

웹 프로그래밍

학습내용 및 목표

학습 내용

1. 크롤링을 통한 인터넷 자료 수집
2. 수집한 자료를 활용한 지도 시각화

학습 목표

1. 파이썬에서 크롤링하여 19대 대선 결과 자료를 수집할 수 있다.
2. 전국의 행정구역 별 대선 결과를 나타내는 지도를 그릴 수 있다.

웹 크롤링

■ 웹 크롤링

- 웹의 데이터를 자동 수집하는 작업
- HTML 구조를 파악하여 라이브러리를 이용하거나, HTML 구조를 파악하지 않고 API를 이용하는 방법이 있음

■ 크롤링 라이브러리

- BeautifulSoup
 - 웹사이트 크롤링에서 가장 많이 사용되는 라이브러리
 - 내부적으로 사용되는 파서를 목적에 맞게 변경 가능
- selenium
 - 복잡한 구조의 웹사이트 크롤링 가능
 - 크롬과 같은 브라우저를 이용하여 동작
- lxml
 - BeautifulSoup보다 빠른 속도로 웹사이트 크롤링 실행

웹 크롤링

■ 웹 페이지 유형

- 크롤링 하려는 페이지 내에서 개발자 모드(또는 `ctrl + U`)으로 HTML 구조 확인

정적 웹 페이지



- 웹 서버에 미리 저장된 파일을 그대로 전달하는 웹 페이지
- 사용 라이브러리: BeautifulSoup

동적 웹 페이지



- 주소(url)만 이용하여 접속할 수 없는 웹 페이지
- 실시간 데이터 업데이트에 따른 콘텐츠 변화
- 사용 라이브러리: BeautifulSoup, selenium

웹 크롤링

■ 웹 크롤링 허용 여부 판단

- 웹사이트에서 크롤링 허용 여부를 알려주는 역할을 하는 텍스트 파일(robots.txt) 확인
- 검색엔진 크롤러의 과도한 크롤링 및 요청으로 인한 과부하 방지 역할



크롤링할 주소/robots.txt



모든 접근 허용

```
User-agent: *  
Allow: /
```

모든 접근 차단

```
User-agent: *  
Disallow: /
```

특정 디렉토리
접근 차단

```
User-agent: *  
Disallow: /user/
```

모든 문서 접근 차단,
첫 페이지만 허용

```
User-agent: *  
Disallow: /  
Allow : /$
```

웹 크롤링

■ HTML5 구조

- 크롤링 하려는 페이지 내에서 개발자 모드(또는 `ctrl + U`)으로 HTML 구조 확인

개발자모드



ctrl + U



웹 크롤링

■ HTML5 구조

• 기본 용어

명칭	설명
요소	<div><ul style="list-style-type: none">제목, 본문, 이미지 등 HTML 페이지에 위치</div> <div><h1> 콘텐츠 </h1></div> <div>태그 요소 태그</div>
태그	<div><ul style="list-style-type: none">요소를 생성할 때 사용하는 기호소문자로 작성태그의 콘텐츠로 태그 포함 가능</div>
속성	<div><ul style="list-style-type: none">태그에 추가 정보 부여</div> <div>속성</div> <div><h1 title = "header"> 제목 </h1></div> <div>속성이름 속성 값</div>
주석	<div><ul style="list-style-type: none">코드 작성자의 메모 추가 목적 작성</div> <div><!-- 메모 콘텐츠 --></div>

웹 크롤링

■ HTML5 구조

• 기본 용어

명칭	설명
요소	<ul style="list-style-type: none">제목, 본문, 이미지 등 HTML 페이지에 위치
태그	<ul style="list-style-type: none">요소를 생성할 때 사용하는 기호소문자로 작성태그의 콘텐츠로 태그 포함 가능
속성	<ul style="list-style-type: none">태그에 추가 정보 부여
주석	<ul style="list-style-type: none">코드 작성자의 메모 추가 목적 작성



예시
<pre><!DOCTYPE html></pre> ▶ 문서 형식 선언
<pre><html lang="ko"></pre> ▶ 태그, <태그명 속성명 = "속성 값"> 형태
<pre><head></pre> ▶ 태그
<pre><meta charset="utf-8"></pre> ▶ 요소(인코딩 문법)
<pre><title> 문서 제목 </title></pre> ▶ <태그> 요소 </태그>
<pre></head></pre> ▶ 태그
<pre><body></pre> ▶ 태그
웹페이지에 표시할 콘텐츠(또는 태그) 입력 요소
<pre><!-- 메모 콘텐츠 --></pre> ▶ 주석
<pre></body></pre> ▶ 태그
<pre></html></pre> ▶ 태그

웹 크롤링

▪ selenium

- 동적 웹 페이지를 크롤링하며 chromedriver를 제어하거나, 원하는 정보를 얻기 위해 사용하는 라이브

```
import selenium
from selenium import webdriver
from selenium.webdriver.common.keys import Keys #selenium으로 키 조작
from selenium.webdriver.common.by import By #특정 HTML 요소 불러오기

driver = webdriver.Chrome() #오류발생시 webdriver.Chrome(service = Service(ChromeDriverManager().install()))
driver.get("URL 주소")

search_box = driver.find_element( By.속성, 'HTML 속성 값') #여러 속성 값을 찾으려면 find_elements 사용
```

- selenium 4.6 이전 버전에서는 사전에 webdriver_manager 설치 요구됨
- selenium 3.x 버전에서는 Chrome driver를 별도 다운로드 후 진행 가능

[참고] print(selenium.__version__) 으로 selenium버전 확인 가능

웹 크롤링

■ selenium

• 웹 요소 선택 함수

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

driver.find_element(By.속성, 'HTML 속성 값')
```

함수	요소 및 태그를 찾는 기준	함수	요소 및 태그를 찾는 기준
By.CLASS_NAME	클래스 명	By.CSS_SELECTOR	CSS 선택자
By.ID	ID 속성	By.XPATH	XML 쿼리
By.NAME	name 속성	By.LINK_TEXT	텍스트 내용
By.TAG_NAME	태그 명	By.PARTIAL_LINK_TEXT	텍스트 내용 일부

❗ By. 이후 작성되는 속성은 대문자 작성 필수

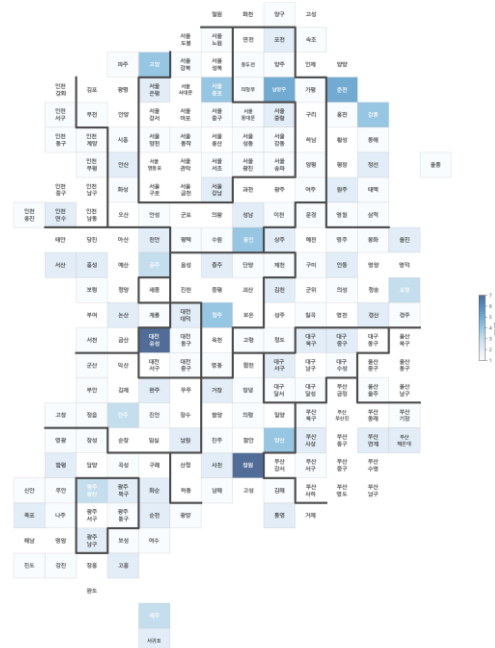
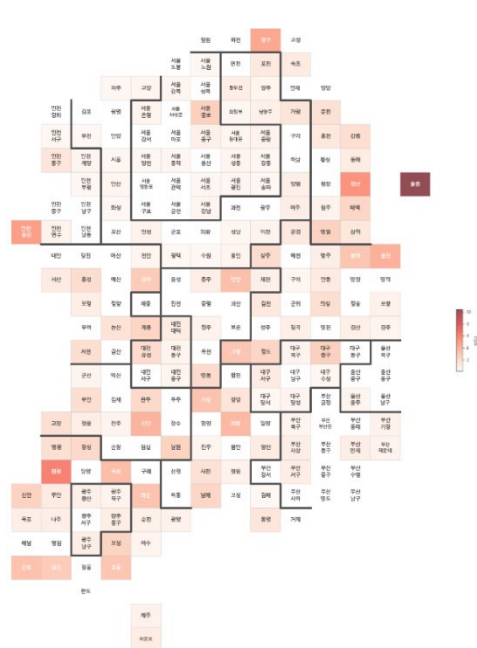
지리 정보 시각화

지리 정보 시각화

■ 블록 맵

- 행정구역의 경계선을 단순화하여 블록 형태로 지리 정보 표현
- 행정구역별 데이터 크기를 시각화할 때 유용한 표현

■ 사용자 정의 함수를 이용한 블록 맵 표현



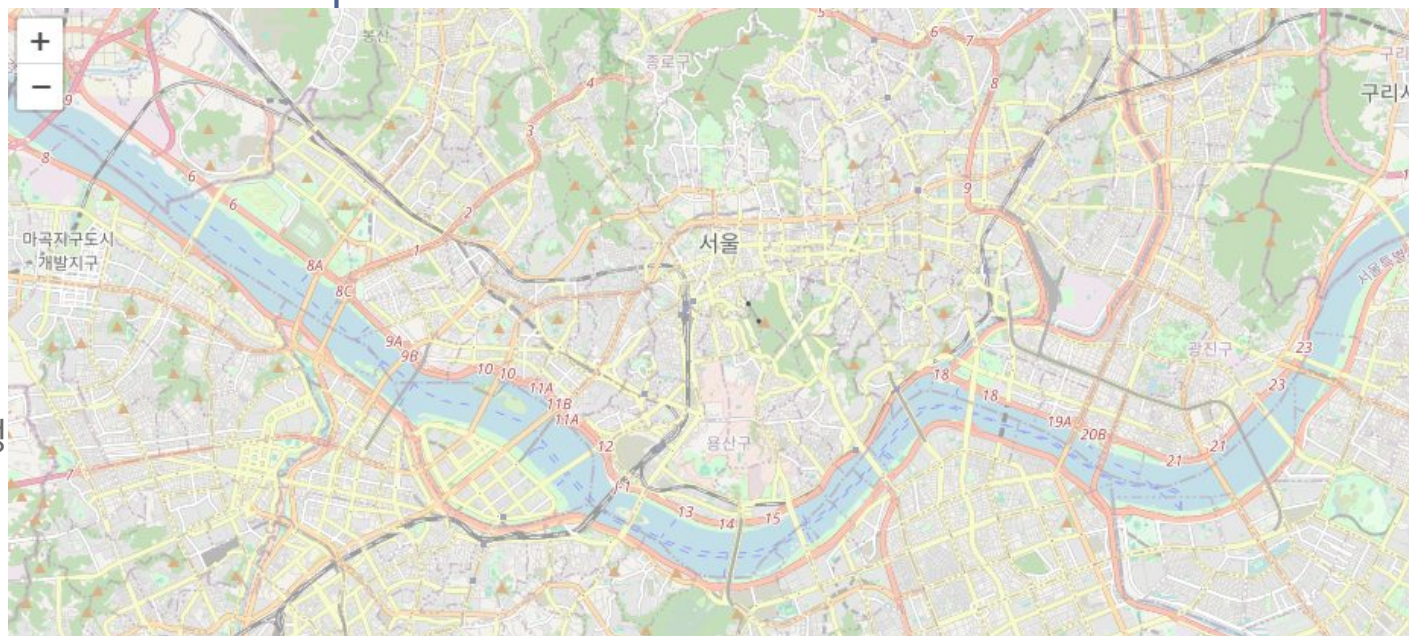
지리 정보 시각화

■ Map함수를 통한 지도객체 생성

```
import folium
```

```
folium.Map( location = [위도, 경도], zoom_start = 화면 확대 비율, tiles = "스타일")
```

- 위도와 경도의 GPS 좌표를 중심으로 지도 객체 생성
- 생성된 지도는 화면 내 확대 및 이동 가능
- tiles에 따른 스타일 변경 가능(지원 여부 확인 필요)



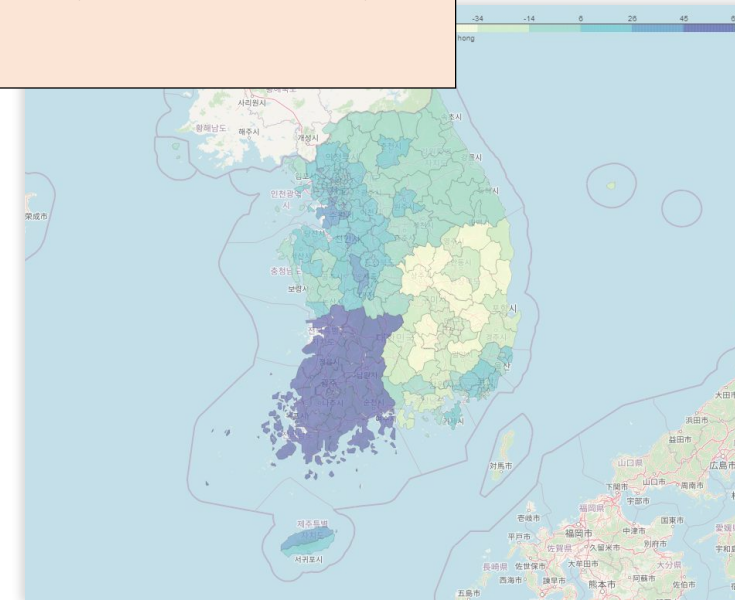
지리 정보 시각화

■ 지도 영역 내 단계구분도 표현

- 지도 상 어떠한 경계로 둘러싸인 영역에 대하여 색을 칠하거나, 음영으로 표현하는 방법

■ Folium의 Choropleth를 이용한 단계구분도 표현

```
folium.Choropleth( geo_data = 지도 경계, data = 표현하려는 데이터, columns = [표현하려는 데이터의 열],  
                  key_on = 지리 영역을 구분하는 열 이름, fill_color = "내부 색상", fill_opacity = 투명도, bins = 색의 구간,  
                  line_color = "경계선 색상", line_opacity = 경계선의 투명도 )
```



- 다양한 정보를 지역별 구분하여 색상 출력
- Polygon 단위의 데이터를 이용하여 지역별 전반적인 분포 확인

19대 대선 결과 분석 실습

▪ 19대 대선 결과 분석

- 운영 체제에 따라 시각화에 필요한 한글 폰트 지정

```
import platform
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc

path = "./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')

plt.rcParams['axes.unicode_minus'] = False
```


실습


■ 19대 대선 결과 분석

• 데이터 수집

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

driver = webdriver.Chrome()
driver.get("http://info.nec.go.kr/main/showDocument.xhtml?electionId=0000000000&topMenuId=VC&secondMenuId=VCCP09")
```

Chrome이 자동화된 테스트 소프트웨어에 의해 제어되고 있습니다.

