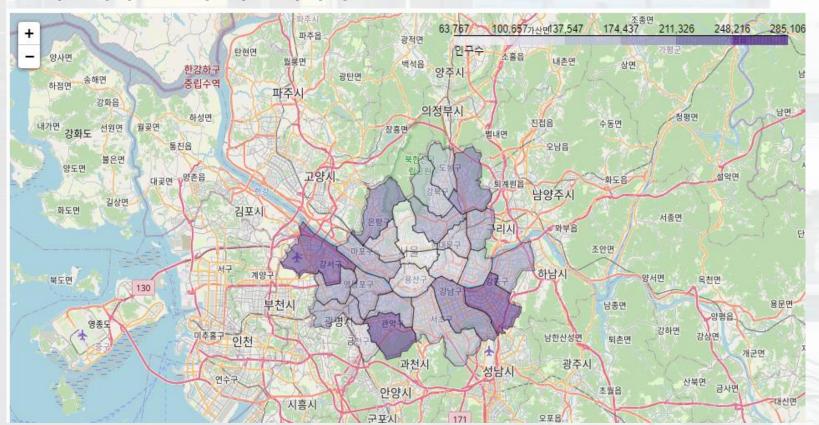




# 오늘의 학습

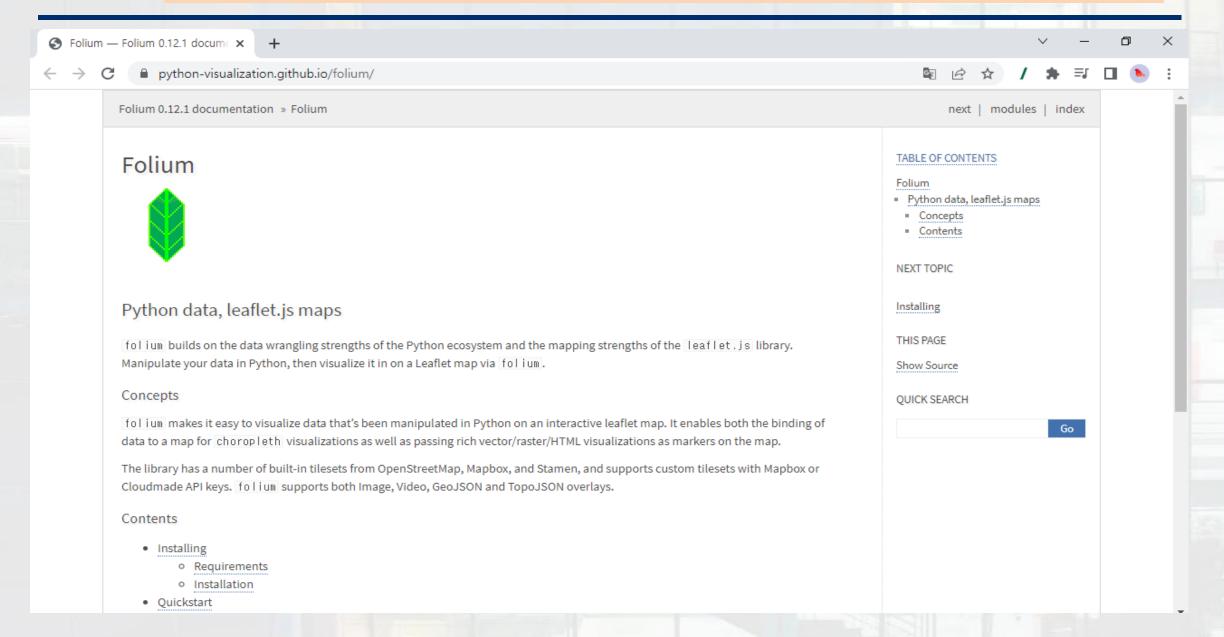
# 학습내용

- 지도 기반 데이터 다루기 (Python의 Folium 패키지)
- 서울시 구별 인구 지도 시각화





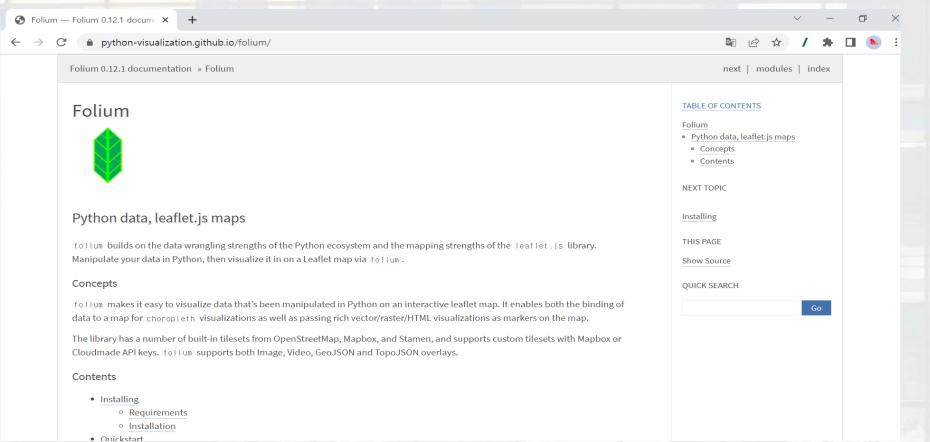
#### folium folium은 leaflet.js 기반으로 만들어진 Python 지도 시각화 라이브러리



#### folium

- location이라는 속성을 이용해 위도와 경도를 기본으로 입력
- 해당 위치를 중심으로 한 지도가 생성

아래 Github에서 더 많은 정보를 얻을 수 있다. <a href="https://python-visualization.github.io/folium/">https://python-visualization.github.io/folium/</a>



# \_folium설치 및 확인

- 설치 : pip install folium
- Anaconda Powershell Prompt (Anaconda3)

(base) PS C:₩Users₩노은희> pip install folium

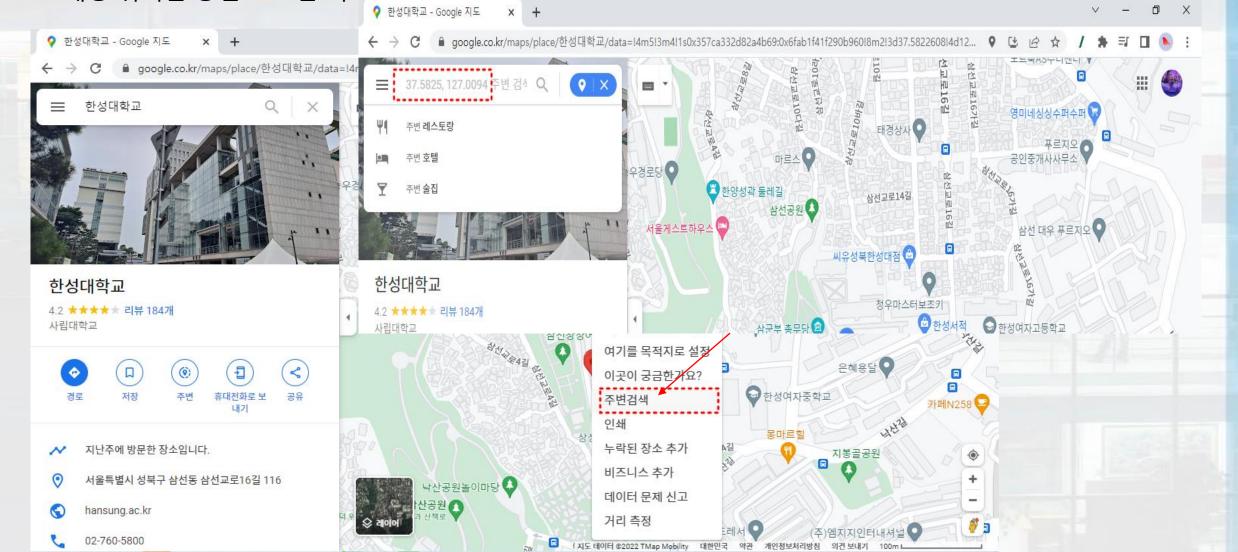
- 설치 확인 : pip list
- Anaconda Powershell Prompt (Anaconda3)

(base) PS C:₩Users₩노은희> Package	pip list Version
alabaster	0.7.12
anaconda-client	1.9.0
anaconda-navigator	2.1.2
anaconda-project	0.10.1
anyio	2.2.0
Anaconda Powershell Prompt (Anac	conda3)
decorator defusedxml	5.1.0 0.7.1

defusedxml	0.7.1
diff-match-patch	20200713
distributed	2021.10.0
docutils	0.17.1
enable	5.3.0
entrypoints	0.3
et-xmlfile	1.1.0
fastcache	1.1.0
filelock	3.3.1
flake8	3.9.2
Flask	1.1.2
folium	0.12.1.post1
fonttools	4.25.0
fsspec	2021.10.1
future	0.18.2
aevent	21.8.0

### -- 구글맵에서 위도와 경도 알아내기

- location이라는 속성을 이용해 위도와 경도를 기본으로 입력
- 해당 위치를 중심으로 한 지도가 생성





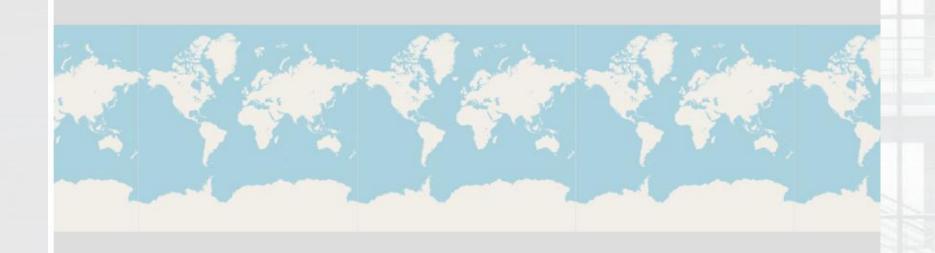
# 기본 지도 그리기

- location이라는 속성을 이용해 위도와 경도를 기본으로 입력
- 해당 위치를 중심으로 한 지도가 생성

zoom\_start 정보를 지정하여 확대의 정도를 지정 zoom\_start의 범위는 최대 18

```
1 import folium
2 m = folium.Map()
3 m
```





# **■ 한성대를 중심으로 지도 보기**



# 지도 스타일 변경

```
- Open street map (기본 값입니다.)
  m = folium.Map(location=[latitude, longitude],
              zoom_start=15,
                                            - Map Quest Open
10
              width=750,
              height=500.
                                            - MapQuest Open Aerial
             tiles = "Stamen Terrain"
                                            - Mapbox Bright
                                            - Mapbox Control Room
                                             Stamenterrain
                                            - Stamentoner
                                             Stamenwatercolor
                                            - cartodbpositron
                                            - cartodbdark matter
```



### ▂ 지도에 마커 추가하기

- location: 마커를 추가할 위도/경도 좌표를 입력 후
- popup: 표기할 팝업 문구 지정 (마우스 클릭시 표기되는 문구)
- tooltip: 표기할 툴팁 지정 (마우스 오버시 표기되는 문구) 마커를 생성 후 기존에 생성된 지도 m에 추가

```
# 지도에 마귀추가하기
import folium
latitude = 37.5822608
longitude = 127.0094254
m = folium.Map(location=[latitude, longitude],
              zoom_start=15,
              width=750.
              height=500,
              tiles = "Open street map"
folium.Marker([latitude, longitude],
```



```
folium.Marker([latitude, longitude],
popup="한성대학교",
tooltip="서울특별시 성북구 삼선동 삼선교로16길 116",
icon=folium.lcon('red', icon='star')).add_to(m)
```



