 저렴한지 확인해보기
plt.figure(figsize=(12,8))
sns.boxplot(x="상표", y="가격", hue="셀프", data=stations, palette="Set3")
plt.show()
```



Swarmplot을 이용하여 더 확실히 데이터의 분포를 살펴보기

Swarmplot : 데이터의 분산까지 고려하여 그리고, 데이터가 퍼져있는 정도를 입체적으로 볼 수 있다.

```
# swarmplot를 이용하여 더 확실히 데이터의 분포를 살펴보기
plt.figure(figsize=(12,8))
sns.boxplot(x="상표", y="가격", data=stations, palette="Set3")
sns.swarmplot(x="상표", y="가격", data=stations, color=".6")
plt.show()
```



5 서울시 구별 주유 가격 확인하기

05 서울시 구별 주유 가격 확인하기

필요한 모듈 설치

```
# 필요한 모듈 설치  
!pip install folium  
!pip install googlemaps
```

필요한 모듈 import

```
# 필요한 모듈 import  
import json  
import folium  
import googlemaps  
import warnings  
warnings.simplefilter(action = "ignore", category = FutureWarning) # 경고 인쇄 X
```

- warnings.simplefilter(~)코드 : 경고 제어 문구

05 서울시 구별 주유 가격 확인하기

서울시에서 주유 가격이 비싼 상위 10개 주유소 보기

```
# 서울시에서 가장 주유 가격이 비싼 주유소 상위 10곳 보기
stations.sort_values(by='가격', ascending=False).head(10)
```

Out[98]:

	Oil_store	주소	가격	셀프	상표	구
325	서남주유소	서울 중구 통일로 30 (봉래동1가)	2132.0	N	SK에너지	<u>중구</u>
324	장충주유소	서울 중구 장충단로 202 (장충동1가)	2130.0	N	SK에너지	<u>중구</u>
124	양재주유소	서울 서초구 바우뒀로 178 (양재동)	2128.0	N	SK에너지	서초구
532	뉴서울(강남)	서울 강남구 언주로 716 (논현동)	2120.0	N	SK에너지	<u>강남구</u>
531	오천주유소	서울 강남구 봉은사로 503 (삼성동)	2107.0	N	SK에너지	<u>강남구</u>
313	재동주유소	서울특별시 종로구 을곡로 58 (경운동)	2106.0	N	SK에너지	종로구
285	청파주유소	서울특별시 용산구 청파로 311 (청파동1가)	2106.0	N	SK에너지	<u>용산구</u>
284	갈월동주유소	서울특별시 용산구 한강대로 322 (갈월동)	2106.0	N	SK에너지	<u>용산구</u>
283	강변주유소	서울특별시 용산구 원효로 9 (청암동)	2106.0	N	SK에너지	<u>용산구</u>
323	통일주유소	서울 중구 동호로 296 (장충동2가)	2106.0	N	SK에너지	<u>중구</u>

05 서울시 구별 주유 가격 확인하기

서울시에서 주유 가격이 낮은 하위 10개 주유소 보기

```
# 서울시에서 가장 주유 가격이 낮은 주유소 상위 10곳 보기
stations.sort_values(by='가격', ascending=True).head(10)
```

Out[99]:

	Oil_store	주소	가격	셀프	상표	구
83	만남의광장주유소	서울 서초구 양재대로12길 73-71 (원지동)	1490.0	N	알뜰(ex)	서초구
344	강서오곡셀프주유소	서울특별시 강서구 별말로 254 (오곡동)	1497.0	Y	SK에너지	강서구
451	태릉솔밭주유소	서울특별시 노원구 노원로 49 (공릉동)	1497.0	Y	S-OIL	노원구
258	수유동주유소	서울특별시 강북구 도봉로 395 (수유동)	1498.0	Y	GS칼텍스	강북구
227	(주)강서오일	서울 영등포구 도신로 151 (도림동)	1499.0	N	현대오일뱅크	영등포구
225	도림주유소	서울 영등포구 도림로 343 (도림동)	1499.0	Y	알뜰주유소	영등포구
226	(주)대청에너지 대청주유소	서울 영등포구 가마산로 328 (대림동)	1499.0	N	GS칼텍스	영등포구
415	플페이주유소	서울특별시 구로구 경인로 41 (온수동)	1499.0	N	SK에너지	구로구
326	신일셀프주유소	서울 중랑구 상봉로 58 (망우동)	1499.0	Y	SK에너지	중랑구
40	서경주유소	서울 동작구 대림로 46 (신대방동)	1499.0	N	현대오일뱅크	동작구

05 서울시 구별 주유 가격 확인하기

pivot_table을 이용해서 구별 가격 정보로 변경
가격은 평균값으로 정리

