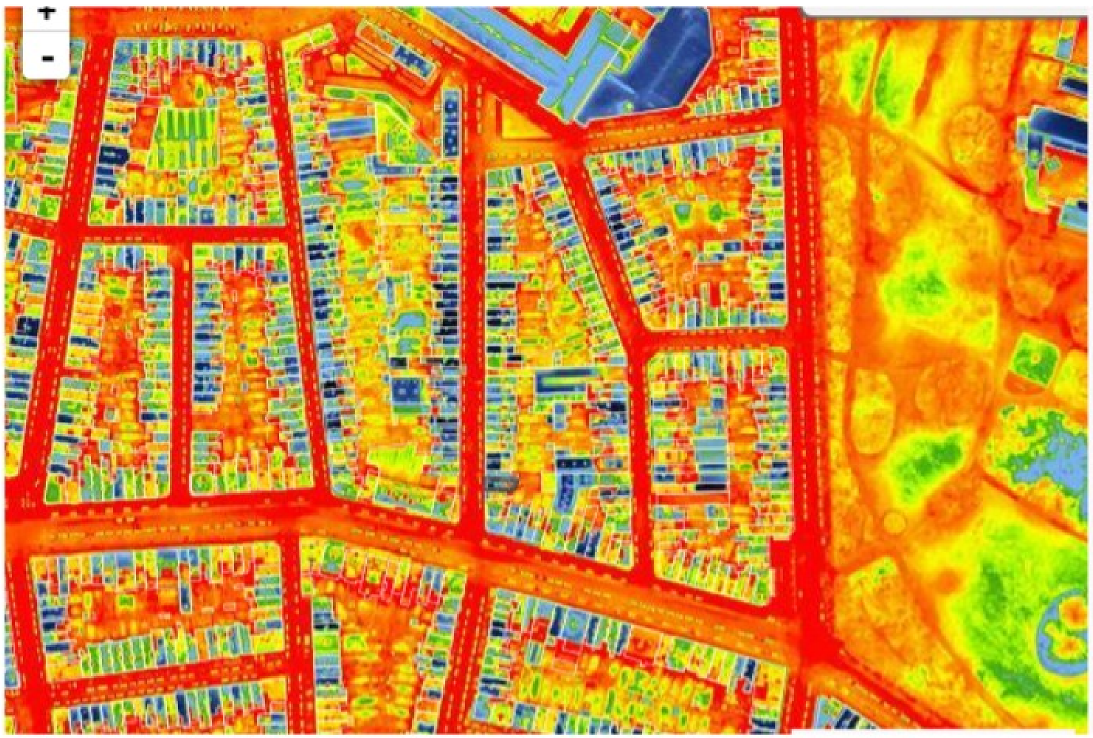
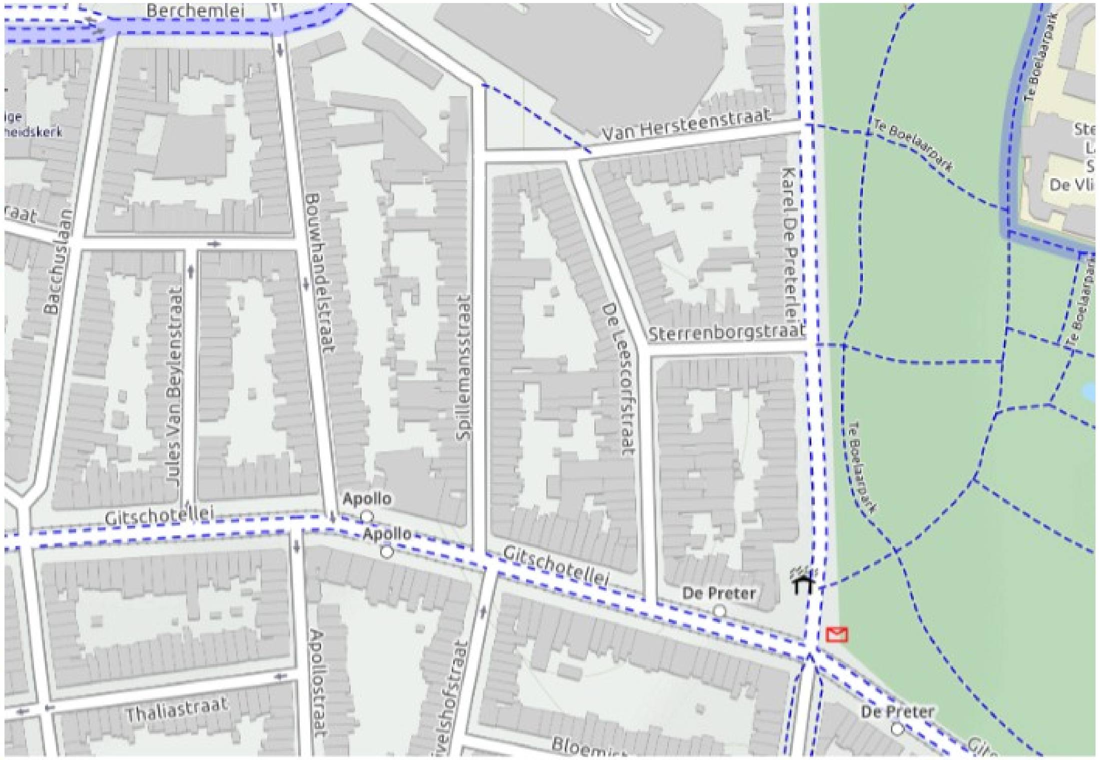
Geographic visualization with Python

Raster vs vector data

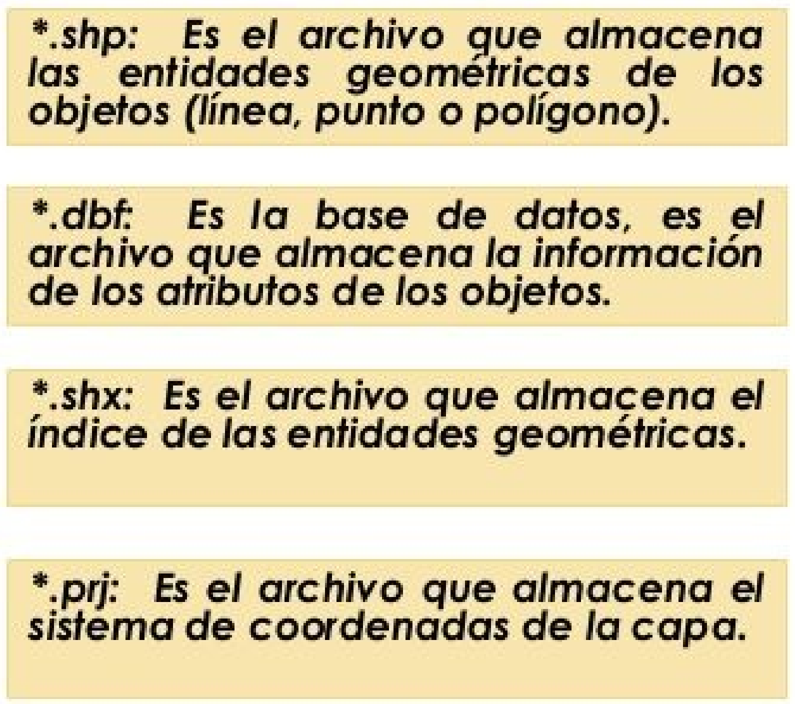
- simple features (points, linestrings, polygons) with attributes





Shp file

- Formato de almacenamiento donde se guarda la localización de los elementos geográficos y los atributos asociados a ellos.



*.shp: Es el archivo que almacena las entidades geométricas de los objetos (línea, punto o polígono).

*.dbf: Es la base de datos, es el archivo que almacena la información de los atributos de los objetos.

*.shx: Es el archivo que almacena el índice de las entidades geométricas.

*.prj: Es el archivo que almacena el sistema de coordenadas de la capa.

*.json, *.geojson, *.topojson

```
"FeatureCollection"
[
  {
    "type": "Feature",
    "geometry": {
      "type": "Point",
      "coordinates": [
        580388.2073999997,
        4721926.0122
      ]
    },
    "properties": {
      "FEATURE": 50200001,
      "CLUGAR": 31,
      "LUGAR": "Fosa del Caserío de Echávarri",
      "TIPO": "Intervenida",
      "UBICACION": "La fosa podría ubicarse junto en las proximidades al
                    Caserío de Echávarri, en la carretera NA-122 que va
                    de Estella a Aberin.",
      "FECHASUCE": "",
      "URL": "http://fosas.navarra.es/?idfosa=31",
      "NUMVICTIMA": 8,
      "MUNICIPIO": "Estella-Lizarra",
      "RADIO": 250,
      "BEGINLIFE": "30/11/2019"
    }
  },
…
]
```

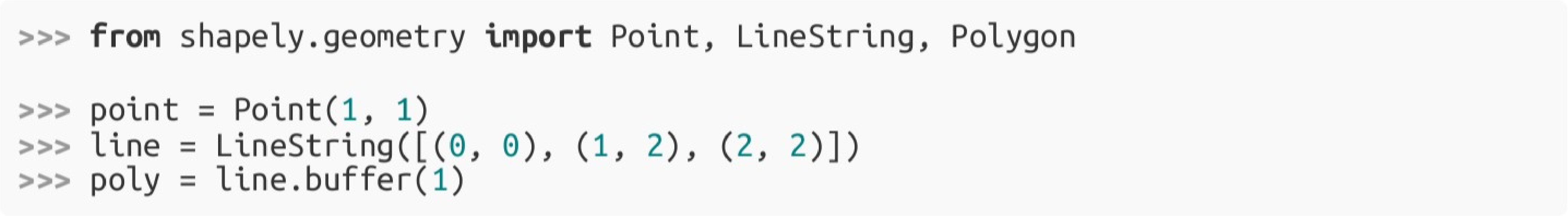
Python geospatial packages

- Interfaces to widely used libraries:
  - Python bindings to GDAL/OGR (from osgeo import gdal, ogr)
  - pyproj : python interface to PROJ.4.
  - Pythonic binding to GDAL/OGR:
    - rasterio for GDAL
    - fiona for OGR
    - shapely : python package based on GEOS.

Shapely

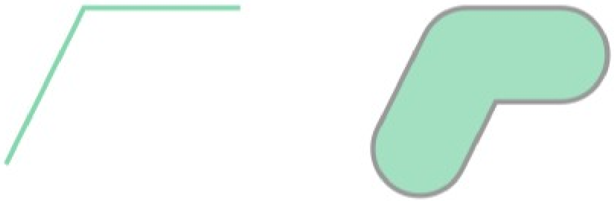
Python package for the manipulation and analysis of geometric objects

Pythonic interface to GEOS

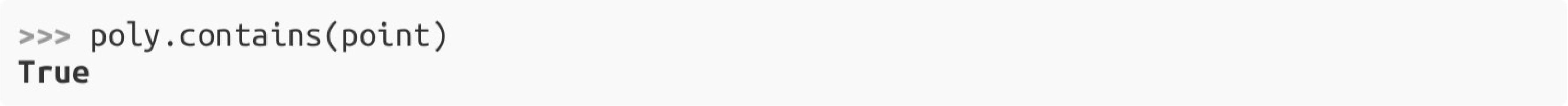


```
>>> from shapely.geometry import Point, LineString, Polygon

>>> point = Point(1, 1)
>>> line = LineString([(0, 0), (1, 2), (2, 2)])
>>> poly = line.buffer(1)
```



single objects, no attributes

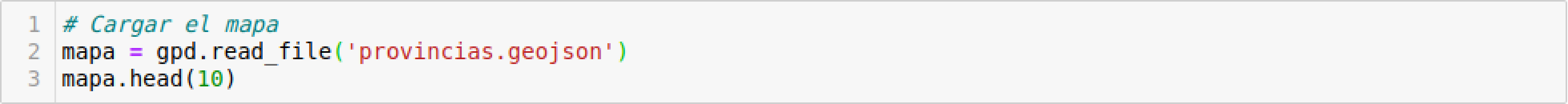


```
>>> poly.contains(point)
True
```

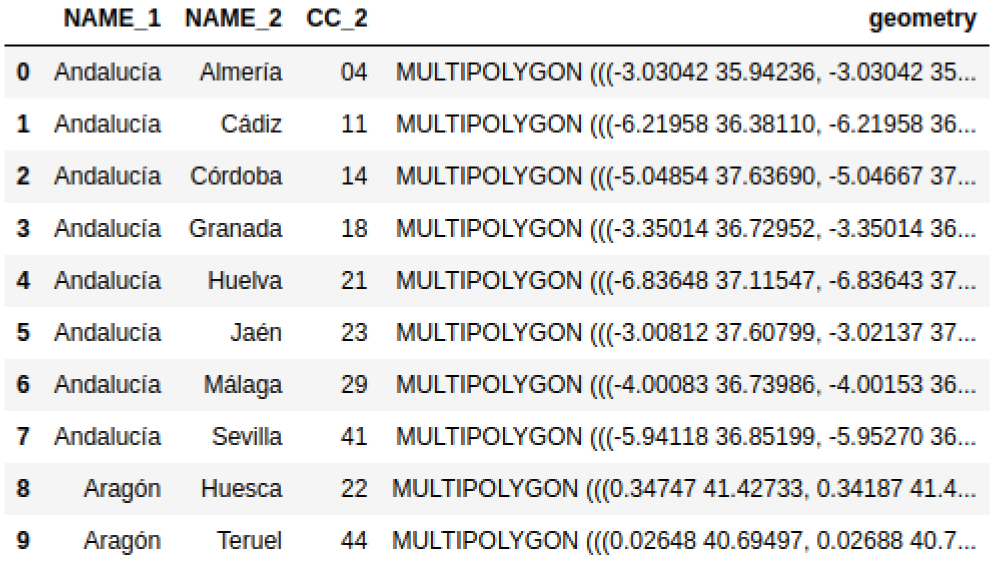
Geopandas

- Extends the pandas data analysis library to work with geographic objects and spatial operations
- Combines the power of whole ecosystem of (geo) tools (pandas, geos, shapely, gdal, fiona, pyproj, rtree, ...)
- Documentation: http://geopandas.readthedocs.io/

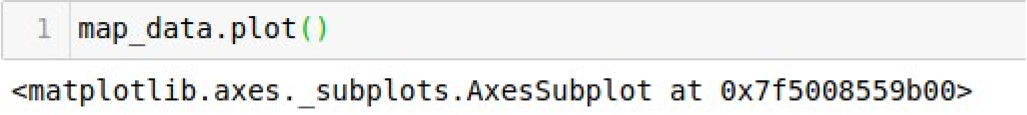
Geopandas



```python
# Cargar el mapa
mapa = gpd.read_file('provincias.geojson')
mapa.head(10)
```

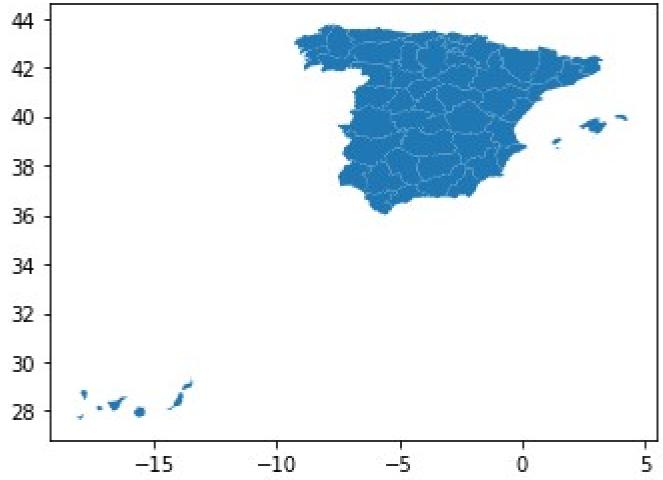


	NAME_1	NAME_2	CC_2	geometry
0	Andalucía	Almería	04	MULTIPOLYGON (((-3.03042 35.94236, -3.03042 35....
1	Andalucía	Cádiz	11	MULTIPOLYGON (((-6.21958 36.38110, -6.21958 36....
2	Andalucía	Córdoba	14	MULTIPOLYGON (((-5.04854 37.63690, -5.04667 37....
3	Andalucía	Granada	18	MULTIPOLYGON (((-3.35014 36.72952, -3.35014 36....
4	Andalucía	Huelva	21	MULTIPOLYGON (((-6.83648 37.11547, -6.83643 37....
5	Andalucía	Jaén	23	MULTIPOLYGON (((-3.00812 37.60799, -3.02137 37....
6	Andalucía	Málaga	29	MULTIPOLYGON (((-4.00083 36.73986, -4.00153 36....
7	Andalucía	Sevilla	41	MULTIPOLYGON (((-5.94118 36.85199, -5.95270 36....
8	Aragón	Huesca	22	MULTIPOLYGON (((0.34747 41.42733, 0.34187 41.4....
9	Aragón	Teruel	44	MULTIPOLYGON (((0.02648 40.69497, 0.02688 40.7....



```python
map_data.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5008559b00>
```

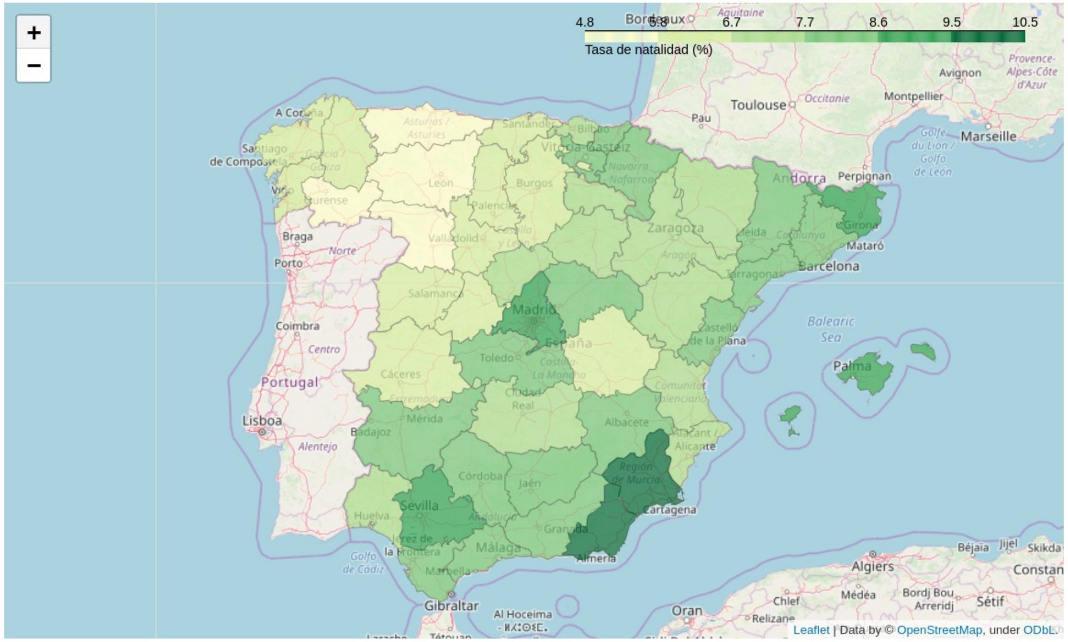


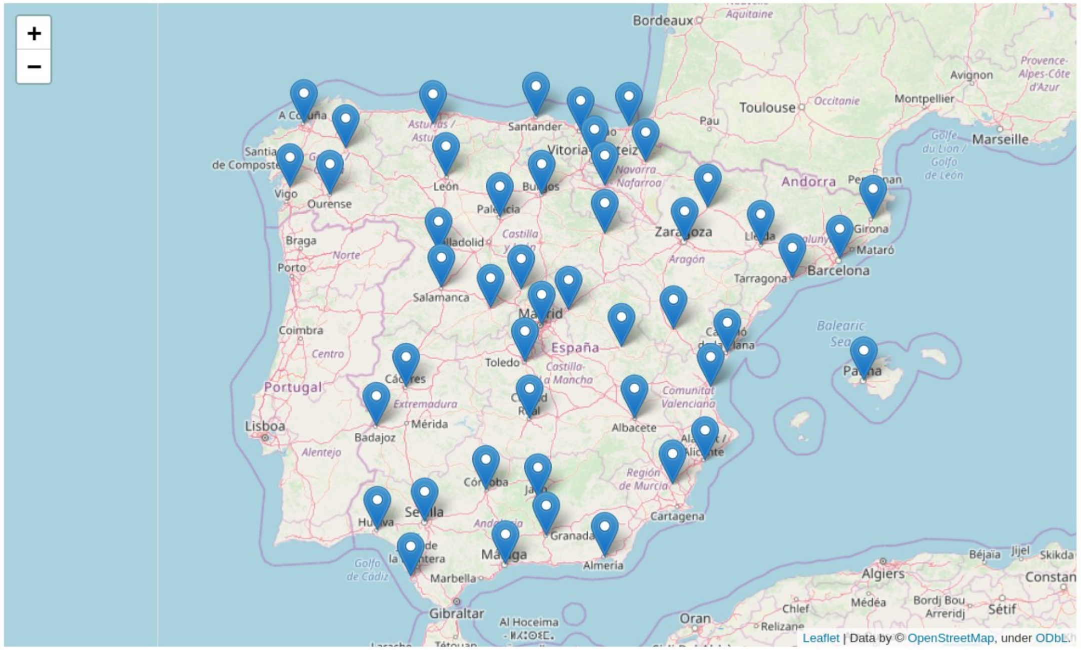
Folium

- Folium is built on the data wrangling strengths of the Python ecosystem and the mapping strengths of the **Leaflet.js** library. You can manipulate your data in python, then visualize it in a Leaflet map via folium. Folium is turning out be an amazing library for plotting spatial data. You can also generate heat maps and choropleth maps using folium.

Folium

m = fl.Map(location=[40.409120, -3.700144], zoom_start=6)

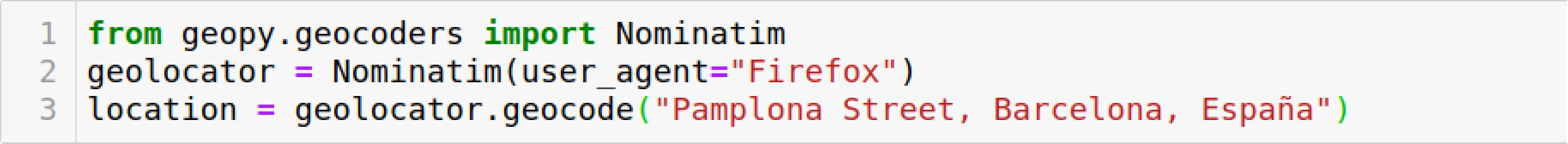




```
fl.Choropleth(
    geo_data='./data/provincias.geojson',
    data=natalidad,
).add_to(m)
```

```
fl.Marker(lat, long, popup).add_to(m)
```

Geocoding



```python
from geopy.geocoders import Nominatim
geolocator = Nominatim(user_agent="Firefox")
location = geolocator.geocode("Pamplona Street, Barcelona, España")
```