



Phần 3: Làm việc với dữ liệu không gian địa lý

Đỗ Thành Long
dtlong@opengis.vn

OPENGIS
Discover the world, Learn with maps

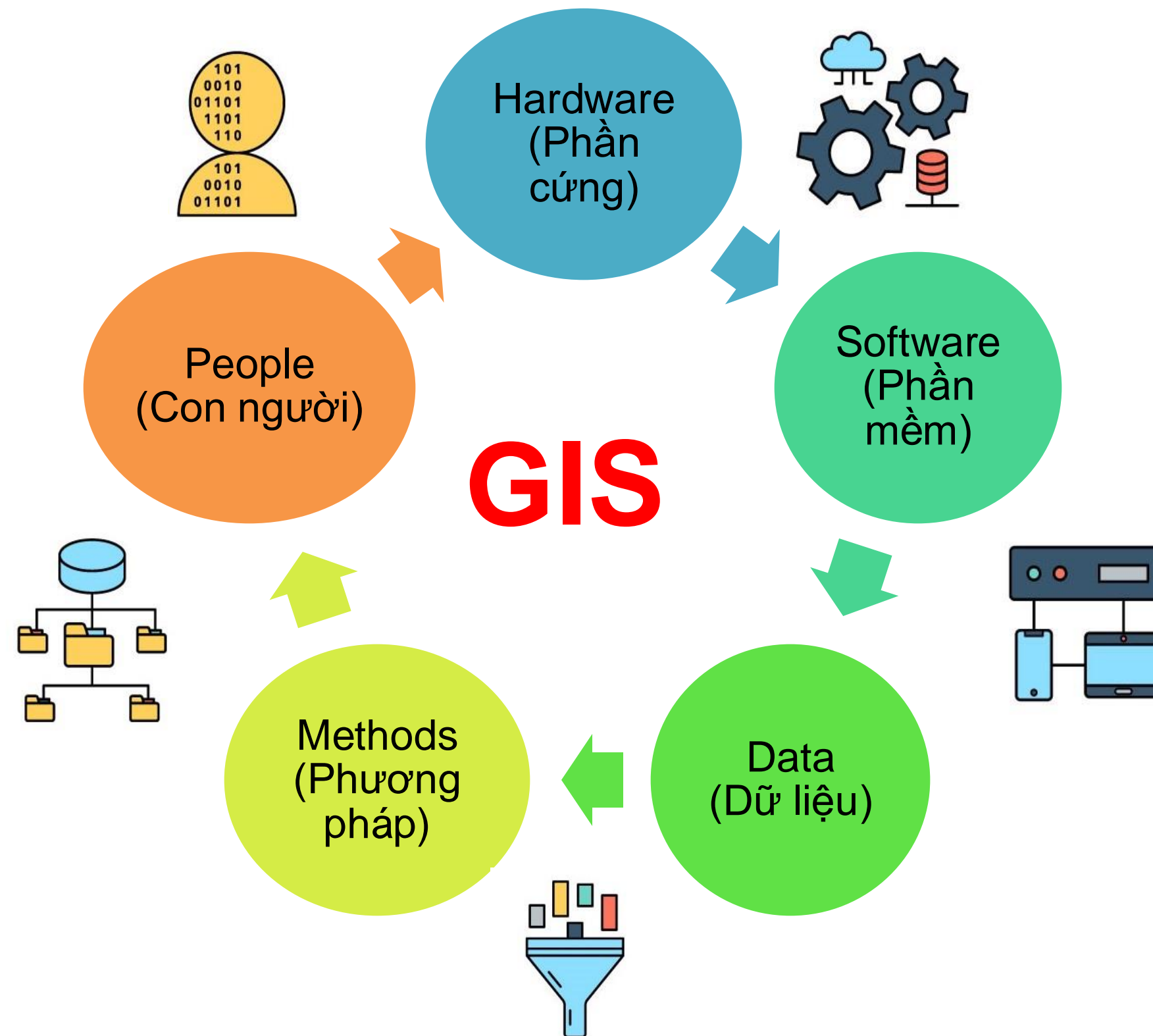
 <https://opengis.vn>

Nội dung

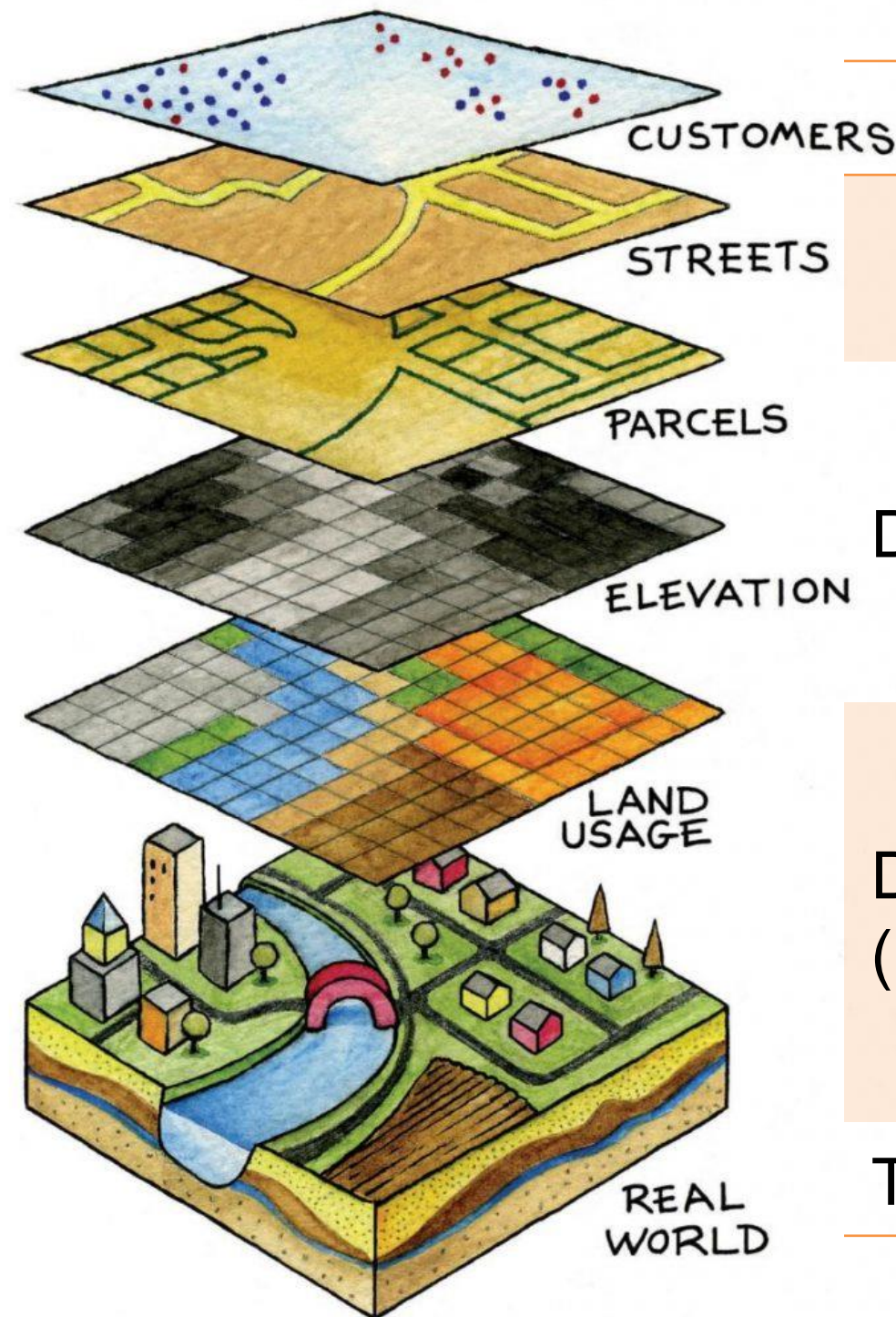
1. Giới thiệu về dữ liệu không gian địa lý
2. Giới thiệu về Python và các thư viện GIS
3. Đọc và hiển thị dữ liệu không gian địa lý
4. Xử lý dữ liệu không gian địa lý
5. Phân tích dữ liệu không gian địa lý



1. Giới thiệu về dữ liệu không gian địa lý

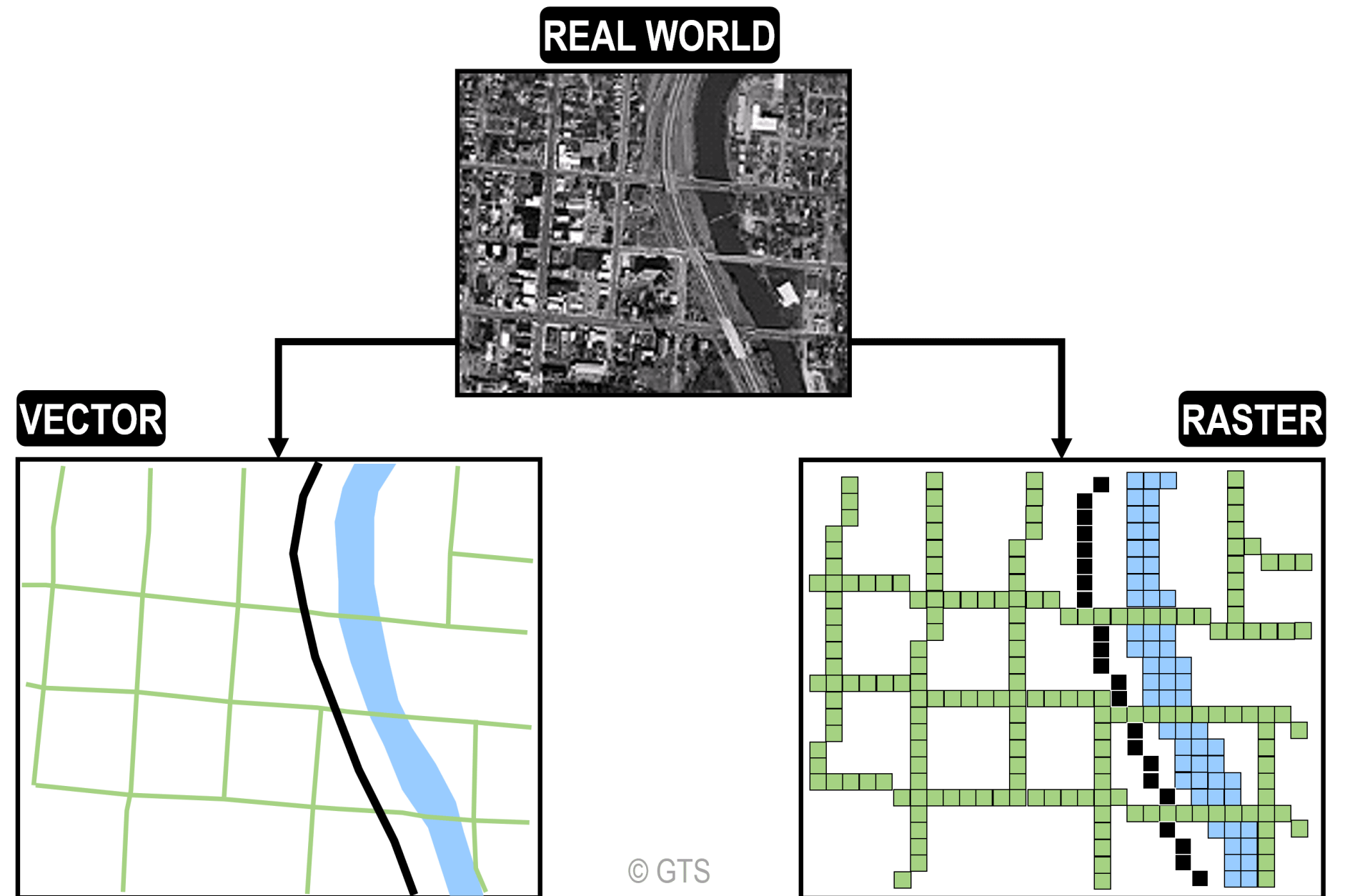
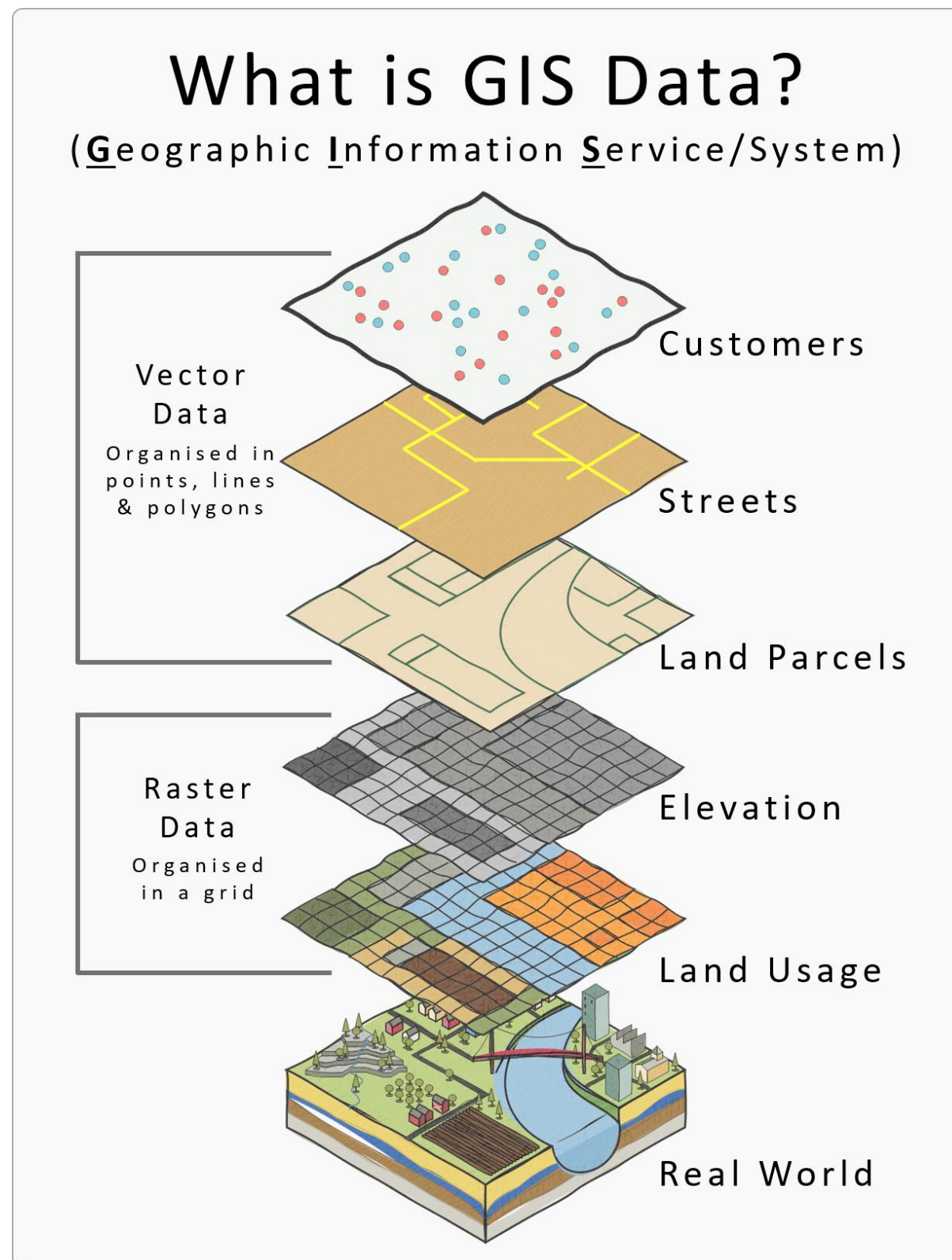


1. Giới thiệu về dữ liệu không gian địa lý

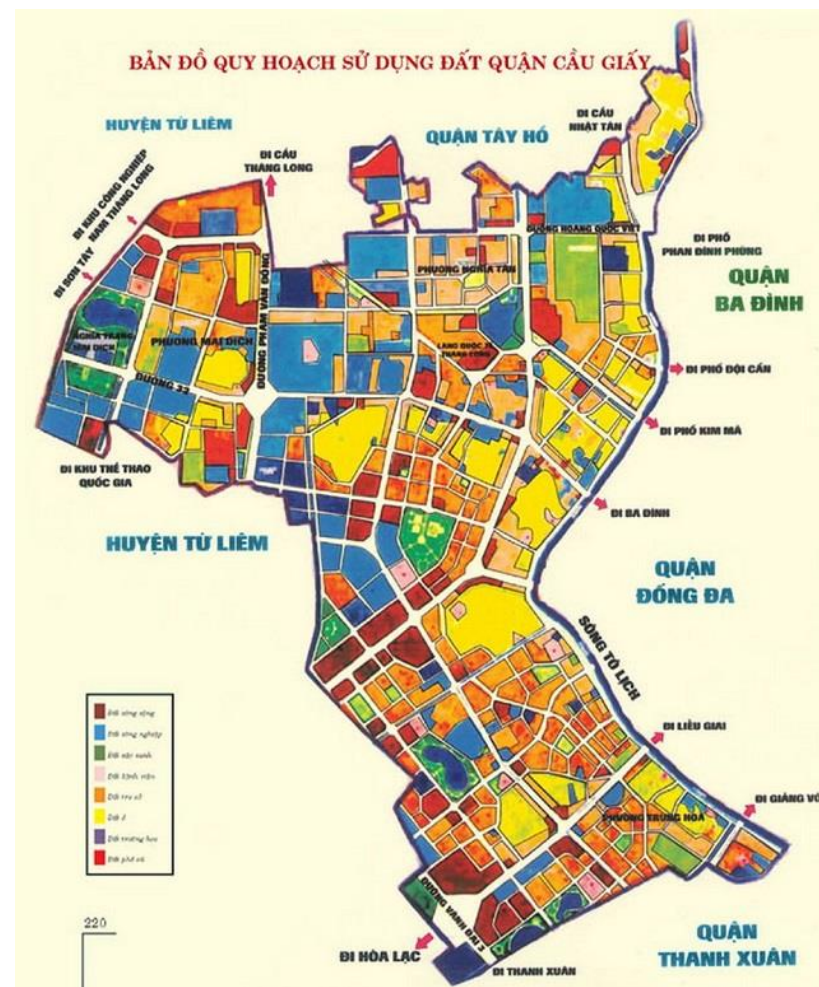


Dữ liệu địa lý	
3 thành phần chính	Nội dung
Dữ liệu không gian	<ul style="list-style-type: none"> - Điểm (Point: UBND, Trạm bus, Trường học,...). - Đường (Line: đường giao thông, đường thủy,...). - Vùng (Polygon: Nhà ở, thửa đất,...). - Thể hiện bằng tọa độ (VN2000, UTM, WGS84,...)
Dữ liệu thuộc tính (Bảng – Table)	<ul style="list-style-type: none"> - Dân số. - Chiều cao/cao độ. - Tọa độ. - Chiều dài. - Bề rộng,....
Thời gian	<ul style="list-style-type: none"> - Gắn liền với thời điểm/khoảng thời gian nhất định.

1. Giới thiệu về dữ liệu không gian địa lý



1. Giới thiệu về dữ liệu không gian địa lý



- Bản đồ giấy

	A	B	C	D	E	F	G	H
1	Prame ñ	X	Y	Z (g)	Z (m)	Lat	Lon	Date
1276	Z-133	-463696.41	-1190386.78		510	18.48060	49.04492	
1277	Z-134	-458539.62	-1189617.81		370	18.55005	49.05563	
1278	Z-135	-458928.38	-1189530.03		360	18.54465	49.05613	
1279	Z-136	-454229.03	-1187860.09		470	18.60688	49.07455	
1280	Z-137	-454596.50	-1187662.76		478	18.60165	49.07605	
1281	Z-138	-456419.24	-1187290.65		440	18.57637	49.07805	

- Excel

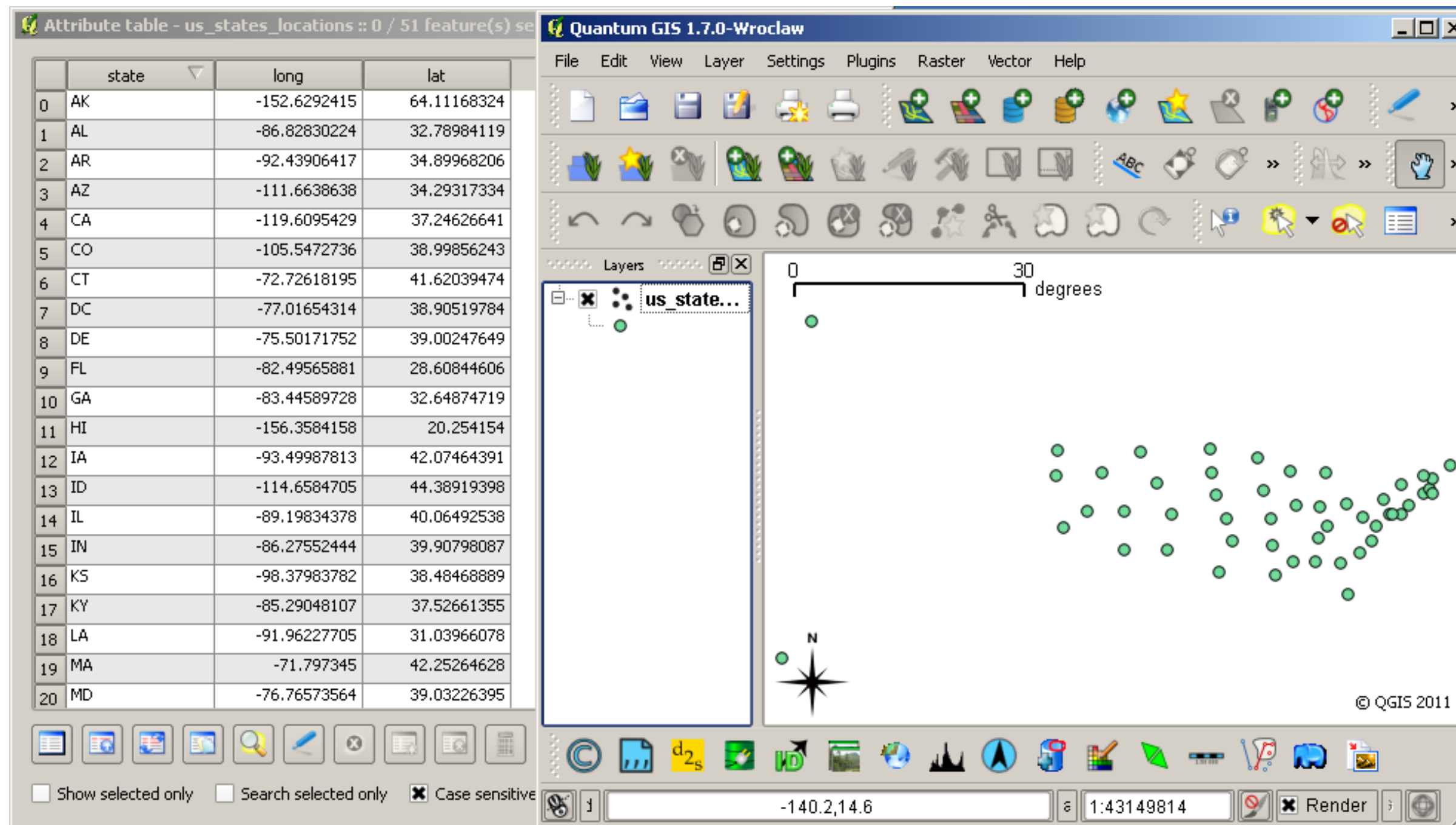
<i>Kiểu dữ liệu</i>	<i>Mô tả</i>
Text	Kiểu văn bản, từ 0 đến 255 ký tự
Memo	Kiểu văn bản, từ 0 đến 64.000 ký tự
Number	Kiểu số, xác định cụ thể tại <i>Field Size</i>
Date/Time	Kiểu ngày và/hoặc giờ
Currency	Kiểu tiền tệ
AutoNumber	Kiểu số được phát sinh tự động và duy nhất
Yes/No	Kiểu logic (1: Yes/true; 0: No/false)
OLE Object	Kiểu đối tượng (hình ảnh, âm thanh, ...)
Hyperlink	Kiểu liên kết
Attachment	Kiểu file đính kèm
Calculated	Kiểu biểu thức

- Văn bản

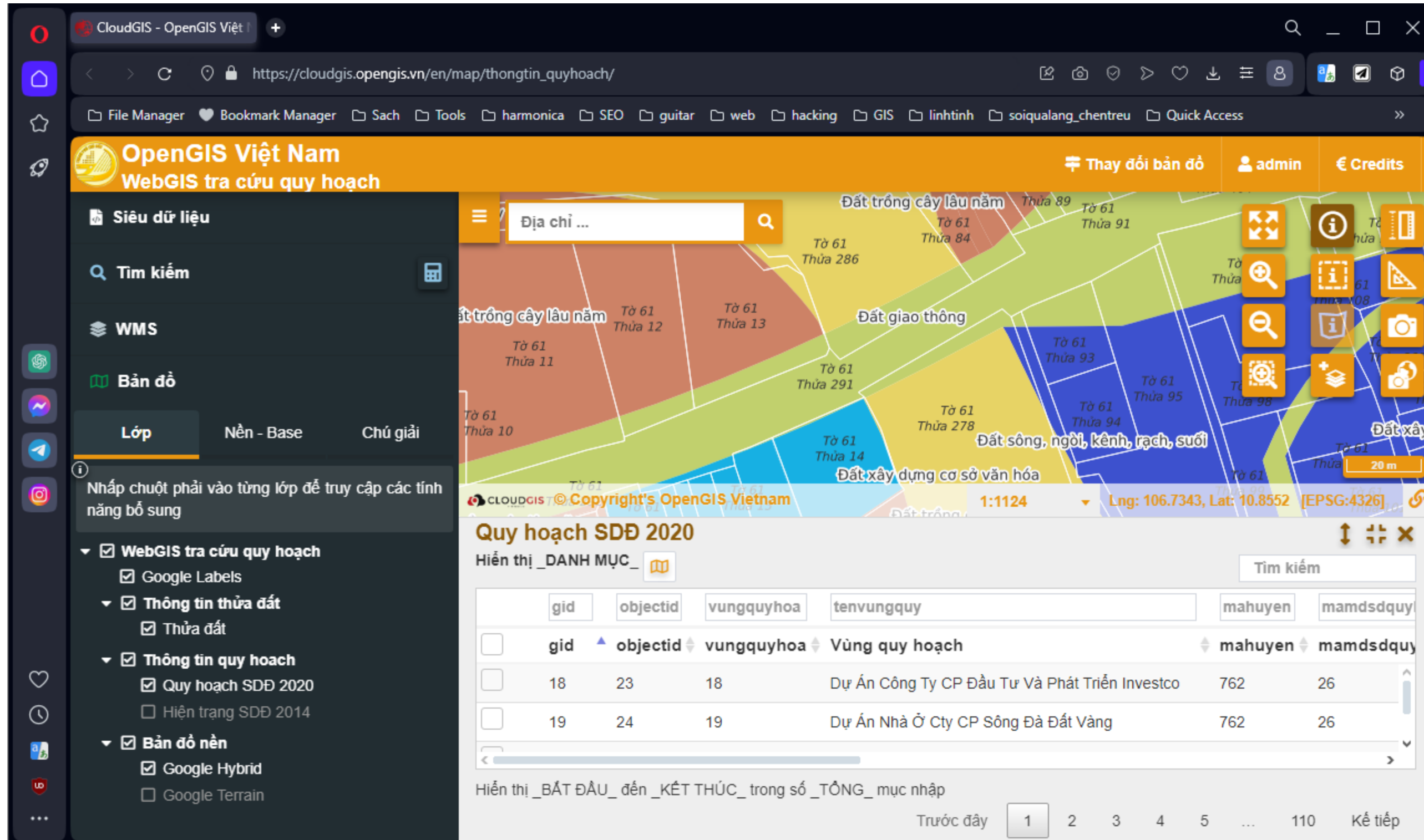


- Ảnh raster (dạng Pixel – 5m)

1. Giới thiệu về dữ liệu không gian địa lý



1. Giới thiệu về dữ liệu không gian địa lý



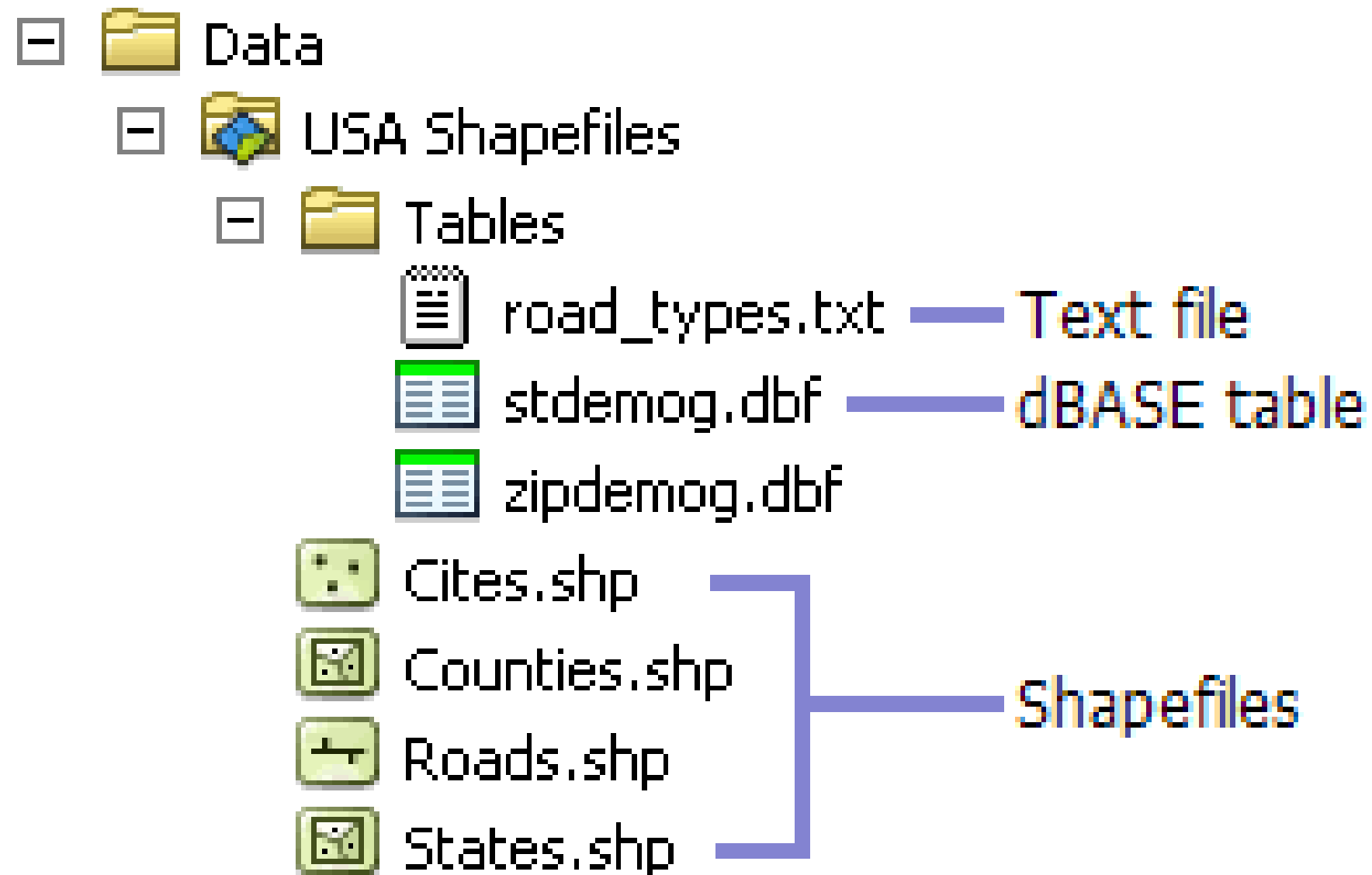
The screenshot displays the OpenGIS Việt Nam WebGIS interface. The main map area shows a land use planning map with various colored regions and labels such as "Đất trồng cây lâu năm", "Đất giao thông", "Đất sông, ngòi, kênh, rạch, suối", and "Đất xây dựng cơ sở văn hóa". The map is overlaid with a grid of land parcels, each labeled with "Tờ 61" and a "Thửa" number (e.g., Thửa 11, Thửa 12, Thửa 13, Thửa 14, Thửa 10, Thửa 286, Thửa 84, Thửa 91, Thửa 291, Thửa 278, Thửa 93, Thửa 94, Thửa 95, Thửa 98, Thửa 99).

The interface includes a sidebar on the left with navigation options: "Siêu dữ liệu", "Tìm kiếm", "WMS", "Bản đồ", and "Lớp". The "Lớp" section is currently selected, showing a list of layers to be displayed. Below the map, there is a table titled "Quy hoạch SDĐ 2020" (Land Use Planning 2020) with columns for "gid", "objectid", "vungquyhoa", "tenvungquy", "mahuyen", and "mamdsdquy". The table lists two planning areas:

gid	objectid	vungquyhoa	tenvungquy	mahuyen	mamdsdquy
18	23	18	Dự Án Công Ty CP Đầu Tư Và Phát Triển Investco	762	26
19	24	19	Dự Án Nhà Ở Cty CP Sông Đà Đất Vàng	762	26

The bottom of the interface shows a pagination control with "Trước đây" (Previous), a list of page numbers (1, 2, 3, 4, 5, ..., 110), and "Kế tiếp" (Next).

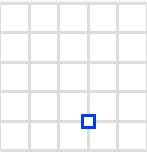
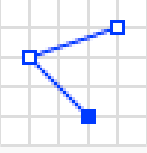
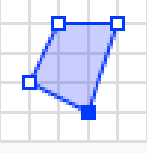
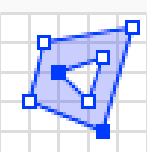
1. Giới thiệu về dữ liệu không gian địa lý



Shapefile

- Đây là một định dạng dữ liệu không gian địa lý phổ biến nhất.
- Được phát triển bởi **Esri** và sử dụng để lưu trữ hình dạng địa lý của các yếu tố như điểm, đường và vùng.
- Shapefile bao gồm một tập tin chính (**.shp**) để lưu trữ hình dạng địa lý và các tập tin phụ (**.shx**, **.dbf**, **.prj**) để lưu trữ thông tin thuộc tính và các thông số khác.

1. Giới thiệu về dữ liệu không gian địa lý

Type	Examples	
Point		<pre>{ "type": "Point", "coordinates": [30, 10] }</pre>
LineString		<pre>{ "type": "LineString", "coordinates": [[30, 10], [10, 30], [40, 40]] }</pre>
Polygon		<pre>{ "type": "Polygon", "coordinates": [[[30, 10], [40, 40], [20, 40], [10, 20], [30, 10]]] }</pre>
		<pre>{ "type": "Polygon", "coordinates": [[[35, 10], [45, 45], [15, 40], [10, 20], [35, 10]], [[20, 30], [35, 35], [30, 20], [20, 30]]] }</pre>

- Dựa trên **JSON** (JavaScript Object Notation).
- GeoJSON cho phép lưu trữ thông tin về hình dạng địa lý và thuộc tính của các yếu tố như điểm, đường và vùng.
- Cú pháp đơn giản và rất phổ biến trong việc trao đổi dữ liệu không gian địa lý trên **web**



1. Giới thiệu về dữ liệu không gian địa lý

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Dinagat Islands"
  }
}
```

- Gồm 3 thuộc tính chính, đó là **type**, **geometry**, **properties**.
- **type**: kiểu dữ liệu của file GeoJSON này. Gồm các loại sau:
 - Feature: Chỉ 1 đối tượng địa lý.
 - FeatureCollection: Tập hợp nhiều đối tượng địa lý.

1. Giới thiệu về dữ liệu không gian địa lý

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Dinagat Islands"
  }
}
```

- Gồm 3 thuộc tính chính, đó là **type**, **geometry**, **properties**.
 - **geometry**: toạ độ của đối tượng. Gồm các loại sau:
 - Point: Một điểm. Thường được dùng việc hiển thị marker.
 - LineString: Một đường thẳng.
 - MultiLineString: Tập hợp nhiều đường thẳng.
 - Polygon: Một đa giác.
 - MultiPolygon: Tập hợp nhiều đa giác.
 - GeometryCollection: Tập hợp nhiều loại hình học khác nhau.

1. Giới thiệu về dữ liệu không gian địa lý

```
{  
  "type": "Feature",  
  "geometry": {  
    "type": "Point",  
    "coordinates": [125.6, 10.1]  
  },  
  "properties": {  
    "name": "Dinagat Islands"  
  }  
}
```

- Gồm 3 thuộc tính chính, đó là **type**, **geometry**, **properties**.
 - **properties**: danh sách thuộc tính của đối tượng.

1. Giới thiệu về dữ liệu không gian địa lý



2. Giới thiệu về Python và các thư viện GIS



- Ngôn ngữ lập trình **thông dịch**
- Tạo ra bởi Guido van Rossum
- Công bố vào năm 1991
- Đặc trưng bởi tính **linh hoạt và sự dễ học**
- Thư viện phong phú: **NumPy, Pandas, Matplotlib, Django, Flask, Tkinter, ...**



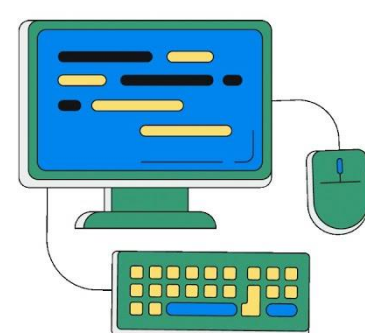
2. Giới thiệu về Python và các thư viện GIS



- Phát triển ứng dụng **desktop**, ứng dụng **web**, ứng dụng **di động, trò chơi**, công cụ **phân tích** dữ liệu, trí tuệ nhân tạo
- Là ngôn ngữ yêu thích cho việc làm việc với dữ liệu và tích hợp với các công nghệ khác như **GIS, viễn thám, trí tuệ nhân tạo**,...



DESKTOP APP



IN PYTHON



2. Giới thiệu về Python và các thư viện GIS

Variables in Python

C Programming

Variable of type int
`int x = 10;`
`x = 4.55;`
Can't store a float

Python Programming

Variable of type int
`x = 10`
`x = 4.55`
Not a problem

Definition Initialization

`data_type variable_name = value`

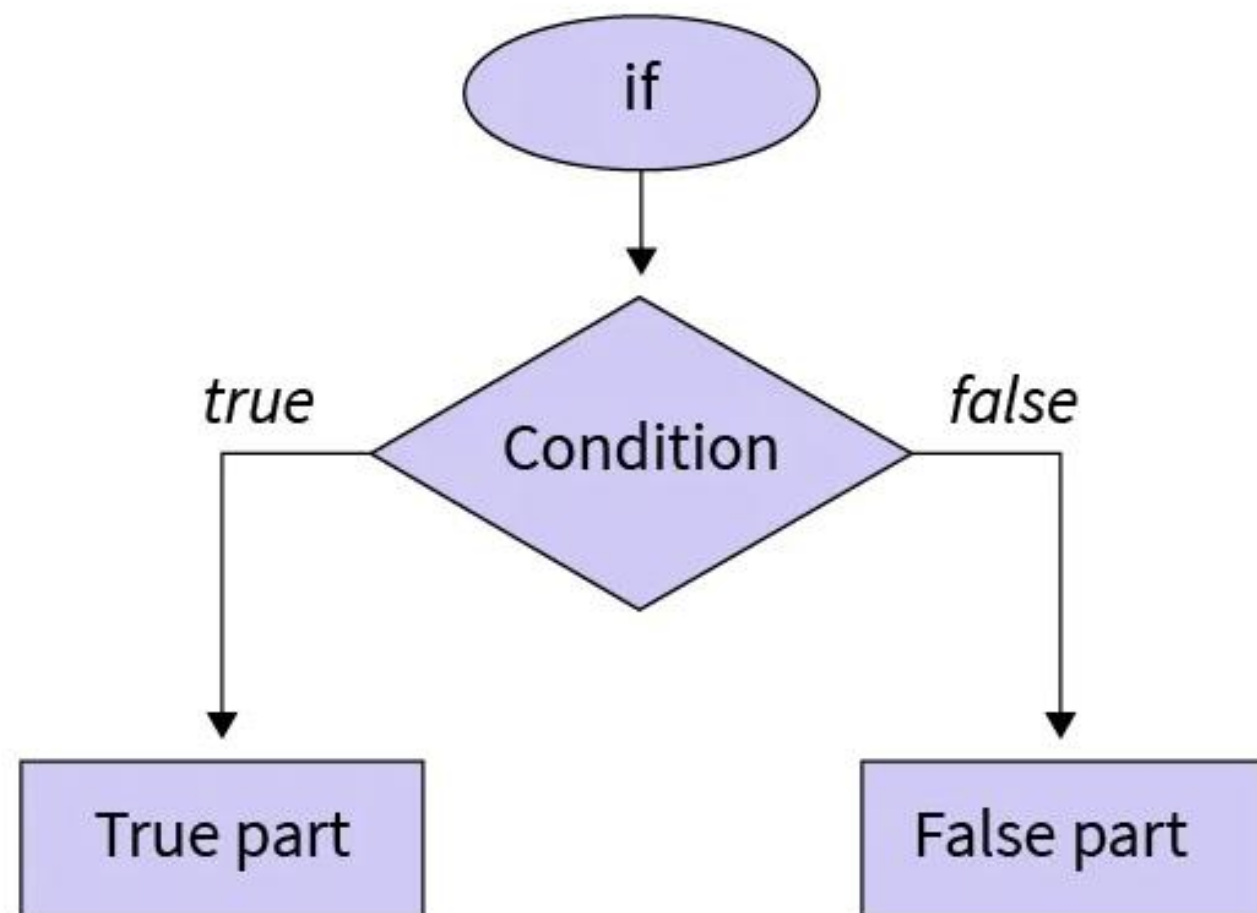
Khai báo biến

2. Giới thiệu về Python và các thư viện GIS

CONTROL STRUCTURES



2. Giới thiệu về Python và các thư viện GIS



Conditional statements

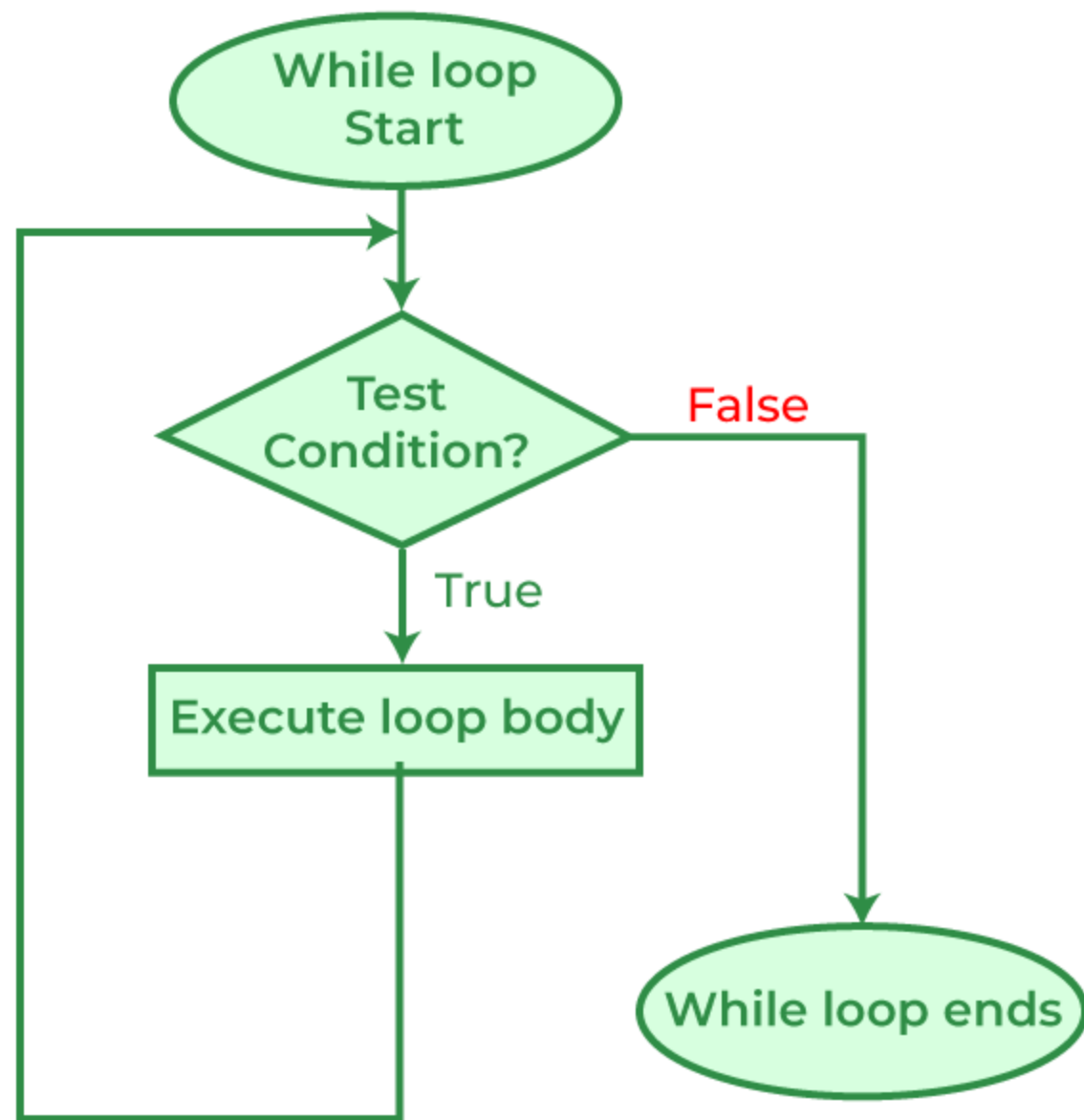
Cấu trúc rẽ nhánh


```
python
```

```
if condition:
    # thực hiện hành động nếu condition đúng
else:
    # thực hiện hành động nếu condition sai
```

Copy

2. Giới thiệu về Python và các thư viện GIS



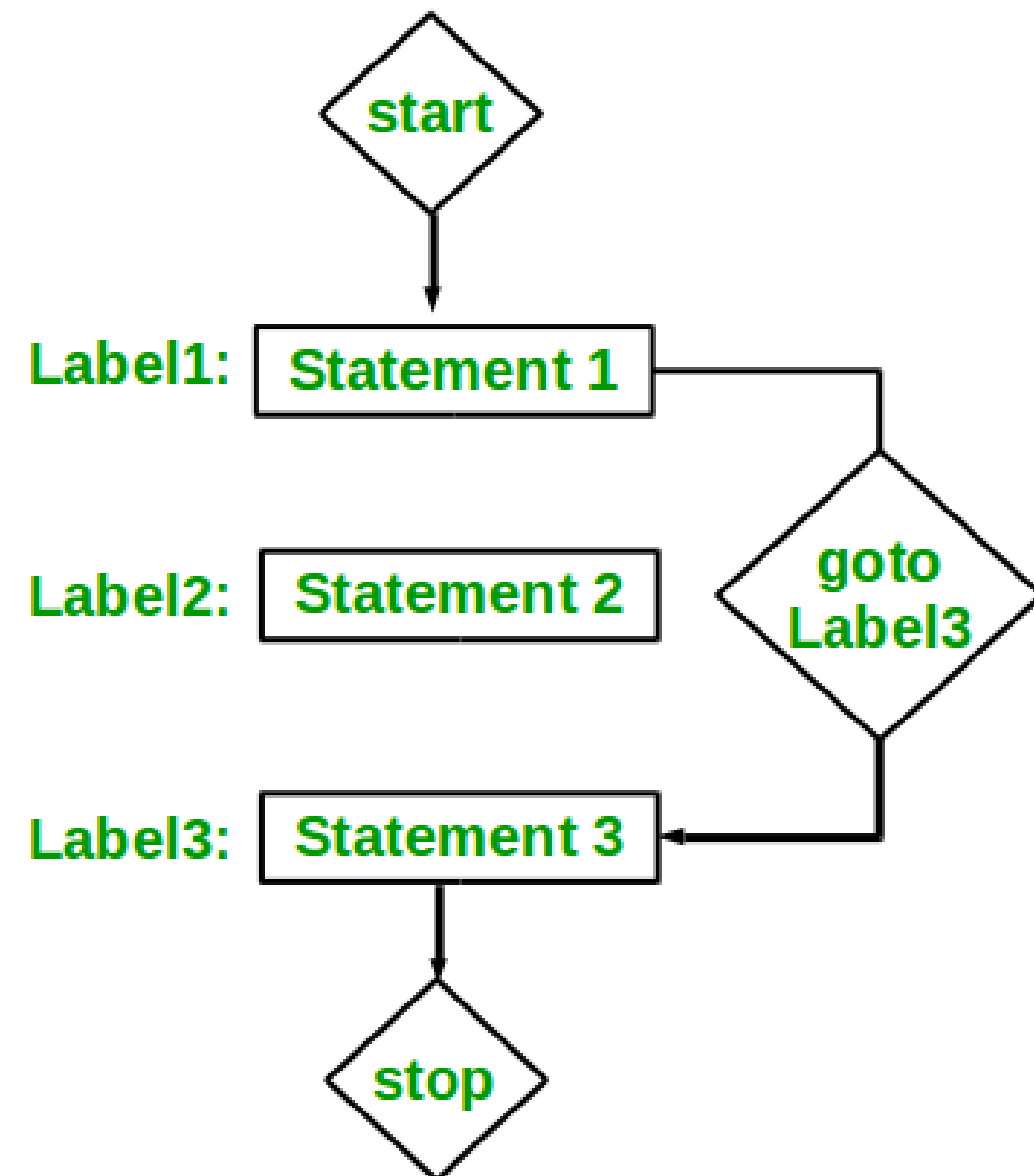
python  Copy

```
for i in range(5):  
    # thực hiện hành động 5 lần  
  
while condition:  
    # thực hiện hành động cho đến khi condition sai
```

Loop statements
Cấu trúc vòng lặp



2. Giới thiệu về Python và các thư viện GIS



python

 Copy

```
for i in range(10):
    if i == 5:
        break # thoát khỏi vòng lặp khi i = 5
    if i == 3:
        continue # bỏ qua các hành động còn lại và tiếp tục vòng lặp
    print(i)

def function():
    # thực hiện một loạt các câu lệnh
    return # thoát khỏi hàm và trả về giá trị
```

Jump statements
Cấu trúc nhảy

2. Giới thiệu về Python và các thư viện GIS

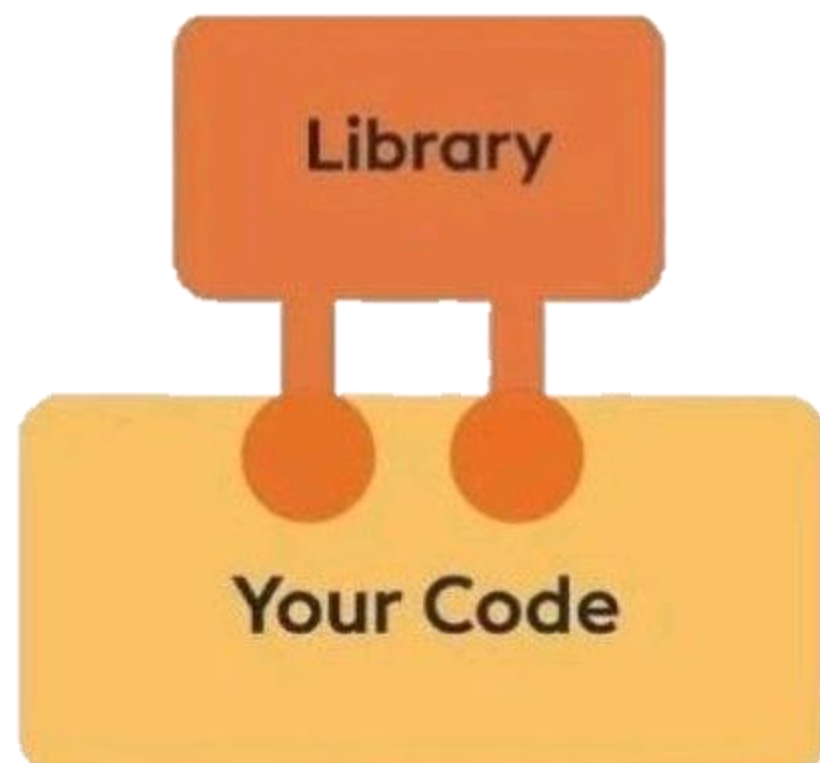
$f(x)$

```
python Copy  
  
def greet(name):  
    print("Hello, " + name + "!")  
  
greet("Steven") # Kết quả: Hello, Steven!
```

Function
Hàm



2. Giới thiệu về Python và các thư viện GIS



Library

Thư viện lập trình

- Tập hợp các mã nguồn đã được viết sẵn
- Thực hiện các tác vụ cụ thể trong ứng dụng
- Gọi các hàm hoặc lớp đã được định nghĩa trong thư viện để thực hiện các chức năng mong muốn



proj.4



2. Giới thiệu về Python và các thư viện GIS



Geospatial Data
Abstraction Library

- Thư viện mã nguồn mở mạnh mẽ và phổ biến trong lĩnh vực xử lý dữ liệu địa lý
- Đọc, ghi và xử lý các tập tin địa lý trong nhiều định dạng khác nhau như **shapefile**, **GeoTIFF**, **JPEG**, **NetCDF**, ...
- Hỗ trợ trích xuất thông tin không gian, chuyển đổi định dạng, phân tích dữ liệu vị trí,...
- Cung cấp các công cụ tiện ích bổ sung như **ogr2ogr**, **gdalinfo**, **gdal_translate**, **gdalwarp**, giúp bạn thực hiện các tác vụ xử lý dữ liệu địa lý một cách dễ dàng từ dòng lệnh.



2. Giới thiệu về Python và các thư viện GIS



- Thư viện mã nguồn mở được sử dụng để đọc và ghi dữ liệu địa lý từ các tập tin định dạng vector như **Shapefile**, **GeoJSON** và **GML**
- Xây dựng dựa trên thư viện **GDAL/OGR**
- Có thể truy cập các thuộc tính, hình học và dữ liệu không gian của các đối tượng và thực hiện các thao tác thêm, sửa và xóa trên chúng



2. Giới thiệu về Python và các thư viện GIS

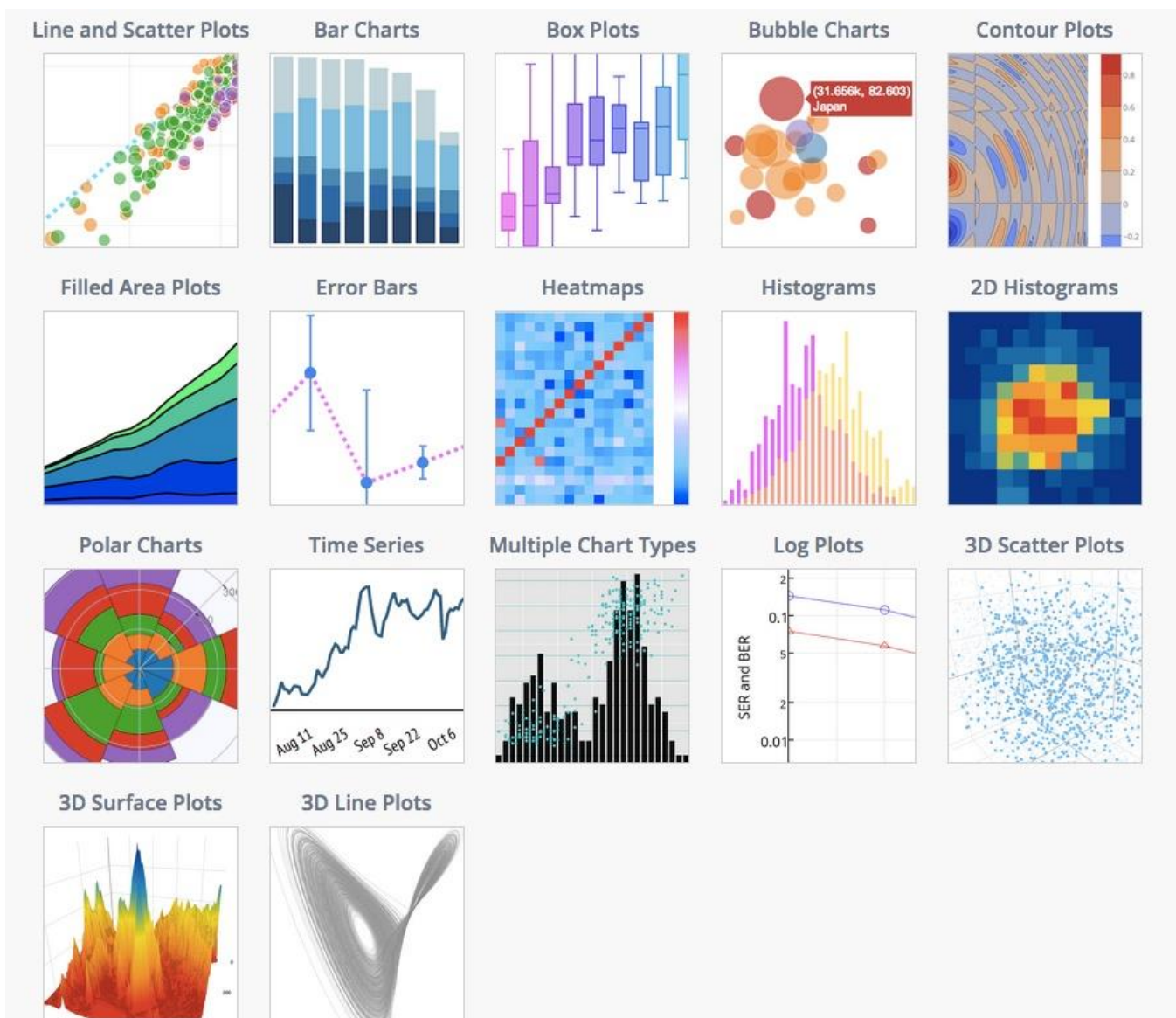


- Thư viện mã nguồn mở trong ngôn ngữ lập trình Python
- làm việc với dữ liệu địa lý dạng bảng (**tabelar**) và dữ liệu địa lý vector
- Kết hợp sức mạnh của hai thư viện phổ biến là **Pandas** và **Shapely** để cung cấp các công cụ mạnh mẽ cho xử lý và phân tích dữ liệu địa lý
- Đọc, ghi và xử lý dữ liệu địa lý **vector**
- Hỗ trợ dữ liệu vector như **Shapefile**, **GeoJSON**, **KML**, và cả cơ sở dữ liệu địa lý **PostgreSQL**,...



2. Giới thiệu về Python và các thư viện GIS

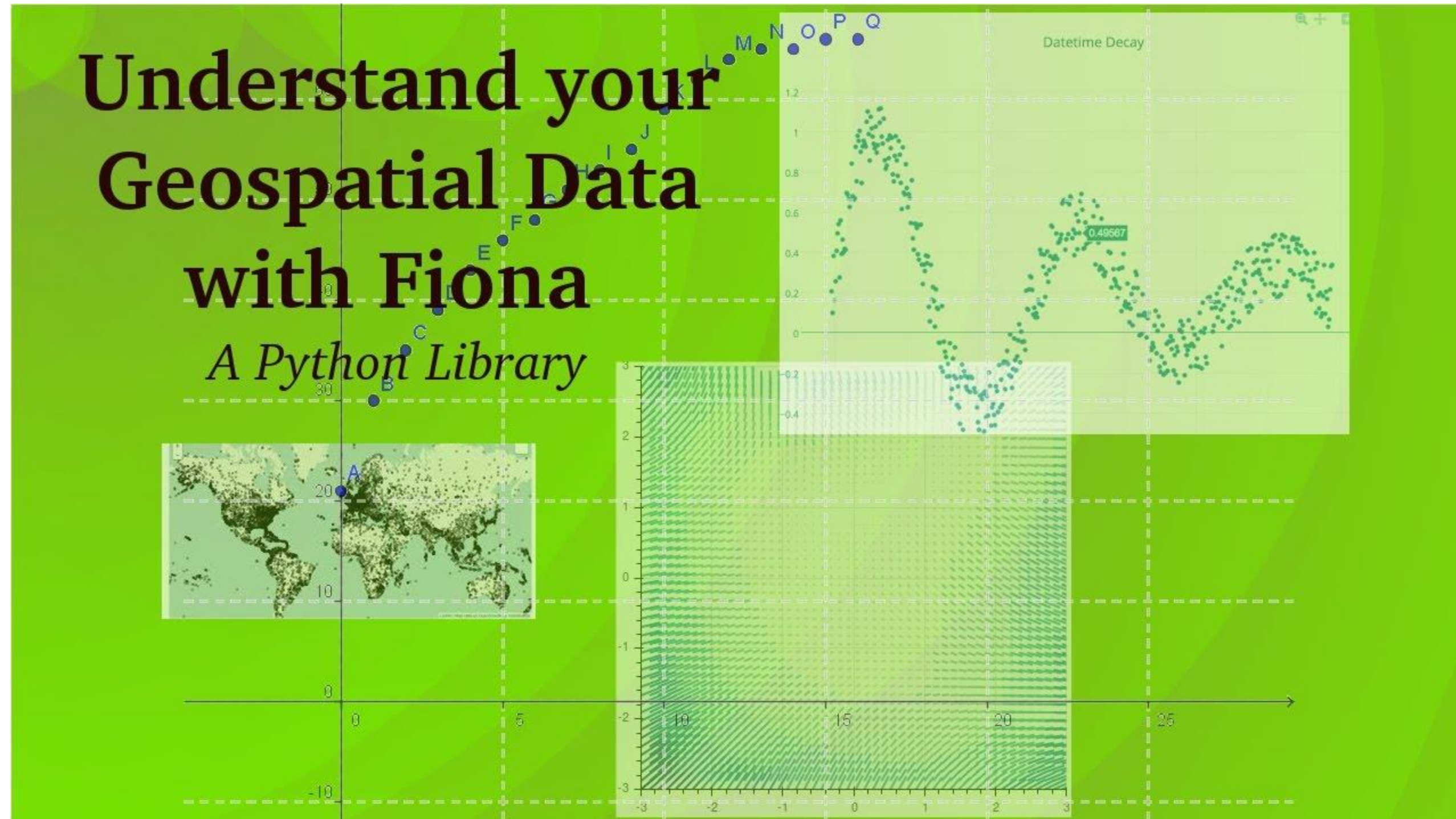
matplotlib



- Thư viện Python rất mạnh mẽ và phổ biến được sử dụng để tạo ra các **biểu đồ và đồ thị** phong phú
- Cho phép bạn **tùy chỉnh** mọi khía cạnh của biểu đồ, từ các thành phần như *tiêu đề, nhãn trục, đơn vị đo, màu sắc, kiểu đường, vùng chụp, kích thước và vị trí* các phần tử trong biểu đồ
- Tương thích với **NumPy và Pandas**, dễ dàng tạo biểu đồ từ dữ liệu được lưu trữ trong các mảng NumPy hoặc DataFrame Pandas.
- Hỗ trợ xuất **PDF, SVG, PNG** và nhiều định dạng khác
- Công cụ quan trọng trong **phân tích dữ liệu và trực quan hóa trong lĩnh vực khoa học dữ liệu và GIS**



3. Đọc và hiển thị dữ liệu không gian địa lý





Geospatial Data
Abstraction Library

```
pip install gdal
```

3. Đọc và hiển thị dữ liệu không gian địa lý

```
python
```

```
from osgeo import ogr

# Đường dẫn đến file shapefile
shapefile_path = "path/to/your/shapefile.shp"

# Mở shapefile
shapefile = ogr.Open(shapefile_path)

# Lấy layer đầu tiên trong shapefile
layer = shapefile.GetLayer(0)

# Lặp qua từng feature trong layer
for feature in layer:
    # Lấy thuộc tính của feature
    attributes = feature.GetField("attribute_name")
    print(attributes)

# Đóng shapefile
shapefile = None
```


3. Đọc và hiển thị dữ liệu không gian địa lý



```
pip install Fiona
```

```
python
```

```
import fiona

# Đường dẫn đến file shapefile
shapefile_path = "path/to/your/shapefile.shp"

# Mở shapefile để đọc
with fiona.open(shapefile_path, 'r') as shapefile:
    # Lấy thông tin về thuộc tính và hình học của từng đối tượng
    for feature in shapefile:
        attributes = feature['properties']
        geometry = feature['geometry']

        # Xử lý dữ liệu địa lý theo nhu cầu của bạn
        # ...

# Kết thúc việc đọc shapefile
```

3. Đọc và hiển thị dữ liệu không gian địa lý



```
pip install geopandas matplotlib
```

python

```
import geopandas as gpd
import matplotlib.pyplot as plt

# Đường dẫn đến file shapefile
shapefile_path = "path/to/your/shapefile.shp"

# Đọc dữ liệu từ shapefile bằng GeoPandas
data = gpd.read_file(shapefile_path)

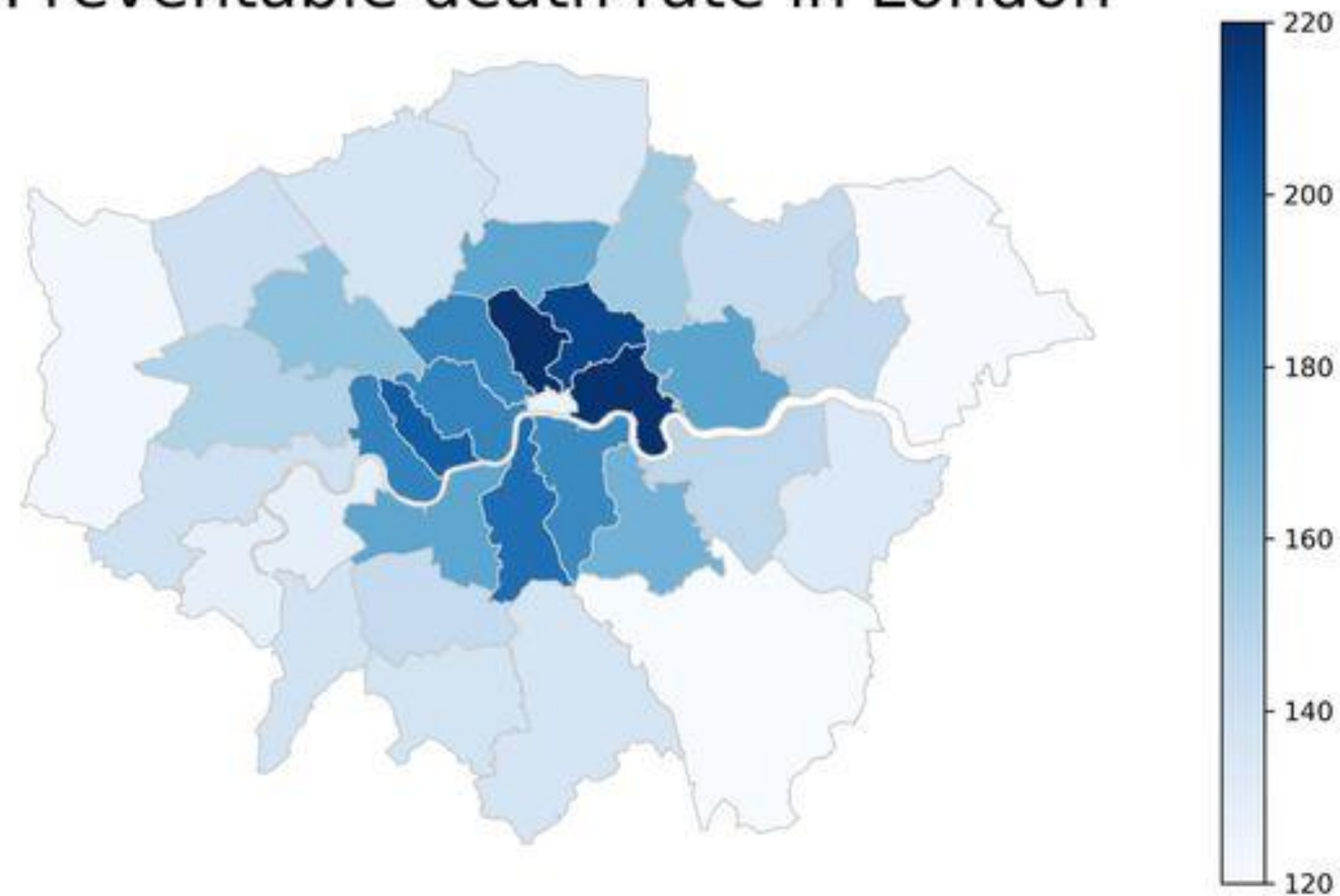
# Vẽ bản đồ với Matplotlib
fig, ax = plt.subplots()
data.plot(ax=ax)

# Vẽ biểu đồ
data.plot.bar(x='column_name', y='column_name')

# Hiển thị bản đồ và biểu đồ
plt.show()
```

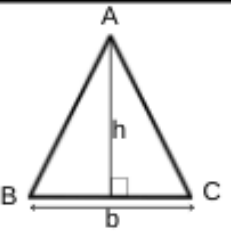

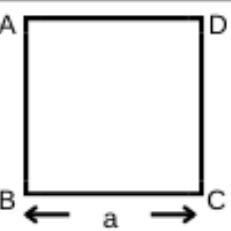
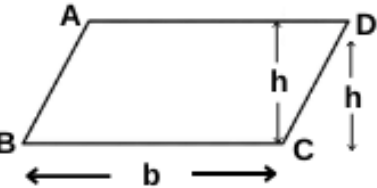
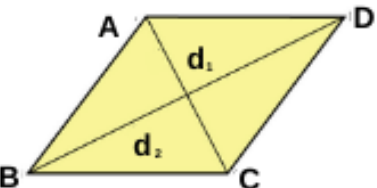
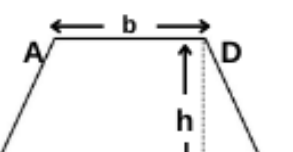
3. Đọc và hiển thị dữ liệu không gian địa lý

Preventable death rate in London



Source: London Datastore, 2014

4. Xử lý dữ liệu không gian địa lý

Triangle		Area (A) = $\frac{1}{2} (b \times h)$ here, b = base, h = height
Rectangle		Area (A) = $w \times l$ here, w = width, l = length
Square		Area (A) = a^2 here, a = side
Parallelogram		Area (A) = $b \times h$ here, b = base, h = height
Rhombus		Area (A) = $\frac{d_1 \times d_2}{2}$ here, d ₁ and d ₂ are the diagonals
Trapezoid		Area (A) = $\frac{1}{2} (a + b) \times h$ here, a = long base

python

```
import geopandas as gpd

# Đường dẫn đến tệp GeoJSON
geojson_path = "path/to/your/geojson.geojson"

# Đọc dữ liệu từ GeoJSON bằng GeoPandas
data = gpd.read_file(geojson_path)

# Tính toán diện tích của các polygon
data['area'] = data.geometry.area

# In kết quả
print(data['area'])
```

Tính diện tích

4. Xử lý dữ liệu không gian địa lý

python

```
import geopandas as gpd
from pyproj import CRS

# Đường dẫn đến tệp shapefile hoặc GeoJSON
input_path = "path/to/your/input/file.shp"

# Đọc dữ liệu đầu vào bằng GeoPandas
data = gpd.read_file(input_path)
```

```
# Hiển thị hệ tọa độ ban đầu
print("Hệ tọa độ ban đầu:")
print(data.crs)

# Tạo một đối tượng CRS mới cho hệ tọa độ đích
target_crs = CRS.from_epsg(4326) # Ví dụ: WGS84

# Chuyển đổi hệ tọa độ
data = data.to_crs(target_crs)

# Hiển thị hệ tọa độ sau khi chuyển đổi
print("Hệ tọa độ sau khi chuyển đổi:")
print(data.crs)
```

Chuyển đổi hệ tọa độ

5. Phân tích dữ liệu không gian địa lý

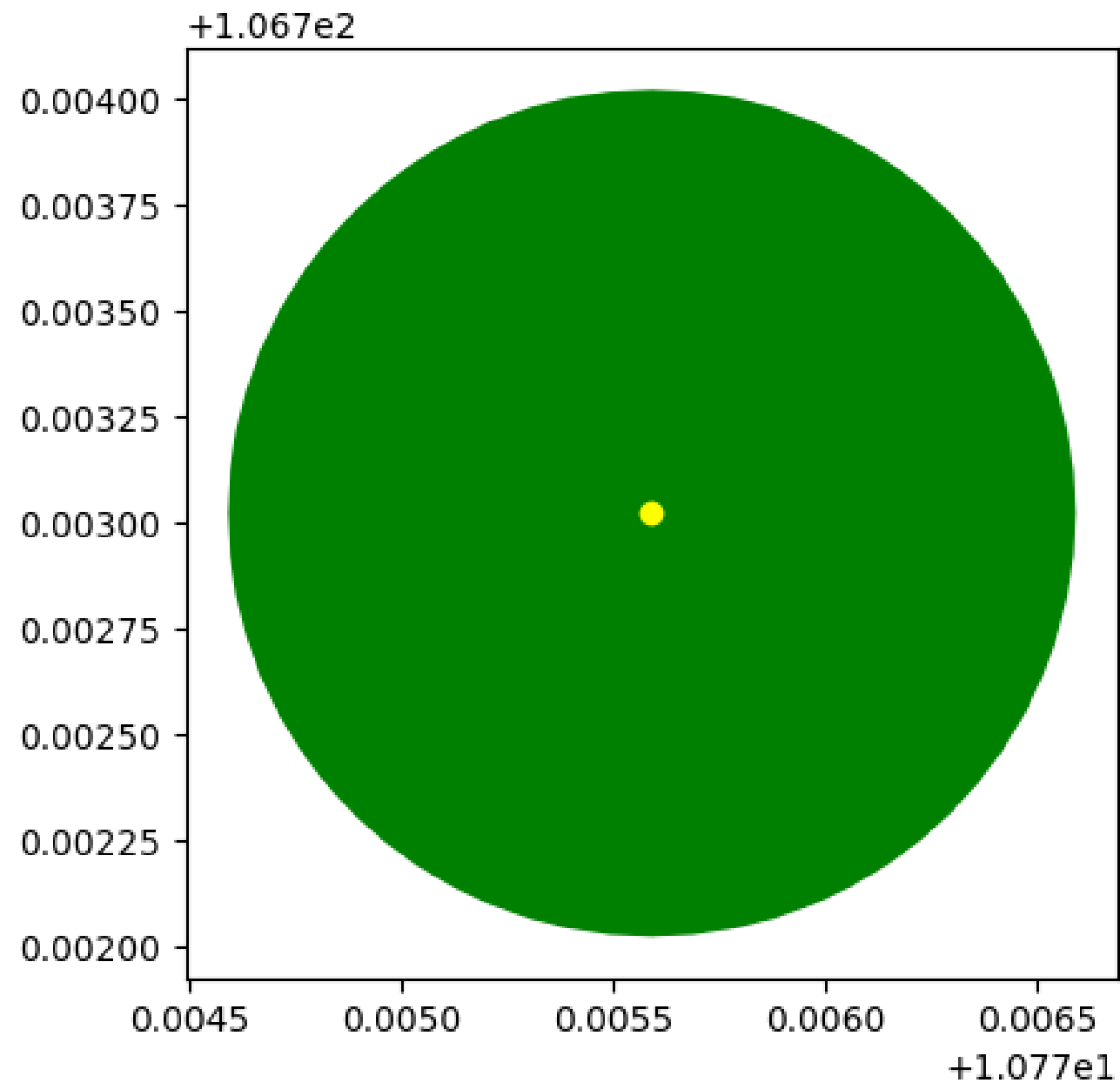


Tính khoảng cách

```
1  # Tính khoảng cách giữa 2 điểm
2  |
3  import geopandas as gpd
4  from shapely.geometry import Point
5
6  # Tạo 2 điểm
7  point1 = Point(-122.224167, 37.176389)
8  point2 = Point(-122.419416, 37.774929)
9
10 # Tính khoảng cách giữa 2 điểm
11 distance = point1.distance(point2)
12
13 # Show kết quả
14 print(distance)
15
```

0.6295810540359369

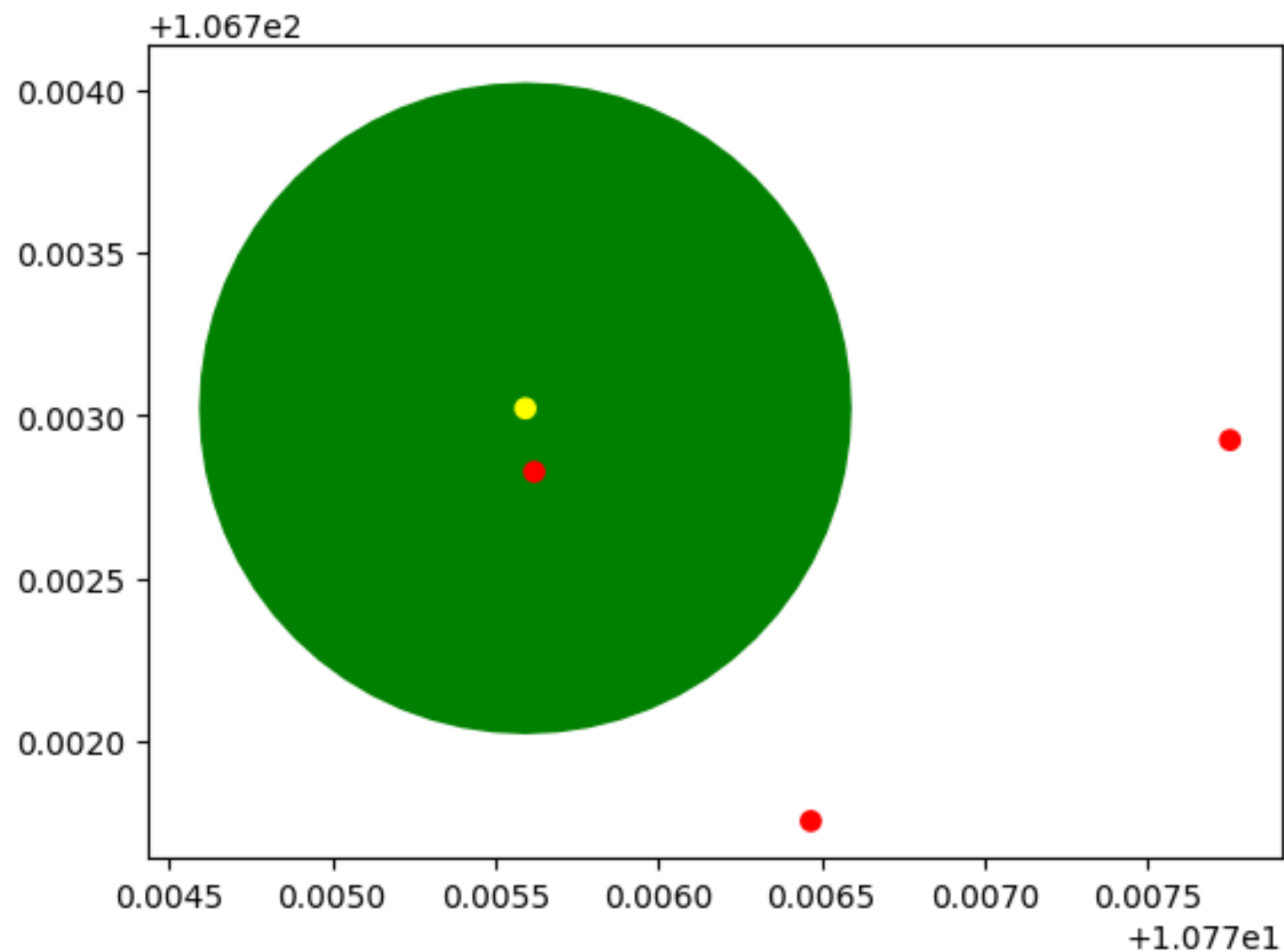
5. Phân tích dữ liệu không gian địa lý



Tạo buffer

```
2
3 import pandas as pd
4 import geopandas as gpd
5 from shapely.geometry import Point, Polygon
6
7 # Tạo điểm
8 point = Point(10.775592424689332, 106.70302168729125)
9
10 # Tạo vùng đệm xung quanh điểm với bán kính 1000 mét
11 buffer = point.buffer(0.001)
12
13 # Tạo GeoDataFrame từ điểm và vùng đệm
14 geometry = [point, buffer]
15 gdf = gpd.GeoDataFrame(geometry=geometry, columns=['geometry'])
16
17 # Hiển thị điểm và vùng đệm trên bản đồ
18 gdf.plot(color=['yellow', 'green'])
19
```

5. Phân tích dữ liệu không gian địa lý



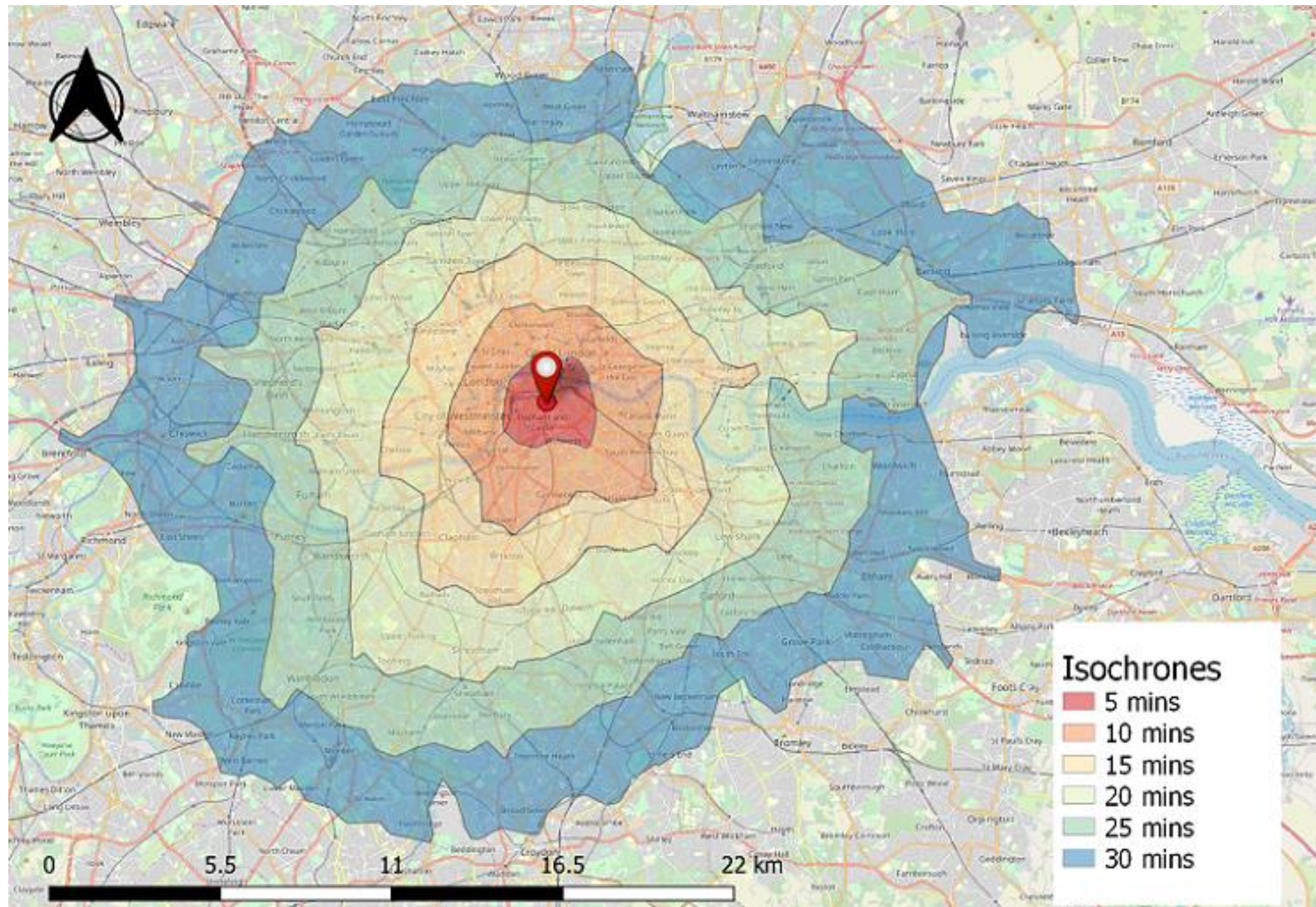
Tìm vị trí nằm trong buffer

```

14 # Tạo điểm từ tọa độ
15 point1 = Point(10.775619761168818, 106.70282689719542)
16 point2 = Point(10.776467190801428, 106.70175555166836)
17 point3 = Point(10.777751998919879, 106.70292429224334)
18
19 # Kiểm tra xem các điểm có nằm trong vùng đệm không
20 is_inside1 = buffer.contains(point1)
21 is_inside2 = buffer.contains(point2)
22 is_inside3 = buffer.contains(point3)
23
24 # Hiển thị vị trí các điểm so với vùng đệm
25 print(f"Point 1 is inside the buffer: {is_inside1}")
26 print(f"Point 2 is inside the buffer: {is_inside2}")
27 print(f"Point 3 is inside the buffer: {is_inside3}")
28
29 # Vẽ các điểm và vùng đệm trên bản đồ
30
31 geometry = [point, point1, point2, point3, buffer]
32 gdf = gpd.GeoDataFrame(geometry=geometry, columns=['geometry'])
33
34 gdf.plot(color=['yellow', 'red', 'red', 'red', 'green'])
35

```


5. Phân tích dữ liệu không gian địa lý



Isochrones

5. Phân tích dữ liệu không gian địa lý

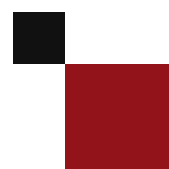
Ví dụ

https://github.com/soiqualang/colab/blob/master/OpenGIS_Vietnam_Work_with_geospatial_data.ipynb





OPENGIS



THANK YOU