```
import numpy as np

# pivot_table을 이용하여 구별 가격 정보로 변경
# 가격은 평균값으로 정리
gu_data = pd.pivot_table(stations, index=["구"], values=["가격"],
                          aggfunc=np.mean)
gu_data.head()
```

- pivot_table : 데이터 재구조화
- pd.pivot_table(aggfunc=np.mean) : 데이터값을 평균으로 정리

Out[100]:

가격	
구	
구	
강남구	1791.000000
강동구	1656.722222
강북구	1534.333333
강서구	1576.428571
관악구	1603.055556

05 서울시 구별 주유 가격 확인하기

서울시 구별 정보에 대해 지도로 표현

```
# 앞선 정보를 지도로 표현
geo_path = '../data/02. skorea_municipalities_geo_simple.json'
geo_str = json.load(open(geo_path, encoding='utf-8'))

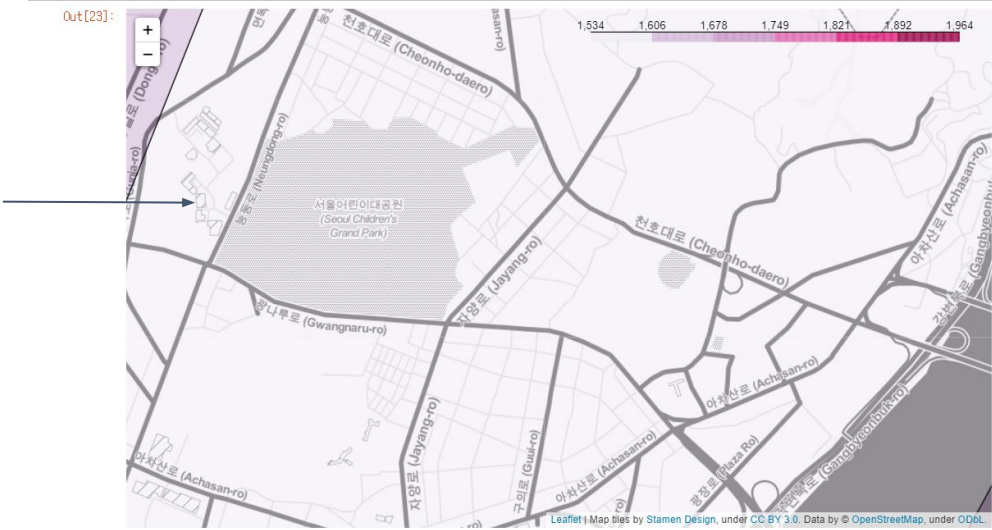
map = folium.Map(location=[37.5502, 126.982], zoom_start=10.5,
                  tiles='Stamen Toner')

map.choropleth(geo_data = geo_str,
               data = gu_data,
               columns=[gu_data.index, '가격'],
               fill_color='PuRd', #PuRd, YlGnBu
               key_on='feature.id')

map
```



세종대학교



6 서울시 주유 가격 상하위 10개 주유소 지도에 표기하기

06 서울시 주유 가격 상하위 10개 주유소 지도에 표기하기

```
oil_price_top10 = stations.sort_values(by='가격', ascending=False).head(10)
oil_price_top10
```

	Oil_store	가격	상표	셀프	주소	구
325	서남주유소	2132.0	SK에너지	N	서울 중구 통일로 30 (봉래동1가)	중구
324	장충주유소	2130.0	SK에너지	N	서울 중구 장충단로 202 (장충동1가)	중구
124	양재주유소	2128.0	SK에너지	N	서울 서초구 바우뒀로 178 (양재동)	서초구
532	뉴서울(강남)	2120.0	SK에너지	N	서울 강남구 언주로 716 (논현동)	강남구
531	오천주유소	2107.0	SK에너지	N	서울 강남구 봉은사로 503 (삼성동)	강남구
313	재동주유소	2106.0	SK에너지	N	서울특별시 종로구 율곡로 58 (경운동)	종로구
285	청파주유소	2106.0	SK에너지	N	서울특별시 용산구 청파로 311 (청파동1가)	용산구
284	갈월동주유소	2106.0	SK에너지	N	서울특별시 용산구 한강대로 322 (갈월동)	용산구
283	강변주유소	2106.0	SK에너지	N	서울특별시 용산구 원효로 9 (청암동)	용산구
323	통일주유소	2106.0	SK에너지	N	서울 중구 동호로 296 (장충동2가)	중구

06 서울시 주유 가격 상하위 10개 주유소 지도에 표기하기

```
oil_price_bottom10 = stations.sort_values(by='가격', ascending=True).head(10)
oil_price_bottom10
```

	Oil_store	가격	상표	셀프	주소	구
83	만남의광장주유소	1490.0	알뜰(ex)	N	서울 서초구 양재대로12길 73-71 (원지동)	서초구
344	강서오곡셀프주유소	1497.0	SK에너지	Y	서울특별시 강서구 별말로 254 (오곡동)	강서구
451	태릉솔밭주유소	1497.0	S-OIL	Y	서울특별시 노원구 노원로 49 (공릉동)	노원구
258	수유동주유소	1498.0	GS칼텍스	Y	서울특별시 강북구 도봉로 395 (수유동)	강북구
227	(주)강서오일	1499.0	현대오일뱅크	N	서울 영등포구 도신로 151 (도림동)	영등포구
225	도림주유소	1499.0	알뜰주유소	Y	서울 영등포구 도림로 343 (도림동)	영등포구
226	(주)대청에너지 대청주유소	1499.0	GS칼텍스	N	서울 영등포구 가마산로 328 (대림동)	영등포구
415	풀페이주유소	1499.0	SK에너지	N	서울특별시 구로구 경인로 41 (온수동)	구로구
326	신일셀프주유소	1499.0	SK에너지	Y	서울 중랑구 상봉로 58 (망우동)	중랑구
40	서경주유소	1499.0	현대오일뱅크	N	서울 동작구 대림로 46 (신대방동)	동작구

06 서울시 주유 가격 상하위 10개 주유소 지도에 표기하기

```
gmap_key = "*****" # 2장에서 구한 자신의 key를 입력
gmaps = googlemaps.Client(key=gmap_key)
```

```
from tqdm import tqdm_notebook

lat = []
lng = []

for n in tqdm_notebook(oil_price_top10.index):
    try:
        tmp_add = str(oil_price_top10['주소'][n]).split('(')[0]
        tmp_map = gmaps.geocode(tmp_add)

        tmp_loc = tmp_map[0].get('geometry')
        lat.append(tmp_loc['location']['lat'])
        lng.append(tmp_loc['location']['lng'])

    except:
        lat.append(np.nan)
        lng.append(np.nan)
        print("Here is nan !")

oil_price_top10['lat'] = lat
oil_price_top10['lng'] = lng
oil_price_top10
```

	Oil_store	가격	상표	셀프	주소	구	lat	lng
325	서남주유소	2132.0	SK에너지	N	서울 중구 통일로 30 (봉래동1가)	중구	37.558348	126.972090
324	장충주유소	2130.0	SK에너지	N	서울 중구 장충단로 202 (장충동1가)	중구	37.562719	127.007010
124	양재주유소	2128.0	SK에너지	N	서울 서초구 바우뒀로 178 (양재동)	서초구	37.478071	127.039610
532	뉴서울(강남)	2120.0	SK에너지	N	서울 강남구 언주로 716 (논현동)	강남구	37.517636	127.035756
531	오천주유소	2107.0	SK에너지	N	서울 강남구 봉은사로 503 (삼성동)	강남구	37.513458	127.053786
313	재동주유소	2106.0	SK에너지	N	서울특별시 종로구 율곡로 58 (경운동)	종로구	37.576388	126.985851
285	청파주유소	2106.0	SK에너지	N	서울특별시 용산구 청파로 311 (청파동1가)	용산구	37.547451	126.969904
284	갈월동주유소	2106.0	SK에너지	N	서울특별시 용산구 한강대로 322 (갈월동)	용산구	37.546955	126.972218
283	강변주유소	2106.0	SK에너지	N	서울특별시 용산구 원효로 9 (청암동)	용산구	37.534910	126.945869
323	통일주유소	2106.0	SK에너지	N	서울 중구 동호로 296 (장충동2가)	중구	37.560714	127.003177

06 서울시 주유 가격 상하위 10개 주유소 지도에 표기하기

