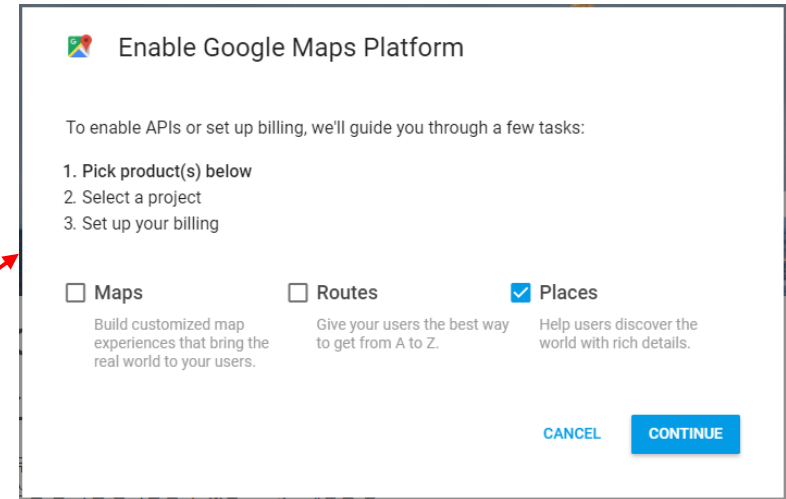
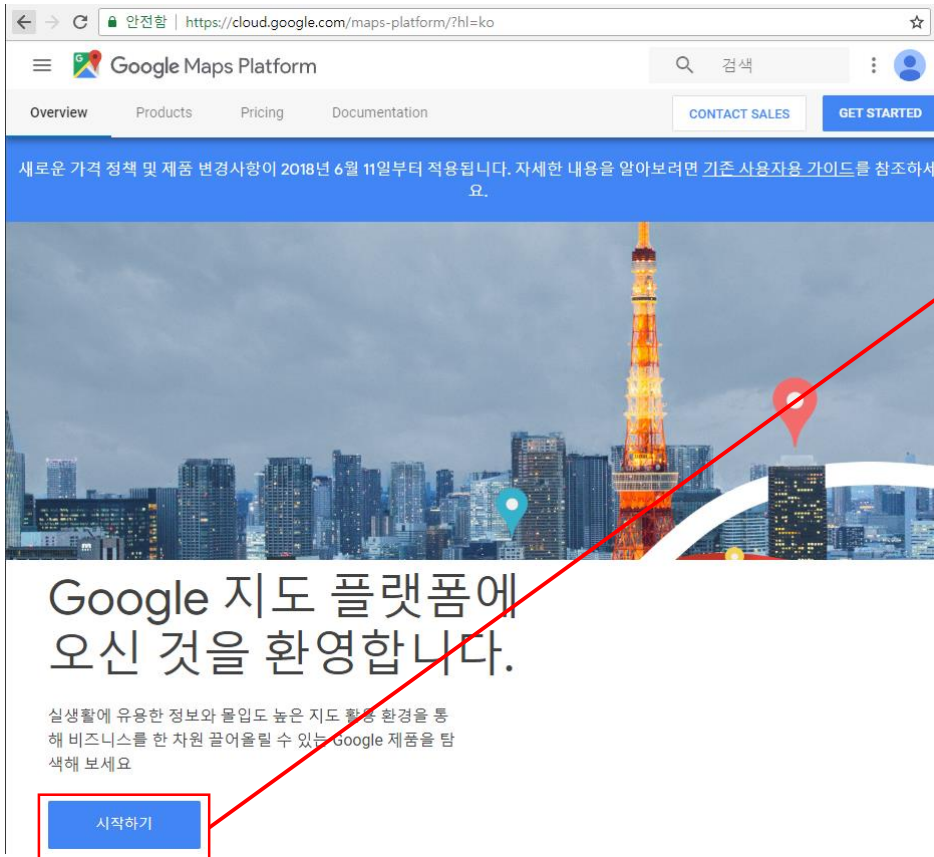


지리 데이터 다루기

최필주

Google maps로 지리 정보 얻기

📍 <https://cloud.google.com/maps-platform/?hl=ko> 접속

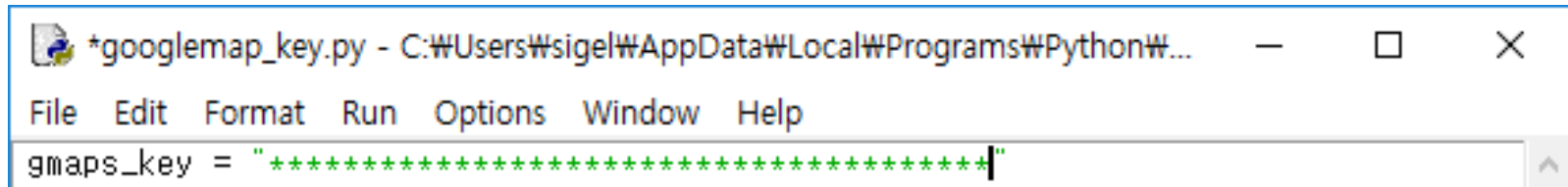


결제 계정 만들기

- 12개월 무료 평가판
- 카드 정보 필요

○ 키 값 안전하게 따로 저장

○ 별도의 파일에 저장



The screenshot shows a window titled '*googlemap_key.py - C:\Users\sigel\AppData\Local\Programs\Python\...'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code editor displays the line: `gmaps_key = "*****"`. The asterisks are green, and a cursor is at the end of the string.

- 파일을 열어 gmaps_key 변수에 키 값 대입(*로 표시)
- 기본 위치에 googlemap_key.py로 저장

○ 키 사용 시

- googlemap_key를 importing한 후 googlemap_key.gmaps_key 사용

○ 서울 중부경찰서 지리 정보 얻기

- 키 입력: gmaps = googlemaps.Client(key="개발자용키")

```
In [1]: import googlemaps  
import googlemap_key  
gmaps = googlemaps.Client(key=googlemap_key.gmaps_key)
```

◎ 서울 중부경찰서 지리 정보 얻기

- 지리정보얻어오기: `gmaps.geocode("검색장소", language='ko')`

```
In [2]: gmaps.geocode('서울중부경찰서', language='ko')
```

```
Out [2]: [{'address_components': [{'long_name': '27',  
    'short_name': '27',  
    'types': ['premise']}],  
  {'long_name': '수표로',  
    'short_name': '수표로',  
    'types': ['political', 'sublocality', 'sublocality_level_4']}],  
  {'long_name': '을지로동',  
    'short_name': '을지로동',  
    'types': ['political', 'sublocality', 'sublocality_level_2']}],  
  {'long_name': '중구',  
    'short_name': '중구',  
    'types': ['political', 'sublocality', 'sublocality_level_1']}],  
  {'long_name': '서울특별시',  
    'short_name': '서울특별시',  
    'types': ['administrative area level 1', 'political']}]
```

○ 서울 중부경찰서 지리 정보 얻기

- 지리정보얻어오기: `gmaps.geocode("검색장소", language='ko')`

```
{ 'long_name': '대한민국',  
  'short_name': 'KR',  
  'types': ['country', 'political'] },  
{ 'long_name': '100-032',  
  'short_name': '100-032',  
  'types': ['postal_code'] },  
  'formatted_address': '대한민국 서울특별시 중구 을지로동 수표로 27',  
  'geometry': { 'location': { 'lat': 37.5636465, 'lng': 126.9895796 },  
    'location_type': 'ROOFTOP',  
    'viewport': { 'northeast': { 'lat': 37.56499548029149,  
      'lng': 126.9909285802915 },  
      'southwest': { 'lat': 37.56229751970849, 'lng': 126.9882306197085 } } },  
  'place_id': 'ChIJc-9q5uSiFDURLhQmr5wkXmc',  
  'plus_code': { 'compound_code': 'HX7Q+FR 대한민국 서울특별시',  
    'global_code': '8Q98HX7Q+FR' },  
  'types': ['establishment', 'point_of_interest', 'police'] }
```

◎ 지리 정보 추출 함수

```
In [3]: def getAddr(name):  
        map = gmaps.geocode(name, language='ko')  
        addr = map[0].get('formatted_address')  
        tmp_loc = map[0].get('geometry').get('location')  
        return addr, tmp_loc['lat'], tmp_loc['lng']
```

```
In [ ]: addr, lat, lng = getAddr('중부경찰서')  
        print(addr, lat, lng)
```


지리 정보 그리기

○ Folium

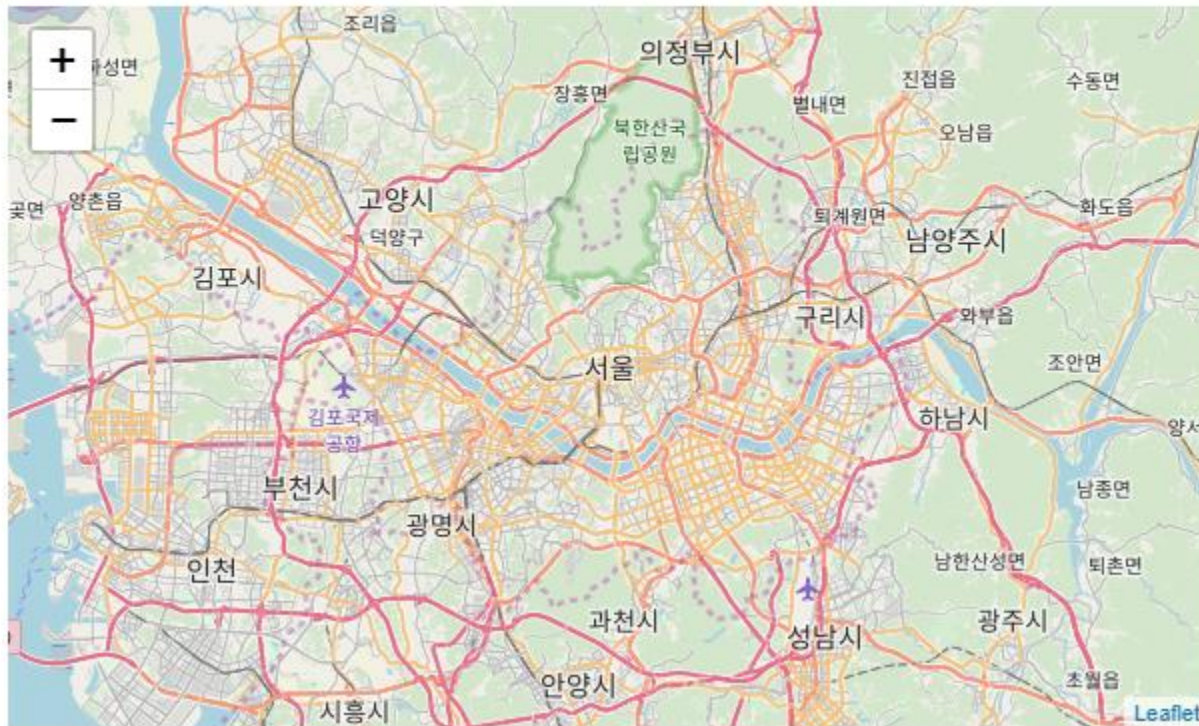
- 지도를 그려주는 라이브러리
- 원하는 위치에 다양한 표현 가능
- 결과를 html로 보여줌
- 참고
 - <http://folium.readthedocs.io/en/latest/>
 - <http://python-visualization.github.io/folium/docs-v0.5.0/modules.html>

○ 기본 지도 그리기

- 지도변수 = folium.Map(location = [위도, 경도])
- 저장: 지도변수.save('파일명')

```
In [6]: import folium
map = folium.Map(location = Seoul_loc)
map.save('c:/python/html/Seoul.html') #별도 html로 저장할 때
map
```

Out [6]:

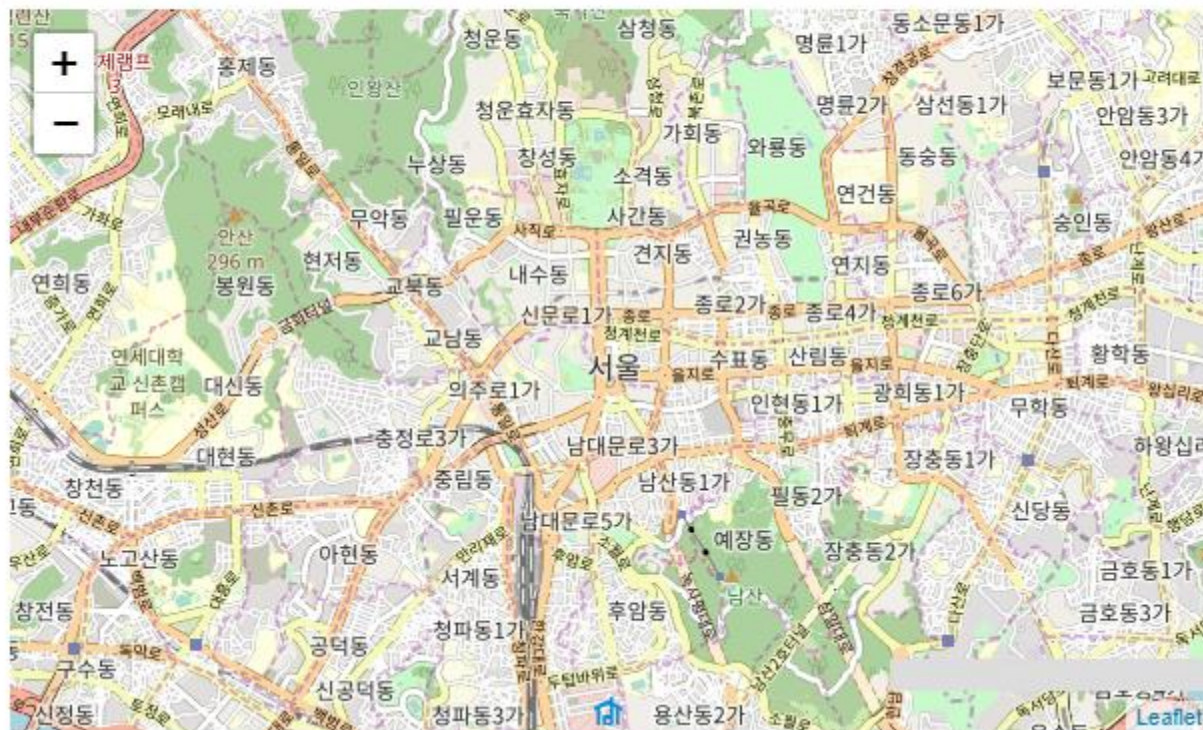


Map함수의 zoom 조절하기

- zoom_start = 숫자

```
In [7]: map = folium.Map(location = Seoul_loc, zoom_start = 13)  
map
```

Out [7]:



○ Map함수의 tile 모양 바꾸기

- tiles = '타일모양'
- 타일모양: 'Stamen Toner' 'cartodbpositron' 'Stamen Terrain' 등

```
In [8]: map = folium.Map(location = Seoul_loc, zoom_start = 13, tiles = 'Stamen Toner')  
map
```

Out [8]:

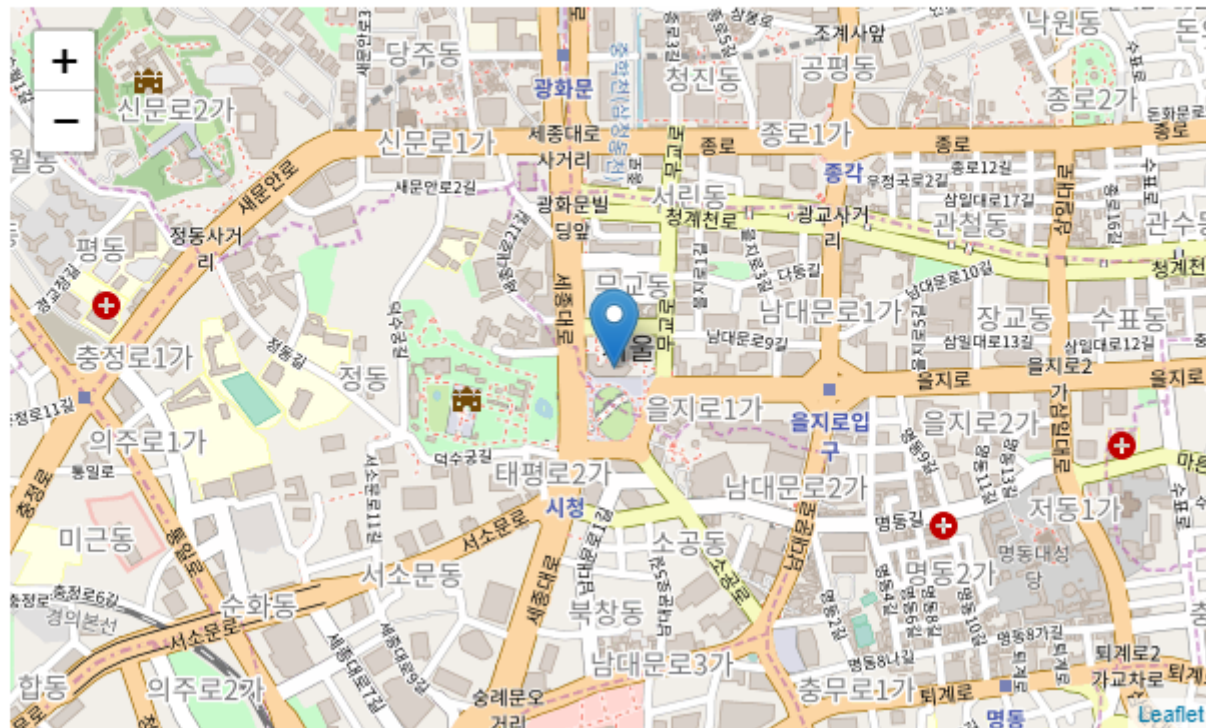


Marker

- folium.Marker([위도, 경도], popup='팝업정보').add_to(지도변수)

```
In [9]: map = folium.Map(location = Seoul_loc, zoom_start = 15)
        folium.Marker(Seoul_loc, popup='서울특별시청').add_to(map)
        map
```

Out [9]:

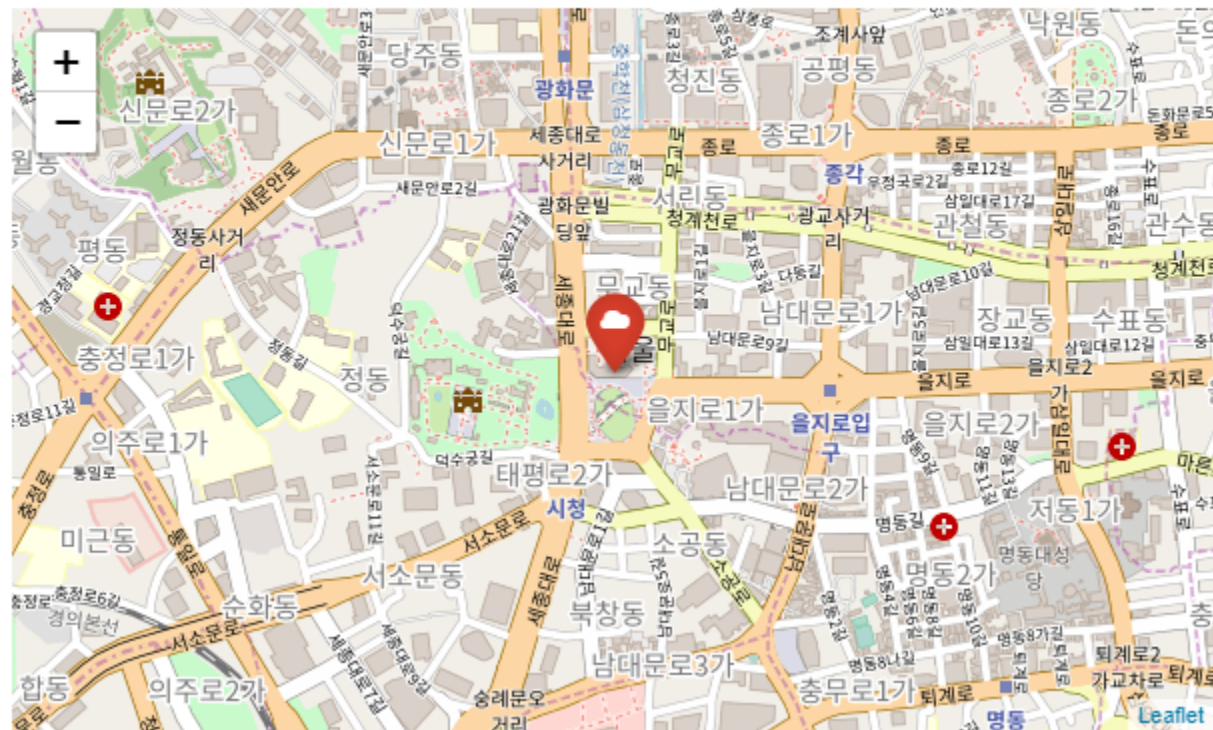


Marker – Icon 바꾸기

- `icon=folium.Icon(icon='모양', color = '색')` 추가

```
In [12]: map = folium.Map(location = Seoul_loc, zoom_start = 15)
          folium.Marker(Seoul_loc, popup='서울특별시청',
                        icon=folium.Icon(icon='cloud', color = 'red')).add_to(map)
          map
```

