

지도 시각화

Folium 라이브러리 - 지도 활용

- 라이브러리 설치 필요

```
conda install -c conda-forge folium
```

```
pip install folium
```

```
import folium
```

```
# 서울 지도 만들기
```

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

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

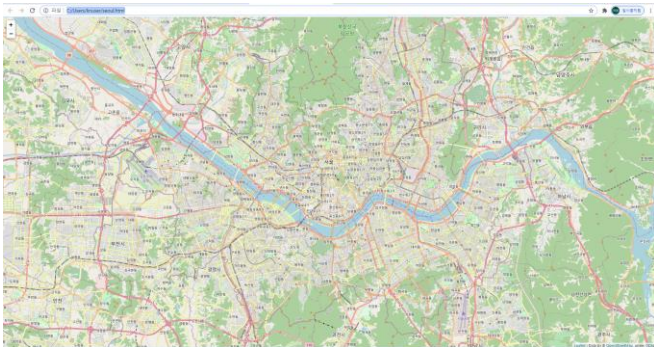
```
seoul_map3 = folium.Map(location=[37.55,126.98],  
                        tiles='Stamen Toner', zoom_start=15)
```

```
# 지도를 HTML 파일로 저장하기
```

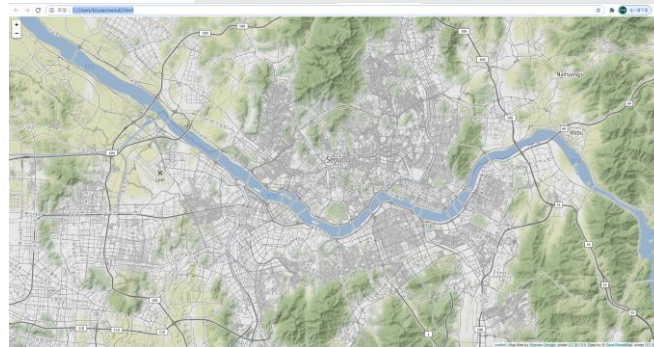
```
seoul_map.save('./seoul.html')
```

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

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



< 기본 설정 >



< Stamen Terrain >



< Stamen Toner >

Folium 라이브러리 - 지도 활용

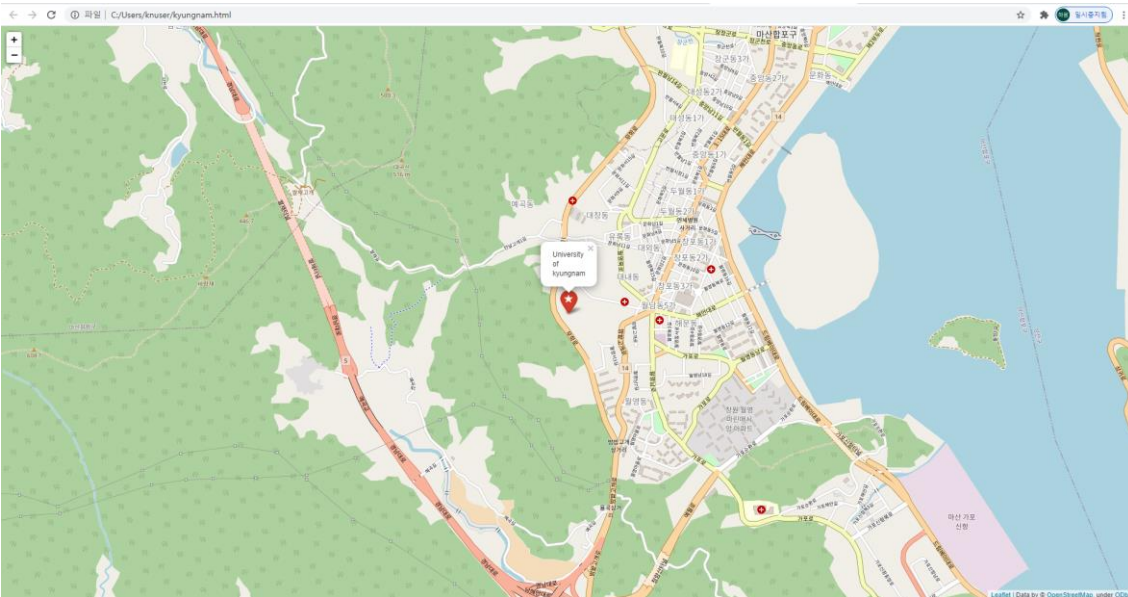
- map()함수 속성

```
folium.Map(  
    location=None,  
    width='100%',  
    height='100%',  
    left='0%',  
    top='0%',  
    position='relative',  
    tiles='OpenStreetMap',  
    attr=None,  
    min_zoom=0,  
    max_zoom=18,  
    zoom_start=10,  
    min_lat=-90,  
    max_lat=90,  
    min_lon=-180,  
    max_lon=180,  
    max_bounds=False,  
    crs='EPSG3857',  
    control_scale=False,  
    prefer_canvas=False,  
    no_touch=False,  
    disable_3d=False,  
    png_enabled=False,  
    zoom_control=True,  
    **kwargs,  
)
```