 중앙선거관리위원회
선거통계시스템

최근선거 | **역대선거** | 통합검색 (예비)후보자 성명 입력

기본현황 | 선거인수 | 후보자 | 투·개표 | 당선인

사전투표율 변동추이 | 사전투표현황 | 투표율 변동추이 | 투표현황 | **개표현황** | 개표현황(음면동별) | 개표현황(투표구별)

개표현황

실습

■ 19대 대선 결과 분석

• 데이터 수집



http://info.nec.go.kr/main/showDocument.xhtml?electionId=0000000000&topMenuId=VC&secondMenuId=VCCP09



Chrome이 자동화된 테스트 소프트웨어에 의해 제어되고 있습니다.



중앙선거관리위원회
NATIONAL ELECTION COMMISSION

선거통계시스템

최근선거

역대선거

통합검색

(메뉴)후보자 성명 입력



기본현황

선거인수

후보자

투·개표

당선인

사전투표를 변동주이 | 사전투표현황 | 투표율 변동주이 | 투표현황 | **개표현황** | 개표현황(음변동별) | 개표현황(투표구별)

개표현황

» 역대선거정보 > 투·개표 > 개표현황

a#electionType1.on 77.39 x 29.48

선거유형	대통령선거 국회의원선거 지방선거 재·보궐선거
조회조건	



조회 자료가 없습니다.
자료를 **조회**하시려면 **조회구분** 유형 선택에서
조건들을 선택해 주세요.

통합검색

개인정보 처리방침

이용지침

FAQ

재보궐선거 실시사유 확정상황

관련사이트

이동



24시

[13809] 경기도 과천시 중문로 44 중앙선거관리위원회 대표전화 : 02-503-1114 / 선거 법률·신고제도 : 국번없이 1390
COPYRIGHT © National Election Commission, All Rights Reserved. 본 사이트는 IE9 이상, 크롬에 최적화 되어 있습니다.

Windows 정품 인증

[설정]으로 이동하여 Windows를 정품 인증합니다.



DevTools is now available in Korean!

Always match Chrome's language

Switch DevTools to Korean

Don't show again



Elements

Console

Sources

Network

Performance

Memory

Application

Security

Lighthouse

Recorder

>>

```
<script type="text/javascript" charset="utf-8" src="/ubi4/js/ubi.js?ver=240201"></script>
<script type="text/javascript"></script>
<script type="text/javascript"></script>
<div id="container">
  <div id="container_inner" class="clearfix">
    <div id="contentarea">
      <div class="titlearea"></div>
      <div class="bodyarea">
        <div class="filter">
          <form id="searchForm" action="/electioninfo/electionInfo_report.xhtml" method="post">
            <input type="hidden" id="electionId" name="electionId" value="0000000000">
            <input type="hidden" id="requestURI" name="requestURI" value="/electioninfo/0000000000/vc/vccp09.jsp">
            <input type="hidden" id="topMenuId" name="topMenuId" value="VC">
            <input type="hidden" id="secondMenuId" name="secondMenuId" value="VCCP09">
            <input type="hidden" id="menuId" name="menuId" value="VCCP09">
            <input type="hidden" id="statementId" name="statementId" value="">
            <input type="hidden" id="oldElectionType" name="oldElectionType" value="">
          </form>
          <div class="f_select">
            <div>
              <dt>선거유형</dt>
              <dd>
                <a href="javascript:setElectionType(1);" id="electionType1" class="on" title="선택됨">대통령선거</a>
                <a href="javascript:setElectionType(2);" id="electionType2">국회의원선거</a>
                <a href="javascript:setElectionType(4);" id="electionType4">지방선거</a>
                <a href="javascript:setElectionType(0);" id="electionType0">재·보궐선거</a>
              </dd>
            </div>
          </div>
          <div class="f_search"></div>
          <div class="result_initial"></div>
        </div>
      </div>
    </div>
  </div>
</div>
```

HTML 구조 확인 방법

F12

또는

ctrl

+


U

으로 확인

실습

■ 19대 대선 결과 분석

• 데이터 수집

 <http://info.nec.go.kr/main/showDocument.xhtml?electionId=0000000000&topMenuId=VC&secondMenuId=VCCP09> ☆

Chrome이 자동화된 테스트 소프트웨어에 의해 제어되고 있습니다.

중앙선거관리위원회 선거통계시스템

최근선거 역대선거 통합검색 (예비)후보자 성명 입력

기본현황 선거인수 후보자 투·개표 당선인

사전투표율 변동추이 | 사전투표현황 | 투표율 변동추이 | 투표현황 | **개표현황** | 개표현황(읍면동별) | 개표현황(투표구별)

개표현황

▲ > 역대선거정보 > 투·개표 > 개표현황

선거유형	<input type="text" value="select#electionName"/> 121.97 x 39.98 8월선거
조회조건	<input type="text" value="선택"/>

조회 결과가 없습니다.
자료를 조회하시려면 **조회구분 유형 선택**에서 조건들을 선택해 주세요.

통합검색 | 개인정보 처리방침 | 이용지침 | FAQ | 재보궐선거 실시사유 확정상황

관련사이트



DevTools is now available in Korean! [Always match Chrome's language](#) [Switch DevTools to Korean](#) [Don't show again](#)

Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder

```
<div id="contentarea">
  <div class="titlearea">
  <div class="bodyarea">
    <div class="filter">
      <form id="searchForm" action="/electioninfo/electionInfo_report.xhtml" method="post">
        <input type="hidden" id="electionId" name="electionId" value="0000000000">
        <input type="hidden" id="requestURL" name="requestURL" value="/electioninfo/0000000000/vc/vccp09.jsp">
        <input type="hidden" id="topMenuId" name="topMenuId" value="VC">
        <input type="hidden" id="secondMenuId" name="secondMenuId" value="VCCP09">
        <input type="hidden" id="menuId" name="menuId" value="VCCP09">
        <input type="hidden" id="statementId" name="statementId" value="">
        <input type="hidden" id="oldElectionType" name="oldElectionType" value="">
      <div class="f_search">
        <div class="f_search">
          <div>
            <dt>조회조건</dt>
            <dd>
              <span id="wub_select" style="display:none;">
                <div class="wub_select">
                  <select name="electionType" id="electionType" style="display:none;" title="선거유형">
                  <span id="spanElectionName" style="display:inline;">
                    <select name="electionName" id="electionName" title="선거회차">
                      <option value="-1">선택</option>
                      <option value="20220309">제20대</option>
                      <option value="20170509">제19대</option>
                      <option value="20121219">제18대</option>
                      <option value="20071219">제17대</option>
                      <option value="20021219">제16대</option>
                      <option value="19971218">제15대</option>
                      <option value="19921218">제14대</option>
                      <option value="19871216">제13대</option>
                      <option value="19810225">제12대</option>
                      <option value="19800827">제11대</option>
                      <option value="19791206">제10대</option>
                      <option value="19780706">제9대</option>
                      <option value="19721223">제8대</option>
                      <option value="19710427">제7대</option>
                      <option value="19670503">제6대</option>
                      <option value="19631015">제5대</option>
                      <option value="19600315">제4대</option>
                      <option value="19560515">제3대</option>
                      <option value="19520805">제2대</option>
                      <option value="19480720">제1대</option>
                    </select>
                  </span>
                </div>
              </dd>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

HTML 구조 확인 방법

F12 또는 ctrl + U 으로 확인

실습

■ 19대 대선 결과 분석

- 데이터 수집

<http://info.nec.go.kr/main/showDocument.xhtml?electionId=0000000000&topMenuId=VC&secondMenuId=VCCP09>



Chrome이 자동화된 테스트 소프트웨어에 의해 제어되고 있습니다.



선거통계시스템

최근선거

등합검색

(예비)후보자 성명 입력

Q

사전투표율 변동추이 | 사전투표현황 | 투표율 변동추이 | 투표현황 | **개표현황** | 개표현황(읍면동별) | 개표현황(투표구별)

개표현황

☞ > 역대선거정보 > 두·개표 > 개표현황

선거유형	<div> <div>대통령선거</div> <div>국회의원</div> <div> <div>select#electionCode</div> <div>121.97 x 39.98</div> </div> </div>
조희조건	<div> <div>제19대</div> <div>선거</div> <div>대통령선거</div> <div>시도</div> <div>선택</div> </div>



조회 자료가 없습니다.
자료를 **조회**하시려면 **조회구분 유형 선택**에서
조건들을 **선택**해 주세요.

[통합검색](#) | [개인정보 처리방침](#) | [이용지침](#) | [FAQ](#) | [재보궐선거 실시사유 확정상황](#)

관련사이트

 중앙선거관리위원회

[13809] 경기도 과천시 홍촌말로 44 중앙선거관리위원회 대표전화 : 02-503-1114 / 선거 방법의-신고제도 : 국번없이 1390
COPYRIGHT © National Election Commission, All Rights Reserved. 본 사이트는 IE9 이상, 크롬에 최적화 되어 있습니다.



 DevTools is now available in Korean!

Always match Chrome's language

Switch DevTools to Korean

Don't show again

```

Elements Sources Network Performance Memory Application Security Lighthouse Recorder >>
<script type="text/javascript" charset="utf-8" src="/ub14/js/ub1.js?ver=242011"></script>
<script type="text/javascript">...</script>
<script type="text/javascript">...</script>
<div id="container">
  <div id="container_inner" class="clearfix">
    <div id="contentarea">
      <div class="titlearea">...</div>
      <div class="bodyarea">
        <div class="filter">
          <form id="searchForm" action="/electioninfo/electionInfo_report.xhtml" method="post">
            <input type="hidden" id="electionId" name="electionId" value="0000000000">
            <input type="hidden" id="requestURI" name="requestURI" value="/electioninfo/0000000000/vc/vccp09.jsp">
            <input type="hidden" id="topMenuId" name="topMenuId" value="VC">
            <input type="hidden" id="secondMenuId" name="secondMenuId" value="VCCP09">
            <input type="hidden" id="menuId" name="menuId" value="VCCP09">
            <input type="hidden" id="statementId" name="statementId" value">
            <input type="hidden" id="oldElectionType" name="oldElectionType" value="1">
          <div class="f_select">...</div>
          <div class="f_search">
            <div>
              <div>조회조건</div>
              <div>
                <span id="wub_select" style">
                  <div class="wub_select1">
                    <select name="electionType" id="electionType" style="display:none;" title="선거유형">...</select>
                    <span id="spanElectionName" style="display: inline;">...</span>
                    <span id="spanElectionCode" style">
                      <label id="electionLabel" for="electionCode">선거</label>
                      <select id="electionCode" name="electionCode" style="display:none;" title="선거구분">...</select>
                      <option value="-1">▽ 선택</option>
                      <option value="1">대통령선거</option>
                    </select>
                  </div>
                <span id="spanCityCode" style">...
                <span id="spanTownCode" style="display:none;">...
                <span id="spanSggCityCode" style">...
                <span id="spanSubmit" style="display:none;">...
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

HTML 구조 확인 방법

F12

또는

ctrl

+

HTML 구조 확인 방법

F12

또는

ctrl

 $+$ | U

으로 확
이

실습

▪ 19대 대선 결과 분석

• 데이터 수집


```
driver.find_element(By.ID, 'electionType1').click()
```

```
driver.find_element(By.ID, 'electionName').send_keys('제19대')
```

```
election_code = WebDriverWait(driver, 60).until(EC.presence_of_element_located((By.ID, "electionCode")))
```

```
election_code.send_keys("대통령선거")
```

- 19대 대선 결과 분석
- 데이터 수집

 <http://info.nec.go.kr/main/showDocument.xhtml?electionId=0000000000&topMenuId=VC&secondMenuId=VCCP09> ☆

개표현황

🏠 > 역대선거정보 > 투·개표 > 개표현황

선거유형	<u>대통령선거</u> 국회의원선거 지방선거 재·보궐선거					
조회조건	제19대 ▾	선거	대통령선거 ▾	시도	서울특별시 ▾	🔍 검색

결과보기

[제19대] [대통령선거] [서울특별시]

PDF 다운

엑셀 다운

구시군명	선거인수	투표수	후보자별 득표수														무효 투표수
			후보자별 득표율 (%)														
			더불어민주당 문재인	자유한국당 홍준표	국민의당 안철수	바른정당 유승민	정의당 심상정	세누리당 조원진	경제애국당 오영국	국민대통합당 장성민	늘푸른한국당 이재오	민중연합당 김선동	한국국민당 이경희	홍익당 윤홍식	무소속 김민찬	계	
합계	8,382,999	6,590,646	2,781,345	1,365,285	1,492,767	476,973	425,459	9,987	789	3,554	1,938	3,416	1,277	2,177	3,950	6,568,917	21,729
			42.34	20.78	22.72	7.26	6.47	0.15	0.01	0.05	0.02	0.05	0.01	0.03	0.06		
종로구	133,769	102,566	42,512	22,325	22,313	7,412	7,113	228	5	78	31	63	26	47	49	102,202	364
			41.59	21.84	21.83	7.25	6.95	0.22	0.00	0.07	0.03	0.06	0.02	0.04	0.04		
중구	109,836	82,852	34,062	17,901	19,372	5,874	4,993	158	12	53	21	55	18	25	51	82,595	257
			41.23	21.67	23.45	7.11	6.04	0.19	0.01	0.06	0.02	0.06	0.02	0.03	0.06		

실습

■ 19대 대선 결과 분석

• 데이터 수집

```
sido_list_raw = driver.find_element(By.XPATH, """/*[@id="cityCode"]""")
sido_list = sido_list_raw.find_elements(By.TAG_NAME, 'option')
sido_names_values = [option.text for option in sido_list]
sido_names_values = sido_names_values[2:] # 시도 명이 2번째 index부터 시작
```



sido_names_values 결과 확인

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



실습

■ 19대 대선 결과 분석

• 데이터 수집

개표현황

홈 > 역대선거정보 > 두·개표 > 개표현황

선거유형	<u>대통령선거</u> 국회의원선거 지방선거 재·보궐선거
조회조건	제19대 ▼ 선거 대통령선거 ▼ 시도 서울특별시 ▼  검색 

크롤링 과정 중 '광역시도' 이름을 리스트로 전달하고, 검색 버튼을 누르는 함수 정의

```
def move_sido(name):  
    element = driver.find_element(By.ID, "cityCode")  
    element.send_keys(name)  
    make_xpath = ""//*[@id="searchBtn"]""  
    WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.XPATH, make_xpath))) #검색 버튼이 클릭 가능할 때까지 대기  
    driver.find_element(By.XPATH, make_xpath).click()
```


실습

19대 대선 결과 분석

데이터 수집

구시군명	선거인수	투표수	후보자별 득표수														무효 투표수	기권수
			후보자별 득표율 (%)															
			더불어민주당 문재인	자유한국당 홍준표	국민의당 안철수	바른정당 유승민	정의당 심상정	새누리당 조원진	경제애국당 오영국	국민대통합당 장성민	늘푸른한국당 이재오	민중연합당 김선동	한국국민당 이경희	홍익당 윤홍식	무소속 김민찬	계		
합계	8,382,999	6,590,646	2,781,345	1,365,285	1,492,767	476,973	425,459	9,987	789	3,554	1,938	3,416	1,277	2,177	3,950	6,568,917	21,729	1,792,353
			42.34	20.78	22.72	7.26	6.47	0.15	0.01	0.05	0.02	0.05	0.01	0.03	0.06			
종로구	133,769	102,566	42,512	22,325	22,313	7,412	7,113	228	5	78	31	63	26	47	49	102,202	364	31,203
			41.59	21.84	21.83	7.25	6.95	0.22	0.00	0.07	0.03	0.06	0.02	0.04	0.04			
중구	109,836	82,852	34,062	17,901	19,372	5,874	4,993	158	12	53	21	55	18	25	51	82,595	257	26,984
			41.23	21.67	23.45	7.11	6.04	0.19	0.01	0.06	0.02	0.06	0.02	0.03	0.06			

천 단위의 쉼표(,) 제거하는 함수 정의

```
def get_num(tmp):  
    return float(tmp.replace(',',''))
```

- 19대 대선 결과 분석
- 데이터 수집

개표현황

🏠 > 역대선거정보 > 투·개표 > 개표현황

선거유형

대통령선거 | 국회의원선거 | 지방선거 | 재·보궐선거

조회조건

제19대 | 선거 | 대통령선거 | 시도 | 서울특별시 | 🔍 검색

결과보기

[제19대] [대통령선거] [서울특별시]

PDF 다운

엑셀 다운

구시군명	선거인수	투표수	후보자별 득표수														무효 투표수	기권수
			후보자별 득표율 (%)															
			더불어민주당 문재인	자유한국당 홍준표	국민의당 안철수	바른정당 유승민	정의당 심상정	새누리당 조원진	경제애국당 오영국	국민대통합당 장성민	늘푸른한국당 이재오	민중연합당 김선동	한국국민당 이경희	홍익당 윤홍식	무소속 김민찬	계		
합계	8,382,999	6,590,646	2,781,345	1,365,285	1,492,767	476,973	425,459	9,987	789	3,554	1,938	3,416	1,277	2,177	3,950	6,568,917	21,729	1,792,353
			42.34	20.78	22.72	7.26	6.47	0.15	0.01	0.05	0.02	0.05	0.01	0.03	0.06			
종로구	133,769	102,566	42,512	22,325	22,313	7,412	7,113	228	5	78	31	63	26	47	49	102,202	364	31,203
			41.59	21.84	21.83	7.25	6.95	0.22	0.00	0.07	0.03	0.06	0.02	0.04	0.04			
중구	109,836	82,852	34,062	17,901	19,372	5,874	4,993	158	12	53	21	55	18	25	51	82,595	257	26,984
			41.23	21.67	23.45	7.11	6.04	0.19	0.01	0.06	0.02	0.06	0.02	0.03	0.06			

: 수집할 정보

실습

■ 19대 대선 결과 분석

• 데이터 수집

```
# 데이터 수집하는 함수 정의
def append_data(rows, sido_name, data):
    for i, row in enumerate(rows[5:], start = 2): #5번째 행부터 진행하되, 행 번호를 2로 간주
        if i % 2 == 0: # 짝수 행(투표 수)만 선택
            cells = row.find_elements(By.TAG_NAME, 'td')

            data['광역시도'].append(sido_name)
            data['시군'].append(cells[0].text.strip())
            data['pop'].append(get_num(cells[2].text.strip())) # (전체) 투표 수
            data['moon'].append(get_num(cells[3].text.strip())) # 더불어민주당 문재인 투표 수
            data['hong'].append(get_num(cells[4].text.strip())) # 자유한국당 홍준표 투표 수
            data['ahn'].append(get_num(cells[5].text.strip())) # 국민의당 안철수 투표 수
```

실습

▪ 19대 대선 결과 분석

• 데이터 수집

```
# 정의한 함수를 이용하여 데이터 수집
election_result_raw = {'광역시도': [], '시군': [], 'pop': [], 'moon': [], 'hong': [], 'ahn': []}

for each_sido in sido_names_values:
    move_sido(each_sido)
    WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.TAG_NAME, "table")))
    table = driver.find_element(By.TAG_NAME, "table")
    rows = table.find_elements(By.TAG_NAME, "tr")
    append_data(rows, each_sido, election_result_raw)

election_result = pd.DataFrame(election_result_raw, columns = ['광역시도', '시군', 'pop', 'moon', 'hong', 'ahn'])

driver.close() #수집 완료 후 활성화된 웹 크롤링 브라우저 종료

election_result.to_csv("./크롤링_수집.csv", index = False, encoding = 'utf-8-sig')
```

실습

■ 19대 대선 결과 분석

• 데이터 정제

```
# '광역시도'의 요약 명칭(앞 두 글자 추출)
sido_candi = election_result['광역시도']
sido_candi = [name[:2] if name[:2] in ['서울', '부산', '대구', '광주', '인천', '대전', '울산'] else " " for name in sido_candi]

# '시군'의 명칭이 2글자 이상이며, 식별 가능한 지명인 경우 요약 명칭(앞 두글자 추출)
# ex. '중구', '남구' → '중구', '남구', '중랑구', '서초구' → '중랑', '서초'
def cut_char_sigu(name):
    return name if len(name) == 2 else name[:1]

# 일부 '시군' 변경: '수원', '성남', '안양', '안산', '고양', '용인', '청주', '천안', '전주', '포항', '창원' → '안양 만안', '안양 동안'
import re
sigun_candi = [""] * len(election_result)

for n in election_result.index:
    each = election_result['시군'][n]
    if each[:2] in ['수원', '성남', '안양', '안산', '고양', '용인', '청주', '천안', '전주', '포항', '창원']:
        sigun_candi[n] = re.split('시', each)[0] + ' ' + cut_char_sigu(re.split('시', each)[1])
    else:
        sigun_candi[n] = cut_char_sigu(each)
```

실습

■ 19대 대선 결과 분석

• 데이터 정제

```
# '광역시도' + '시군'
# '시군'변수의 첫 글자가 공백인 경우, 그 다음 글자부터 읽어오기, (예외) '세종시' → 별도 정제

ID_candi = [sido_candi[n] + ' ' + sigun_candi[n] for n in range(0, len(sigun_candi))]
ID_candi = [name[1:] if name[0] == ' ' else name for name in ID_candi]
ID_candi = [name[:2] if name[:2] == '세종' else name for name in ID_candi]

election_result['ID'] = ID_candi
```

	광역시도	시군	pop	moon	hong	ahn
0	서울특별시	종로구	102566.0	42512.0	22325.0	22313.0
1	서울특별시	중구	82852.0	34062.0	17901.0	19372.0
2	서울특별시	용산구	148157.0	58081.0	35230.0	32109.0



	광역시도	시군	pop	moon	hong	ahn	ID
0	서울특별시	종로구	102566.0	42512.0	22325.0	22313.0	서울 종로
1	서울특별시	중구	82852.0	34062.0	17901.0	19372.0	서울 중구
2	서울특별시	용산구	148157.0	58081.0	35230.0	32109.0	서울 용산

19대 대선 결과 분석

득표율 계산

```
# 득표율 계산
election_result[['rate_moon', 'rate_hong', 'rate_ahn']] = election_result[['moon', 'hong', 'ahn']].div(election_result['pop'], axis = 0)
election_result[['rate_moon', 'rate_hong', 'rate_ahn']] *= 100
```

	광역시도	시군	pop	moon	hong	ahn	ID
0	서울특별시	종로구	102566.0	42512.0	22325.0	22313.0	서울 종로
1	서울특별시	중구	82852.0	34062.0	17901.0	19372.0	서울 중구
2	서울특별시	용산구	148157.0	58081.0	35230.0	32109.0	서울 용산



	광역시도	시군	pop	moon	hong	ahn	ID	rate_moon	rate_hong	rate_ahn
0	서울특별시	종로구	102566.0	42512.0	22325.0	22313.0	서울 종로	41.448433	21.766472	21.754773
1	서울특별시	중구	82852.0	34062.0	17901.0	19372.0	서울 중구	41.111862	21.605996	23.381451
2	서울특별시	용산구	148157.0	58081.0	35230.0	32109.0	서울 용산	39.202333	23.778829	21.672280

■ 19대 대선 결과 분석

• 후보자별 득표율이 높은 지역 파악

```
# 문재인 후보가 높은 비율로 득표한 지역
election_result.sort_values(['rate_moon'], ascending = [False]).head(10)
```

	광역시도	시군	pop	moon	hong	ahn	ID	rate_moon	rate_hong	rate_ahn
1	전라남도	순천시	181451	122595	4525	40429	순천	67.5637	2.493786	22.28095
2	전라북도	전주시덕진구	187921	125375	5183	40188	전주 덕진	66.71687	2.758074	21.38558
3	전라북도	전주시완산구	236383	157637	7003	50506	전주 완산	66.68711	2.962565	21.36617
4	전라북도	장수군	16079	10714	717	3353	장수	66.6335	4.459233	20.85329
5	전라남도	광양시	96384	63544	4100	20080	광양	65.92796	4.253818	20.83333
6	전라북도	진안군	18107	11918	819	3904	진안	65.81985	4.523113	21.56072
7	전라북도	완주군	62470	41057	2107	13897	완주	65.72275	3.372819	22.24588
8	전라북도	익산시	192208	123422	6470	45737	익산	64.21273	3.366145	23.79558
9	전라북도	남원시	55371	35539	1939	13854	남원	64.18342	3.501833	25.02032
10	광주광역시	광산구	248209	159119	3630	65402	광주 광산	64.10686	1.462477	26.34957

■ 19대 대선 결과 분석

- 후보자별 득표율이 높은 지역 파악

```
# 홍준표 후보가 높은 비율로 득표한 지역
election_result.sort_values(['rate_hong'], ascending = [False]).head(10)
```

	광역시도	시군	pop	moon	hong	ahn	ID	rate_moon	rate_hong	rate_ahn
1	경상북도	군위군	17627	2251	11651	1939	군위	12.77018	66.09746	11.00017
2	경상북도	의성군	37855	5365	23790	4767	의성	14.1725	62.84507	12.59279
3	경상북도	영덕군	26125	3786	16314	3231	영덕	14.49187	62.44593	12.36746
4	경상남도	합천군	33021	7143	19699	3077	합천	21.63169	59.65598	9.318313
5	경상북도	고령군	22396	3754	13248	2600	고령	16.76192	59.15342	11.60922
6	경상북도	예천군	32124	5261	18863	4427	예천	16.37716	58.71934	13.78097
7	경상북도	청도군	30398	5323	17678	3654	청도	17.51102	58.15514	12.02053
8	경상북도	청송군	18418	3218	10669	2387	청송	17.47204	57.92703	12.96015
9	경상남도	창녕군	42878	10310	24464	3877	창녕	24.04497	57.0549	9.041933
10	경상북도	문경시	49113	8616	27832	6905	문경	17.54322	56.66931	14.05941

실습

■ 19대 대선 결과 분석

• 후보자별 득표율이 높은 지역 파악

```
# 안철수 후보가 높은 비율로 득표한 지역  
election_result.sort_values(['rate_ahn'], ascending = [False]).head(10)
```

	광역시도	시군	pop	moon	hong	ahn	ID	rate_moon	rate_hong	rate_ahn
1	전라남도	진도군	21189	10392	511	8855	진도	49.04432	2.411629	41.79055
2	전라남도	신안군	28950	14370	713	12000	신안	49.63731	2.462867	41.45078
3	전라남도	강진군	25175	12476	753	10152	강진	49.5571	2.991063	40.32572
4	전라남도	해남군	48351	25901	1158	18157	해남	53.5687	2.394987	37.55248
5	전라남도	영암군	36402	18999	825	13610	영암	52.19219	2.266359	37.38806
6	전라남도	목포시	145476	77896	2584	53303	목포	53.5456	1.776238	36.64041
7	광주광역시	동구	66287	37053	1308	23438	광주 동구	55.89784	1.973238	35.35837
8	전라남도	장흥군	27149	14821	636	9593	장흥	54.59133	2.342628	35.33464
9	전라남도	보성군	29967	16666	732	10514	보성	55.61451	2.442687	35.08526
10	전라남도	무안군	52516	29516	983	18052	무안	56.20382	1.87181	34.37429

실습

■ 19대 대선 결과 분석

• 시각화를 위한 정제

행정 구역이 단순화된 지역 정보 데이터 활용

```
draw_korea = pd.read_csv("./draw_korea.csv", encoding = 'utf-8', index_col = 0)
```

draw_korea의 ID와 election_result의 ID가 서로 일치하는지 확인

```
set(draw_korea['ID'].unique()) - set(election_result['ID'].unique())
```

결과: {'고성(강원)', '고성(경남)', '부천 소사', '부천 오정', '부천 원미', '창원 합포', '창원 회원'}

```
set(election_result['ID'].unique()) - set(draw_korea['ID'].unique())
```

결과: {'고성', '부천', '창원 마산합포', '창원 마산회원'}

실습

■ 19대 대선 결과 분석

• 시각화를 위한 정제

```
# 강원도 고성, 경상남도 고성 구분
election_result[election_result['ID'] == '고성']

election_result.loc[125, 'ID'] = '고성(강원)'
election_result.loc[233, 'ID'] = '고성(경남)'

election_result[election_result['시군'] == '고성군'] # 확인
```

광역시도	시군	pop	moon	hong	ahn	ID	rate _moon	rate _hong	rate _ahn
강원도	고성군	18692	5664	6511	3964	고성	30.302	34.833	21.207
경상남 도	고성군	34603	9848	16797	4104	고성	28.460	48.542	11.860



광역시도	시군	pop	moon	hong	ahn	ID	rate _moon	rate _hong	rate _ahn
강원도	고성군	18692	5664	6511	3964	고성(강원)	30.302	34.833	21.207
경상남 도	고성군	34603	9848	16797	4104	고성(경남)	28.460	48.542	11.860

실습

■ 19대 대선 결과 분석

• 시각화를 위한 정제

```
# 경남 창원시 지명 구분
```

```
election_result[election_result['광역시도'] == '경상남도']
```

```
election_result.loc[228, 'ID'] = '창원 합포'
```

```
election_result.loc[229, 'ID'] = '창원 회원'
```

```
election_result[election_result['광역시도'] == '경상남도'] # 확인
```

광역시도	시군	pop	moon	hong	ahn	ID	rate_moon	rate_hong	rate_ahn
경상남도	창원시 의창구	164047	60757	56887	22830	창원 의창	37.036	34.677	13.917
경상남도	창원시 성산구	153327	63717	42052	22923	창원 성산	41.556	27.426	14.950
경상남도	창원시 마산합포구	119281	35592	54488	14686	창원 마산합포	29.839	45.680	12.312
경상남도	창원시 마산회원구	136757	45014	56340	17744	창원 마산회원	32.915	41.197	12.975
경상남도	창원시 마산합포구	119281	35592	54488	14686	창원 마산합포	29.839	45.680	12.312



광역시도	시군	pop	moon	hong	ahn	ID	rate_moon	rate_hong	rate_ahn
경상남도	창원시 의창구	164047	60757	56887	22830	창원 의창	37.036	34.677	13.917
경상남도	창원시 성산구	153327	63717	42052	22923	창원 성산	41.556	27.426	14.950
경상남도	창원시 마산합포구	119281	35592	54488	14686	창원 합포	29.839	45.680	12.312
경상남도	창원시 마산회원구	136757	45014	56340	17744	창원 회원	32.915	41.197	12.975
경상남도	창원시 마산합포구	119281	35592	54488	14686	창원 합포	29.839	45.680	12.312

실습

■ 19대 대선 결과 분석

• 시각화를 위한 정제

```
# 경기도 부천시 지명 구분 - 2016년 6월부터 부천시는 소사, 오정, 원미구를 폐지하고 책임동제를 실시하는 지역으로 변경됨
election_result[election_result['시군'] == '부천시']

# 원칙적으로는 각 동의 인구를 다시 조사하고 이를 옛 지역구에 맞춰야 하지만, 단순히 '3'으로 나눠서 진행
ahn_tmp = election_result.loc[85, 'ahn'] / 3
hong_tmp = election_result.loc[85, 'hong'] / 3
moon_tmp = election_result.loc[85, 'moon'] / 3
pop_tmp = election_result.loc[85, 'pop'] / 3

rate_moon_tmp = election_result.loc[85, 'rate_moon']
rate_hong_tmp = election_result.loc[85, 'rate_hong']
rate_ahn_tmp = election_result.loc[85, 'rate_ahn']

election_result.loc[250] = ['경기도', '부천시', pop_tmp, moon_tmp, hong_tmp, ahn_tmp, '부천 소사', rate_moon_tmp, rate_hong_tmp, rate_ahn_tmp]
election_result.loc[251] = ['경기도', '부천시', pop_tmp, moon_tmp, hong_tmp, ahn_tmp, '부천 오정', rate_moon_tmp, rate_hong_tmp, rate_ahn_tmp]
election_result.loc[252] = ['경기도', '부천시', pop_tmp, moon_tmp, hong_tmp, ahn_tmp, '부천 원미', rate_moon_tmp, rate_hong_tmp, rate_ahn_tmp]

election_result[election_result['시군'] == '부천시'] # 확인 1
election_result.drop([85], inplace = True) # 남은 부천시 제거

election_result[election_result['시군'] == '부천시'] # 확인 2
```

실습

■ 19대 대선 결과 분석

• 시각화를 위한 정제

```
# draw_korea의 ID와 election_result의 ID가 서로 일치하는지 재확인
set(draw_korea['ID'].unique()) - set(election_result['ID'].unique()) # 결과: set()
set(election_result['ID'].unique()) - set(draw_korea['ID'].unique()) # 결과: set()

# 최종 데이터 구성
final_elect_data = pd.merge(election_result, draw_korea, how = 'left', on = ['ID'] )
```

	광역시도	시군	pop	moon	hong	ahn	ID	rate _moon	rate _hong	rate _ahn	y	x
0	서울특별시	종로구	102566.0	42512.0	22325.0	22313.0	서울 종로	41.448433	21.766472	21.754773	4	6
1	서울특별시	중구	82852.0	34062.0	17901.0	19372.0	서울 중구	41.111862	21.605996	23.381451	5	6
2	서울특별시	용산구	148157.0	58081.0	35230.0	32109.0	서울 용산	39.202333	23.778829	21.672280	6	6

실습

■ 19대 대선 결과 분석

• 시각화를 위한 정제

각 후보 간 득표율 차이 계산

`final_elect_data['moon_vs_hong'] = final_elect_data['rate_moon'] - final_elect_data['rate_hong']` # 문재인 후보 - 홍준표 후보

`final_elect_data['moon_vs_ahn'] = final_elect_data['rate_moon'] - final_elect_data['rate_ahn']` # 문재인 후보 - 안철수 후보

`final_elect_data['ahn_vs_hong'] = final_elect_data['rate_ahn'] - final_elect_data['rate_hong']` # 안철수 후보 - 홍준표 후보

	광역시도	시군	pop	moon	hong	ahn	ID	rate_moon	rate_hong	rate_ahn	y	x	moon_vs_hong	moon_vs_ahn	ahn_vs_hong
0	서울특별시	종로구	102566.0	42512.0	22325.0	22313.0	서울 종로	41.448433	21.766472	21.754773	4	6	19.682	19.694	-0.012
1	서울특별시	중구	82852.0	34062.0	17901.0	19372.0	서울 중구	41.111862	21.605996	23.381451	5	6	19.506	17.730	1.775
2	서울특별시	용산구	148157.0	58081.0	35230.0	32109.0	서울 용산	39.202333	23.778829	21.672280	6	6	15.424	17.530	-2.107

실습

■ 19대 대선 결과 분석

• 시각화

지도의 경계선을 나타내는 변수

BORDER_LINES = [[(5, 1), (5, 2), (7, 2), (7, 3), (11, 3), (11, 0)], # 인천

[(5, 4), (5, 5), (2, 5), (2, 7), (4, 7), (4, 9), (7, 9), (7, 7), (9, 7), (9, 5), (10, 5), (10, 4), (5, 4)], # 서울

[(1, 7), (1, 8), (3, 8), (3, 10), (10, 10), (10, 7), (12, 7), (12, 6), (11, 6), (11, 5), (12, 5), (12, 4), (11, 4), (11, 3)], # 경기도

[(8, 10), (8, 11), (6, 11), (6, 12)], # 강원도

[(12, 5), (13, 5), (13, 4), (14, 4), (14, 5), (15, 5), (15, 4), (16, 4), (16, 2)], # 충청북도

[(16, 4), (17, 4), (17, 5), (16, 5), (16, 6), (19, 6), (19, 5), (20, 5), (20, 4), (21, 4), (21, 3), (19, 3), (19, 1)], # 전라북도

[(13, 5), (13, 6), (16, 6)], # 대전

[(13, 5), (14, 5)], # 세종

[(21, 2), (21, 3), (22, 3), (22, 4), (24, 4), (24, 2), (21, 2)], # 광주

[(20, 5), (21, 5), (21, 6), (23, 6)], # 전라남도

[(10, 8), (12, 8), (12, 9), (14, 9), (14, 8), (16, 8), (16, 6)], # 충청북도

[(14, 9), (14, 11), (14, 12), (13, 12), (13, 13)], # 경상북도

[(15, 8), (17, 8), (17, 10), (16, 10), (16, 11), (14, 11)], # 대구

[(17, 9), (18, 9), (18, 8), (19, 8), (19, 9), (20, 9), (20, 10), (21, 10)], # 부산

[(16, 11), (16, 13)], # 울산

[(27, 5), (27, 6), (25, 6)]]

▪ 19대 대선 결과 분석

• 시각화

```
# 시각화 사용자 정의 함수: 본 슬라이드의 35페이지까지 하나의 함수로 지정
def drawKorea(targetData, blockedMap, cmapname):
    whitelabelmin = 20.
    datalabel = targetData

    tmp_max = max([np.abs(min(blockedMap[targetData])), np.abs(max(blockedMap[targetData]))])
    vmin, vmax = -tmp_max, tmp_max

    mapdata = blockedMap.pivot_table(index = 'y', columns = 'x', values = targetData)
    masked_mapdata = np.ma.masked_where(np.isnan(mapdata), mapdata)

    plt.figure(figsize = (9, 11))
    plt.pcolor(masked_mapdata, vmin = vmin, vmax = vmax, cmap = cmapname, edgecolor = '#aaaaaa', linewidth = 0.5)
```

실습

■ 19대 대선 결과 분석

• 시각화

```
# 지역명칭 표시 → (광역시: 시단위 명칭 함께 표시)
for idx, row in blockedMap.iterrows():
    if len(row['ID'].split()) == 2:
        dispname = '{}\n{}'.format(row['ID'].split()[0], row['ID'].split()[1])
    elif row['ID'][:2] == '고성':
        dispname = '고성'
    else:
        dispname = row['ID']

# 지역명칭 3글자 이상이면 글자 크기 조정
if len(dispname.splitlines())[-1] >= 3:
    fontsize, linespacing = 10.0, 1.1
else:
    fontsize, linespacing = 11, 1.

annocolor = 'white' if np.abs(row[targetData]) > whitelabelmin else 'black'
plt.annotate(dispname, (row['x'] + 0.5, row['y'] + 0.5), weight = 'bold', fontsize = fontsize, ha = 'center', va = 'center', color = annocolor, linespacing = linespacing)
```

실습

▪ 19대 대선 결과 분석

• 시각화

```
# 시도 경계 표시
for path in BORDER_LINES:
    ys, xs = zip(*path)
    plt.plot(xs, ys, c = 'black', lw = 2)

plt.gca().invert_yaxis()

plt.axis('off')

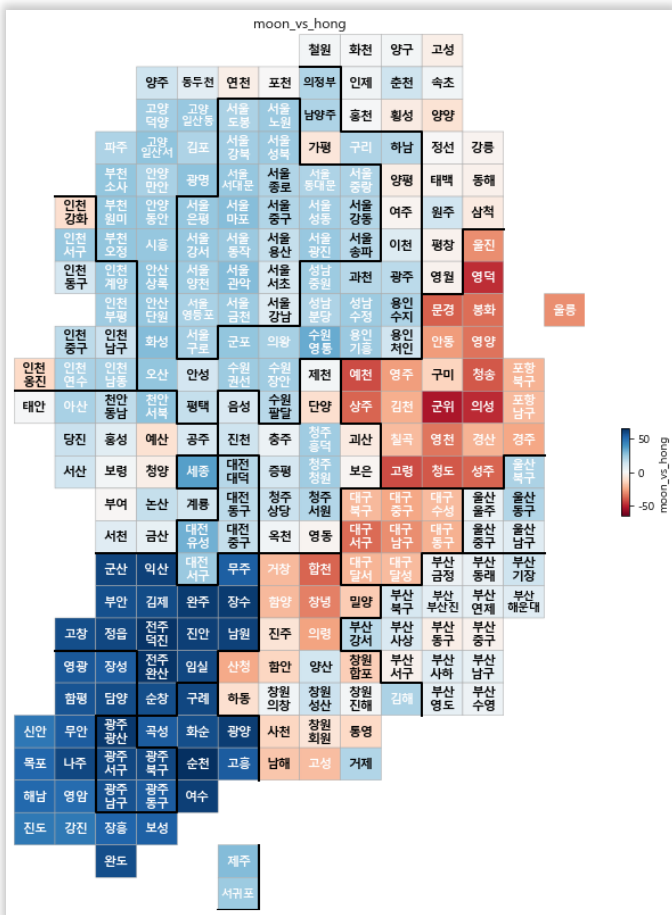
cb = plt.colorbar(shrink = 0.1, aspect = 10)
cb.set_label(datalabel)

plt.tight_layout()
plt.title(targetData)
plt.show()
```

실습

■ 19대 대선 결과 분석

- 시각화



"문재인 후보 vs 홍준표 후보" 득표율 격차 시각화

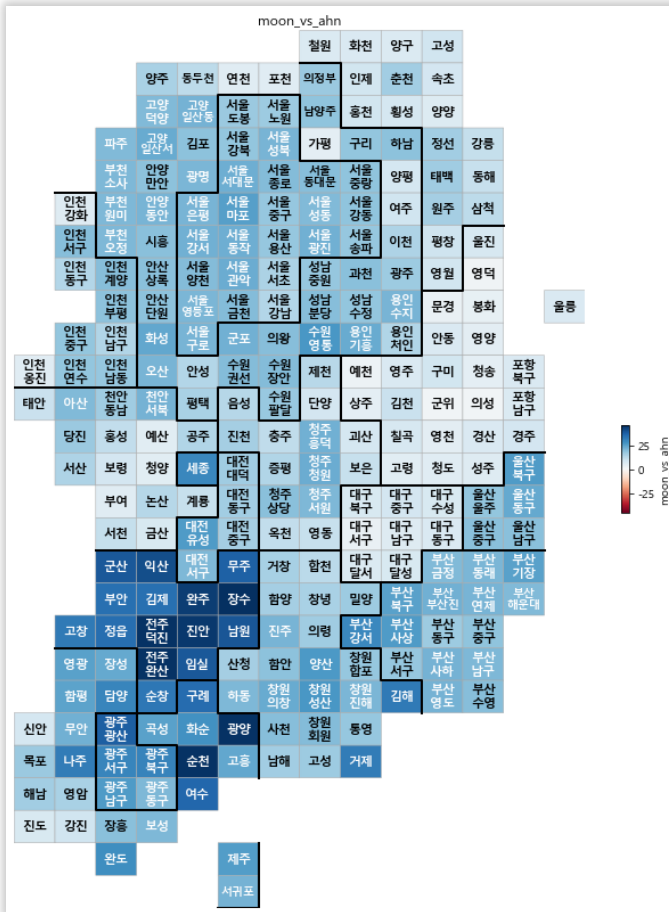
```
drawKorea('moon_vs_hong', final_elect_data, 'RdBu')
```

- 행정구역 별 19대 대선에서 문재인 후보와 홍준표 후보 간 득표율 격차를 기준으로 색상 지정
- **문재인 후보**의 경우 전라도, 서울·경기에서 홍준표 후보에 비해 높은 득표를 보임
- **홍준표 후보**의 경우 경남 창원, 김해, 부산광역시에서 문재인 후보에 비해 높은 득표를 보임

실습

19대 대선 결과 분석

시각화



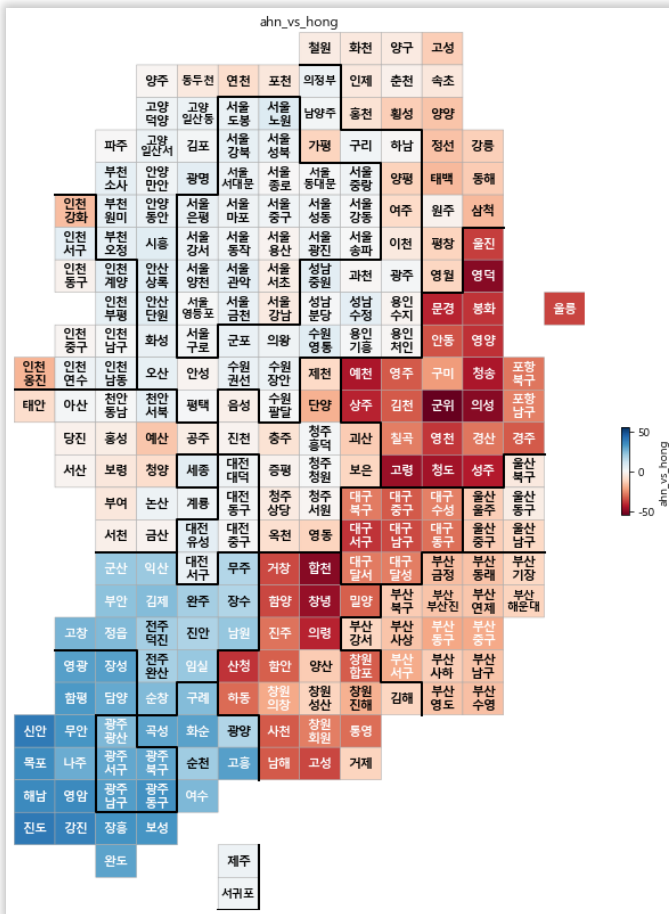
"문재인 후보 vs 안철수 후보" 득표율 격차 시각화
`drawKorea('moon_vs_ahn', final_elect_data, 'RdBu')`

- 행정구역 별 19대 대선에서 문재인 후보와 안철수 후보 간 득표율 격차를 기준으로 색상 지정
- 문재인 후보의 경우 전반적으로 안철수 후보에 비해 높은 득표를 보임
- 안철수 후보의 경우 전반적으로 문재인 후보에 비해 낮은 득표를 보임

실습

■ 19대 대선 결과 분석

• 시각화



"안철수 후보 vs 홍준표 후보" 득표율 격차 시각화
drawKorea('ahn_vs_hong', final_elect_data, 'RdBu')

- 행정구역 별 19대 대선에서 안철수 후보와 홍준표 후보 간 득표율 격차를 기준으로 색상 지정
- 안철수 후보의 경우 전라도에서 홍준표 후보에 비해 높은 득표를 보임
- 홍준표 후보의 경우 대구광역시 및 경북에서 안철수 후보에 비해 높은 득표를 보임

실습

▪ 19대 대선 결과 분석

• 시각화

```
# folium을 이용해서 "문재인 후보 vs 홍준표 후보" 득표율 격차 시각화
import folium
import json

pop_folium = final_elect_data.set_index('ID')

geo_path = "./skorea_municipalities_geo_simple.json"
geo_str = json.load(open(geo_path, encoding = 'utf-8'))

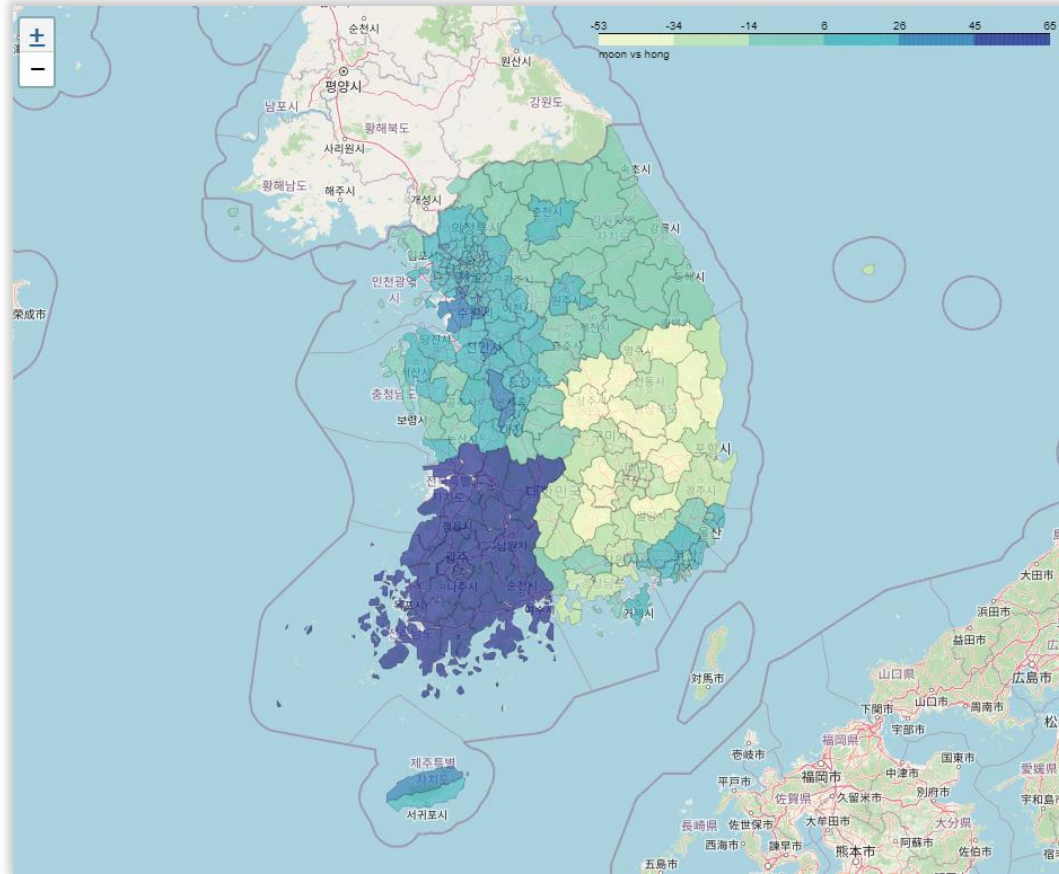
moon_hong_map = folium.Map(location = [36.2002, 127.054], zoom_start = 6)
folium.Choropleth( geo_data = geo_str, data = pop_folium, columns = [pop_folium.index, 'moon_vs_hong'], key_on = 'feature.id',
                   fill_color = 'PuBu', fill_opacity = 0.7, line_opacity = 0.2, legend_name = 'moon vs hong').add_to(moon_hong_map)

moon_hong_map.save('./moon_vs_hong.html')
```


실습

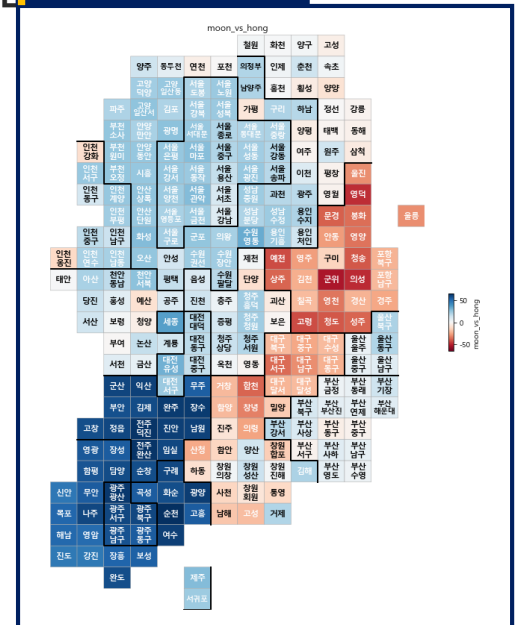
19대 대선 결과 분석

시각화



• 블록맵과 동일한 결과를 folium을 통해 시각화할 수 있음

• 문재인 후보의 경우 전라도, 서울경기, 홍준표 후보의 경우 경남 창원, 김해, 부산
상대 후보에 비해 높은 득표를 보임



정리하기

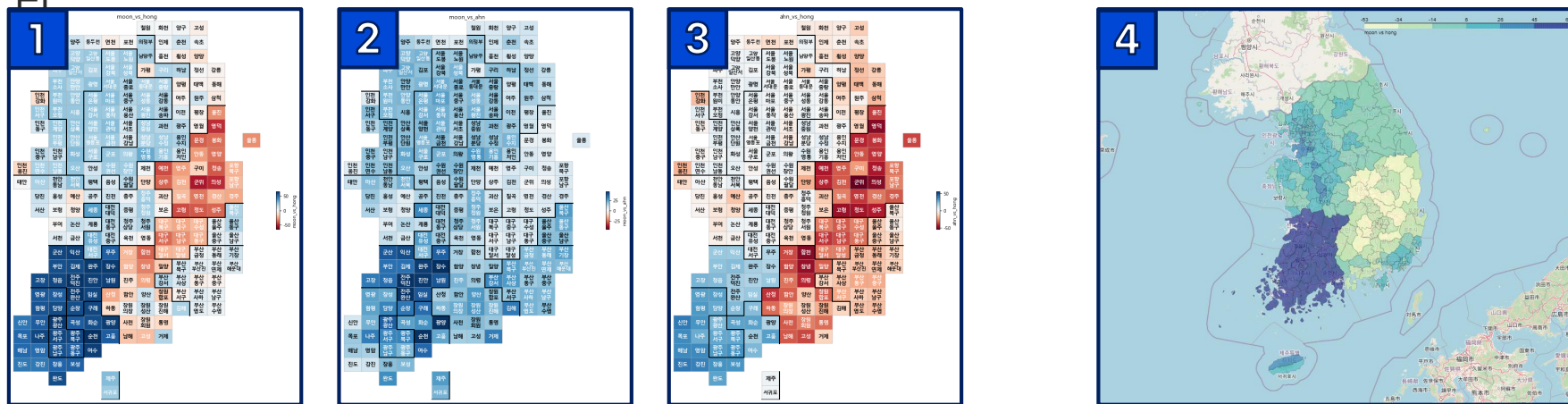
학습 내용

1. 파이썬에서 selenium을 이용하여 인터넷 자료를 수집하였다.



광역시도	시군	pop	moon	hong	ahn	ID	rate_moon	rate_hong	rate_ahn
0	서울특별시 종로구	102566.0	42512.0	22325.0	22313.0	서울 종로	41.448	21.766	21.755
1	서울특별시 중구	82852.0	34062.0	17901.0	19372.0	서울 중구	41.112	21.606	23.381
2	서울특별시 용산구	148157.0	58081.0	35230.0	32109.0	서울 용산	39.202	23.779	21.672

2. 사용자 정의 함수(블록맵)과 folium 패키지를 이용하여 행정구역 별 수집된 대선 결과 자료를 시각화하였다.



크롤링 및 지도 시각화

학습내용 및 목표

학습 내용

1. 크롤링을 통한 인터넷 자료 수집
2. 수집한 자료를 활용한 지도 시각화

학습 목표

1. 파이썬에서 크롤링하여 서울시의 주유 가격 자료를 수집할 수 있다.
2. 서울시에서 셀프 주유 여부에 따른 가격 차이를 비교할 수 있다.
3. 서울시 구별 주유 가격을 나타내는 지도를 그릴 수 있다.

실습

■ 파이썬의 크롤링 모듈 비교

- BeautifulSoup과 selenium 모듈 비교

	BeautifulSoup	selenium
정의	html 문서를 다루기 쉬운 형태로 만들어 필요한 자료를 추출하는 라이브러리	chromedriver를 제어하거나, 원하는 정보를 얻기 위해 사용하는 모듈
기본 문 법	<pre>from bs4 import BeautifulSoup import urllib.request as ur soup = BeautifulSoup(ur.urlopen("URL 주소").read(), 'html.parser')</pre>	<pre>import selenium from selenium import webdriver driver = webdriver.Chrome() driver.get("URL 주소")</pre>
필요 정보 추 출	<pre>target = soup.find('태그명', attrs, recursive, string) 또는 soup.find_all('태그명', attrs, recursive, string) 또는 soup.select('태그명') 또는 soup.select_one('태그명') target.string 또는 target.text 또는 target.get_text()</pre>	<pre>from selenium.webdriver.common.keys import Keys from selenium.webdriver.common.by import By target = driver.find_element(By.키의 대문자)</pre>

※ 여기서 태그명이란, HTML 요소를 의미함

※ 여기서 키란, HTML에서 특정 속성을 의미함

실습

서울시 주유 가격 파악

데이터 수집

 <http://www.opinet.co.kr/searRgSelect.do>



Opinet

싼 주유소찾기 국내유가통계 유가관련정보 불법행위공표 이용안내 로그인 회원가입

지역별 경로별 도로별 면세유

select#SIDO_NMO 62 x 20

지역: 서울 시/군/구 읍/면/동

검색방법: 지역 지도 상호 환경

형태: 일반 선택 불합 인증 전산

상표: 전체 SK, GS, GS24, HD, S-Oil, OIL, 91, 95, 98

부가정보: 세차장, 경주차, 편의점, 24시간

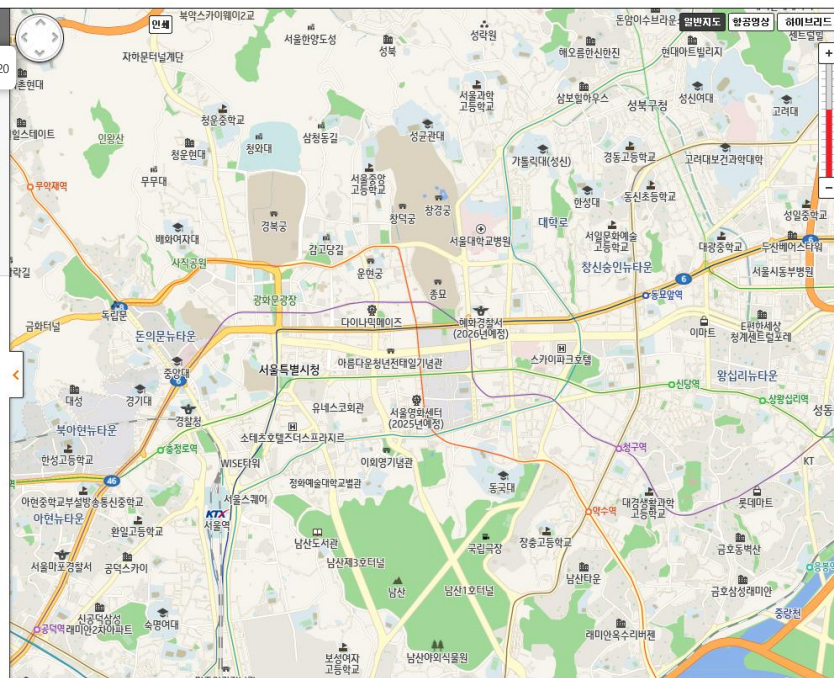
조외

고급위법유 보통위법유 경유 일반유

검색결과 (총 33개) 불합 선택 인증 전산

주유소명	휘발유	경유
평균가격	1711	1530
한일주유소	1674	1504
그린주유소	1675	1495
문화주유소	1675	1495
수성현대셀프주유소	1675	1505
기분좋은주유소	1683	1503
KK(주)중앙2주...	1685	1505
출구석유(주)만촌	1685	1505
태영주유소	1685	1499
메트로팔레스주유소	1685	1499
버디클럽오일뱅크	1695	1515

* 본 가격정보는 특정시점에 수집된 가격에 실제 판매가격과 다소 차이가 있을 수 있습니다.



HTML 구조 확인 방법

F12 또는 ctrl + U 으로 확
인

DevTools is now available in Korean!

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

Elements Console Sources Network Performance Memory

```
<input type="hidden" id="SIDO_NMO" name="SIDO_NMO" value="14ub">
<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="LP6_YN" name="LP6_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 class="search_form">
        <div>
          <div class="fir">
            <div class="fir">
              <span class="hidden">
                <select style="width:62px;" id="SIDO_NMO" name="SIDO_NMO">
              </select>
              <span class="hidden">
                <select style="width:102px;" id="SIGUNGU_NMO" name="SIGUNGU_NMO">
              </select>
              <span class="hidden">
                <select style="width:70px;" id="DONG_NM" name="DONG_NM">
              </select>
              <input id="GIS_X_COORD" name="GIS_X_COORD" type="hidden" value>
              <input id="GIS_Y_COORD" name="GIS_Y_COORD" type="hidden" value>
              <input id="GIS_X_COORD_S" name="GIS_X_COORD_S" type="hidden" value>
              <input id="GIS_Y_COORD_S" name="GIS_Y_COORD_S" type="hidden" value>
              <input id="GIS_X_COORD_E" name="GIS_X_COORD_E" type="hidden" value>
              <input id="GIS_Y_COORD_E" name="GIS_Y_COORD_E" type="hidden" value>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```


실습

서울시 주유 가격 파악

데이터 수집

 <http://www.opinet.co.kr/searRgSelect.do>



Opinet

싼 주유소찾기 국내유가통계 유가관련정보 불법행위공표 이용안내 로그인 회원가입

지역별 경로별 도로별 면세유

주유소 select#SIGUNGU_NMO 102 x 20

지역: 서울 > 시/군/구: 강남구 > 읍/면/동: 역삼동

검색방법: ☒ 지역 ☐ 지도 ☐ 상호 ☐ 환경

필터: ☒ 일반 ☒ 셀프 ☒ 불법 ☒ 인증 ☒ 전산

상표: ☒ SK ☒ GS ☒ GS 9700 ☒ HD ☒ S-OIL ☒ 98 ☒ 95 ☒ 90

부가정보: ☐ 세차장 ☐ 경정비 ☐ 편의점 ☐ 24시간

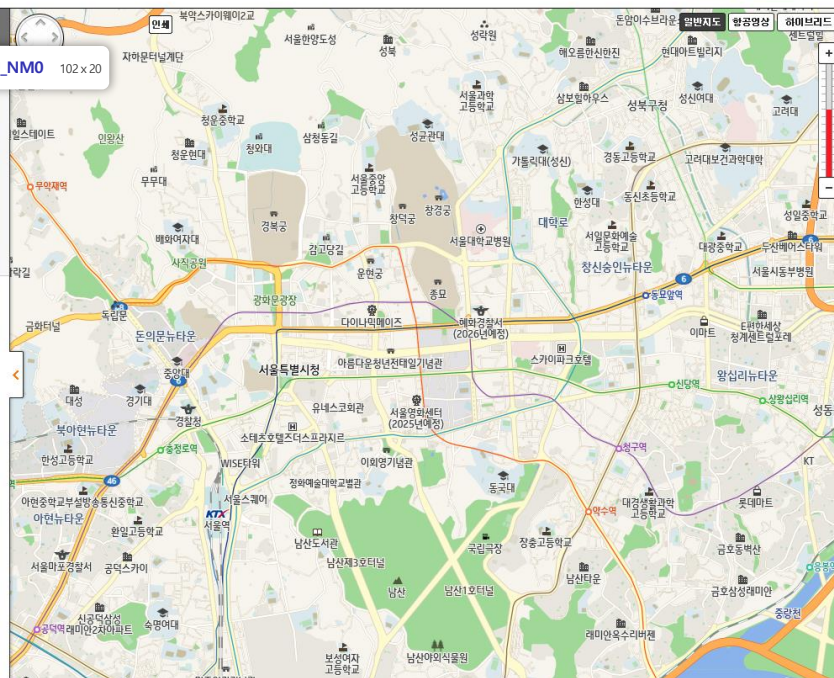
조회

고급회원유 보통회원유 경유 선택회원유

검색결과 (총 33개) ☒ 불법 ☒ 셀프 ☒ 인증 ☒ 전산

주유소명	휘발유	경유
평균가격	1711	1530
SK 한일주유소	1674	1504
GS 그린주유소	1675	1495
GS 문화주유소	1675	1495
GS 수성현대셀프주유소	1675	1505
GS 기본충전주유소	1683	1503
SK KK(주)중앙2주...	1685	1505
GS 송구석유(주)만촌	1685	1505
GS 태영주유소	1685	1499
GS 메트로팔레스주유소	1685	1499
GS 버디컴포알뜰뱅크	1695	1515

* 본 가격정보는 특정시점에 수집된 가격에 실제 판매가격과 다소 차이가 있을 수 있습니다.



HTML 구조 확인 방법

F12 또는 ctrl + U 으로 확
인

DevTools is now available in Korean!

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

Elements Console Sources Network Performance Memory

```
<input type="hidden" id="SIGUNGU" name="SIGUNGU" value="14Ub">
<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>
<div class="search_gis">
  <div class="inner">
    <div class="tab_type1">...</div>
    <div class="search_form">
      <div>
        <div class="fir">...</div>
        <div class="fir">
          <span class="hidden">...</span>
          <select style="width:62px;" id="SIDO_NMO" name="SIDO_NMO">...</select>
          <span class="hidden">...</span>
          <select style="width:102px;" id="SIGUNGU_NMO" name="SIGUNGU_NMO">...</select>...</div>
          <span class="hidden">...</span>
          <select style="width:70px;" id="DONG_NM" name="DONG_NM">...</select>
          <input id="GIS_X_COORD" name="GIS_X_COORD" type="hidden" value>
          <input id="GIS_Y_COORD" name="GIS_Y_COORD" type="hidden" value>
          <input id="GIS_X_COORD_S" name="GIS_X_COORD_S" type="hidden" value>
          <input id="GIS_X_COORD_E" name="GIS_X_COORD_E" type="hidden" value>
          <input id="GIS_Y_COORD_S" name="GIS_Y_COORD_S" type="hidden" value>
          <input id="GIS_Y_COORD_E" name="GIS_Y_COORD_E" type="hidden" value>
        </div>
      </div>
    </div>
  </div>
</div>
```

실습

▪ 서울시 주유 가격 파악

• 데이터 수집

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import Select

driver = webdriver.Chrome()
driver.get("http://www.opinet.co.kr/searRgSelect.do")

# 검색 지역 - 서울(시/도) 지정
sido_select_element = driver.find_element(By.ID, "SIDO_NM0")
sido_select = Select(sido_select_element).select_by_visible_text('서울')

# 서울의 행정구역 리스트 생성
gu_list_raw = driver.find_element(By.XPATH, """/*[@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("")
# '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'
```


실습

서울시 주유 가격 파악

데이터 수집

 <http://www.opinet.co.kr/searRgSelect.do>



Opinet

싼 주유소찾기 국내유가통계 유가관련정보 불법행위공표 이용안내 로그인 회원가입

지역별 | 경로별 | 도로별 | 면세유

주유소 | 충전소

지역: 서울 | 시/군/구: | 읍/면/동: | 검색방법: ☒ 지체 ☐ 지도 ☐ 상하 ☐ 반경

할대: ☒ 일반 ☒ 선택 ☒ 불법 ☒ 인증 ☒ 전산

상표: ☒ 전체 ☒ SK ☒ GS ☒ S-OIL ☒ VPS ☒ HD

부가정보: ☐ 세차장 ☐ 경정비 ☐ 편의점

a#searRgSelect.btn_type1 62 x 27

조회

고급휘발유	보통휘발유	경유	심해경유
검색결과 (총 33개)	별도	별도	별도
주유소명	위발유	경유	
평균가격	1711	1530	
SK 한일주유소	1674	1504	
그린주유소	1675	1495	
문화주유소	1675	1495	
수성현대셀프주유소	1675	1505	
기분좋은주유소	1683	1503	
KK(주)중앙2구	1685	1505	
솔구석유(주)관촌	1685	1505	
태영주유소	1685	1499	
메트로플러스주유소	1685	1499	
바다를잡을뱅크	1695	1515	

* 본 가격정보는 통상시점에 수집된 가격으로 실제 판매가격과 다소 차이가 있을 수 있습니다.

HTML 구조 확인 방법

F12 또는 ctrl + U 으로 확인



DevTools is now available in Korean!

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

Elements Console Sources Network Performance Memory >>

```
<div class="fly_box" style="height: 874px;>
  <div class="fly_inner" style="height: 839px;>
    <div class="header_path_gsi">
      <form id="searr90" 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="SIGUNGM_NM" name="SIGUNGM_NM" value="강남구">
        <input type="hidden" id="SIDO_CD" name="SIDO_CD" value="14">
        <input type="hidden" id="SIGUN_CD" name="SIGUN_CD" value="1406">
        <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="LP6_YN" name="LP6_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 class="search_form">
              <input type="text" value="" />
            </div>
          </div>
          <div class="t_right">
            <a class="btn_type1" href="#" id="searRgSelect" title="조회">
              <span>조회</span>
            </a>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

실습

서울시 주유 가격 파악

데이터 수집

 <http://www.opinet.co.kr/searRgSelect.do>



Opinet

싼 주유소찾기 국내유가통계 유가관련정보 불법행위공표 이용안내 로그인 회원가입

지역별 경로별 도로별 면세유

주유소 충전소

지역 서울 시/군/구 읍/면/동

검색방법 지역 지도 상하 환경

할대 일반 선택 선택 불법 인증 전산

상표 전체 SK OIL GS ENEC HD

부가정보 세차장 경정비 편의점 24시간

조회

고급검색	보통검색	경유	심해경유
검색결과 (총 33개)	불법 선택 인증 전산		
주유소명	위발유	경유	
평균가격	1711	1530	
한일주유소	1674	1504	
그린주유소	1675	1495	
문화주유소	1675	1495	
수성현대셀프주유소	1675	1505	
기본충원주유소	1683	1503	
KK(주)중앙2구	1685	1505	
출구석유(주)한솔	1685	1505	
태양주유소	1685	1499	
메트로주유소	168		
바다를잡을뱅크	169		

a#btn_type6_ex_save 71.13 x 20

* 본 가격정보는 특정시간에 수집된 가격으로 실제 판매가격과 다소 차이가 있을 수 있습니다.

HTML 구조 확인 방법

F12 또는 ctrl + U 으로 확인



DevTools is now available in Korean!

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

```
Elements Console Sources Network Performance Memory >>
<input id="btn_UIV_SIH" name="btn_UIV_SIH" 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="SIGUN_NM" name="SIGUN_NM" value="강남구">
<input type="hidden" id="SIDO_CD" name="SIDO_CD" value="14">
<input type="hidden" id="SIGUN_CD" name="SIGUN_CD" value="1405">
<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=">
<div class="search_gis">
<!-- //search_gis -->
<div class="result_gis">
<div class="inner" id="templ_list0">
<div class="tab_type2_mgb_7">
<div class="desc_box">
<!-- 일반위발유 -->
<div id="os_price1" class="t_center over flow_gis_mgb_7" style="height:320px;">
<!-- 고급위발유 -->
<div id="os_price2" class="t_center over flow_gis_mgb_7" style="height:320px; display:none;">
<!-- 경유 -->
<div id="os_price3" class="t_center over flow_gis_mgb_7" style="height:320px; display:none;">
<!-- LPG -->
<div id="os_price4" class="t_center over flow_gis_mgb_7" style="height:320px; display:none;">
<div class="desc_box">
<div class="d_right">
<div class="btn_type6_ex_save" href="javascript:if_excel_download('os_btn_1');" title="사양열람:역설지장">
</div>
</div>
</div>
</div>
```

실습

▪ 서울시 주유 가격 파악

• 데이터 수집

```
import time

for gu in gu_names:
    element = driver.find_element(By.ID, "SIGUNGU_NM0")
    element.send_keys(gu)
    time.sleep(2)

    element_sel_gu = driver.find_element(By.XPATH, "//*[@id='searRgSelect']").click() # 조회 버튼 클릭
    time.sleep(1)

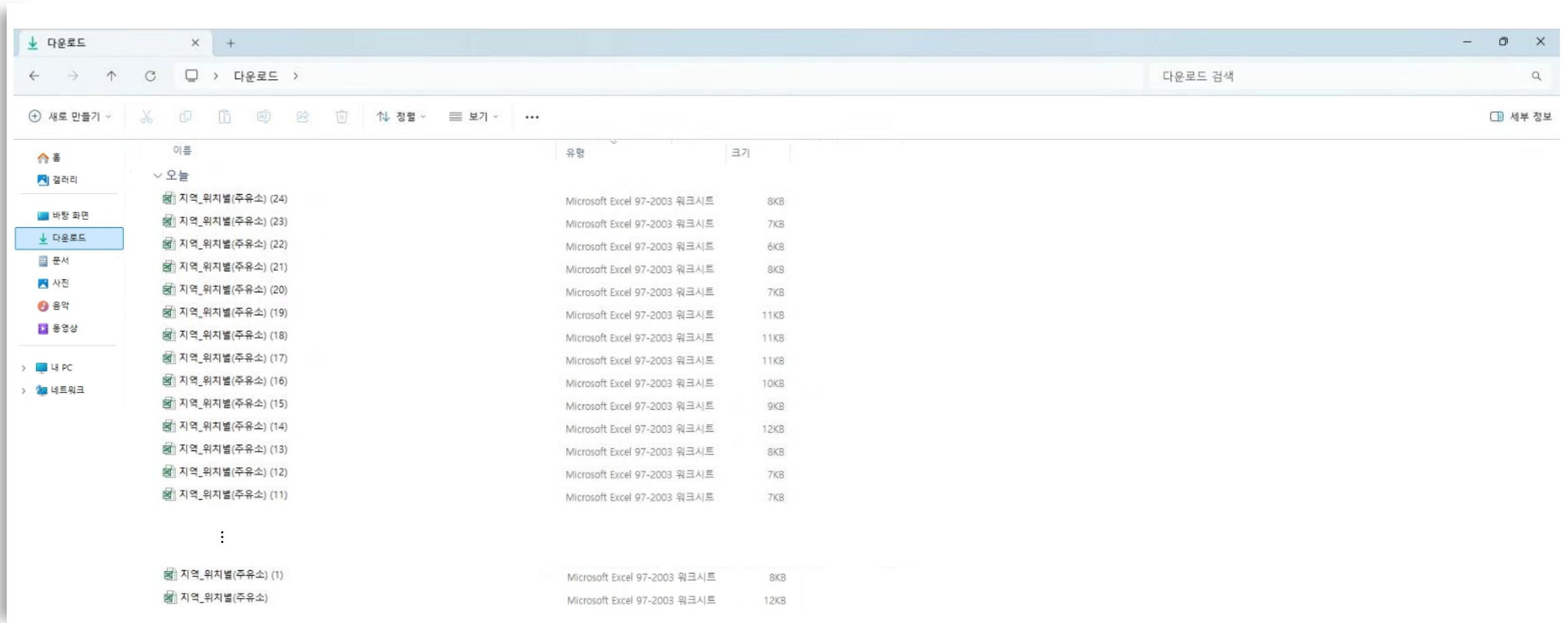
    element_get_excel = driver.find_element(By.XPATH, "//a[@class='btn_type6_ex_save']").click() # 엑셀 저장 버튼 클릭
    time.sleep(1)

driver.close() #수집 완료 후 활성화된 웹 크롤링 브라우저 종료
```

실습

■ 서울시 주유 가격 파악

• 데이터 수집



실습

▪ 서울시 주유 가격 파악

• 데이터 정제

```
import glob
import pandas as pd

stations_files = glob.glob('./지역_위치별(주유소) *xls')

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)

stations = station_raw[['상호', '주소', '휘발유', '셀프여부', '상표']]
stations.rename(columns = {'상호':'Oil_store', '휘발유':'가격', '셀프여부':'셀프'}, inplace = True)
```



	Oil_store	주소	가격	셀프	상표
0	면목SELF주유소	서울 중랑구 동일로 627 (면목동)	1667	Y	SK에너지
1	대원주유소	서울 중랑구 동일로 600	1667	Y	GS칼텍스
2	우림주유소	서울 중랑구 용마산로 487 (망우제3동)	1667	Y	GS칼텍스

실습

■ 서울시 주유 가격 파악

• 데이터 정제

```
# 주소에서 구 이름 추출
stations['구'] = [eachAddress.split()[1] for eachAddress in stations['주소']]
stations['구'].unique()
```



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

```
# 가격변수에서 '-'으로 표기된 결측값 확인 후 float(실수형)으로 변경
stations['가격'].unique()
stations['가격'] = [float(value) for value in stations['가격']]
```



1667, 1677, 1687, 1699, 1713, 1717, 1718, 1758, 1728, 1739, 1742, 1747, 1749, 1755, 1756, 1786, 1787, 1795, 1799, 1828, 1834, 1845,
1860, 1887, 1914, 1928, 1959, 2060, 2178, 2228, 2273, 2288, 2296, 2395, 2498, 2593, 1719, 1738, 1745, 1766, 1780, 1785, 1790, 1895,
1979, 2018, 1665, 1675, 1695, 1735, 1668, 1684, 1685, 1688, 1696, 1698, 1715, 1737, 1740, 1748, 1768, 1777, 1798, 1993, 1725, 1729,
1743, 1825, 1898, 2059, 1689, 1694, 1733, 1796, 1958, 2290, 1693, 1753, 1759, 1760, 1855, 1679, 1697, 1732, 1765, 1767, 1704, 1709,
1778, 1744, 1754, 1779, 1875, 1764, 1774, 1808, 1726, 1659, 1734, 1736, 1757, 1893, 2068, 2168, 1669, 1678, 1707, 1818, 1838, 1843,
1995, 2145, 2281, 1673, 1723, 1829, 2014, 1730, 1968, 1682, 1691, 1724, 1727, 1789, 1864, 1763, 1854, 2093, 2195, 2295, 1937, 1985,
2130, 2169, 2170, 2188, 2199, 2287, 2362, 2829, 1664, 1674, 1840, 1849, 1890, 2298, 2345, 1878, 1896, 2078, 2198, 2539, 2752

서울시 주유 가격 파악

데이터 정제

```
# 최종 데이터셋 구성
stations.reset_index(inplace = True)
del stations['index']
```



	Oil_store	주소	가격	셀프	상표	구
0	면목SELF주유소	서울 중랑구 동일로 627 (면목동)	1667	Y	SK에너지	중랑구
1	대원주유소	서울 중랑구 동일로 600	1667	Y	GS칼텍스	중랑구
2	우림주유소	서울 중랑구 용마산로 487 (망우제3동)	1667	Y	GS칼텍스	중랑구
3	(주)자연에너지 대창주유소	서울 중랑구 동일로 636 (면목동)	1667	Y	HD현대오일뱅크	중랑구
4	오천만주유소	서울 중랑구 동일로 547 (면목동)	1667	Y	S-OIL	중랑구
⋮	⋮	⋮	⋮	⋮	⋮	⋮
428	서남주유소	서울 중구 통일로 30	2752	N	SK에너지	중구

실습

▪ 서울시 주유 가격 파악

- 운영 체제에 따라 시각화에 필요한 한글 폰트 지정

```
import platform
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc

path = "./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')

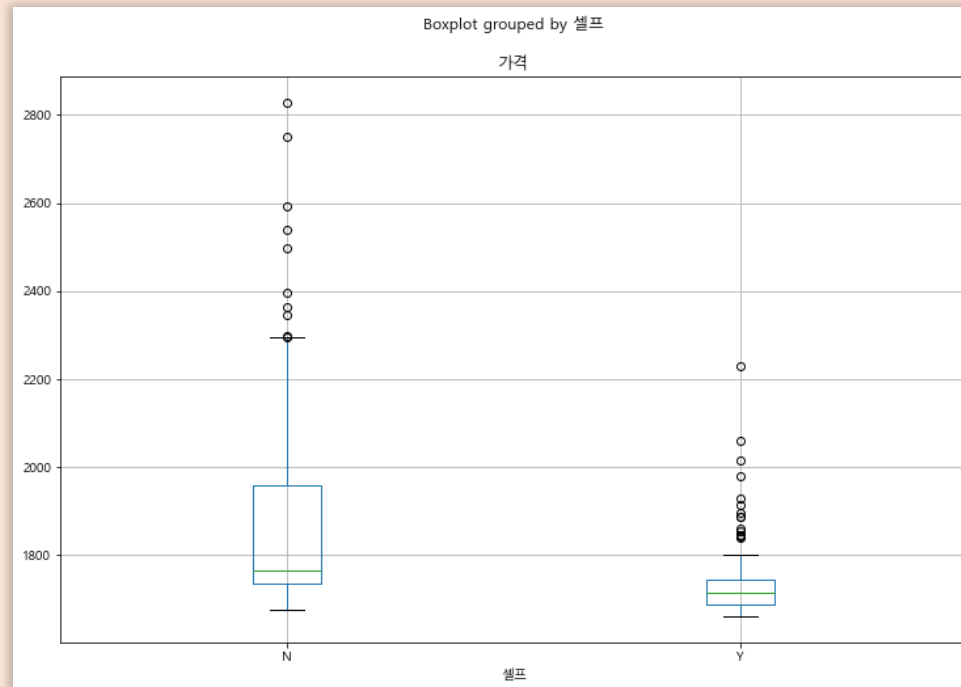
plt.rcParams['axes.unicode_minus'] = False
```


실습

▪ 서울시 주유 가격 파악

- 주유 셀프 여부에 따른 가격 분포 확인 (1)

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



실습

▪ 서울시 주유 가격 파악

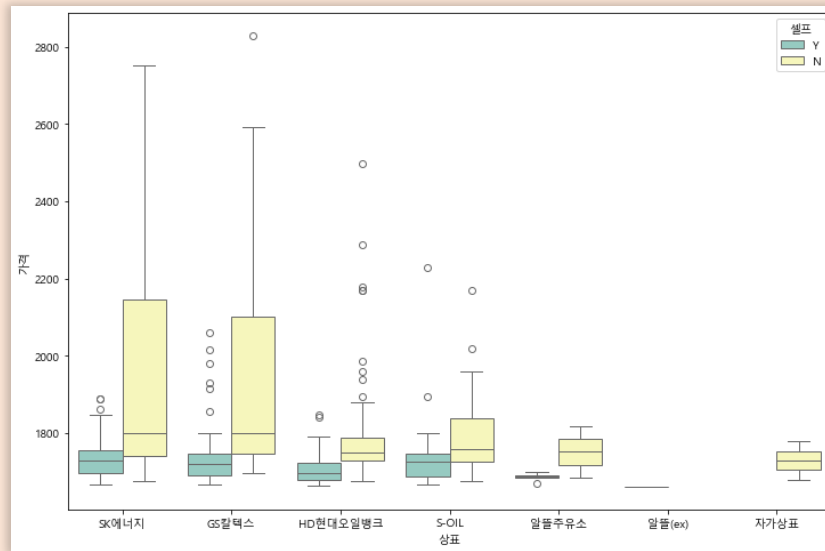
- 주유 셀프 여부에 따른 가격 분포 확인 (2)

```
import seaborn as sns
```

```
plt.figure(figsize = (12,8))
```

```
sns.boxplot(x = "상표", y = "가격", hue = "셀프", data = stations, palette = "Set3")
```

```
plt.show()
```

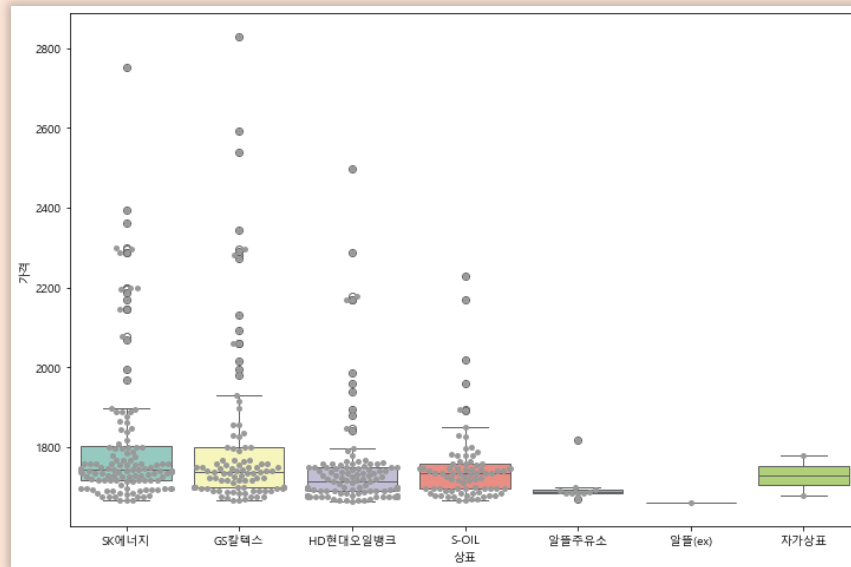


실습

▪ 서울시 주유 가격 파악

- 주유 셀프 여부에 따른 가격 분포 확인 (3)

```
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()
```



실습

▪ 서울시 주유 가격 파악

• 서울시 구별 주유 가격 시각화

```
import json
import folium
import numpy as np

stations.sort_values(by = '가격', ascending = False).head(10)
gu_data = pd.pivot_table(stations, index = ["구"], values = ["가격"], aggfunc = np.mean)

# 시각화
geo_str = json.load(open('./skorea_municipalities_geo_simple.json', encoding='utf-8'))
oil_map = folium.Map(location = [37.5502, 126.982], zoom_start = 10.5)
folium.Choropleth( geo_data = geo_str, data = gu_data, columns = [gu_data.index, '가격'], key_on = 'feature.id',
                    fill_color = 'PuBu', fill_opacity = 0.7, line_opacity = 0.2).add_to(oil_map)

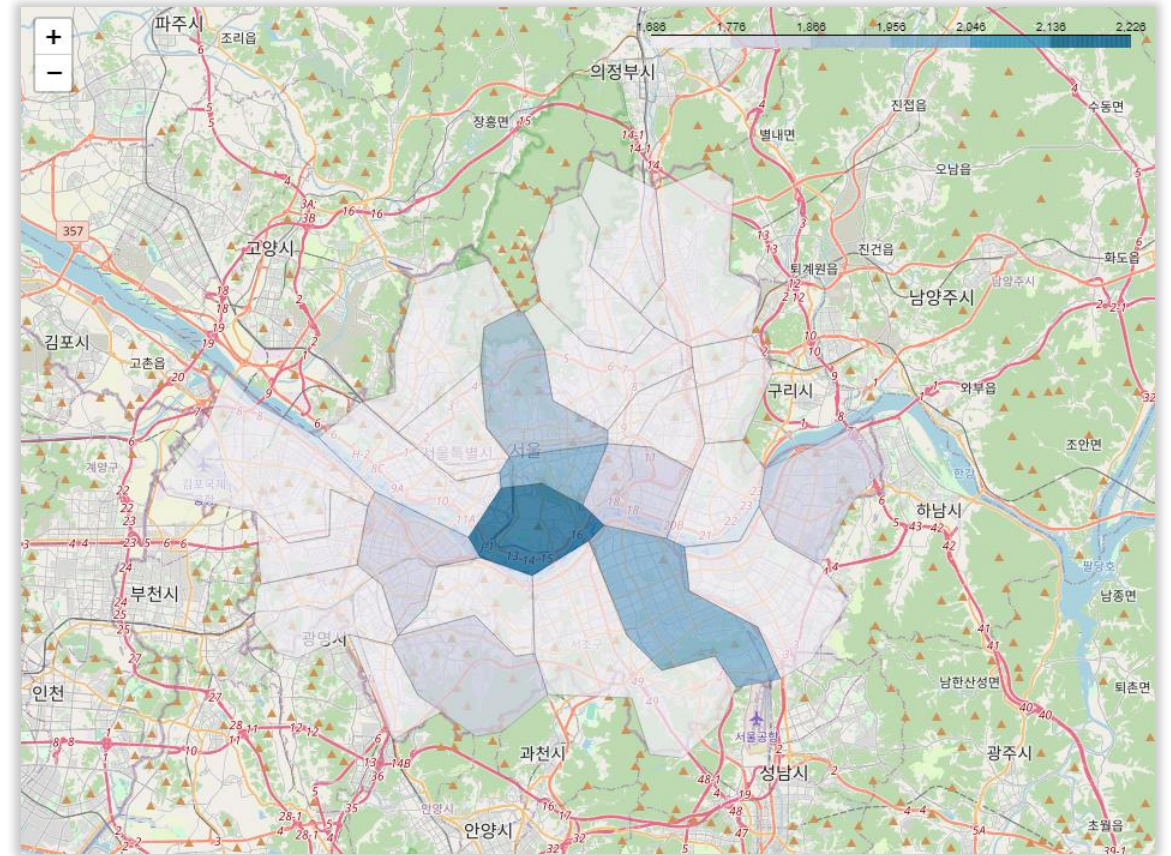
oil_map.save('./서울시 구별 주유 가격.html')
```

실습

■ 서울시 주유 가격 파악

• 서울시 구별 주유 가격 시각화

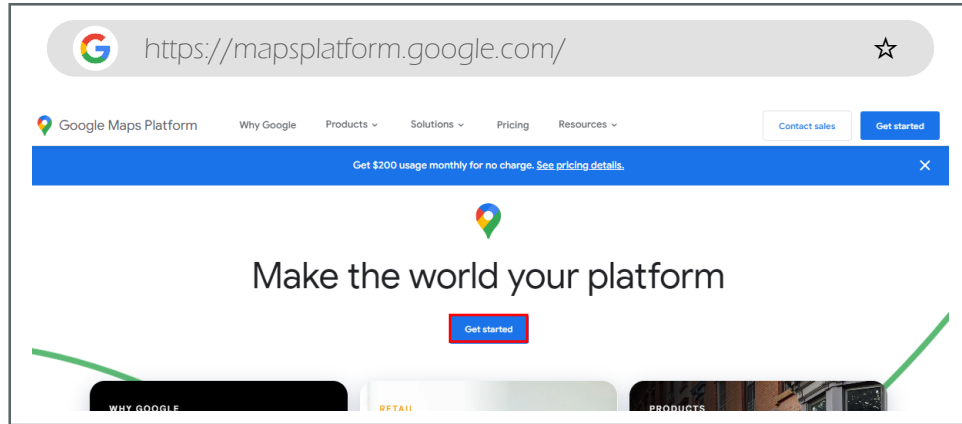
순위	구	가격	순위	구	가격
1	용산구	2226.083	14	강서구	1735.061
2	종구	2046.100	15	노원구	1731.571
3	강남구	1970.138	16	금천구	1729.091
4	종로구	1939.375	17	성북구	1717.391
5	강동구	1809.846	18	양천구	1716.292
6	성동구	1809.500	19	동대문구	1710.813
7	영등포구	1787.370	20	광진구	1709.833
8	관악구	1777.214	21	서대문구	1709.667
9	서초구	1775.806	22	은평구	1700.533
10	동작구	1768.778	23	도봉구	1697.867
11	마포구	1746.333	24	중랑구	1694.000
12	송파구	1745.071	25	강북구	1686.273
13	구로구	1744.556			



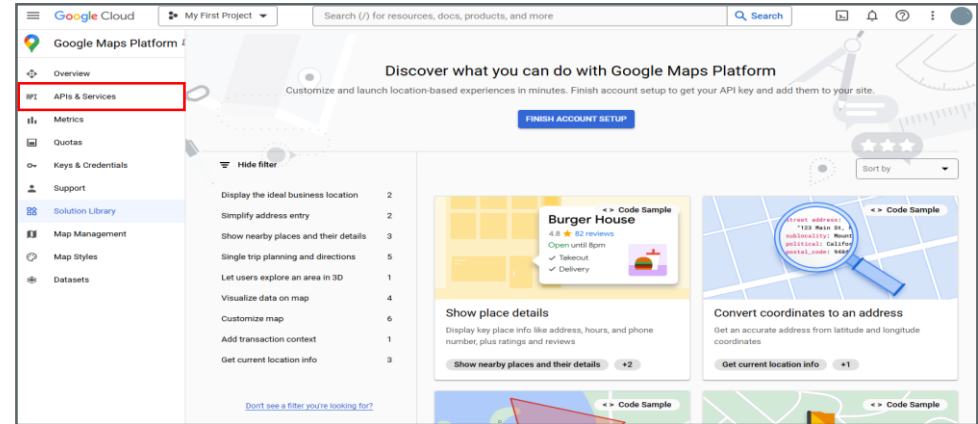
실습

■ 서울시 주유 가격 파악

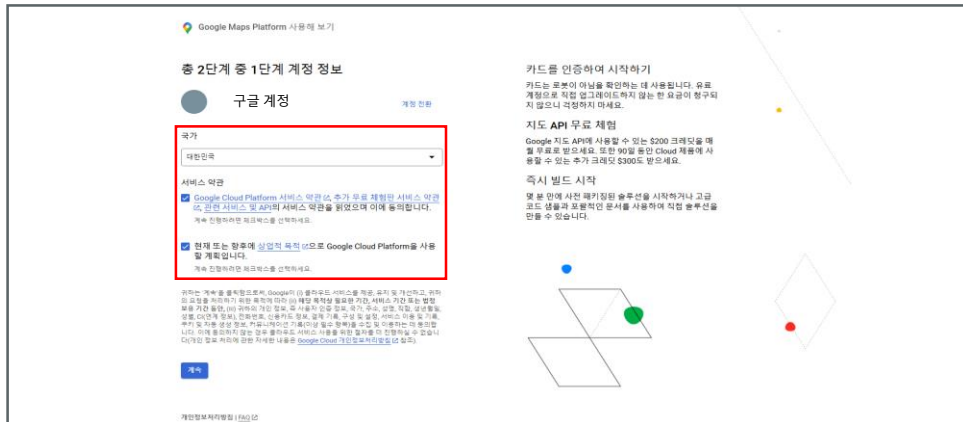
• 구글 맵 개인 API 키 발급



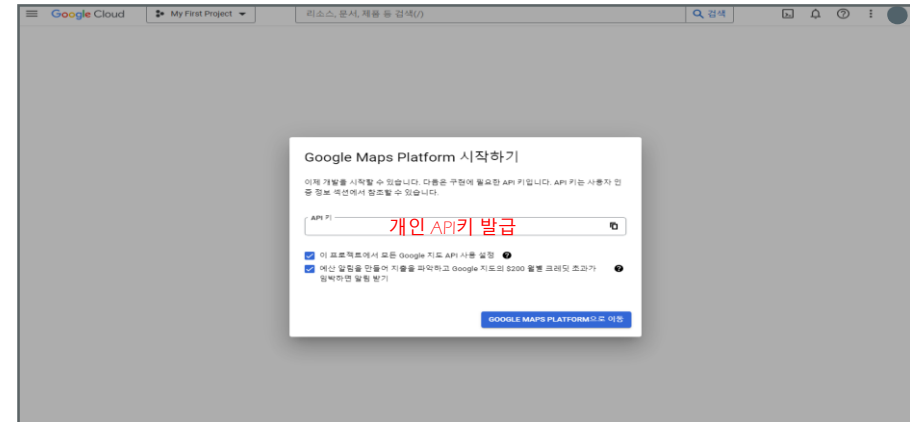
▶ 구글 맵 플랫폼 사이트로 이동 후 시작하기 클릭



▶ API & Services(API 및 서비스) 이동



▶ 결제 정보 입력(총 2단계)



▶ 개인 API 키 발급

서울시 주유 가격 파악

- 서울시 주유 가격 상·하위 10곳의 주유소 위치 표기

서울시 주유 가격 상·하위 10개 자료 추출

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

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



상위 순위	Oil_store	구	가격	상위 순위	Oil_store	구	가격
1	서계주유소	용산구	2829	6	SK논현주유소	강남구	2395
2	서남주유소	중구	2752	7	한석주유소	용산구	2362
3	(주)만정에너지 삼보주유소	강남구	2593	8	(주)대양씨앤씨 사직주유소	종로구	2345
4	필동주유소	중구	2539	9	SK북악주유소	종로구	2298
5	(주)제이제이네트웍스 제이제이주유소	강남구	2498	10	대청주유소	강남구	2296

하위 순위	Oil_store	구	가격	하위 순위	Oil_store	구	가격
1	만남의광장주유소	서초구	1659	6	(주)자연에너지 대창주유소	중랑구	1667
2	(주)아이에스피에너지 (서울서부고속주유소)	은평구	1664	7	오천만주유소	중랑구	1667
3	덕릉로주유소	강북구	1665	8	면목SELF주유소	중랑구	1667
4	(주)자연에너지 번동주유소	강북구	1665	9	(주)자연에너지 중원주유소	광진구	1667
5	우림주유소	중랑구	1667	10	서울중앙고속주유소	광진구	1667

실습

▪ 서울시 주유 가격 파악

- 서울시 주유 가격 상·하위 10곳의 주유소 위치 표기

```
import googlemaps
gmaps = googlemaps.Client(key = "*****") # 개인 key 입력

# 서울시 주유 가격 상위 10개 주유소의 시각화를 위한 좌표 생성
lat = []; lng = []
for n in 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
```


▪ 서울시 주유 가격 파악

- 서울시 주유 가격 상·하위 10곳의 주유소 위치 표기

```
# 지도 생성
oil_topbottom = folium.Map(location = [37.5202, 126.975], zoom_start = 10.5)

# 상·하위 10개 주유소 마커 추가
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(oil_topbottom)

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(oil_topbottom)

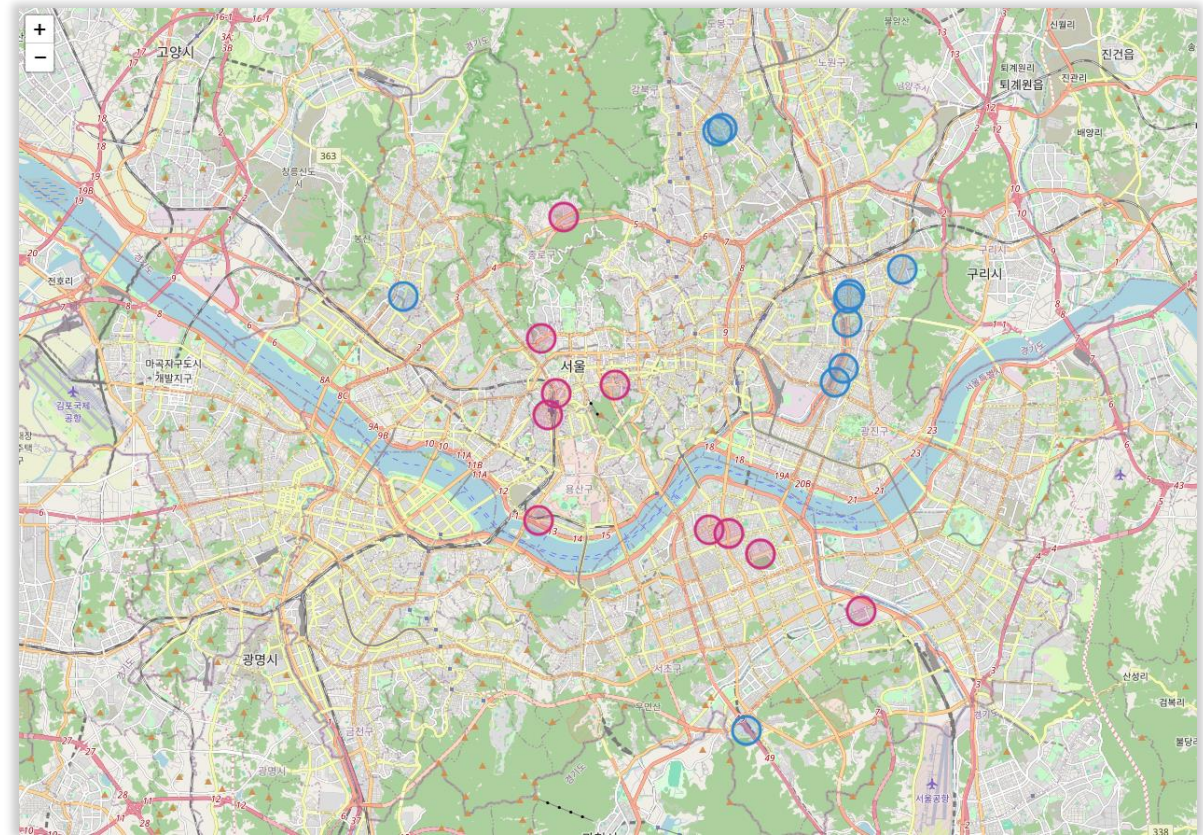
oil_topbottom.save('./서울시 주유 상·하위 가격.html')
```

실습

서울시 주유 가격 파악

서울시 주유 가격 상하위 10곳의 주유소 위치 표기

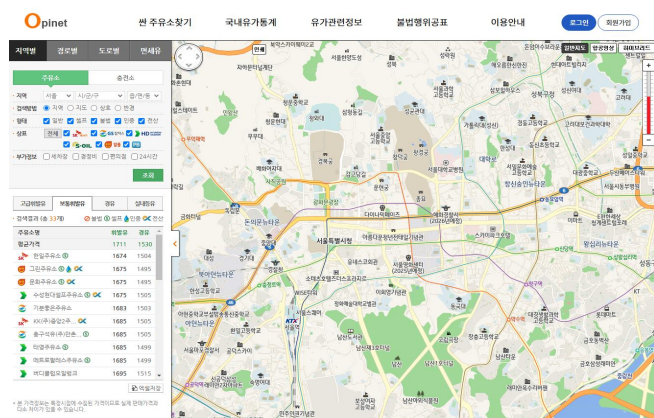
상위	Oil_store	구	가격	하위	Oil_store	구	가격
1	서계주유소	용산구	2829	1	만남의광장주유소	서초구	1659
2	서남주유소	중구	2752	2	(주)아이에스피에너지 (서울서부고속주유소)	은평구	1664
3	(주)만정에너지 삼보주유소	강남구	2593	3	덕릉로주유소	강북구	1665
4	필동주유소	중구	2539	4	(주)자연에너지 번동주유소	강북구	1665
5	(주)제이제이네트웍스 제이제이주유소	강남구	2498	5	우림주유소	중랑구	1667
6	SK논현주유소	강남구	2395	6	(주)자연에너지 대창주유소	중랑구	1667
7	한석주유소	용산구	2362	7	오천만주유소	중랑구	1667
8	(주)대양씨앤씨 사직주유소	종로구	2345	8	면목SELF주유소	중랑구	1667
9	SK북악주유소	종로구	2298	9	(주)자연에너지 중원주유소	광진구	1667
10	대청주유소	강남구	2296	10	서울중앙고속주유소	광진구	1667



정리하기

학습 내용

1. 파이썬에서 `selenium`을 이용하여 인터넷 자료를 수집하였다.



	Oil_store	주소	가격	셀프	상표	구
0	면목SELF주유소	서울 중랑구 동일로 627 (면목동)	1667	Y	SK에너지	중랑구
1	대원주유소	서울 중랑구 동일로 600	1667	Y	GS칼텍스	중랑구
2	우림주유소	서울 중랑구 용마산로 487 (망우제3동)	1667	Y	GS칼텍스	중랑구
3	(주)자연에너지 대창주유소	서울 중랑구 동일로 636 (면목동)	1667	Y	HD현대오일뱅크	중랑구
4	오천만주유소	서울 중랑구 동일로 547 (면목동)	1667	Y	S-OIL	중랑구
:	:	:	:	:	:	:
428	서남주유소	서울 중구 통일로 30	2752	N	SK에너지	중구

2. 수집한 자료를 활용하여 상자그림과 지도 시각화를 작성하였다.