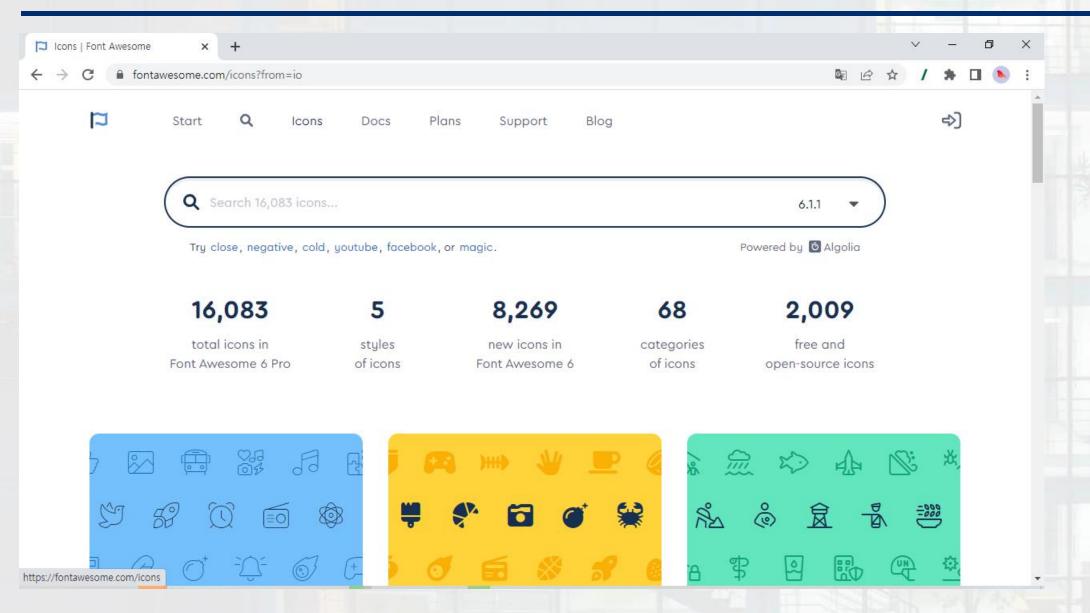
# 지도에 마커 추가하기

```
14 folium.Marker([latitude, longitude].
15 popup ='<iframe width="560" height="315" src="https://www.hansung.ac.kr"></iframe> ,
16 tooltip="처울특별서 성북구 삼선동 삼선교로 18월 116";
17 icon=folium.lcon('red', icon='star')).add_to(m)
```





### 아이콘 변경하기 https://fontawesome.com/icons?from=io



# ■ 지도 마커 이미지로 추가하기

```
class folium.features.Customlcon(
    icon_image,
    icon_size=None,
    icon_anchor=None,
    shadow_image=None,
    shadow_size=None,
    shadow_anchor=None,
    popup_anchor=None
```

```
import folium
from folium.features import CustomIcon
atitude = 37.5822608
longitude = 127.0094254
m = folium.Map(location=[atitude , longitude ], zoom start=12)
################ 상상부기
icon_image1 = 'hsu.png' #상상부기 이미지
icon1 = CustomIcon(icon image1,
         icon size=(30, 30),
  icon_anchor=(10, 30),
  popup anchor=(30, -30)
marker1 = folium.Marker(location=[atitude, longitude],
   popup='한성 대학교',
   icon=icon1,
m.add child(marker1)
m
```



### 지도에 원하는 위치에 원 그리기

```
1 # CircleMarker() radius와 color로 지정하면 원을 그립니다.
2 latitude = 37.5822608
3 longitude = 127.0094254
4 folium.CircleMarker([latitude, longitude],
6 color='red',
7 radius = 100,
8 tooltip='한성대역 삼권').add_to(m)
```

# \_\_ 지도에 원하는 위치에 원 그리기

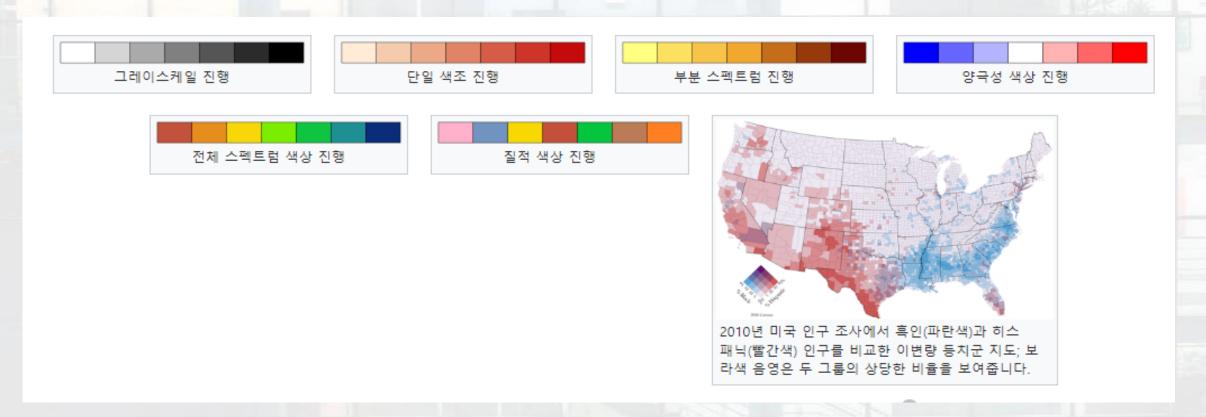
```
# CircleMarker() radius와 color로 지정하면 원을 그립니다.
 | latitude = 37.5822608|
  longitude = 127.0094254
  folium.CircleMarker([latitude, longitude],
6
                     color='red',
                     radius = 100,
                     tooltip='한성대역 상권').add_to(m)
8
```