- Open street map (기본 값입니다.)
- Map Quest Open
- MapQuest Open Aerial
- Mapbox Bright
- Mapbox Control Room
- Stamenterrain
- Stamentoner
- Stamenwatercolor
- cartodbpositron
- cartodbdark_matter

Folium 라이브러리 - 지도 활용

- 지도에 마커 표시하기



```
import folium
```

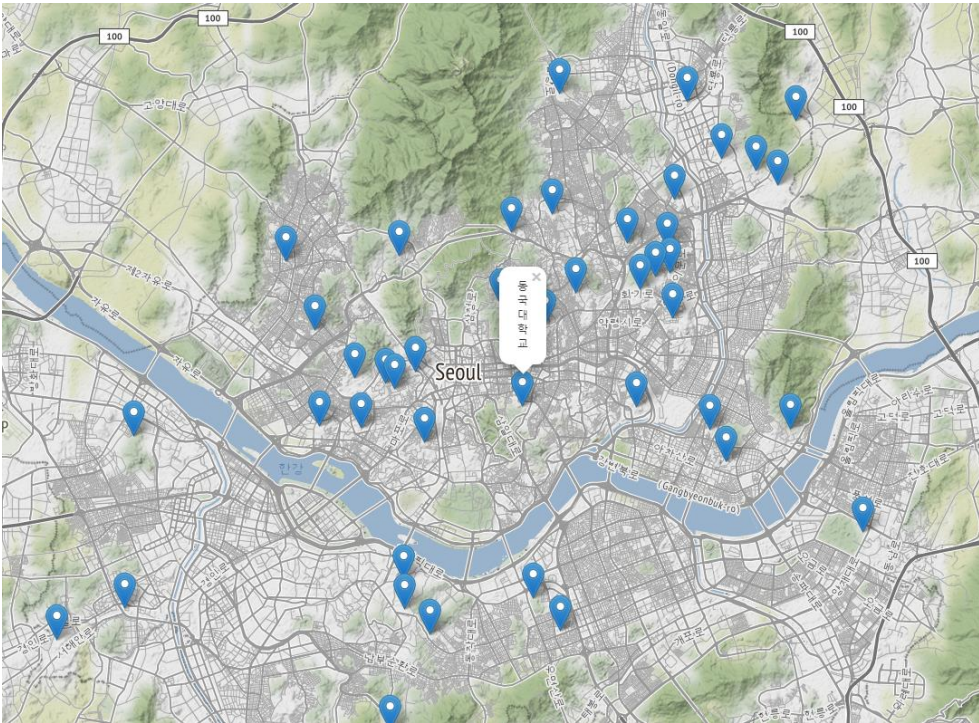
```
m = folium.Map(  
    location=[35.1804486,128.5523014],  
    zoom_start=15  
)
```

```
folium.Marker(  
    location=[35.1804486,128.5523014],  
    popup='University of kyungnam',  
    icon=folium.Icon(color='red',icon='star')  
) .add_to(m)
```

```
m.save('kyungnam.html')
```


Folium 라이브러리 - 지도 활용

- 지도에 마커 표시하기



```
df = pd.read_excel('data/서울지역 대학교 위치.xlsx')
df.set_index('학교명', inplace=True)
```

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

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

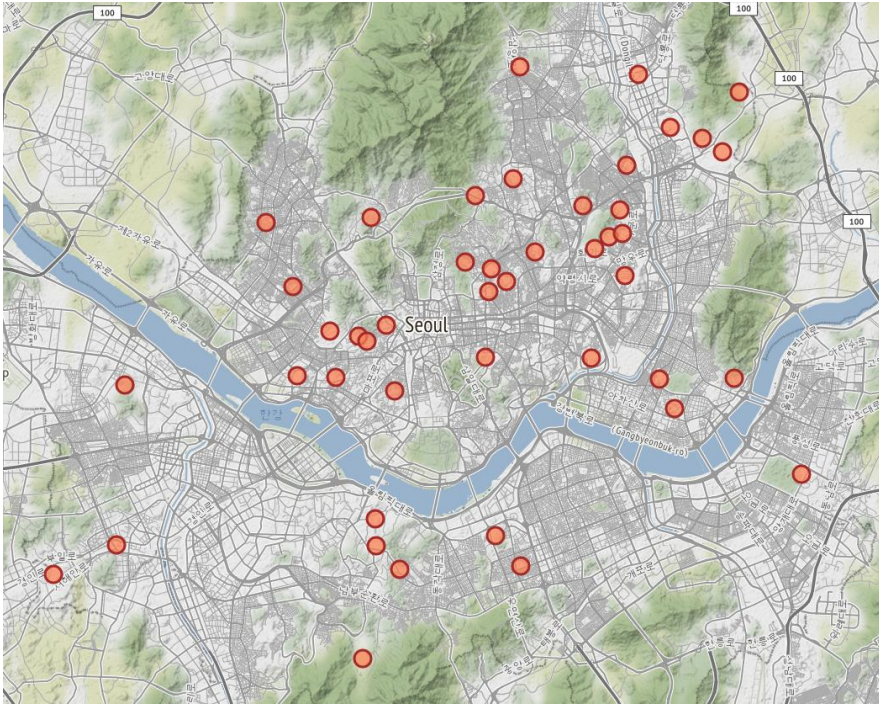
```
# 지도보기
seoul_map
```

한글 깨짐

```
!pip install git+https://github.com/python-visualization/branca.git@master
```

Folium 라이브러리 - 지도 활용

- 지도에 마커 표시하기



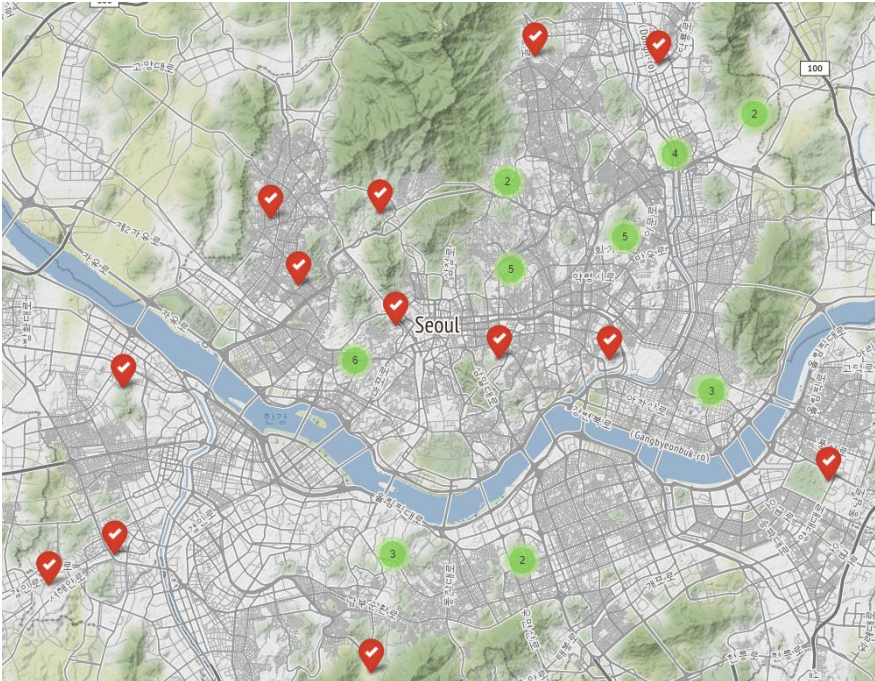
```
# 서울 지도 만들기
seoul_map = folium.Map(location=[37.55,126.98],
                        tiles='Stamen Terrain',
                        zoom_start=12)
```

```
# 대학교 위치정보를 CircleMarker로 표시
for name, lat, lng in zip(df.index, 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)
```

```
# 지도보기
seoul_map
```


Folium 라이브러리 - 지도 활용

- 지도에 마커 표시하기(MarkerCluster)
 - 가까운 거리의 Marker들을 군집시켜서 해당 건수를 표현
 - 특정 클러스터를 클릭하면 해당 위치 확대



```
from folium.plugins import MarkerCluster
```

```
df = pd.read_excel('data/서울지역 대학교 위치.xlsx')  
df.set_index('학교명', inplace=True)
```

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

```
marker_cluster = MarkerCluster().add_to(seoul_map)
```

```
# 대학교 위치정보를 Marker로 표시  
for name, lat, lng in zip(df.index, df.위도, df.경도):  
    folium.Marker([lat, lng],  
                  icon=folium.Icon(color='red', icon='ok'),  
                  popup=name).add_to(marker_cluster)
```

```
# 지도보기  
marker_cluster
```

Folium 라이브러리 - 지도 활용

■ Circle Marker

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

```
folium.CircleMarker(  
    location = df.loc[0, ['위도', '경도']],  
    tooltip = df.loc[0, '학교명'],  
    radius = 100  
)
```

seoul_map



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

```
folium.CircleMarker(  
    location = df.loc[0, ['위도', '경도']],  
    tooltip = df.loc[0, '학교명'],  
    weight = 10, # 추가한 부분
```

```
    radius = 100  
)
```

seoul_map



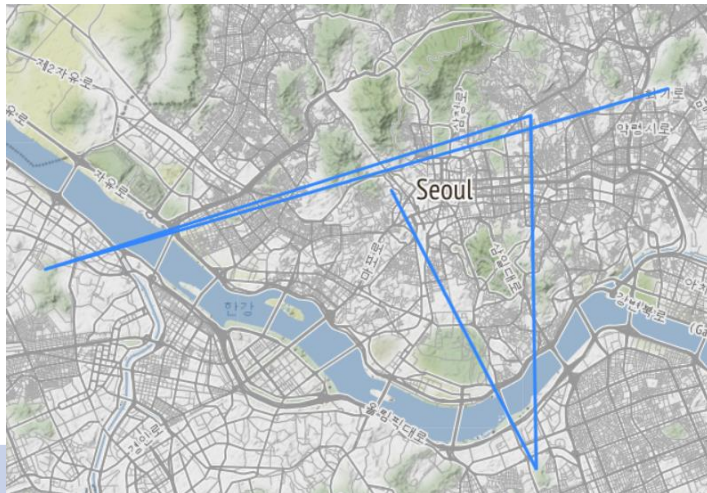
Folium 라이브러리 - 지도 활용

■ PolyLine

```
lines = df[['위도', '경도']].values[:5].tolist()
seoul_map = folium.Map(location=[37.55,126.98],
                        tiles='Stamen Terrain', zoom_start=12)
```

```
folium.PolyLine(
    locations = lines,
    tooltip = 'PolyLine'
).add_to(seoul_map)
```

seoul_map

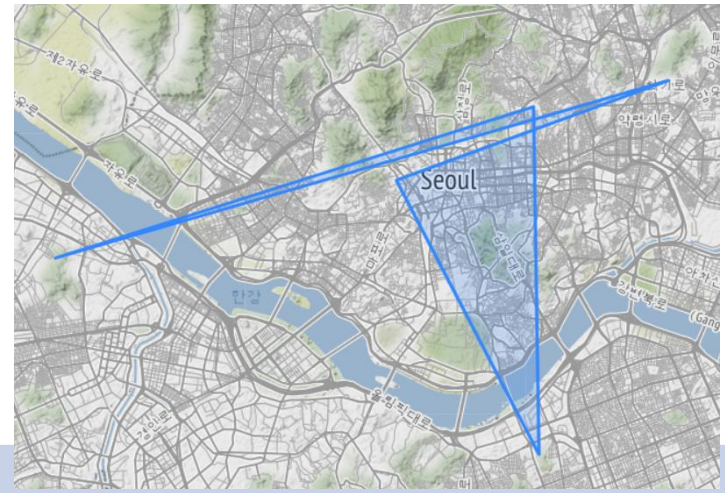


■ Polygon

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

```
folium.Polygon(
    locations = lines,
    fill=True,
    tooltip = 'Polygon'
).add_to(seoul_map)
```

seoul_map



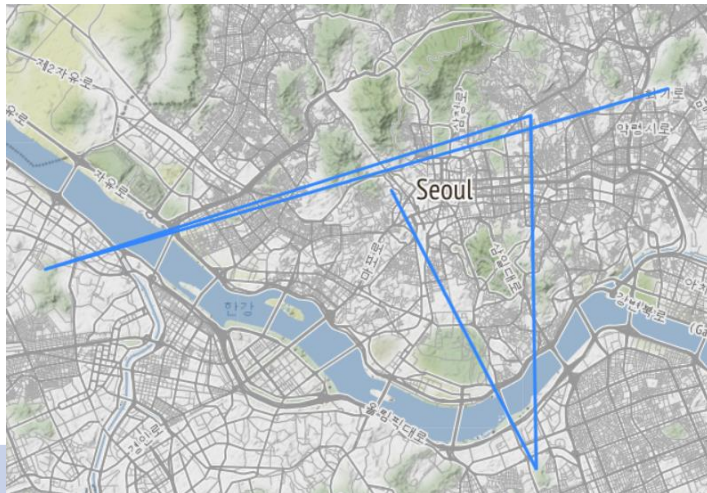
Folium 라이브러리 - 지도 활용

■ Rectangle

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

```
folium.Rectangle(  
    bounds = lines,  
    tooltip = 'Rectangle'  
) .add_to(seoul_map)
```

```
seoul_map
```

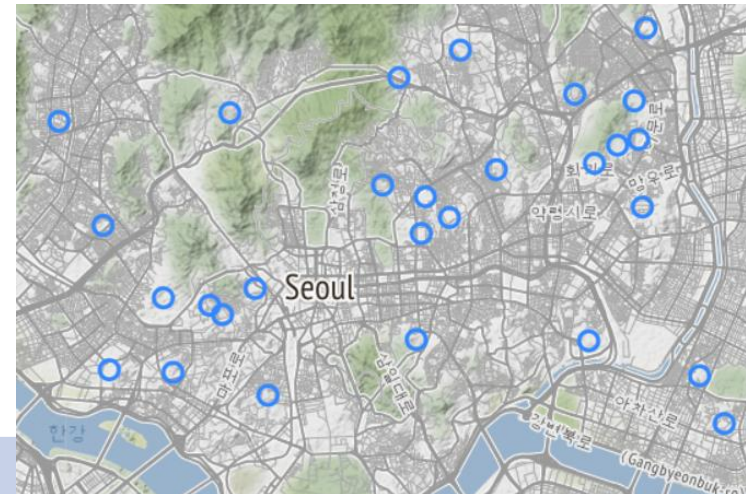


■ Circle

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

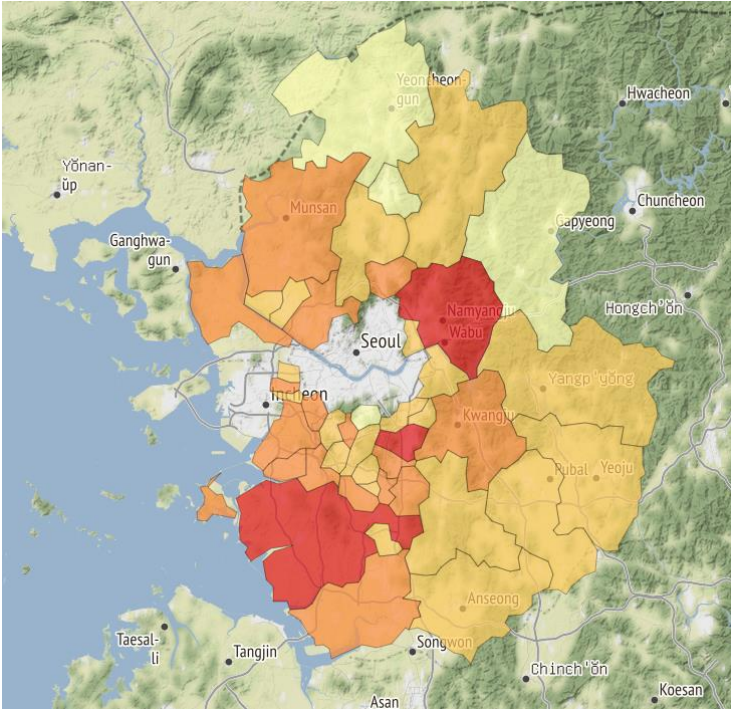
```
for i in df.index[:1000]:  
    folium.Circle(  
        location = df.loc[i, ['위도', '경도']],  
        tooltip = df.loc[i, '학교명'],  
        radius = 200  
    ) .add_to(seoul_map)
```

```
seoul_map
```



Folium 라이브러리 - 지도 활용

- 영역 구분(choropleth map) 표시



```
file_path = 'data/경기도인구데이터.xlsx'
df = pd.read_excel(file_path, index_col='구분')
df.columns = df.columns.map(str)

geo_path = 'data/경기도행정구역경계.json'
geo_data = json.load(open(geo_path, encoding='utf-8'))

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

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

# Choropleth 클래스로 단계구분도 표시하기
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',
                  legend_name='인구수'
                  ).add_to(g_map)

# 지도를 HTML 파일로 저장하기
g_map.save('./gyonggi_population_' + year + '.html')
```

Folium 라이브러리 - 지도 활용

- Choropleth map 주요 속성

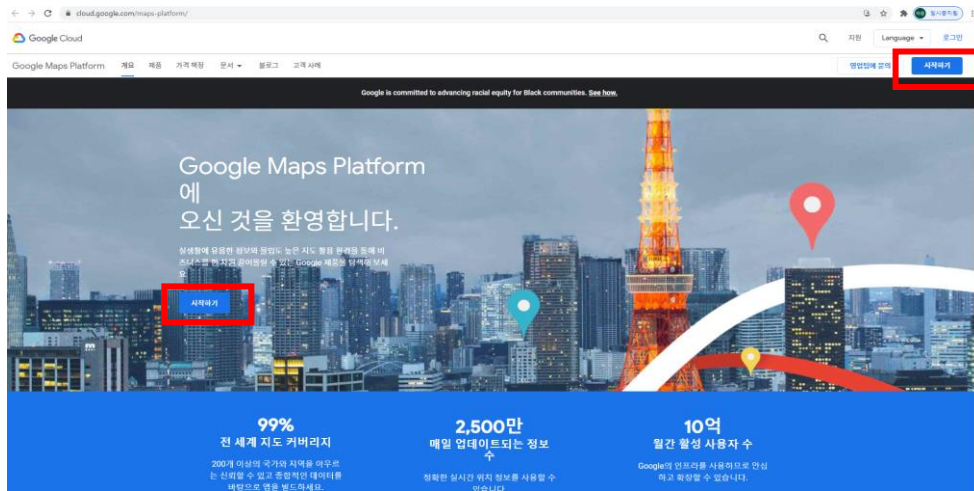
```
folium.Choropleth(  
    geo_data = "지도 데이터 파일 경로 (.geojson, geopandas.DataFrame)"  
    data = "시각화 하고자 하는 데이터파일. (pandas.DataFrame)"  
    columns = (지도 데이터와 매핑할 값, 시각화 하고자하는 변수),  
    key_on = "feature.데이터 파일과 매핑할 값",  
    fill_color = "시각화에 쓰일 색상",  
    legend_name = "칼라 범주 이름"  
).add_to(m)
```


Google map API 설정

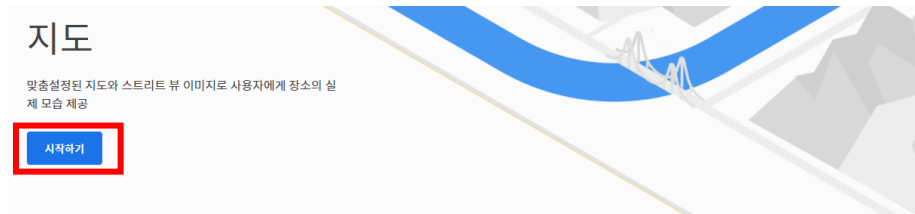
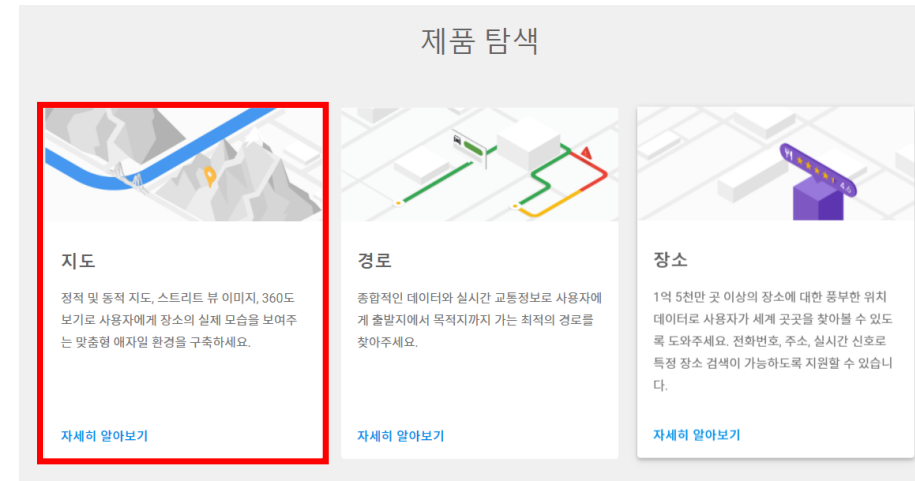
■ Google Maps API 접속

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

■ 시작하기 클릭



■ 제품 선택



99%
전 세계 지도 커버리지

200개 이상의 국가와 지역을 아우르는 신뢰할 수 있고 종합적인 데이터를 바탕으로 앱을 빌드하세요.

2,500만
매일 업데이트되는 정보 수


정확한 실시간 위치 정보를 사용할 수 있습니다.

10억
월간 활성 사용자 수

Google의 인프라를 사용하므로 혁신을 갖고 확장할 수 있습니다.

Google map API 설정

로그인


로그인
Google 클라우드 플랫폼으로 이동

[이메일을 잊으셨나요?](#)

내 컴퓨터가 아닌가요? 게스트 모드를 사용하여 비공개로 로그인하세요. [자세히 알아보기](#)

[계정 만들기](#) [다음](#)

한국어 ▾

도움말

개인정보처리방침

약관

유료 계정 활성화

유료 계정 활성화



클라우드가 중단 없이 계속 실행됩니다.



무료 체험판 기간 동안 사용할 수 있는 크레딧 잔액이 유지됩니다.



사용한 만큼만 비용 지불: 무료 체험판이 종료되면 자동 결제가 시작됩니다.

[취소](#)

[활성화](#)

하용님, 환영합니다.

Google Maps Platform으로 할 수 있는 작업에는 제한이 없습니다.

아래에서 제품 선택

API 사용 설정은 무료이며 사용량에 따라 요금이 청구됩니다.

☒ 지도

사용자에게 장소의 실제 모습을 보여주는 맞춤형 지도 환경을 구축하세요.

☒ 경로

사용자에게 출발지에서 목적지까지 가는 최적의 경로를 알려주세요.

☒ 장소

풍부한 상세정보로 사용자의 장소 탐색을 도와주세요.

[사용 설정](#)

[취소](#)

Google map API 설정

■ Google Maps Platform 사용 설정



Google Maps Platform 사용 설정

보다 나은 서비스를 제공할 수 있도록 몇 가지 질문에 답하기

❗ 계속 진행하려면 모든 필드가 올바른지 확인하세요.



Google Maps Platform에 가입하는 주된 목표가 무엇인가요?



다음 중 어떤 업종에 종사하시나요?



업계에서 가장 인기 있는 사용 사례는 다음과 같습니다. 자세히 알아보고 싶은 사용 사례를 선택하세요.

목록에서 사용 사례를 선택하세요. *

위치 공유

제출

이제 개발을 시작할 수 있습니다.

다음은 구현에 필요한 API 키입니다. API 키는 사용자 인증 정보 섹션에서 검색할 수 있습니다.

API 키

본인 API키



① 앱 보안을 강화하려면 다음에서 이 키의 사용량을 제한하세요. [사용자 인증 정보 페이지](#)



월예산 알림을 만들어 지출을 파악하고 Google 지도의 무료 크레딧 \$200를 곧 초과하게 될 때 알림 받기



완료

Google map API 설정

■ API 탐색 및 사용 설정

Google Cloud Platform | My First Project | 제품 및 리소스 검색

Google Maps Platform | Overview | 모든 Google Maps Platform API

Google Maps Platform에 오신 것을 환영합니다

Google Maps Platform을 쉽고 빠르게 시작할 수 있습니다. 준비하는 데 도움이 되는 몇 가지 태스크가 추가되었습니다.

- 시작하기** API 탐색 및 사용 설정
Marketplace를 방문하여 자신에게 가장 적합한 API를 찾아보고 Google Maps Platform을 사용할 수 있는 모든 방법에 대해 알아보세요.
[API 선택](#)
- 시작하기** API 호출
가이드, 튜토리얼, 동영상, 샘플 앱 등이 있는 강력한 라이브러리를 탐색하여 첫 번째 API 호출을 하는 방법을 알아보세요.
[문서로 이동](#)
- 시작하기** 가격 계산
비용을 예측하려면 예산을 초과하지 않으면서 사용량을 극대화할 수 있도록 가격 계산기 사용 방법을 익히는 것이 좋습니다.
[가격 계산](#)

Maps Platform 최대한 활용하기
Google Maps Platform에 대한 니즈에 가장 적합한 환경을 제공해 드릴 수 있도록 몇 가지 질문에 답변해 주세요.
[설문조사 수정](#)

사용 설정된 API 지난 30일

API	요청
Street View Static API	0
Maps Static API	0
Places API	0
Time Zone API	0
Maps SDK for iOS	0
Roads API	0

[즉정항목 보기](#)

예산 알림을 만들었습니다. [보기](#)

API 라이브러리

API 및 서비스 검색

필터링 기준

- 공개 상태: 공개 (335), 비공개 (4)
- 카테고리: 광고 (14), 분석 (3), 빅데이터 (17), 블로그 및 CMS (1), 컴퓨팅 (7), CRM (1), 데이터베이스 (6), 개발자 스택 (2), 개발자 도구 (16), 이메일 (1), 금융 서비스 (1), Firebase (5), Google Cloud API (29), Google Workspace (18), 의료 (4), 머신러닝 (8), **지도 (17)**, 모바일 (12), 모니터링 (4), 네트워킹 (6)













지도


- Maps SDK for Android**
Google
Maps for your native Android app.
- Dialogflow API**
Google
Builds conversational interfaces
- Google Drive API**
Google
The Google Drive API allows clients to access resources from Google Drive.

Google map API 설정

■ API 및 서비스 검색

검색결과 17개

 Directions API Google Directions between multiple locations.	 Distance Matrix API Google Travel time and distance for multiple destinations.	 Geocoding API Google Convert between addresses and geographic coordinates.	 Geolocation API Google Location data from cell towers and WiFi nodes.
 Maps JavaScript API Google Maps for your website	 Maps SDK for Android Google Maps for your native Android app.	 Maps SDK for iOS Google Maps for your native iOS app.	 Maps Static API Google Simple, embeddable map image with minimal code.
 Roads API Google Snap-to-road functionality to accurately trace GPS breadcrumbs.	 Semantic Tile API Google Google Maps geospatial data for Maps SDK for Unity	 Street View Publish API Google Publishes 360 photos to Google Maps, along with position, orientation, and connectivity	 Street View Static API Google Real-world imagery and panoramas.


Geocoding API
Google
Convert between addresses and geographic coordinates.
[관리](#) API 사용 설정됨

[개요](#) [가격 책정](#) [문서](#) [지원](#)

개요

Convert addresses into geographic coordinates (geocoding), which you can use to place markers or position the map. This API also allows you to convert geographic coordinates into an address (reverse geocoding).

Google 정보

Google's mission is to organize the world's information and make it universally accessible and useful. Through products and platforms like Search, Maps, Gmail, Android, Google Play, Chrome and YouTube, Google plays a meaningful role in the daily lives of billions of people.

추가 세부정보

유형: [APIs & services](#)
최종 업데이트: 21. 3. 19.
카테고리: [Maps](#)
서비스 이름: geocoding-backend.googleapis.com

Google

Time zone data for anywhere in the world.

Google map API 활용

■ googlemaps 라이브러리 다운로드

```
1 !pip install googlemaps
```

```
Requirement already satisfied: googlemaps in c:\programdata\anaconda3\lib\site-packages (4.4.2)
Requirement already satisfied: requests<3.0,>=2.20.0 in c:\programdata\anaconda3\lib\site-packages (from googlemaps) (2.27.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests<3.0,>=2.20.0->googlemaps) (1.25.9)
Requirement already satisfied: idna<3,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests<3.0,>=2.20.0->googlemaps) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\programdata\anaconda3\lib\site-packages (from requests<3.0,>=2.20.0->googlemaps) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests<3.0,>=2.20.0->googlemaps) (2020.6.20)
```

```
import googlemaps
```

```
gmaps_key = ***** # API설정할 때 얻은 key
gmaps = googlemaps.Client(key=gmaps_key)
```

```
gmaps.geocode('경남대학교', language='ko')
```

```
[{'address_components': [{'long_name': '7',
  'short_name': '7',
  'types': ['premise']}],
  {'long_name': '경남대학교',
  'short_name': '경남대학교',
  'types': ['political', 'sublocality', 'sublocality_level_2']}],
  {'long_name': '마산합포구',
  'short_name': '마산합포구',
  'types': ['political', 'sublocality', 'sublocality_level_1']}],
  {'long_name': '창원시',
  'short_name': '창원시',
  'types': ['locality', 'political']}],
  {'long_name': '경상남도',
  'short_name': '경상남도',
  'types': ['administrative_area_level_1', 'political']}],
  {'long_name': '대한민국',
  'short_name': 'KR',
  'types': ['country', 'political']}],
  {'long_name': '631-260',
  'short_name': '631-260',
  'types': ['postal_code']}],
  'formatted_address': '대한민국 경상남도 창원시 마산합포구 경남대학교 7',
  'geometry': {'location': {'lat': 35.18035, 'lng': 128.553385},
  'location_type': 'ROOFTOP',
  'viewport': {'northeast': {'lat': 35.18169898029149,
    'lng': 128.5547339802915},
    'southwest': {'lat': 35.1790010197085, 'lng': 128.5520360197085}}},
  'place_id': 'ChIJe_m8f44ubzURJYVWIKWPT38',
  'plus_code': {'compound_code': '5HJ3+49 대한민국 경상남도 창원시 마산합포구',
    'global_code': '8Q7C5HJ3+49'},
  'types': ['establishment', 'point_of_interest', 'university']}]
```

Google map API 활용

- 각 대학 주소 및 위/경도 획득

```
university_name=['경남대학교', '경상대학교', '창원대학교']
```

```
university_address = []
```

```
university_lat = []
```

```
university_lng = []
```

```
for name in university_name:
```

```
    tmp = gmaps.geocode(name, language='ko')
```

```
    university_address.append(tmp[0].get('formatted_address'))
```

```
    tmp_loc = tmp[0].get('geometry')
```

```
    university_lat.append(tmp_loc['location']['lat'])
```

```
    university_lng.append(tmp_loc['location']['lng'])
```

```
for address, lat, lng, name in zip(university_address, university_lat, university_lng, university_name):
```

```
    print('학교명:', name)
```

```
    print('주소: ', address)
```

```
    print('위도: ', str(lat), ' 경도:', str(lng))
```

```
    print('-' * 70)
```

학교명: 경남대학교

주소: 대한민국 경상남도 창원시 마산합포구 경남대학로 7

위도: 35.18035 경도: 128.553385

학교명: 경상대학교

주소: 대한민국 경상남도 진주시 가좌동 진주대로 501

위도: 35.1531629 경도: 128.0994469

학교명: 창원대학교

주소: 대한민국 경상남도 창원시 의창구 봉림동 창원대학로 20

위도: 35.245595 경도: 128.691953

감사합니다
