**웹의 데이터로 그래프 그리기**

|  |  |
| --- | --- |
|  | 웹페이지의 HTML 가져오기 |
| 크롤링 | ↓ |
|  | 필요한 데이터만 가져와서 파일로 저장 |
|  | ↓ |
| 전처리 | 그래프로 그릴 수 있도록 데이터 가공 |
|  | ↓ |
| 시각화 | 데이터로 그래프 그리기 |
|  |  |

크롤링 : 웰페이지를 가져오고 파일로 저장하는 크롤링과정

전처리 과정 : 데이터를 가공하는 과정

시각화 : 데이터로 그래프를 그리는 시각화 과정

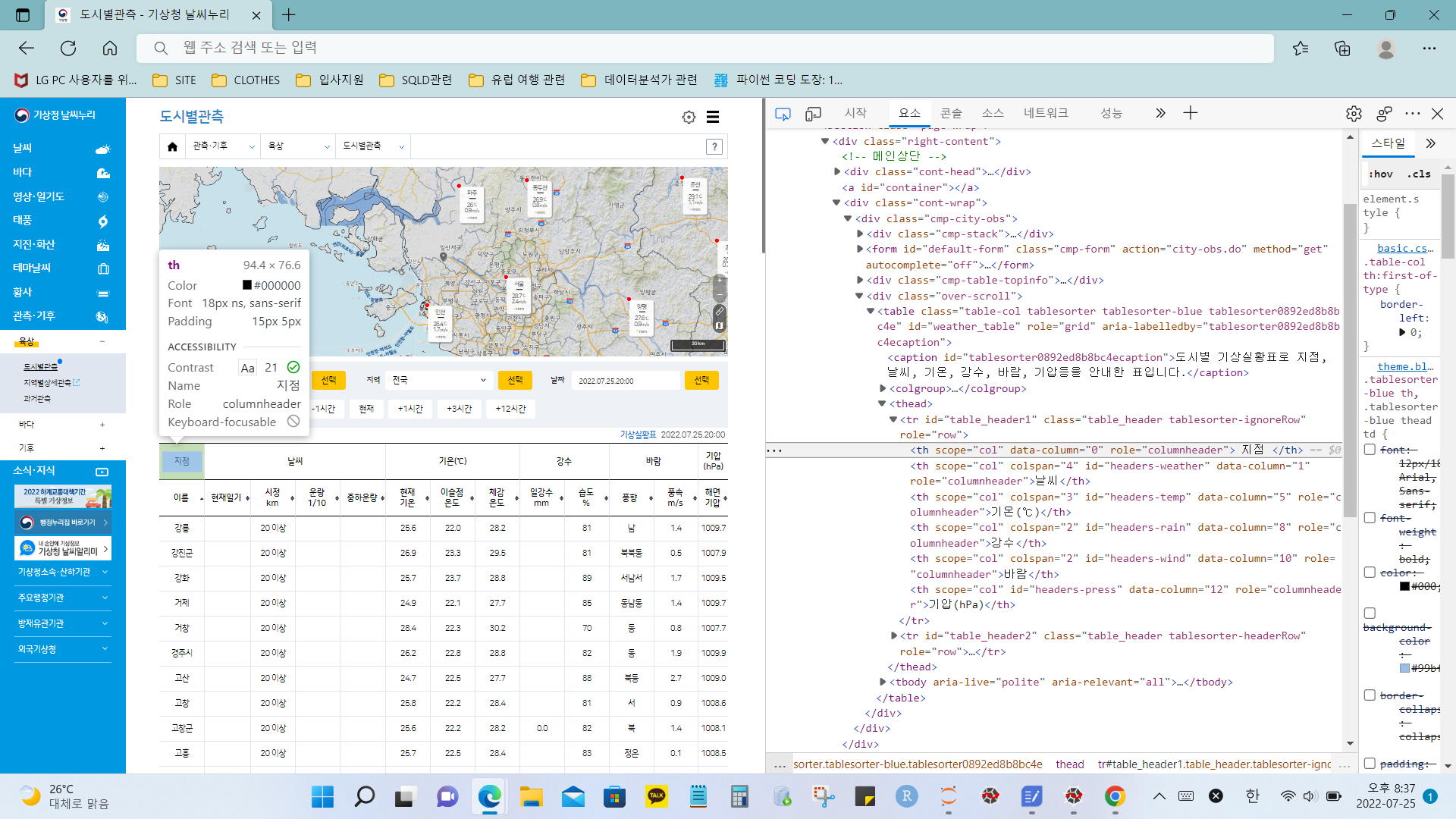
데이터 : 2022.07.25 현재 기준

**1. 웹페이지 HTML 가져오기**

\*\*도시별 현재날씨 > 지상관측자료 > 관측자료 > 날씨 > 기상청  
URL : http://www.kma.go.kr/weather/observation/currentweather.jsp

\*\*F12를 눌러 개발자 도구 표시하기

\*\*Ctrl+Shift+C 클릭한 뒤 현재 날씨 표에서 지점을 클릭하기

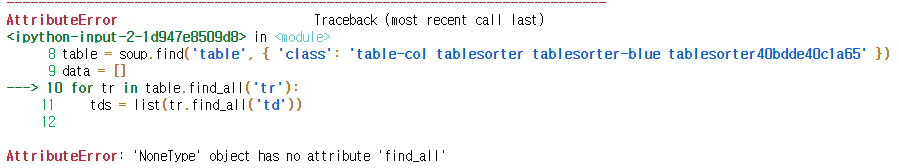




지점에 해당하는 HTML 코드가 표시 되고, 지점부분이 포함된 표 전체가 가져올 부분이다.