# 🥃 등치지도: choropleth 맵

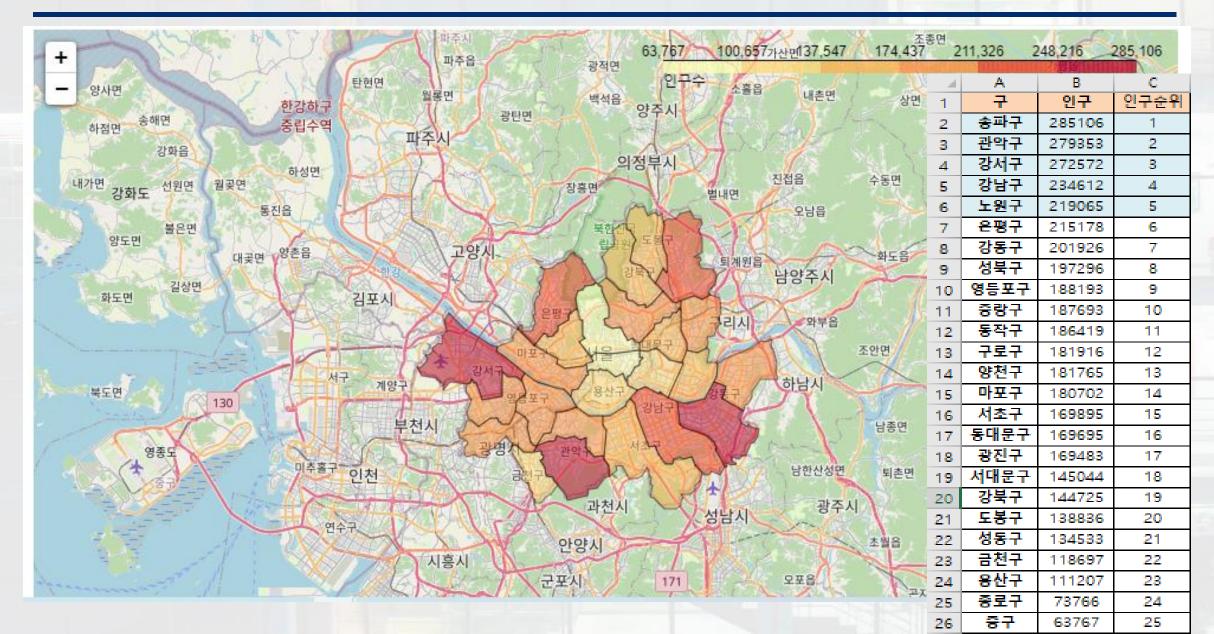
등치 지도는 변수가 지리적 영역에 따라 어떻게 변하는지 시각화하거나 지역 내 변동성 수준을 표시하는 쉬운 방법을 제공

Choropleth 도란 색상이나 패턴을 사용하여 특정 통계에 대한 데이터를 사전 정의된 영역과 관련시켜 시각화한 지도 유형

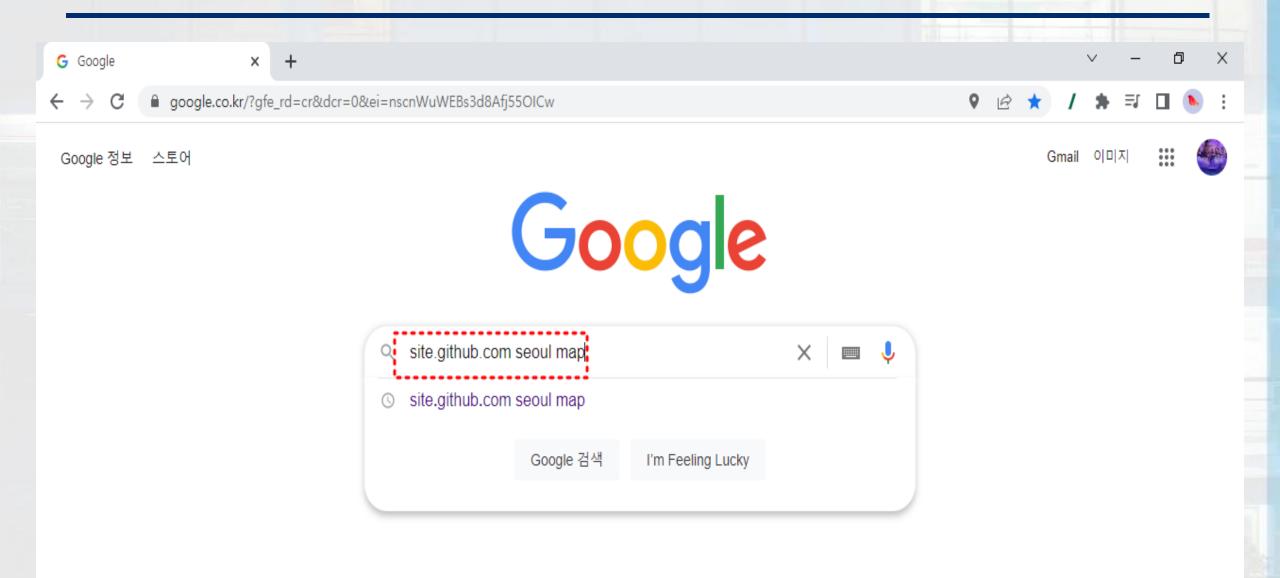




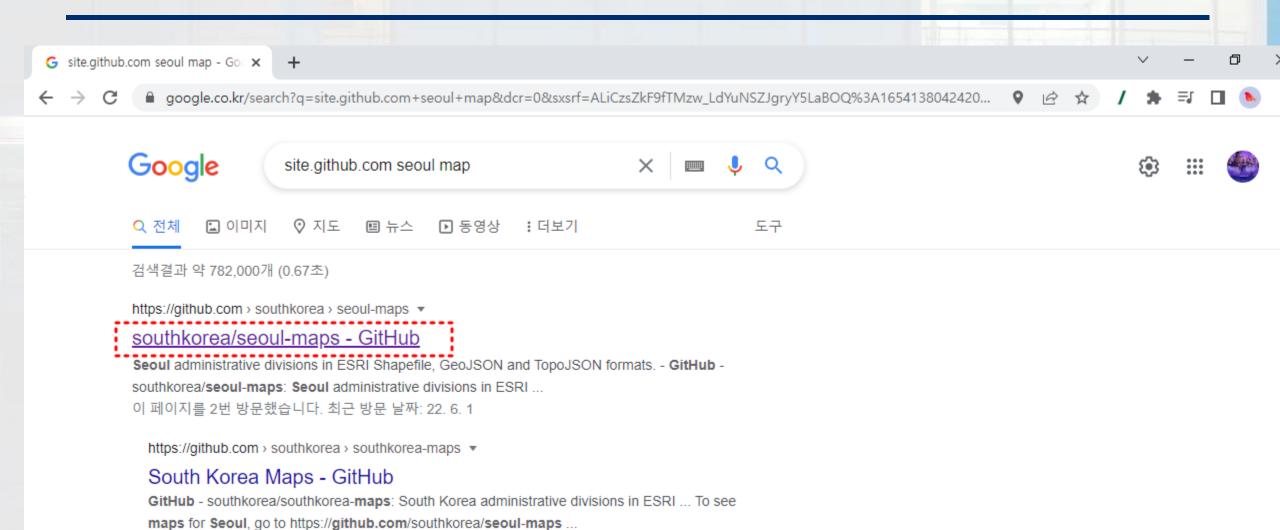
### \_\_ [실습] 서울시 구별 인구 시각화



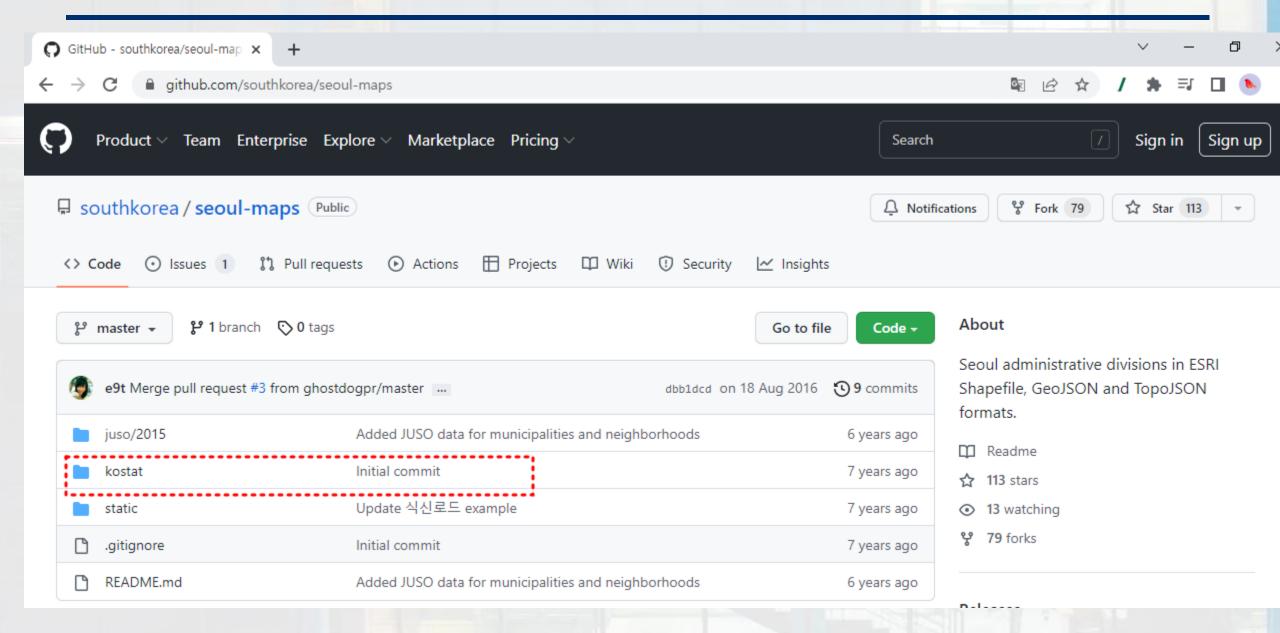




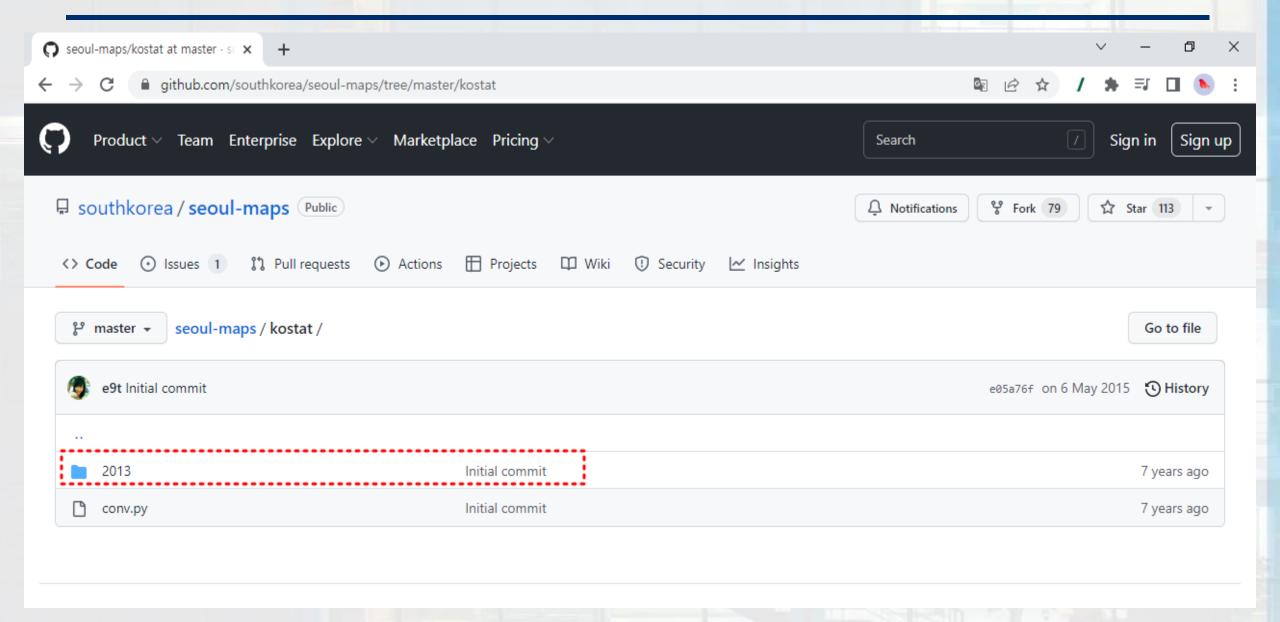




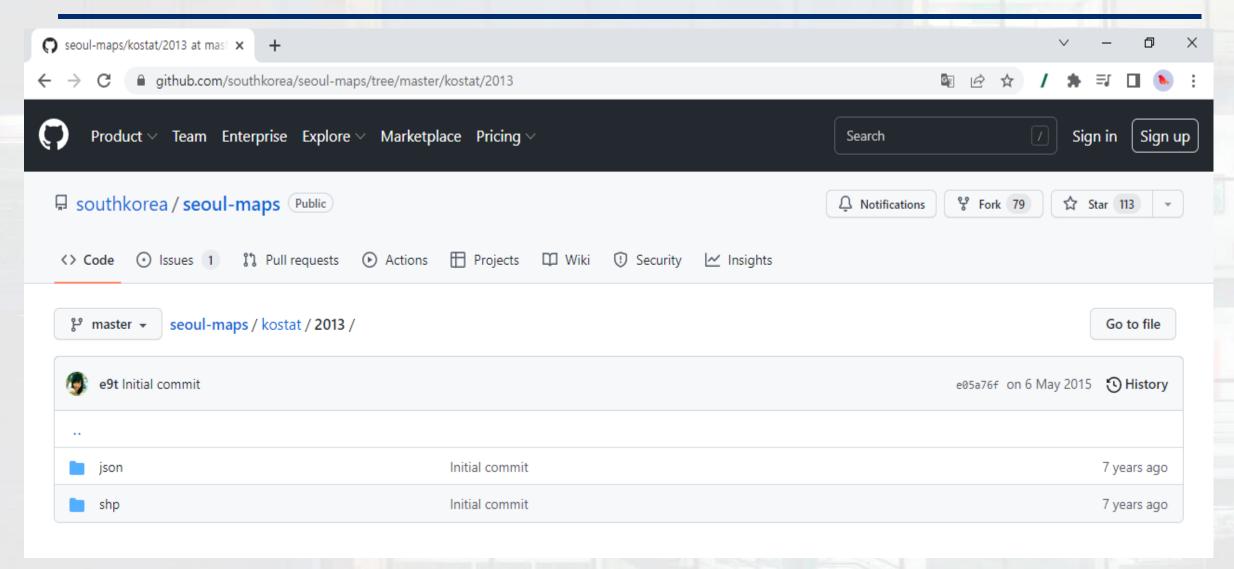




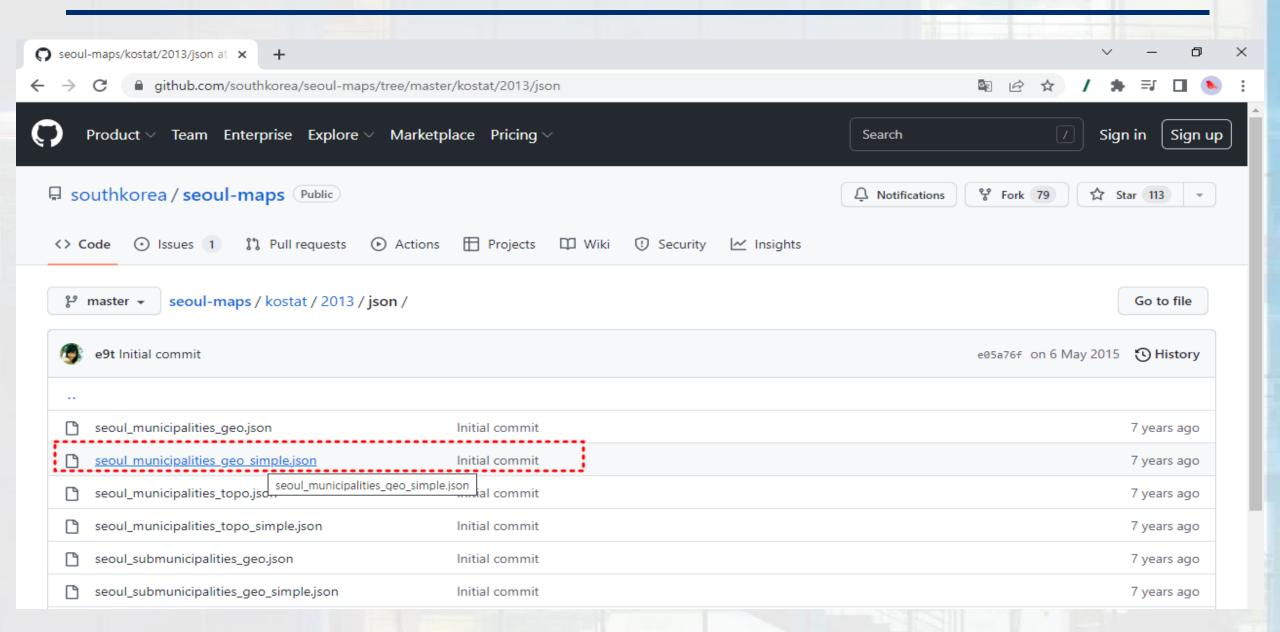




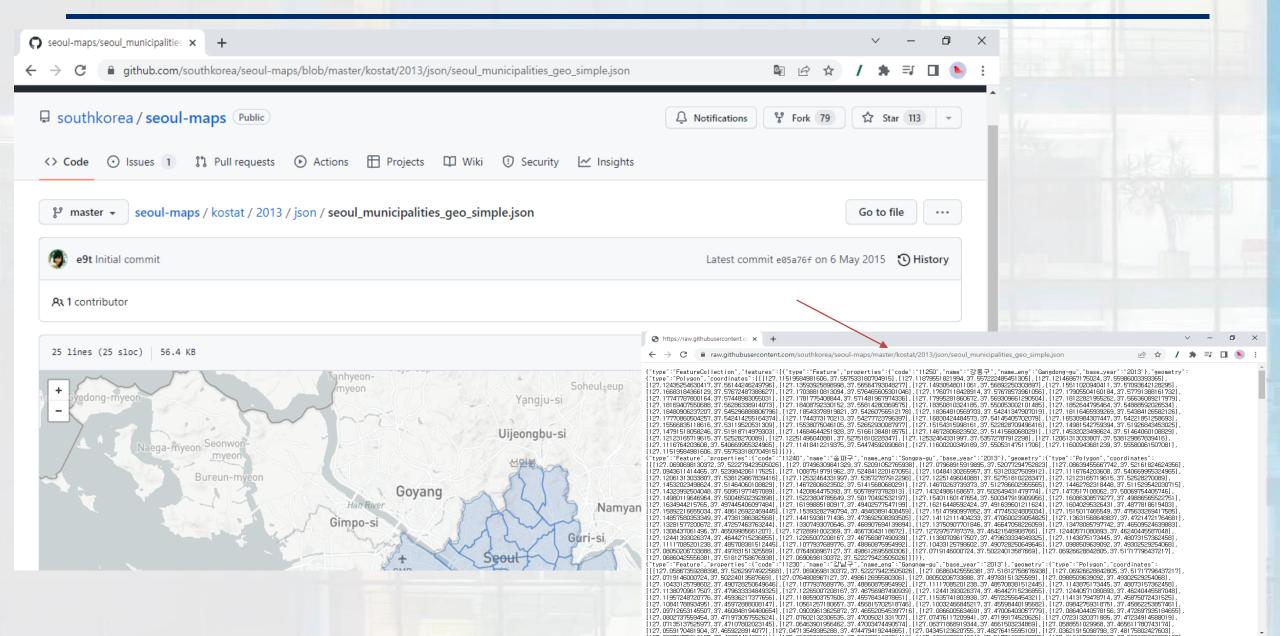












# \_\_\_ [실습] 서울시 구별 인구 시각화\_ Choropleth 시각화

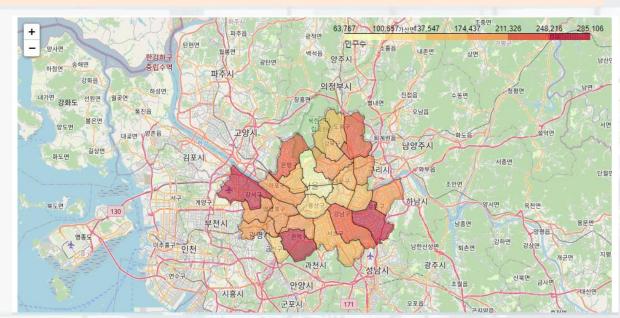
```
folium.Choropleth(
geo data = "지도 데이터 파일 경로(.geojson, geopandas.DataFrame)"
data = "시각화 하고자 하는 데이터파일(pandas.DataFrame)"
columns = (지도 데이터와 매핑할 값, 시각화 하고자하는 변수),
key on = "feature.데이터 파일과 매핑할 값",
fill color = "시각화에 쓰일 색상",
legend name = "칼라 범주 이름", )
.add to (m)
```

- •columns 에는 tuple 형태의 2개의 값만 들어가야 함.
  - 하나는 geo\_data 의 데이터 변수(열) 중, data 의 데이터와 매핑할 변수
  - 아래 key\_on 값과 매핑할 변수
  - 예제에서, 동 으로 매핑
- •key\_on 에는 하나의 String 값이 들어가야 합니다.
  - •각 데이터의 속성(변수)는 properties 라는 dict 형태 안에 저장
  - •예에서 feature.properties.동 으로 접근하여 매칭
- •geo\_data 는 url 이 올 수도 있고, geopandas 로 읽은 데이터프레임이 올 수도 있습니다.

# \_\_\_ [실습] 서울시 구별 인구 시각화\_ Choropleth 시각화

```
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)
```

```
# Choropleth 레이어를 만들고, 맵 m에 추가합니다.
folium.Choropleth(
   geo_data = seoul_geo,
   data = df
   columns = ('구','일구'),
   key_on = 'feature.properties.name',
   fill_color = 'YlOrRd', # 색상 변경도 가능하다.
   fill_{opacity} = 0.5,
   line\_opacity = 0.5,
    Tegend_name = '인구수',
).add to(m)
```