```
lat = []
lng = []

for n in tqdm_notebook(oil_price_bottom10.index):
    try:
        tmp_add = oil_price_bottom10['주소'][n].split('(')[0]
        tmp_map = gmaps.geocode(tmp_add)

        tmp_loc = tmp_map[0]['geometry']
        lat.append(tmp_loc['location']['lat'])
        lng.append(tmp_loc['location']['lng'])

    except:
        lat.append(np.nan)
        lng.append(np.nan)
        print("Here is nan !")

oil_price_bottom10['lat'] = lat
oil_price_bottom10['lng'] = lng
oil_price_bottom10
```

	Oil_store	가격	상표	셀프	주소	구	lat	lng
83	만남의광장주유소	1490.0	알뜰(ex)	N	서울 서초구 양재대로12길 73-71 (원지동)	서초구	37.459906	127.042060
344	강서오곡셀프주유소	1497.0	SK에너지	Y	서울특별시 강서구 별말로 254 (오곡동)	강서구	37.555284	126.768567
451	태릉솔밭주유소	1497.0	S-OIL	Y	서울특별시 노원구 노원로 49 (공릉동)	노원구	37.625918	127.085533
258	수유동주유소	1498.0	GS칼텍스	Y	서울특별시 강북구 도봉로 395 (수유동)	강북구	37.642140	127.029568
227	(주)강서오일	1499.0	현대오일뱅크	N	서울 영등포구 도신로 151 (도림동)	영등포구	37.509969	126.908231
225	도림주유소	1499.0	알뜰주유소	Y	서울 영등포구 도림로 343 (도림동)	영등포구	37.507656	126.900191
226	(주)대청에너지 대청주유소	1499.0	GS칼텍스	N	서울 영등포구 가마산로 328 (대림동)	영등포구	37.498556	126.895791
415	풀페이주유소	1499.0	SK에너지	N	서울특별시 구로구 경인로 41 (온수동)	구로구	37.491009	126.826675
326	신일셀프주유소	1499.0	SK에너지	Y	서울 중랑구 상봉로 58 (망우동)	중랑구	37.590907	127.093834
40	서경주유소	1499.0	현대오일뱅크	N	서울 동작구 대림로 46 (신대방동)	동작구	37.487947	126.908240

06 서울시 주유 가격 상하위 10개 주유소 지도에 표기하기

```
map = folium.Map(location=[37.5202, 126.975], zoom_start=10.5)

for n in oil_price_top10.index:
    if pd.notnull(oil_price_top10['lat'][n]):
        folium.CircleMarker([oil_price_top10['lat'][n], oil_price_top10['lng'][n]],
                             radius=15, color='#CD3181',
                             fill_color='#CD3181',
                             fill=True).add_to(map)

for n in oil_price_bottom10.index:
    if pd.notnull(oil_price_bottom10['lat'][n]):
        folium.CircleMarker([oil_price_bottom10['lat'][n],
                             oil_price_bottom10['lng'][n]],
                             radius=15, color='#3186cc',
                             fill_color='#3186cc',
                             fill=True).add_to(map)
```

map

Folium

Folium은 Leaflet.js를 이용하여 위치 좌표계 값을 지도 위에 시각화할 수 있다.

Map

먼저 Map 클래스를 로드해서 범위를 지정해준다. location 속성으로 주시하고자 하는 지도의 정중앙 좌표를 설정하고,

zoom_start 속성으로 배율을 설정한다.  SMARCLE

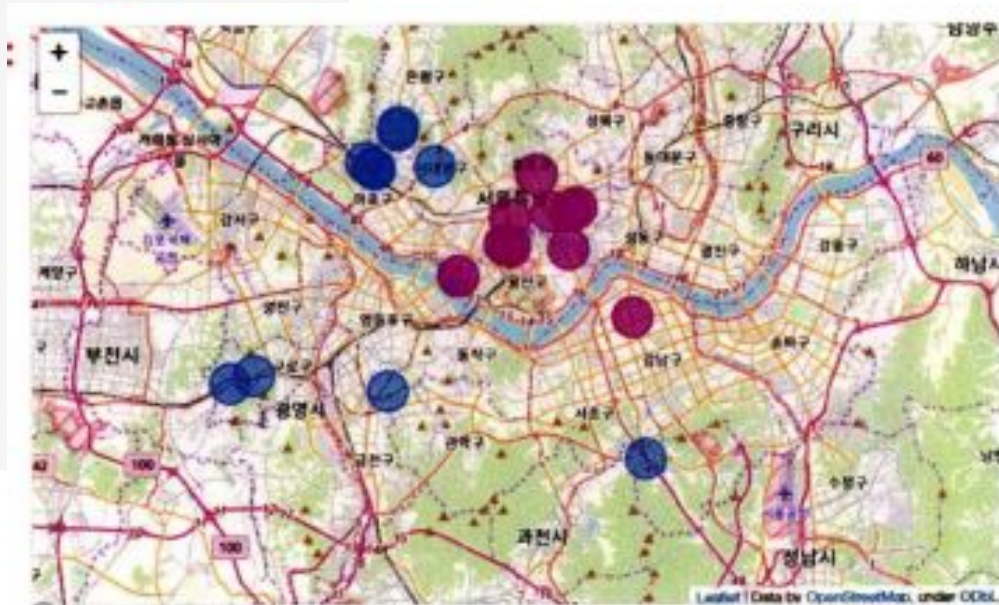
06 서울시 주유 가격 상하위 10개 주유소 지도에 표기하기