Out [12]:

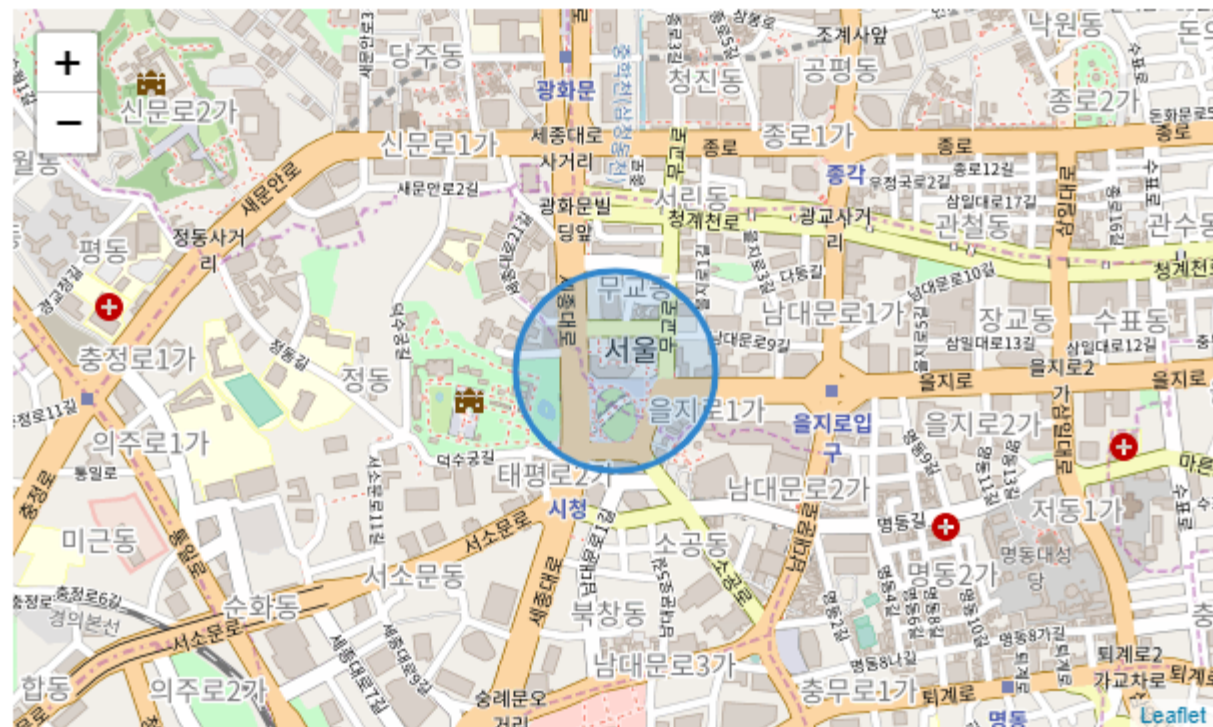


○ Circle Marker

- folium.CircleMarker([위도, 경도], radius = 반지름, popup = '팝업정보', color = '경계색', fill_color = '채움색', fill = True/False).add_to(지도변수)

```
In [13]: map = folium.Map(location = Seoul_loc, zoom_start = 15)
          folium.CircleMarker(Seoul_loc, radius = 50, popup = '서울특별시청', fill = True,
                              color = '#3186cc', fill_color = '#3186cc').add_to(map)
          map
```

Out [13]:



○ 지도에 경계 추가

- json 모듈 가져오기: `import json`
- 경계데이터변수 = `json.load(open('파일명.json', encoding = 'utf-8'))`
- 지도변수.choropleth(`geo_data = 경계데이터변수, fill_opacity = 투명도, fill_color = '채움색'`)

○ 지도에 경계 추가

```
In [14]: import json
geo_str = json.load(open('Seoul_geo.json', encoding = 'utf-8'))
map = folium.Map(location = Seoul_loc, zoom_start = 11)
map.choropleth(geo_data = geo_str, fill_opacity = 0.3, fill_color = 'red')
map
```

Out [14]:

