

지도 활용하기

안화수

지도활용하기

❖ 지도 활용하기

- google API를 이용한 위치정보(위도, 경도) 구하기
- Folium 라이브러리를 이용한 지도 만들기

Google API

❖ Google API 활용하기

구글 지오코딩(geocoding)이란 특정 위치나 주소를 입력하면 위도와 경도 좌표 정보를 제공해 주는 서비스 이다.

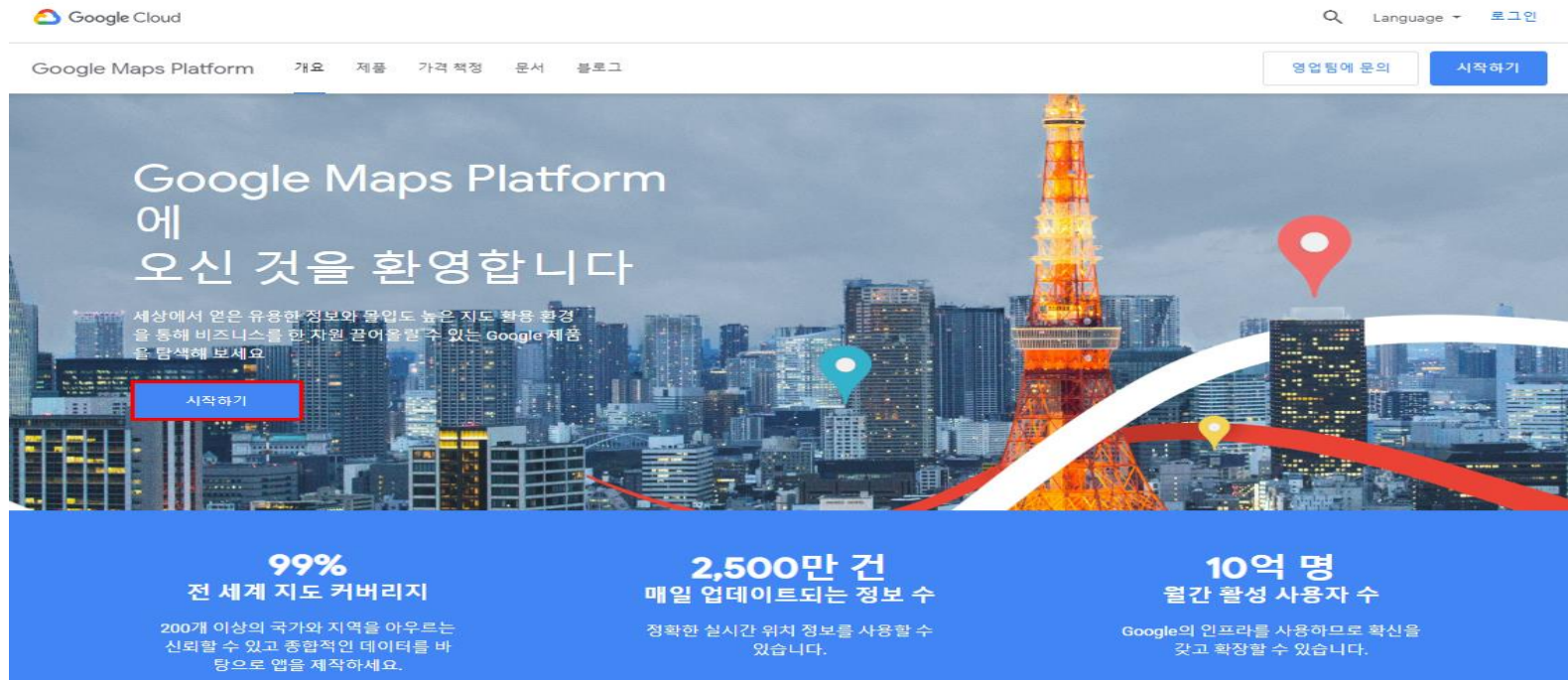
이 서비스를 이용하려면 사용자 인증 후에 google API 키를 발급 받아야 한다.
google API 키를 발급 받는 과정에 본인의 신용카드 정보를 등록해야 한다.

Google API

❖ Google APIkey 등록 절차

1. Google 사이트에 접속 후 로그인

<http://cloud.google.com/maps-platform>



The screenshot shows the Google Cloud Maps Platform homepage. At the top, there's a navigation bar with 'Google Cloud' logo, a search icon, 'Language' dropdown, and '로그인' (Login). Below this is a sub-header 'Google Maps Platform' with links for '개요' (Overview), '제품' (Products), '가격 책정' (Pricing), '문서' (Docs), and '블로그' (Blog). There are two buttons: '영업팀에 문의' (Contact Sales) and '시작하기' (Get Started). The main banner features a cityscape with the Tokyo Tower and the text 'Google Maps Platform 에 오신 것을 환영합니다' (Welcome to Google Maps Platform). Below this, a smaller text block says '세상에서 얻은 유용한 정보와 물어도 높은 지도 활용 환경을 통해 비즈니스를 한 차원 끌어올릴 수 있는 Google 제품을 탐색해 보세요' (Explore Google products that can help you take your business to the next level by leveraging the rich information and high map usage environment you can get from the world). A blue button labeled '시작하기' (Get Started) is highlighted with a red box. The bottom section has three statistics: '99% 전 세계 지도 커버리지' (99% global map coverage), '2,500만 건 매일 업데이트되는 정보 수' (25 million pieces of information updated daily), and '10억 명 월간 활성 사용자 수' (1 billion monthly active users).

Google Cloud

Google Maps Platform

개요 제품 가격 책정 문서 블로그

영업팀에 문의 시작하기

Google Maps Platform
에
오신 것을 환영합니다

세상에서 얻은 유용한 정보와 물어도 높은 지도 활용 환경을 통해 비즈니스를 한 차원 끌어올릴 수 있는 Google 제품을 탐색해 보세요

시작하기

99%
전 세계 지도 커버리지

200개 이상의 국가와 지역을 아우르는
신뢰할 수 있고 종합적인 데이터를 바
탕으로 앱을 제작하세요.

2,500만 건
매일 업데이트되는 정보 수

정확한 실시간 위치 정보를 사용할 수
있습니다.

10억 명
월간 활성 사용자 수

Google의 인프라를 사용하므로 확신을
갖고 확장할 수 있습니다.

Google API

❖ Google APIkey 등록 절차

2. Google 지도 플랫폼 사용 설정



Google 지도 플랫폼 사용 설정

API를 사용 설정하고 결제를 설정하기 위한 단계를 안내해 드립니다.

1. 아래에서 제품 선택
2. 프로젝트 선택
3. 결제 설정

☒ 지도

사용자에게 장소의 실제 모습을 보여주는 맞춤형 지도 환경을 구축하세요.

☒ 경로

사용자에게 출발지에서 목적지까지 가는 최적의 경로를 알려주세요.

☒ 지역 정보

풍부한 상세정보로 사용자의 장소 탐색을 도와주세요.

[취소](#) [계속](#)

[계속](#)

Google API

❖ Google APIkey 등록 절차

3. project 선택후 Next 버튼 클릭



Enable Google Maps Platform

To enable APIs or set up billing, we'll guide you through a few tasks:

1. Pick product(s) below
2. **Select a project**
3. Set up your billing

Projects allow you to use APIs, add collaborators, and manage permissions.

My Project ▼

CANCEL

NEXT

Google API

❖ Google APIkey 등록 절차

4. Google map Key 생성



Google Maps Platform 사용 설정

You're all set!

You're ready to start developing!

YOUR API KEY

AIzaSyC5S4zaVBJEG0Nh4ovde1Q6H-sFh16yRcg



To improve your app's security, restrict this key's usage in the [API Console](#).

DONE

Google API

❖ 구글 지오코딩 API를 이용한 위치정보 구하기 예제

구글 지오코딩 API를 이용하기 위해서는 구글 API키를 발급 받아야 되고, googlemaps 모듈을 설치해야 사용할 수 있다.

1. 구글 API 키 발급 받기
2. googlemaps 모듈 설치하기

```
pip install googlemaps
```


Google API

❖ 구글 지오코딩 API를 이용한 위치정보 구하기 예제

geocoding_api.py (1/2)

```
# -*- coding: utf-8 -*-
```

```
# google 지오코딩 API를 이용해서 위도, 경도 데이터 가져오기
```

```
import googlemaps  
import pandas as pd
```

```
# my_key = "본인이 발급받은 API키를 입력"  
my_key = "AlzaSyC5S4zaVBJEG0Nh4ovde1Q6H-sFh16yRcg"
```

```
# 구글맵스 객체 생성하기
```

```
maps = googlemaps.Client(key=my_key)
```

```
# my_key값 입력
```

```
lat = []
```

```
#위도
```

```
lng = []
```

```
#경도
```

```
# 장소(또는 주소) 리스트
```

```
places = ["서울시청", "국립국악원", "해운대해수욕장", '제주도', '독도']
```

Google API

❖ 구글 지오코딩 API를 이용한 위치정보 구하기 예제

geocoding_api.py (2/2)

```
i=0
for place in places:
    i = i + 1
    try:
        print(i, place)
        # 지오코딩 API 결과값 호출하여 geo_location 변수에 저장
        geo_location = maps.geocode(place)[0].get('geometry')
        lat.append(geo_location['location']['lat'])
        lng.append(geo_location['location']['lng'])
    except:
        lat.append("")
        lng.append("")
        print(i)

# 데이터프레임으로 변환하기
df = pd.DataFrame({'위도':lat, '경도':lng}, index=places)
print(df)
```

Google API

❖ 구글 지오코딩 API를 이용한 위치정보 구하기 예제

➤ 실행결과 화면

- 1 서울시청
- 2 국립국악원
- 3 해운대해수욕장
- 4 제주도
- 5 독도





	위도	경도
서울시청	37.566295	126.977945
국립국악원	37.477759	127.008304
해운대해수욕장	35.158698	129.160384
제주도	33.489011	126.498302
독도	37.242936	131.866842

Folium

- ❖ Folium 라이브러리를 이용한 지도 만들기
- Folium 라이브러리는 지도를 시각화할 때 유용한 도구이다.
- 세계 지도를 기본적으로 지원하고 다양한 스타일의 지도 이미지를 제공하고 있다.
- Folium은 웹 기반으로 지도를 만들기 때문에, PyCharm이나 스파이더(Spyder) 같은 IDE 프로그램에서 실행해도 지도가 표시되지 않는다.
지도를 보려면 지도 객체를 `save()` 함수로 HTML 파일로 저장하고, 웹브라우저로 저장된 HTML파일을 열어서 확인해야 한다.

Jupyter Notebook 등 웹 기반 IDE 프로그램에서는 지도를 바로 확인 할 수 있다.

Folium



❖ Folium 라이브러리

➤ Folium 라이브러리 설치하기

```
pip install folium
```

Folium

❖ 서울 지도 만들기

folium_maps01.py

```
# -*- coding: utf-8 -*-
```

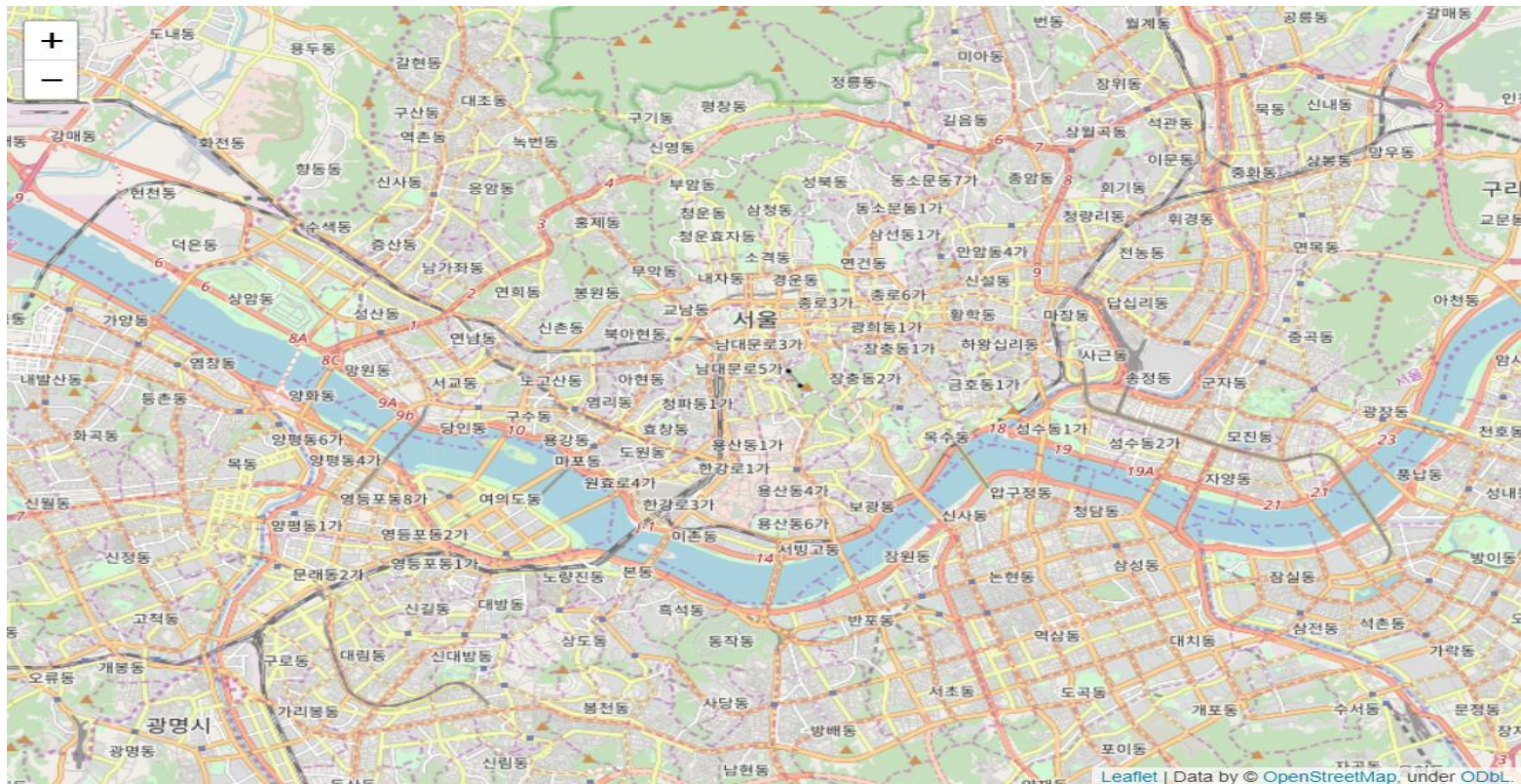
```
# 라이브러리 불러오기  
import folium
```

```
# 서울 지도 만들기  
seoul_map = folium.Map(location=[37.55, 126.98], zoom_start=12)
```

```
# 지도를 HTML 파일로 저장하기  
seoul_map.save('./seoul.html')
```

❖ 서울 지도 만들기

<실행결과>



Folium

- ❖ 서울 지도 만들기

- 지도 스타일 적용하기

Map() 함수에 `tiles` 옵션을 적용하면 지도 스타일을 변경할 수 있다.

```
seoul_map2 = folium.Map(location=[37.55,126.98], tiles='Stamen Terrain',  
                        zoom_start=12)
```

```
seoul_map3 = folium.Map(location=[37.55,126.98], tiles='Stamen Toner',  
                        zoom_start=15)
```


Folium

❖ 서울 지도 만들기 : 지도 스타일 적용하기

folium_maps02.py

```
import folium
```

```
# 서울 지도 만들기
```

```
seoul_map2 = folium.Map(location=[37.55,126.98], tiles='Stamen Terrain',  
                        zoom_start=12)
```

```
seoul_map3 = folium.Map(location=[37.55,126.98], tiles='Stamen Toner',  
                        zoom_start=15)
```

```
# 지도를 HTML 파일로 저장하기
```

```
seoul_map2.save('seoul2.html')
```

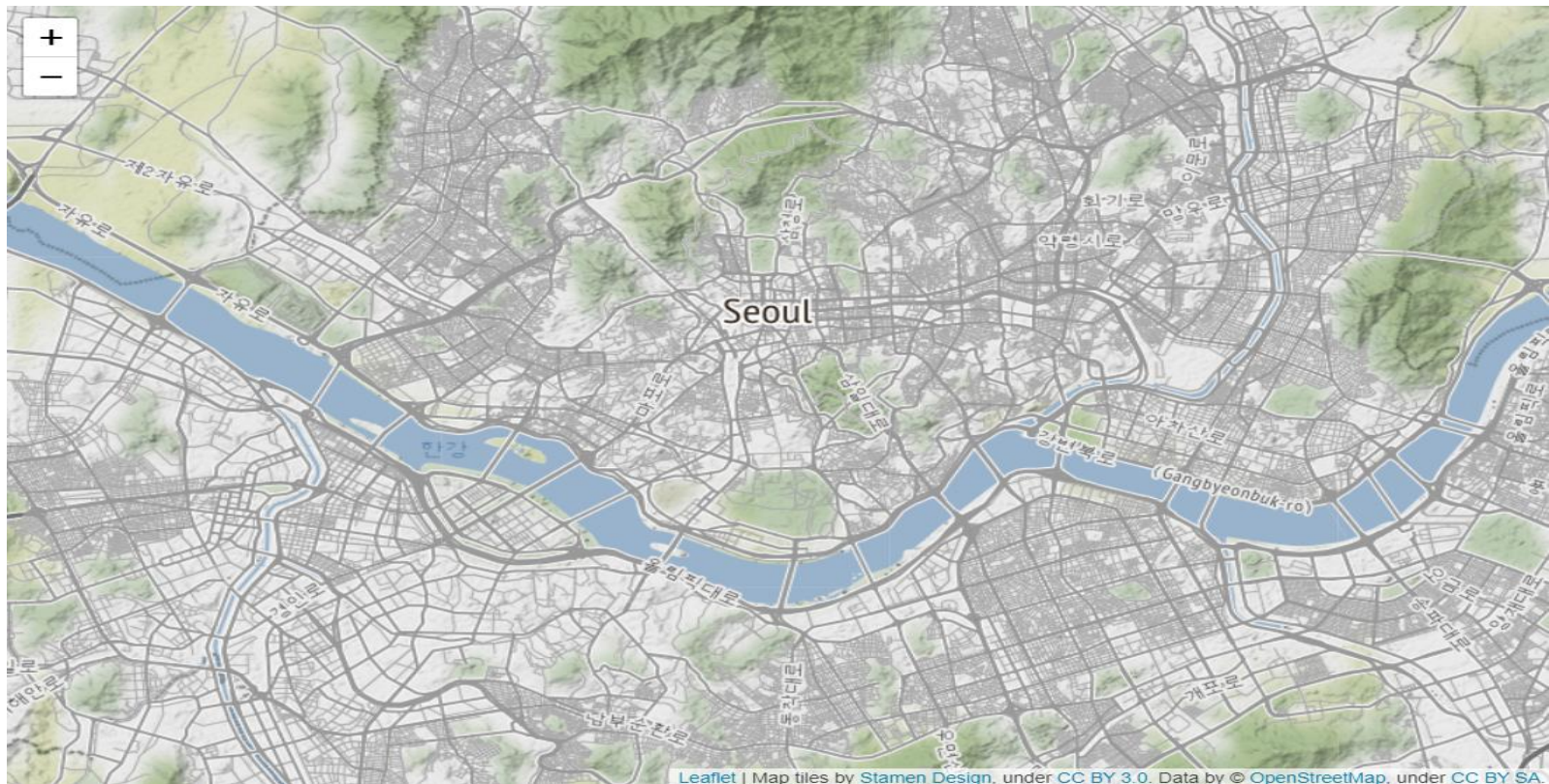
```
seoul_map3.save('seoul3.html')
```

Folium

❖ 서울 지도 만들기

'Stamen Terrain' 맵은 산악 지형 등의 지형이 보다 선명하게 보여준다.

<실행결과> tiles='Stamen Terrain' 적용 (zoom_start=12)



Folium

❖ 서울 지도 만들기

'Stamen Toner' 맵은 흑백 스타일로 도로망을 강조해서 보여준다.

<실행결과> tiles= 'Stamen Toner' 적용 (zoom_start=15)



Folium

❖ 서울 지도 만들기 : 지도에 마커 표시하기

서울 시내 주요 대학교의 위치 데이터 파일(서울지역 대학교 위치.xlsx)을 읽어와서, 해당 대학교에 마커를 표시하고, 해당 대학교의 위치의 마커를 클릭했을 때 팝업 메시지가 나타난다.

➤ 라이브러리 설치하기

```
pip install xlrd
```

```
pip install pandas
```

```
pip install folium
```

➤ 지도에 마커 표시하기

```
for name, lat, lng in zip(df.collage, df.위도, df.경도):  
    folium.Marker([lat, lng], popup=name).add_to(seoul_map)
```

Folium

❖ 서울 지도 만들기 : 지도에 마커 표시하기

folium_marker01.py (1/2)

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

```
# 대학교 리스트를 데이터프레임 변환 (에러 발생시 xlrd 모듈 설치)
# df = pd.read_excel('서울지역 대학교 위치.xlsx', sheet_name='Sheet1')
df = pd.read_excel('서울지역 대학교 위치.xlsx')
print(df.head())                # 앞에서 5개의 데이터를 불러옴
```

```
# 대학교명이 저장된 컬럼명이 없으므로 collage로 추가
df.columns = ['collage', '위도', '경도']
# print(df)
```

```
# 서울 지도 만들기
seoul_map = folium.Map(location=[37.55,126.98], tiles='Stamen Terrain',
                        zoom_start=12)
```

Folium

❖ 서울 지도 만들기 : 지도에 마커 표시하기

folium_marker01.py (2/2)

```
# 대학교 위치정보를 Marker로 표시
for name, lat, lng in zip(df.collage, df.위도, df.경도):
    folium.Marker([lat, lng], popup=name).add_to(seoul_map)
# for index, row in df.iterrows():
#     folium.Marker([row['위도'], row['경도']], popup=row['collage']).add_to(seoul_map)

# 지도를 HTML 파일로 저장하기
seoul_collage = 'seoul_colleges.html'
seoul_map.save(seoul_collage)

# 웹브라우저로 지도 출력
webbrowser.open(seoul_collage)
```


Folium

❖ 서울 지도 만들기 : 지도에 마커 표시하기

<실행결과>





지도에 원형 마커 표시하기

지도에 원형 마커 표시하기

지도에 원형 마커 표시하기

지도에 원형 마커 표시하기

Folium

❖ 서울 지도 만들기 : 지도에 원형 마커 표시하기

folium_circlemarker.py (1/2)

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

```
# 대학교 리스트를 데이터프레임 변환
df = pd.read_excel('서울지역 대학교 위치.xlsx')
```

```
# 대학교명이 저장된 컬럼명이 없으므로 collage로 추가
df.columns = ['collage', '위도', '경도']
```

```
# 서울 지도 만들기
seoul_map = folium.Map(location=[37.55,126.98], tiles='Stamen Terrain',
                        zoom_start=12)
```

Folium

❖ 서울 지도 만들기 : 지도에 원형 마커 표시하기

folium_circlemarker.py (2/2)

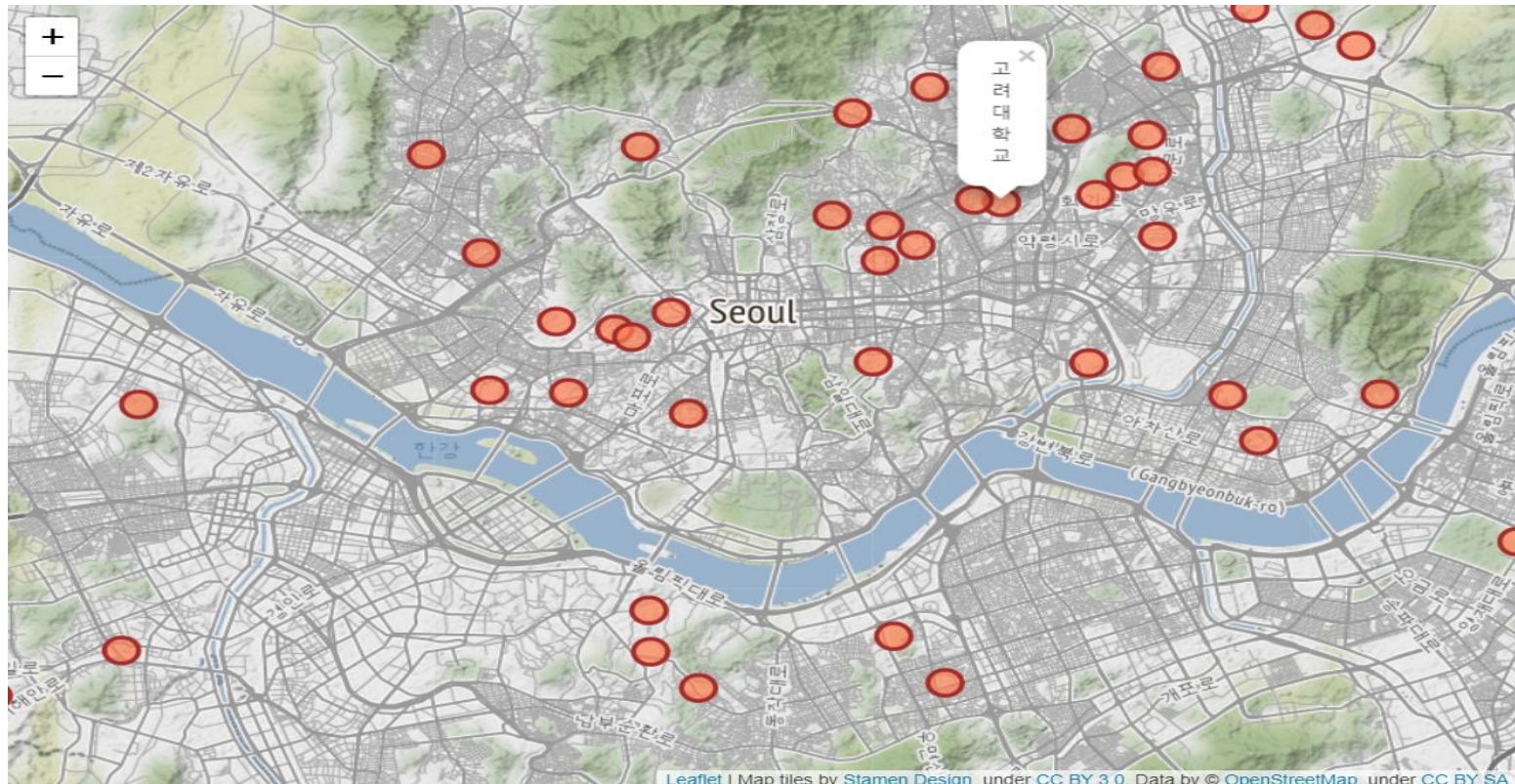
```
# 대학교 위치정보를 CircleMarker로 표시
for name, lat, lng in zip(df.collage, df.위도, df.경도):
    folium.CircleMarker([lat, lng],
                        radius=10,                # 원의 반지름
                        color='brown',            # 원의 둘레 색상
                        fill=True,                 # 원을 채우는 색
                        fill_color='coral',        # 투명도
                        fill_opacity=0.7,
                        popup=name
    ).add_to(seoul_map)

# 지도를 HTML 파일로 저장하기
seoul_map.save('seoul_colleges2.html')
```

Folium

❖ 서울 지도 만들기 : 지도에 원형 마커 표시하기

<실행결과>



❖ 경기도 인구분포 단계구분도(Choropleth Map)

행정구역과 같이 지도 상의 어떤 경계에 둘러싸인 영역에 색을 칠하거나 음영 등으로 정보를 나타내는 시각화 방법이다.

경기도 지역의 시군구별 인구 변화 데이터(2007 ~ 2017년), 경기도 행정구역 경계 지리 정보를 이용하여 인구수에 따른 단계구분도로 시각화 한다.

➤ 단계 구분도

```
folium.Choropleth(geo_data=geo_data,                                # 지도 경계
                  data = df[year],                                # 표시하려는 데이터
                  columns = [df.index, df[year]],                  # 열 지정
                  fill_color='YlOrRd', fill_opacity=0.7, line_opacity=0.3,
                  threshold_scale=[10000, 100000, 300000, 500000, 700000],
                  key_on='feature.properties.name',
                  ).add_to(g_map)
```

Folium

❖ 경기도 인구분포 단계구분도(Choropleth Map)

folium_choropleth.py (1/2)

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

```
# 경기도 인구변화 데이터를 불러와서 데이터프레임으로 변환
```

```
file_path = '경기도인구데이터.xlsx'
```

```
df = pd.read_excel(file_path, index_col='구분')
```

```
# print(df)
```

```
print(df.columns)
```

```
# 컬럼이 정수형
```

```
print(df.columns.map(str))
```

```
# 컬럼을 문자형으로 변환 cf.astype(str)
```

```
df.columns = df.columns.map(str)
```

```
# 컬럼을 정수형에서 문자형으로 변환 2017 -> '2017'
```

```
# 경기도 시군구 경계 정보를 가진 geo-json 파일 불러오기
```

```
geo_path = '경기도행정구역경계.json'
```

```
geo_data = json.load(open(geo_path, encoding='utf-8'))
```

Folium

❖ 경기도 인구분포 단계구분도(Choropleth Map)

folium_choropleth.py (2/2)

경기도 지도 만들기

```
g_map = folium.Map(location=[37.5502,126.982],  
                    tiles='Stamen Terrain', zoom_start=9)
```

출력할 연도 선택 (2007 ~ 2017년 중에서 선택)
year = '2007'

Choropleth 함수로 단계구분도 표시하기

```
folium.Choropleth(geo_data=geo_data,  
                  data = df[year],  
                  columns = [df.index, df[year]],  
                  fill_color='YlOrRd', fill_opacity=0.7, line_opacity=0.3,  
                  threshold_scale=[10000, 100000, 300000, 500000, 700000],  
                  key_on='feature.properties.name',  
                  ).add_to(g_map)
```

지도 경계
표시하려는 데이터
열 지정

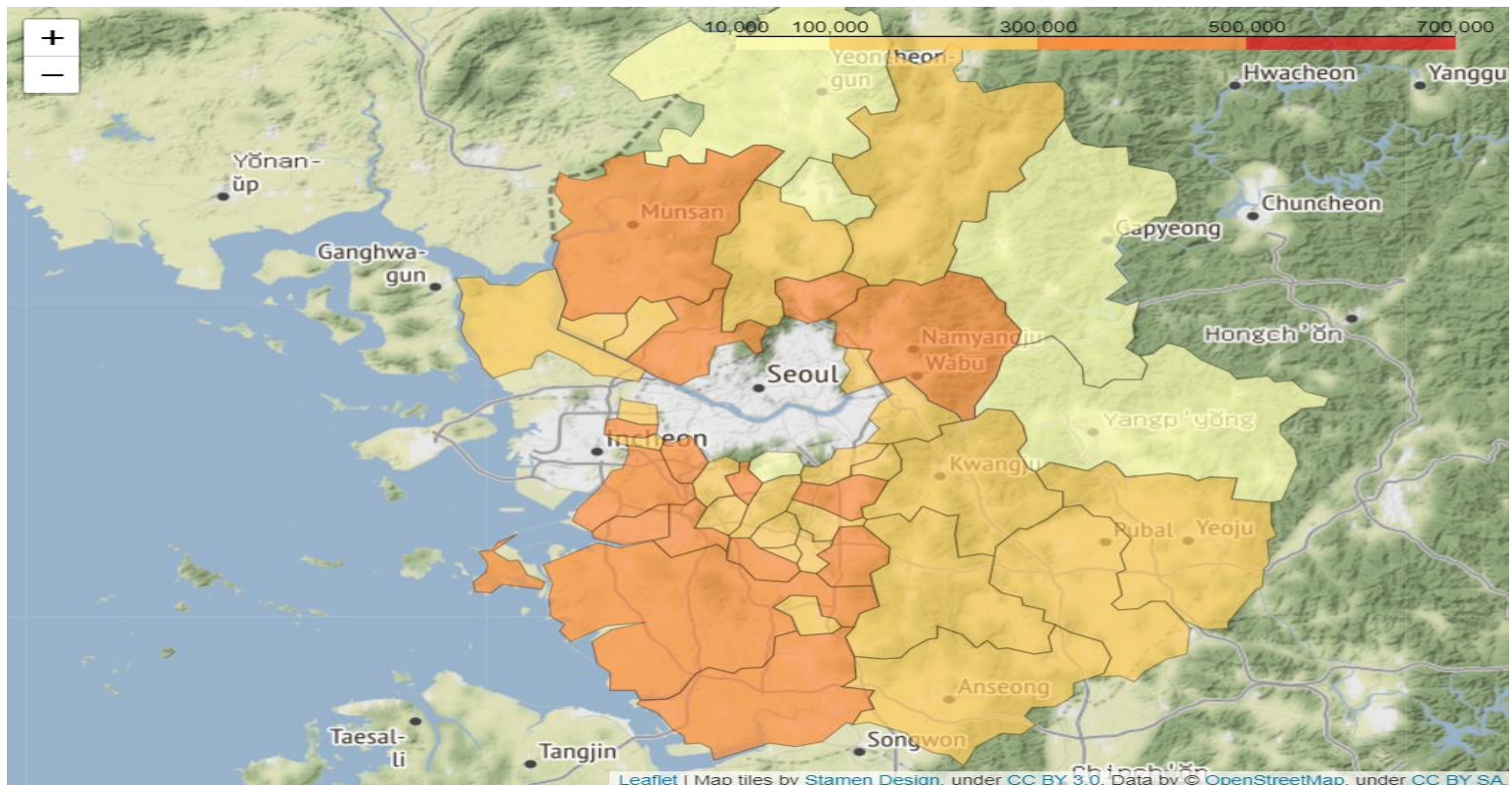
지도를 HTML 파일로 저장하기

```
g_map.save('gyonggi_population_' + year + '.html')
```


Folium

❖ 경기도 인구분포 단계구분도(Choropleth Map)

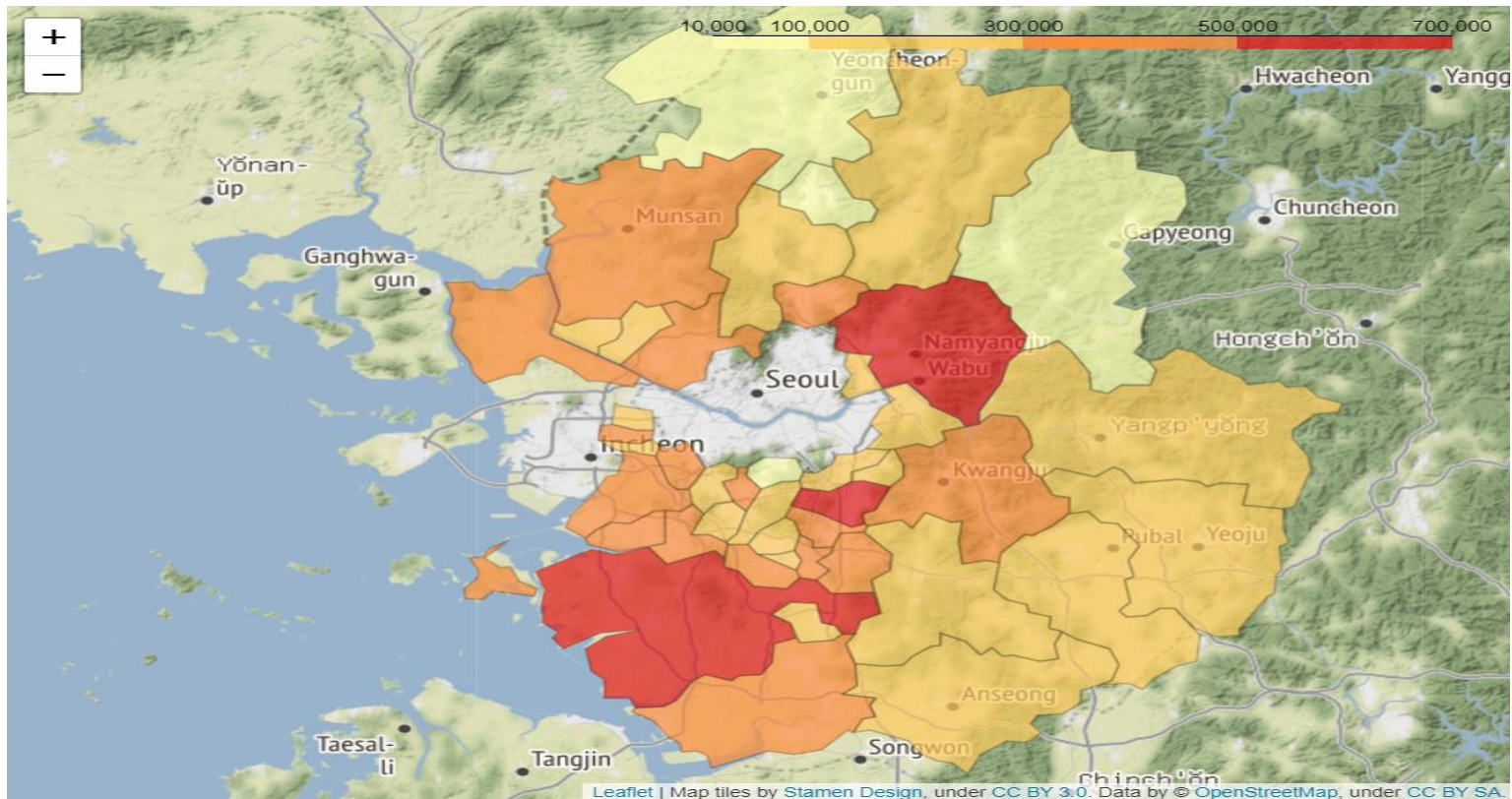
<실행결과> 2007년도 경기도 인구분포 (gyonggi_population_2007.html)



Folium

❖ 경기도 인구분포 단계구분도(Choropleth Map)

<실행결과> 2017년도 경기도 인구분포 (gyonggi_population_2017.html)



❖ 경기도 인구분포 단계구분도(Choropleth Map)

연도 year = '2007' , year = '2017' 을 바꿔가면서 경기도 인구분포를 단계구분도로 표현하였다.

2007년에 비해서 2017년도에는 남양주, 분당, 화성(동탄) 지역의 신도시 개발과 인구 유입으로 인구가 집중되는 현상이 심화된 것을 볼 수 있다.