```
map = folium.Map(location=[37.5202, 126.975], zoom_start=10.5)
```

```
for n in oil_price_top10.index:  
    if pd.notnull(oil_price_top10['lat'][n]):  
        folium.CircleMarker([oil_price_top10['lat'][n], oil_price_top10['lng'][n]],  
                             radius=15, color='#CD3181',  
                             fill_color='#CD3181',  
                             fill=True).add_to(map)
```

```
for n in oil_price_bottom10.index:  
    if pd.notnull(oil_price_bottom10['lat'][n]):  
        folium.CircleMarker([oil_price_bottom10['lat'][n],  
                             oil_price_bottom10['lng'][n]],  
                             radius=15, color='#3186cc',  
                             fill_color='#3186cc',  
                             fill=True).add_to(map)
```

map



06 느낀 점

이은지

웹사이트 테스트를 위한 도구로 브라우저 동작을 자동화할 수 있는 selenium을 사용해봄으로써 웹크롤링에 대해 더 잘 알게 된 것 같다. 현재 적용되지 않은 코드들이 많아 구글링을 하여 더 찾아봐야 했는데 이를 통해 더 잘 알게 된 것 같다.

신아진

모듈들의 코드가 옛날과는 달라진 부분이 있어, 그거에 맞춰서 코드를 수정하고 재구성하는것이 힘들었다. 그렇지만, 이를 해결하기 위해 직접 모듈들의 쓰임을 구글링해보고, 그렇게 스스로 코드의 오류를 고치는 과정을 통해 오히려 모듈들을 더 자세히 이해할 수 있는 계기가 되어 더 뜻깊었다.

현희섭

책에 오류가 가끔씩 있어서 헛갈릴 때도 있었고, 무엇보다도 이러한 분석을 해보는 것은 처음이어서 오류도 많이 나왔다. 그래도 천천히 오류를 해결해가고 최종적으로 다 했을 때는 뿌듯함을 느꼈다. 이러한 활동을 많이 해서 다음에는 더 잘 해보고 싶다.

김지은

selenium이라는 새로운 라이브러리를 경험해볼 수 있어서 의미있었다. 이를 이용하여 웹에서 데이터를 가져오고, 가져온 데이터를 원하는 대로 사용해 본 경험이 생겼기 때문에 앞으로도 비슷한 스터디를 해볼 기회가 생긴다면, 더 익숙하게 진행할 수 있을 것 같다.

셀레니움 네이버 캡차 우회 방법

<https://youtu.be/UenvOvag0B4>



나 개발 한 번 해보려고 2년 전

자동입력방지 문자가 나타날 경우에는 스크립트로 입력하는 방식을 사용하면 되네요



```
driver.execute_script("document.getElementsByName('id')[0].value=\" + id + "\"")
driver.execute_script("document.getElementsByName('pw')[0].value=\" + pw + "\"")
```



7



답글

```
# 아이디와 비밀번호를 입력합니다. (0.5초씩 기다리기)
```

```
sleep(1)
```

```
driver.execute_script("document.getElementsByName('id')[0].value=\\\" + '본인아이디' + '\\\"")
```

```
sleep(1)
```

```
driver.execute_script("document.getElementsByName('pw')[0].value=\\\" + '본인비밀번호' + '\\\"")
```

```
# 자동 클릭
```

```
driver.find_element_by_xpath('//*[@id="frmNIDLogin"]/ul/li/div/div[7]').click()
```

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

```
#메일함 이동
```

```
driver.get("http://mail.naver.com")
```

Thank you