|  |
| --- |
| import requests # 웹 페이지의 HTML을 가져오는 모듈  from bs4 import BeautifulSoup # HTML을 파싱하는 모듈    # 웹 페이지를 가져온 뒤 BeautifulSoup 객체로 만듦  response=requests.get('https://www.weather.go.kr/w/obs-climate/land/city-obs.do?auto\_man=m&stn=0&dtm=&type=t99&reg=100&tm=2022.07.25.20%3A00')  soup = BeautifulSoup(response.content, 'html.parser')    table=soup.find('table',{'class':'table-coltablesorter tablesorter-blue tablesorter40bdde40c1a65' }) # <table class="table\_col ">을 찾음  data = [] # 데이터를 저장할 리스트 생성  for tr in table.find\_all('tr'): # 모든 <tr> 태그를 찾아서 반복(각 지점의 데이터를 가져옴)  tds = list(tr.find\_all('td')) # 모든 <td> 태그를 찾아서 리스트로 만듦  # (각 날씨 값을 리스트로 만듦)  for td in tds: # <td> 태그 리스트 반복(각 날씨 값을 가져옴)  if td.find('a'): # <td> 안에 <a> 태그가 있으면(지점인지 확인)  point = td.find('a').text # <a> 태그 안에서 지점을 가져옴  temperature = tds[5].text  # <td> 태그 리스트의 여섯 번째(인덱스 5)에서 기온을 가져옴  humidity = tds[9].text  # <td> 태그 리스트의 열 번째(인덱스 9)에서 습도를 가져옴  data.append([point, temperature, humidity])  # data 리스트에 지점, 기온, 습도를 추가    data |

위의 코드를 실행했을 때 에러 발생했다.



아래와 같은 에러는 **"검색 되어지는 값이 하나도 없을 때"**나오는 에러였다.

다시 코드를 확인해보니 table 변수 부분을 아래처럼 바꾸니 실행이 되었다.

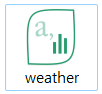
|  |
| --- |
| import requests # 웹 페이지의 HTML을 가져오는 모듈  from bs4 import BeautifulSoup # HTML을 파싱하는 모듈    # 웹 페이지를 가져온 뒤 BeautifulSoup 객체로 만듦  response = requests.get('https://www.weather.go.kr/w/obs-climate/land/city-obs.do?auto\_man=m&stn=0&dtm=&type=t99&reg=100&tm=2022.07.25.20%3A00')  soup = BeautifulSoup(response.content, 'html.parser')    table = soup.find('table', { 'class': 'table-col' }) # <table class="table-col ">을 찾음 |

**재실행 결과**



**2. 데이터를 csv파일에 저장하기**

|  |
| --- |
| with open('weather.csv','w')as file:  file.write('point,temperature,humidity\n') #컬럼 이름 추가  for i in data:  file.write('{0},{1},{2}\n'.format(i[0], i[1], i[2])) |



**3. 데이터 시각화하기**

|  |
| --- |
| # %matplotlib inline을 설정하면 matplotlib.pyplot의 show 함수를 호출하지 않아도  # 주피터 노트북 안에서 그래프가 표시됨  %matplotlib inline  import pandas as pd # 데이터를 저장하고 처리하는 패키지  #pandas의 DataFrame은 CSV파일의 내용을 변도의 처리 없이 그대로 DataFrame으로 가져올 수 있어서 편리하지만 첫줄에 컬럼이름이 들어있어야한다.  import matplotlib as mpl # 그래프를 그리는 패키지  import matplotlib.pyplot as plt # 그래프를 그리는 패키지    # csv 파일을 읽어서 DataFrame 객체로 만듦. 인덱스 컬럼은 point로 설정  df = pd.read\_csv('weather.csv', index\_col='point',encoding='euc-kr')  df |

\*\*모든 데이터를 가지고 시각화하기 어려움으로 특별시와 광역시만 모아서 DataFrame객체를 만든다.

|  |
| --- |
| #특별시와 광역시만 모아서 DataFrame 객체로 만들기  city\_df = df.loc[['서울', '인천', '대전', '대구', '광주', '부산', '울산']]  city\_df  #loc는 DataFrame의 속성으로 특정 인덱스의 데이터만 가져올 수 있다.  #인덱스는 문자열로 지정한다. |

5. 특별시와 광역시 데이터 시각화하기

|  |
| --- |
| # Windows 한글 폰트 설정  font\_name = mpl.font\_manager.FontProperties(fname='C:/Windows/Fonts/malgun.ttf').get\_name()  mpl.rc('font', family=font\_name)    # 차트 종류, 제목, 차트 크기, 범례, 폰트 크기 설정  ax = city\_df.plot(kind='bar', title='날씨', figsize=(12, 4), legend=True, fontsize=12)  ax.set\_xlabel('도시', fontsize=12) # x축 정보 표시  ax.set\_ylabel('기온/습도', fontsize=12,rotation=360) # y축 정보 표시  ax.legend(['기온', '습도'], fontsize=12) # 범례 지정 |