29

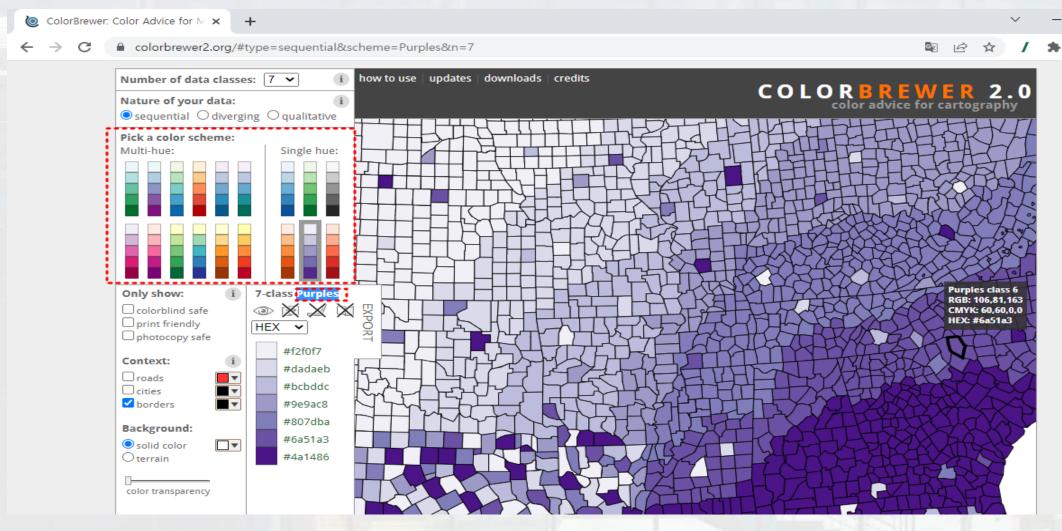
# [실습] 서울시 구별 인구 시각화 전체소스

```
import folium
   import pandas as pd
   # 등치 지도 그리기 #1 서울 행정구역 json raw파일(githubcontent)
   seoul_geo = 'https://raw.githubusercontent.com/southkorea/seoul-maps/master/kostat/2013/json/seoul_municipalities_geo_simple.json'
   df = pd.read_excel('서울시 구별 인구수.xlsx')
   # 서울시 중심부의 위도, 경도
   center = [37.541, 126.986]
   # 맵이 center에 위치하고, zoom 레벨은 10로 시작하는 맵 m을 만듭니다.
   m = folium.Map(location=center, zoom_start=10)
                                                                                                          174.437 211,326 248,216 285,106
14
15
   # Choropleth 레이어를 만들고, 맵 m에 추가합니다.
   folium.Choropleth(
                                                                                                    진접음
                                                        내가면 강화도 선원면 월곶면
18
       geo_data = seoul_geo,
                                                                                                     오남읍
       data = df.
       columns = ('구','인구'),
20
                                                          화도면
                                                                          김포시
       key_on = 'feature.properties.name',
22
       fill_color = 'Purples', # 색상 변경도 가능하다
23
       fill_{opacity} = 0.5
                                                         북도면
24
       line_opacity = 0.5,
       Tegend_name = '인구수',
26
   ).add_to(m).
                                                                                                      광주시
   # 맵 m을 출력합니다.
```

# \_\_\_ [실습] 서울시 구별 인구 시각화\_ Choropleth 시각화

색상 변경

https://colorbrewer2.org/#type=sequential&scheme=Purples&n=7

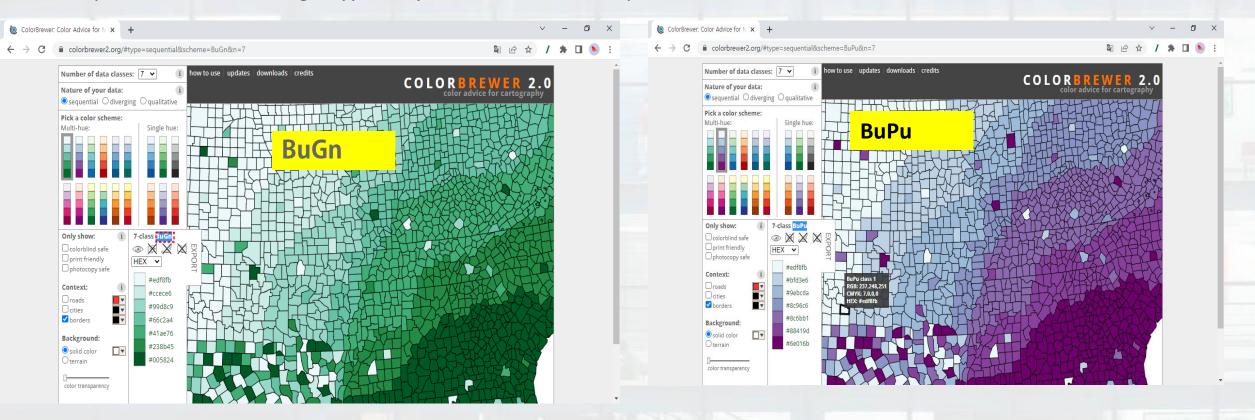




# \_\_\_ [실습] 서울시 구별 인구 시각화\_ Choropleth 시각화

#### 색상 변경

https://colorbrewer2.org/#type=sequential&scheme=Purples&n=7





#### GeoJSON(지오제이슨)

GeoJSON(지오제이슨)은 위치정보를 갖는 점을 기반으로 체계적으로 지형을 표현하기 위해 설계된 개방형 공개 표준 형식

Choropleth map은 geoJson 내용을 사용한다. geoJson은 shapefile로 만들어 지는데 국토교통부의 국가공간정보포털이나 민간 운영 블로그인 GIS DELELOPER 에서 다운로드 받을 수 있다.

### GeoJSON 포맷

```
"type": "FeatureCollection",
"features": []
}
features 안에 실제 데이터가 담기는데, 빈 Feature 요소는 다음과 같다.
{
"type": "Feature",
"geometry": {},
"properties": {}
}
geometry에 종류(type)과 좌표(coordinates) 정보가 저장되고, properties에 속성 정보가 Key-Values로 자유롭게 저장된다.
```