

서울 데이터 지도 시각화 수행하기

- url : <https://www.data.go.kr/data/15051860/fileData.do> (<https://www.data.go.kr/data/15051860/fileData.do>)
- 서울 toilet 위치 표시하기

In [2]:



```
import pandas as pd
import folium
```

01. 데이터 준비



```
dat = pd.read_csv("seoul_public_toilet.csv" , engine = 'python', encoding='EUCKR')
dat
```

Out[3]:

[illegible]

	POI ID	대명칭	중명칭	소명칭	중앙좌표X1	중앙좌표Y1	WSG84X좌표	WSG84Y좌표	등록일자	수정일자
4933	A10008	전국모범운전자연합양천지회	민간개방화장실	NaN	188119.607000	445886.552000	126.865623	37.512379	20110506	20110506
4934	A10009	세림빌딩	민간개방화장실	NaN	187717.295000	446300.534000	126.861066	37.516104	20110506	20110506
4935	A10010	신목동주유소	민간개방화장실	NaN	188601.435000	446685.734000	126.871060	37.519586	20110506	20110506
4936	A10011	등원주유소	민간개방화장실	NaN	188913.820000	448906.293000	126.874561	37.539597	20110506	20110506
4937	A10012	신양천주유소	민간개방화장실	NaN	187929.411000	446269.665000	126.863465	37.515829	20110506	20110506

4938 rows × 10 columns

In [4]:

```
dat.columns
```

Out[4]:

```
Index(['POI ID', '대명칭', '중명칭', '소명칭', '중앙좌표X1', '중앙좌표Y1', 'WSG84X좌표',
      'WSG84Y좌표', '등록일자', '수정일자'],
      dtype='object')
```

In [5]:



```
dat_tmp = dat.loc[ :, [ '대명칭', 'WSG84Y좌표', 'WSG84X좌표' ] ]
dat_tmp
```

Out[5]:

	대명칭	WSG84Y좌표	WSG84X좌표
0	우성스포츠센터	37.492386	126.909832
1	프레곤빌딩	37.486127	126.904575
2	하림빌딩	37.494283	127.016646
3	크레신타워	37.518195	127.018478
4	한주실업빌딩	37.513501	126.943960
...
4933	전국모범운전자연합양천지회	37.512379	126.865623
4934	세림빌딩	37.516104	126.861066
4935	신목동주유소	37.519586	126.871060
4936	등원주유소	37.539597	126.874561
4937	신양천주유소	37.515829	126.863465

4938 rows × 3 columns

02. 컬럼명 변경 및 위치 확인

In [6]:



```
# 중앙 위치 찾기
dat_tmp.columns = [ 'name', 'latitude', 'longitude' ]
print( dat_tmp.latitude.mean(), dat_tmp.longitude.mean() )
dat_tmp.head()
```

37.54478573028457 126.98281813570502

Out[6]:

	name	latitude	longitude
0	우성스포츠센터	37.492386	126.909832
1	프레곤빌딩	37.486127	126.904575
2	하림빌딩	37.494283	127.016646
3	크레신타워	37.518195	127.018478
4	한주실업빌딩	37.513501	126.943960

In [7]:



```
m = folium.Map(location = [37.5447, 126.9828], zoom_start=10)
m
```

Out[7]:

Make this Notebook Trusted to load map: File -> Trust Notebook

03. 일부 데이터 획득하기

In [8]:



```
### 200개 만 얻기
dat_200 = dat_tmp.iloc[0:200, :]
dat_200
```

Out[8]:

	name	latitude	longitude
0	우성스포츠센터	37.492386	126.909832
1	프레곤빌딩	37.486127	126.904575
2	하림빌딩	37.494283	127.016646
3	크레신타워	37.518195	127.018478
4	한주실업빌딩	37.513501	126.943960
...
195	나들이공중화장실	37.571836	126.844277
196	수궁동주민센터	37.493950	126.831472
197	꽃동네공중화장실	37.570206	126.847085
198	온수근린공원화장실	37.660927	127.064176
199	실로암안과병원	37.546186	126.861944

200 rows × 3 columns

In [9]:



```
size = dat_200.name.count()
print(size) # 크기

print(dat_200.columns)
```

```
200
Index(['name', 'latitude', 'longitude'], dtype='object')
```

In [10]:



```
import time
```

04. 데이터 지도위에 표시하기

In [22]:



```
# %%time
now = time.time()

m = folium.Map(location = [37.5447, 126.9828], zoom_start=13)

for i in range(size):
    folium.Marker(location= [ dat_200.loc[i, 'latitude'],
                              dat_200.loc[i, 'longitude'] ] ,
                  tooltip= dat_200.loc[i, 'name'] ,
                  icon = folium.Icon(color='blue', icon='info-sign')
    ).add_to(m)

work_time = time.time() - now
m.save("seoul_data.html")
print( work_time )
m
```

0.024593830108642578

Out[22]:

Make this Notebook Trusted to load map: File -> Trust Notebook

05. 데이터 지도위에 표시하기

- html 을 활용하여 표시하기

In [19]:



```
print( dat_200.columns)
des = dat_200['name']
dat_p1 = list( dat_200['latitude'] )
dat_p2 = list( dat_200['longitude'] )
```

Index(['name', 'latitude', 'longitude'], dtype='object')

In [20]:



```
dat_xy = list(zip(dat_p1, dat_p2))  
dat_xy
```

Out[20]:

```
((37.49238614627172, 126.90983237468619),  
(37.48612718078578, 126.90457509912622),  
(37.49428349959236, 127.01664600669828),  
(37.518195183969375, 127.01847813152573),  
(37.513500887007005, 126.94395989534347),  
(37.48202003578589, 126.94944132673054),  
(37.482241353193785, 126.94605749195469),  
(37.527955229634735, 127.1206261560178),  
(37.53803874941219, 127.12382208485907),  
(37.53505026778164, 127.13856695580252),  
(37.53467648037231, 127.13773224672887),  
(37.53822193303926, 127.12627274769704),  
(37.54619034813657, 127.12645076662243),  
(37.55164308479295, 127.12778197832401),  
(37.55008363501574, 127.12719633394077),  
(37.55026047361081, 127.17271500776569),  
(37.49101959700149, 126.9027716500649),
```


In [21]:



```
# %%time
now = time.time()

m = folium.Map(location = [37.492386, 126.909832], zoom_start=13)

for name, loc in zip(des, dat_xy):
    folium.Marker(location=loc,
                  tooltip= name ,
                  icon=folium.DivIcon(html=f"<div style='font-family: courier new; color: blue'>{name}</div>").add_to(m)

work_time = time.time() - now
m.save("seoul_data_c.html")
print( work_time )
m
```

0.014739036560058594

Out[21]:

Make this Notebook Trusted to load map: File -> Trust Notebook

plugins 이용

In [25]:



```
from folium import plugins
```

In [26]:



```
m = folium.Map(location=[37.492386, 126.909832], zoom_start=10)
plugins.MarkerCluster(dat_xy, popups = des).add_to(m)
m
```

Out[26]:

Make this Notebook Trusted to load map: File -> Trust Notebook

In []:

