미세먼지 농도 시각화

코드를 짜게 된 동기

요즘 사람들이 건강에 많은 신경을 쓰고 있는 만큼 매일매일 날씨를 체크할 때 미세먼지나 황사가 얼마나 있는지로 하루 날씨의 좋고 나쁨을 판단함.

대한민국 지도에 색깔 구분이 되어있는 원으로 시각화하여 사람들이 쉽게 미세먼지 농도를 파악할 수 있도록 함.

코드 설명 1

틀 잡기

- 라이브러리 불러오기

```
import webbrowser
import requests
from urllib.parse import urlencode
import pandas as pd
import requests
from urllib.parse import urlencode
import pandas as pd
```

코드 설명 2

데이터 가공

- 공공 api 사용하기
- 판다스 사용하기

```
base_url = "http://apis.data.go.kr/B552584/UlfptcaAlarmInqireSvc"
params = {
'serviceKey': decoded_api_key,
'returnType': 'json',
'year': 2022,
'pageNo': 1,
'numOfRows': 600
}
response = requests.get(f"{base_url}/getUlfptcaAlarmInfo?
{urlencode(params)}")
date_array = []
code_array = []
val_array = []
district_array = []
if response.status_code == 200:
# JSON 파싱
data = response.json()
# 데이터 활용
for item in data['response']['body']['items']:
data_date = item.get('dataDate', 'N/A')
item_code = item.get('itemCode', 'N/A')
issue_val = item.get('issueVal', 'N/A')
district_name = item.get('districtName', 'N/A')
date_array.append(data_date)
code_array.append(item_code)
val_array.append(issue_val)
district_array.append(district_name)
else:
print("Error:", response.status_code)
a = pd.DataFrame({'발생_일시':date_array, '항목_코드':code_array, '지
역명':district_array, '발령농도': val_array})
a['발령농도'] = a['발령농도'].astype(int)
```

코드 설명 3

데이터 전처리 과정

- 필요한 열과 월 제외하고 나머지 데이터 삭제하기

```
import pandas as pd
csv_test = a
def date_replace(x):
if x[5:7] == '12':
x = '12'
return x
elif x[5:7] == '11':
x = '11'
return x
if x[5:7] == '10':
x = '10'
return x
if x[5:7] == '09':
x = '09'
return x
if x[5:7] == '08':
x = '08'
return x
if x[5:7] == '07':
x = '07'
return x
if x[5:7] == '06':
x = '06'
return x
elif x[5:7] == '05':
x = '05'
return x
elif x[5:7] == '04':
x = '04'
return x
elif x[5:7] == '03':
x = '03'
return x
elif x[5:7] == '02':
x = '02'
return x
elif x[5:7] == '01':
x = '01'
return x
csv_test['발생_일시'] = csv_test['발생_일시'].apply(date_replace)
csv_test = csv_test.groupby(['발생_일시', '항목_코드', '지역명']).mean()
csv_test.reset_index(inplace = True)
```

```
import folium
korea_weedo = 37.27538
korea_gyeongdo = 127.05488
seoul_weedo = 37.5642135
seoul_gyeongdo = 127.269311
incheon_weedo = 37.4562557
incheon_gyeongdo = 126.7052062
gwangju_weedo = 35.126033
gwangju_gyeongdo = 126.831302
daegu_weedo = 35.798838
daegu_gyeongdo = 128.583052
ulsan_weedo = 35.519301
ulsan_gyeongdo = 129.239078
daejeon_weedo = 36.321655
daejeon_gyeongdo = 127.378953
busan_weedo = 35.198362
busan_gyeongdo = 129.053922
gyeongi_weedo = 37.567167
gyeongi_gyeongdo = 127.190292
gangwon_weedo = 37.555837
gangwon_gyeongdo = 128.209315
choongnam_weedo = 36.557229
choongnam_gyeongdo = 126.779757
choongbuk_weedo = 36.628503
choongbuk_gyeongdo = 127.929344
gyeongbuk_weedo = 36.248647
gyeongbuk_gyeongdo = 128.664734
gyeongnam_weedo = 35.259787
gyeongnam_gyeongdo = 128.664734
jeonbuk_weedo = 35.716705
jeonbuk_gyeongdo = 127.144185
jeonnam_weedo = 34.819400
jeonnam_gyeongdo = 126.893113
jeju_weedo = 33.364805
jeju_gyeongdo = 126.542671
dust = input("항목을 입력하세요: ")
date = input("발생일시를 입력하세요: ")
m = None # 지도 객체 초기화
```

코드 설명 4-1

지도 만들기

- 실제로 시각화하기
- 미세먼지의 농도에 따라 색깔이 다름

```
if dust == "PM25":
selected_rows = csv_test[(csv_test['항목_코드'] == 'PM25') & (csv_test['발생_일시'] == date)]
if not selected_rows.empty:
m = folium.Map(
location=[korea_weedo, korea_gyeongdo],
zoom_start=7,
width=750,
height=500
)
folium.Circle([seoul_weedo,seoul_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([incheon_weedo,incheon_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([gwangju_weedo,gwangju_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([daegu_weedo,daegu_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([ulsan_weedo,ulsan_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([daejeon_weedo,daejeon_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([busan_weedo,busan_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([gyeongi_weedo, gyeongi_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([gangwon_weedo, gangwon_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([choongnam_weedo, choongnam_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([gyeongbuk_weedo, gyeongbuk_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([gyeongnam_weedo, gyeongnam_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([jeonnam_weedo, jeonnam_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([jeonbuk_weedo, jeonbuk_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([jeju_weedo, jeju_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
for index, row in selected_rows.iterrows():
if row['발령농도'] >= 76 and row['발령농도']<111:
c = 'red'
elif row['발령농도'] >= 36 and row['발령농도']<75:
c = 'blue'
elif row['발령농도'] >= 0 and row['발령농도']<35:
c = 'green'
elif row['발령농도']>=111:
c = '#730602'
if row['지역명'] == '서울':
folium.Circle([seoul_weedo, seoul_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] == '인천':
folium.Circle([incheon_weedo,incheon_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] == '광주':
folium.Circle([gwangju_weedo,gwangju_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] == '대구':
folium.Circle([daegu_weedo,daegu_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] == '울산':
folium.Circle([ulsan_weedo,ulsan_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='대전':
folium.Circle([daejeon_weedo,daejeon_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='부산':
folium.Circle([busan_weedo,busan_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='경기':
folium.Circle([gyeongi_weedo, gyeongi_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='강원':
folium.Circle([gangwon_weedo, gangwon_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='충남':
folium.Circle([choongnam_weedo, choongnam_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='경북':
folium.Circle([gyeongbuk_weedo, gyeongbuk_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='경남':
folium.Circle([gyeongnam_weedo, gyeongnam_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='전남':
folium.Circle([jeonnam_weedo, jeonnam_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] == '전북':
folium.Circle([jeonbuk_weedo, jeonbuk_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='제주':
folium.Circle([jeju_weedo, jeju_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
else:
break
```

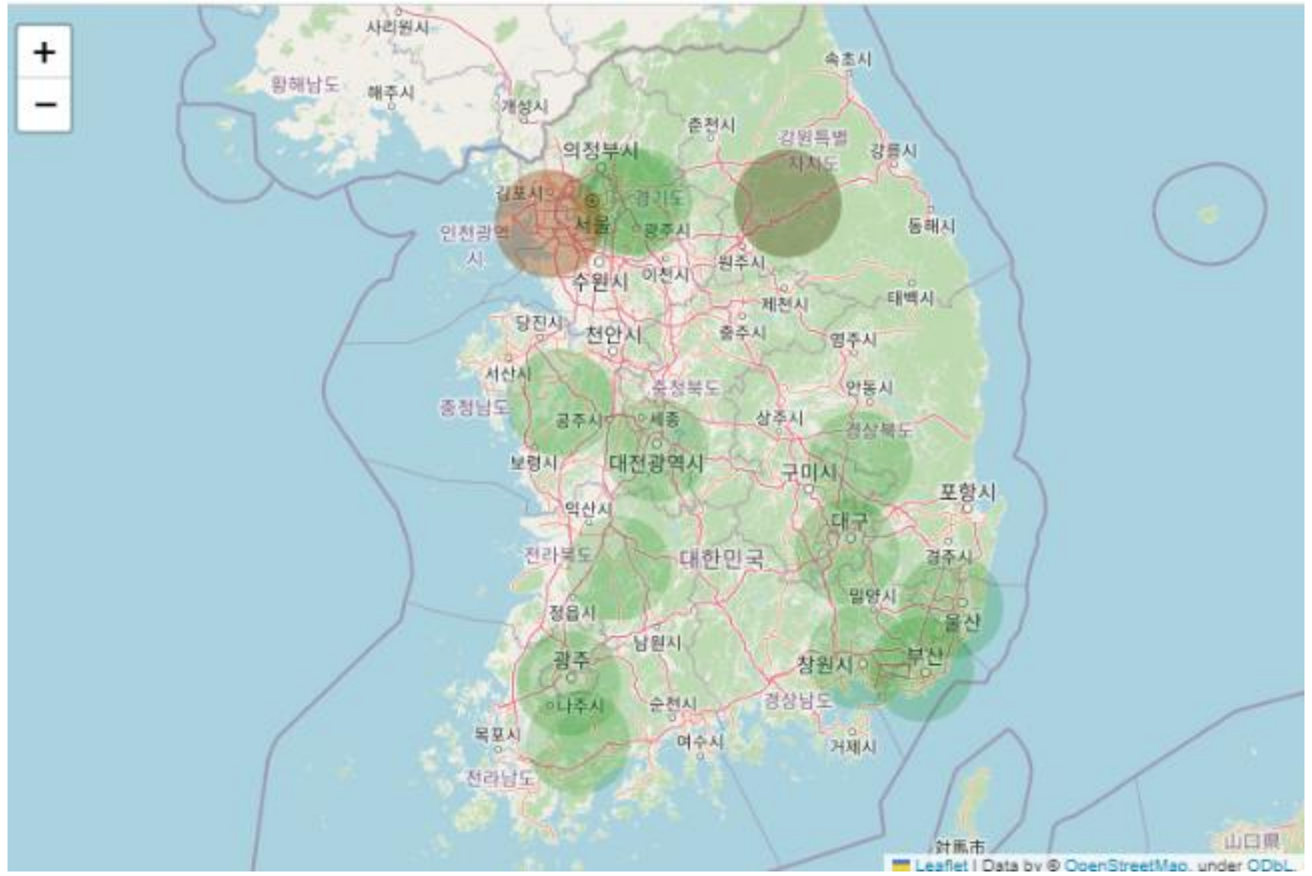
코드 설명 4-2

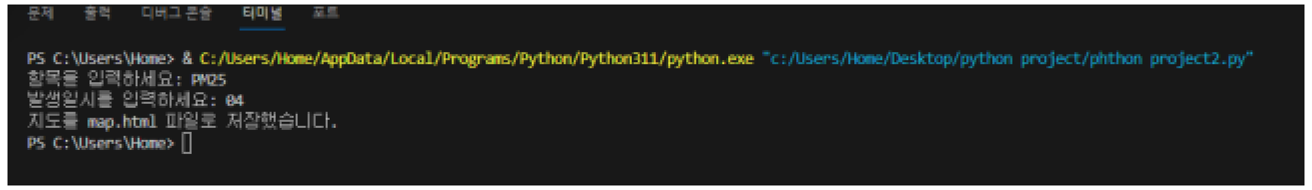
지도 만들기

- 미세먼지의 농도에 따라 색깔이 다름

```
elif dust == "PM10":
selected_rows = csv_test[(csv_test['항목_코드'] == 'PM10') & (csv_test['발생_일시'] == date)]
if not selected_rows.empty:
m = folium.Map(
location=[korea_weedo, korea_gyeongdo],
zoom_start=7,
width=750,
height=500
)
folium.Circle([seoul_weedo,seoul_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([incheon_weedo,incheon_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([gwangju_weedo,gwangju_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([daegu_weedo,daegu_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([ulsan_weedo,ulsan_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([daejeon_weedo,daejeon_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([busan_weedo,busan_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([gyeongi_weedo, gyeongi_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([gangwon_weedo, gangwon_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([choongnam_weedo, choongnam_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([gyeongbuk_weedo, gyeongbuk_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([gyeongnam_weedo, gyeongnam_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([jeonnam_weedo, jeonnam_gyeongdo], popup='PM25', radius=30000,fill_color='green',color=False).add_to(m)
folium.Circle([jeonbuk_weedo, jeonbuk_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
folium.Circle([jeju_weedo, jeju_gyeongdo], popup='PM25', radius=30000, fill_color='green',color=False).add_to(m)
for index, row in selected_rows.iterrows():
if row['발령농도'] >= 151 and row['발령농도']<200:
c = 'red'
elif row['발령농도'] >= 81 and row['발령농도']<151:
c = 'blue'
elif row['발령농도'] >= 31 and row['발령농도']<80:
c = 'green'
elif row['발령농도']>=200:
c = '#730602'
if row['지역명'] == '서울':
folium.Circle([seoul_weedo, seoul_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] == '인천':
folium.Circle([incheon_weedo,incheon_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] == '광주':
folium.Circle([gwangju_weedo,gwangju_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] == '대구':
folium.Circle([daegu_weedo,daegu_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] == '울산':
folium.Circle([ulsan_weedo,ulsan_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='대전':
folium.Circle([daejeon_weedo,daejeon_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='부산':
folium.Circle([busan_weedo,busan_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='경기':
folium.Circle([gyeongi_weedo, gyeongi_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='강원':
folium.Circle([gangwon_weedo, gangwon_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='충남':
folium.Circle([choongnam_weedo, choongnam_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='경북':
folium.Circle([gyeongbuk_weedo, gyeongbuk_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='경남':
folium.Circle([gyeongnam_weedo, gyeongnam_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='전남':
folium.Circle([jeonnam_weedo, jeonnam_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] == '전북':
folium.Circle([jeonbuk_weedo, jeonbuk_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
elif row['지역명'] =='제주':
folium.Circle([jeju_weedo, jeju_gyeongdo], popup='PM25', radius=30000, fill_color=c,color=False).add_to(m)
else:
break
else:
print("유효한 항목을 입력하세요 (PM25 또는 PM10).")
if m:
m.save('map.html')
print("지도를 map.html 파일로 저장했습니다.")
elif dust in ["PM25", "PM10"]:
print("선택된 행이 없어 지도를 생성하지 않았습니다.")
m.save('map.html')
webbrowser.open('map.html')
```

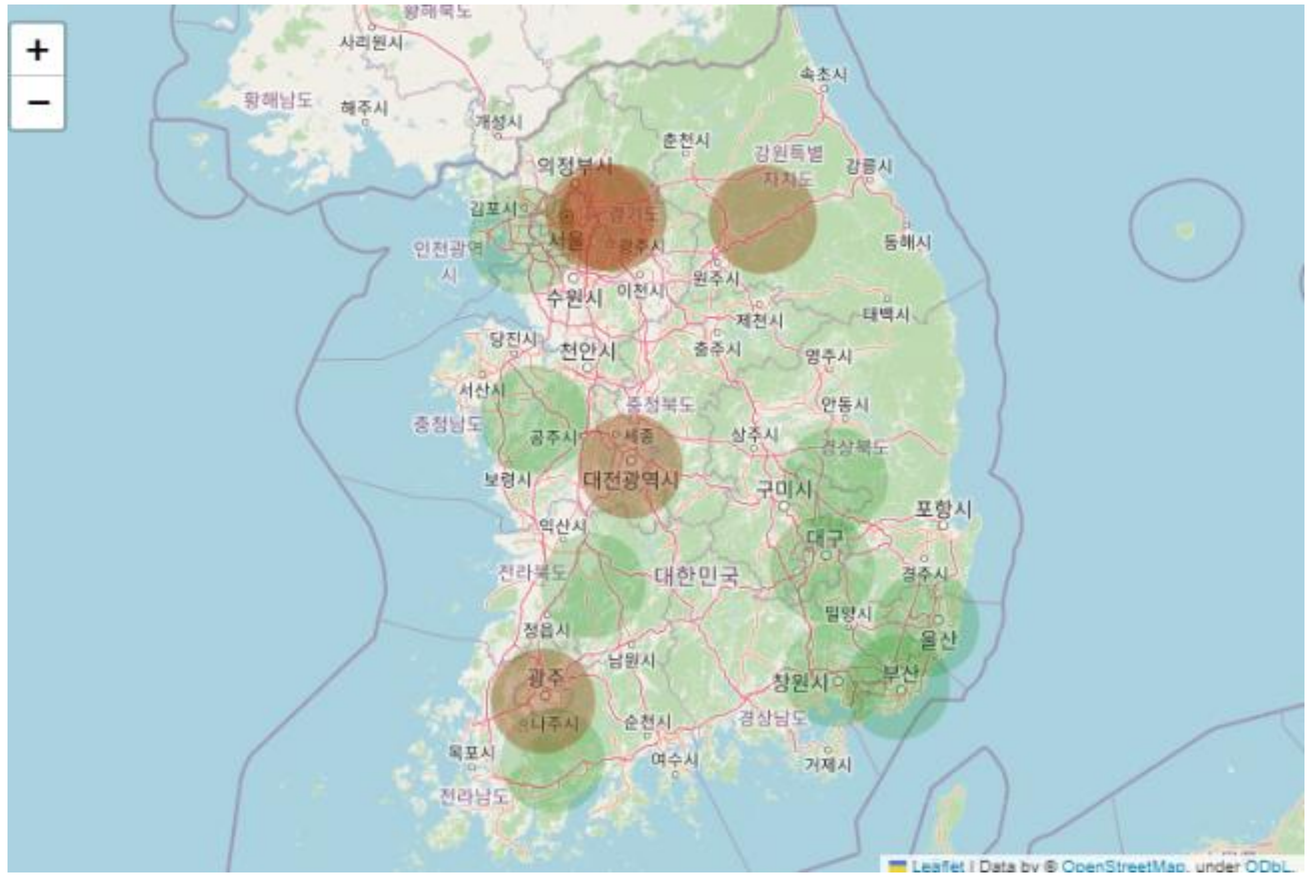
코드 실행

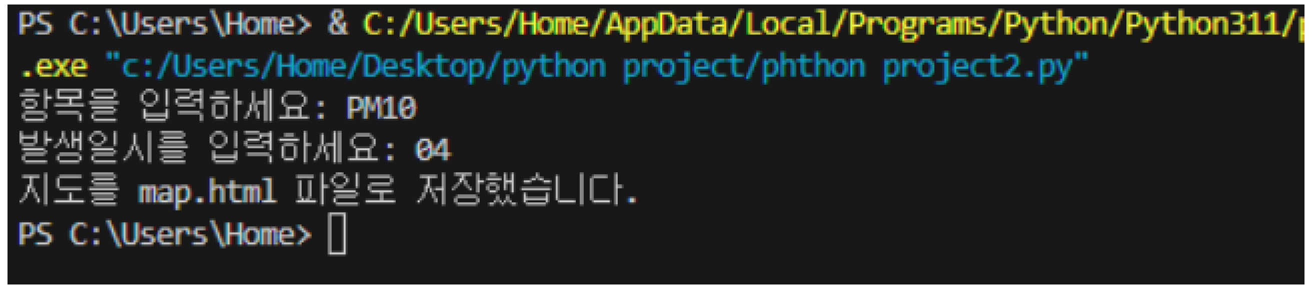




```
PS C:\Users\Home> & C:/Users/Home/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Home/Desktop/python project/phthon project2.py"
항목을 입력하세요: PM25
발생일시를 입력하세요: 04
지도를 map.html 파일로 저장했습니다.
PS C:\Users\Home> []
```

PM25 4월 사진과 코드





```
PS C:\Users\Home> & C:/Users/Home/AppData/Local/Programs/Python/Python311/
.exe "c:/Users/Home/Desktop/python project/phthon project2.py"
항목을 입력하세요: PM10
발생일시를 입력하세요: 04
지도를 map.html 파일로 저장했습니다.
PS C:\Users\Home> []
```

PM10 4월 사진과 코드