SimCenter NHERI

Center for Computational Modeling and Simulation

Data Gathering, Web Automation & GIS

Wael Elhaddad

NHERI SimCenter

Programming Bootcamp 2019 (Day 4)

NSF award: CMMI 1612843

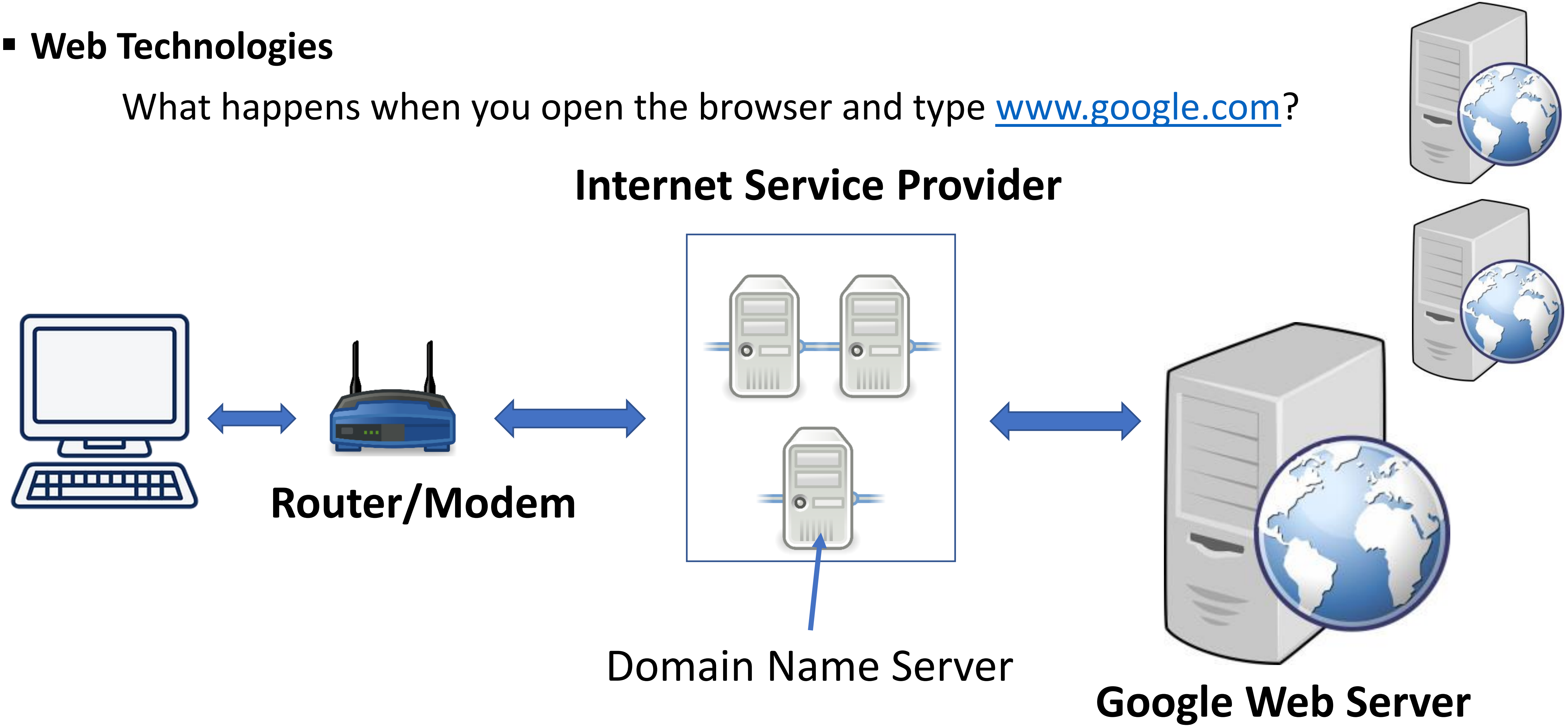
Outline (Day 4)

- **Introduction**
  - Web Technologies & HTTP
  - Web APIs (e.g. REST)
  - JSON
  - Relevant Web Services (Exposure and Hazard Data)
- **Web Automation using Selenium**
  - Tax Assessor’s Data (e.g. Anchorage, Memphis, NJ…etc.)
- **Visualization & Analysis in GIS**
  - Introduction to QGIS
- **AI Applications**
  - Computer Vision
  - Data Enhancement (SURF)
- **Regional Data Gathering Exercise**

Introduction

- **Web Technologies**

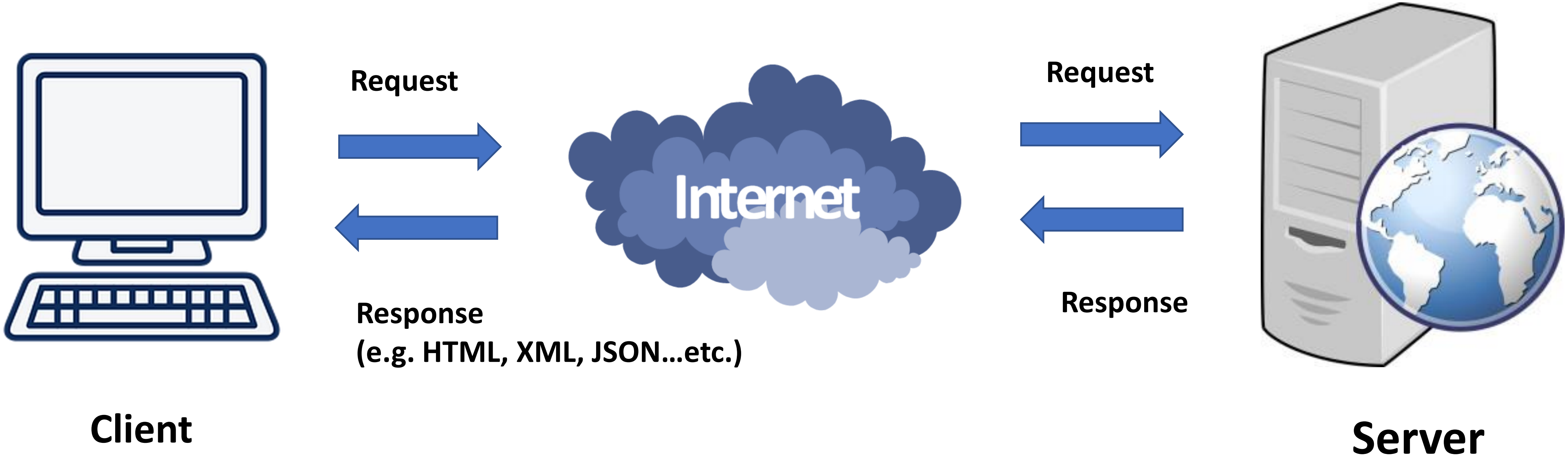
What happens when you open the browser and type www.google.com?



HTTP

- **Hypertext Transfer Protocol (HTTP)**

What happens when you open the browser and type www.google.com?
Then, what happens when you search for something?



Request

Response
(e.g. HTML, XML, JSON…etc.)

Request

Response

Client

Server

Web API

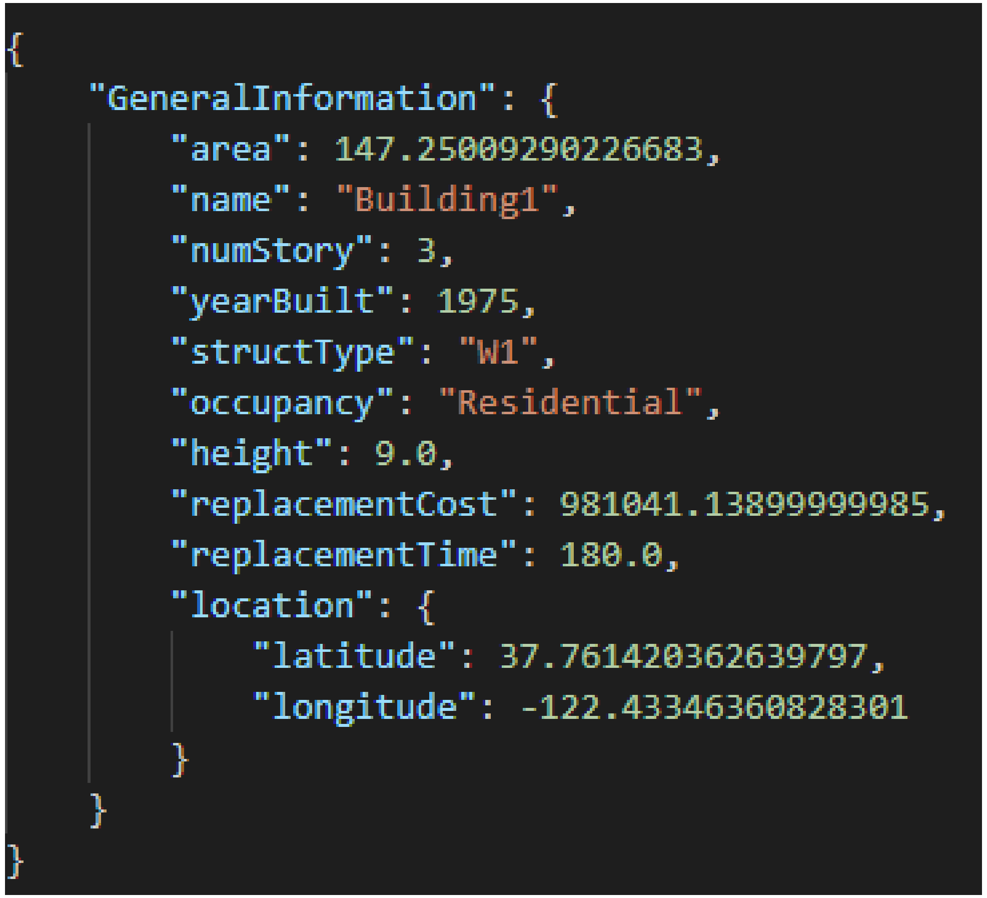
- **Application Programming Interface (API)**
  - Defines a set of methods for communication
- **Web API**
  - Defines the methods for communication between a client and a server
- **REST API**
  - Set some standard rules for web communication (e.g. HTTP)
  - Four methods are defined (GET, POST, PUT, DELETE)
    - **GET: to retrieve data**
    - POST: to create data
    - PUT: to modify data
    - DELETE: to delete data

JSON

- **JavaScript Object Notation**

  File format to describe data in human-readable form

- **The format provides attribute-value pairs**
- **Data Types**
  - Number
  - String
  - Boolean
  - Array
  - Objects
- **Disadvantage:** large size (not efficient)



```
{
    "GeneralInformation": {
        "area": 147.25009290226683,
        "name": "Building1",
        "numStory": 3,
        "yearBuilt": 1975,
        "structType": "W1",
        "occupancy": "Residential",
        "height": 9.0,
        "replacementCost": 981041.13899999985,
        "replacementTime": 180.0,
        "location": {
            "latitude": 37.761420362639797,
            "longitude": -122.43346360828301
        }
    }
}
```

Web Services

- **ATC API**
  - Hazard by Location API: https://hazards.atcouncil.org/api
  - Example: https://api-hazards.atcouncil.org/wind.json?lat=35.4676&lng=-97.5164

- **USGS APIs (NSHMP-ws)**
  - Hazard Service: https://earthquake.usgs.gov/nshmp-haz-ws/
  - Design Maps: https://earthquake.usgs.gov/ws/designmaps/

- **FDNS**
  - Earthquake Catalog: https://earthquake.usgs.gov/fdsnws/event/1/
  - Examples:

Ridgecrest, CA

https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2019-01-01&endtime=2019-07-24&latitude=35.6225&longitude=-117.6709&maxradiuskm=50&minmagnitude=6

Anchorage, AK

https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2018-11-30&endtime=2018-12-01&latitude=61.2181&longitude=-149.9003&maxradiuskm=50&minmagnitude=6

Web Services

- **DataSF Portal**
  - Tall Building Inventory
    - Map: https://data.sfgov.org/Housing-and-Buildings/Map-of-Tall-Buildings/xnf9-cudk
    - Inventory: https://data.sfgov.org/Housing-and-Buildings/Tall-Building-Inventory/5kya-mfst
    - Request: https://data.sfgov.org/resource/5kya-mfst.json

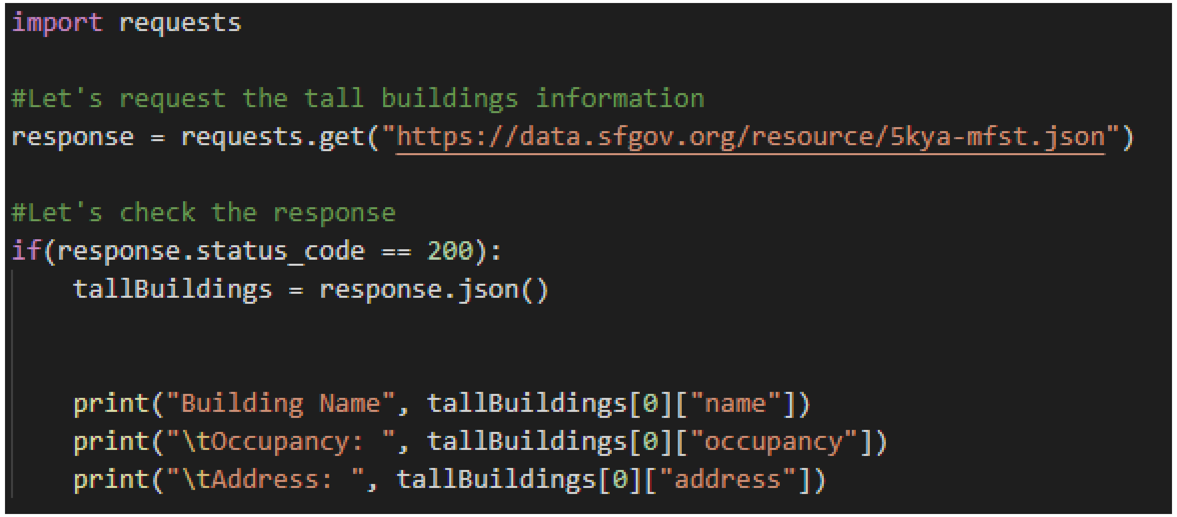
- **Census API**
  - https://www.census.gov/data/developers/data-sets.html

Python Libraries

- **Requests**
  - Submit HTTP requests and get the response
  - Documentation: https://2.python-requests.org/en/master/
- **Selenium**
  - Webdriver to control the web browser
  - Documentation: https://selenium-python.readthedocs.io/getting-started.html
- **BeautifulSoup, lxml**
  - Packages to facilitate processing html
  - Documentation: https://www.crummy.com/software/BeautifulSoup/bs4/doc/#quick-start
- **Census, US**
  - Python package to facilitate querying Census data
  - Documentation: https://github.com/datamade/census

Requests Demo

- **Using requests we will get a list of tall buildings and print one of them to the screen**



```
import requests

#Let's request the tall buildings information
response = requests.get("https://data.sfgov.org/resource/5kya-mfst.json")

#Let's check the response
if(response.status_code == 200):
    tallBuildings = response.json()


    print("Building Name", tallBuildings[0]["name"])
    print("\tOccupancy: ", tallBuildings[0]["occupancy"])
    print("\tAddress: ", tallBuildings[0]["address"])
```

- **Exercise 1**

  Print to the screen the list of buildings including relevant information about the building like structure type, occupancy, number of stories, , total area.

- **Exercise 2**

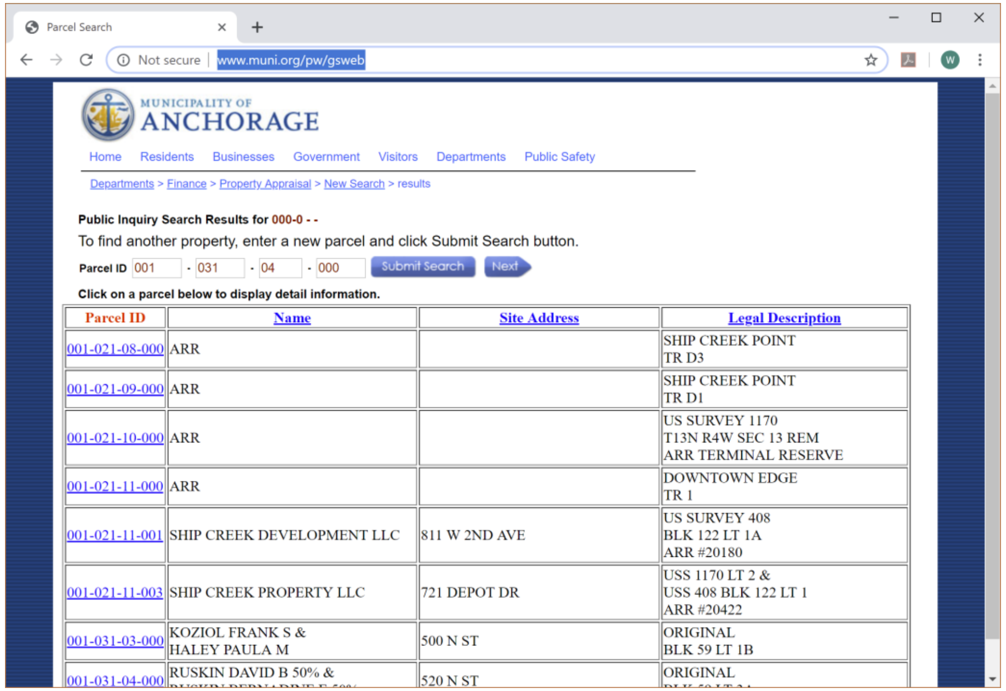
  Write the data from exercise 1 into a csv text file, including the latitude and longitude

- **Exercise 3**

  Can we get PGA from USGS API for each building and include it in the output file

Selenium Demo

- **Using Selenium, we automate browsing the tax assessor’s website**



Departments > Finance > Property Appraisal > New Search > results

Public Inquiry Search Results for 000-0 - -

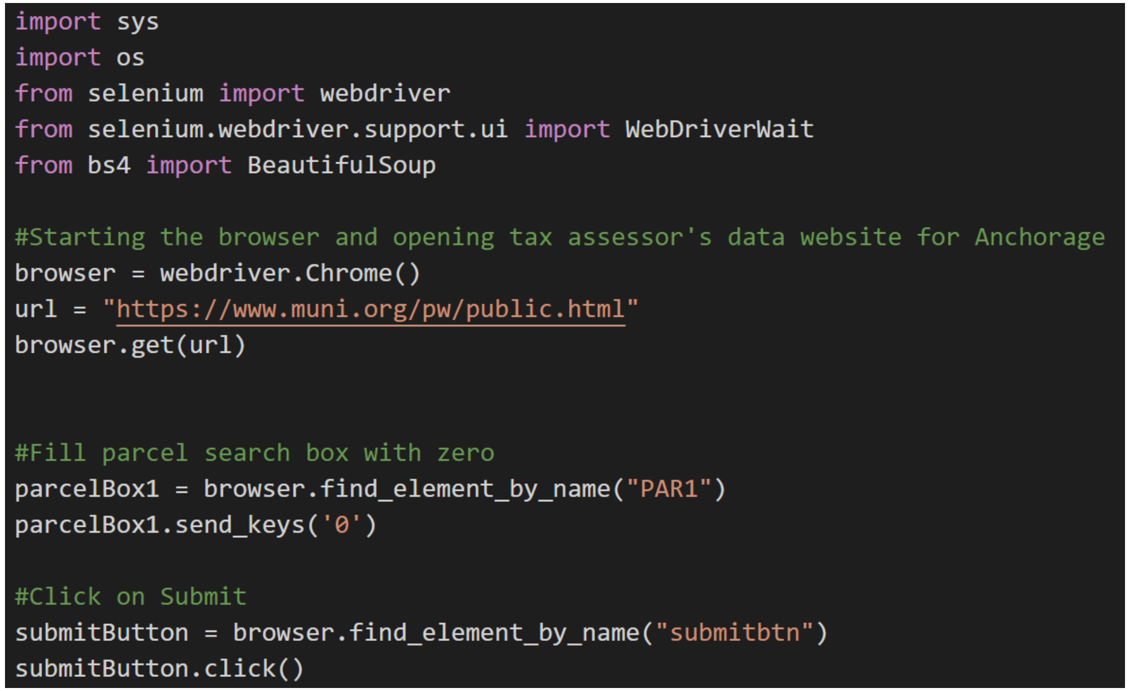
To find another property, enter a new parcel and click Submit Search button.

Parcel ID 001 - 031 - 04 - 000

Click on a parcel below to display detail information.

Parcel ID	Name	Site Address	Legal Description
001-021-08-000	ARR		SHIP CREEK POINT TR D3
001-021-09-000	ARR		SHIP CREEK POINT TR D1
001-021-10-000	ARR		US SURVEY 1170 T13N R4W SEC 13 REM ARR TERMINAL RESERVE
001-021-11-000	ARR		DOWNTOWN EDGE TR 1
001-021-11-001	SHIP CREEK DEVELOPMENT LLC	811 W 2ND AVE	US SURVEY 408 BLK 122 LT 1A ARR #20180
001-021-11-003	SHIP CREEK PROPERTY LLC	721 DEPOT DR	USS 1170 LT 2 & USS 408 BLK 122 LT 1 ARR #20422
001-031-03-000	KOZIOL FRANK S & HALEY PAULA M	500 N ST	ORIGINAL BLK 59 LT 1B
001-031-04-000	RUSKIN DAVID B 50% &	520 N ST	ORIGINAL

http://www.muni.org/pw/gsweb



```python
import sys
import os
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from bs4 import BeautifulSoup

#Starting the browser and opening tax assessor's data website for Anchorage
browser = webdriver.Chrome()
url = "https://www.muni.org/pw/public.html"
browser.get(url)


#Fill parcel search box with zero
parcelBox1 = browser.find_element_by_name("PAR1")
parcelBox1.send_keys('0')

#Click on Submit
submitButton = browser.find_element_by_name("submitbtn")
submitButton.click()
```

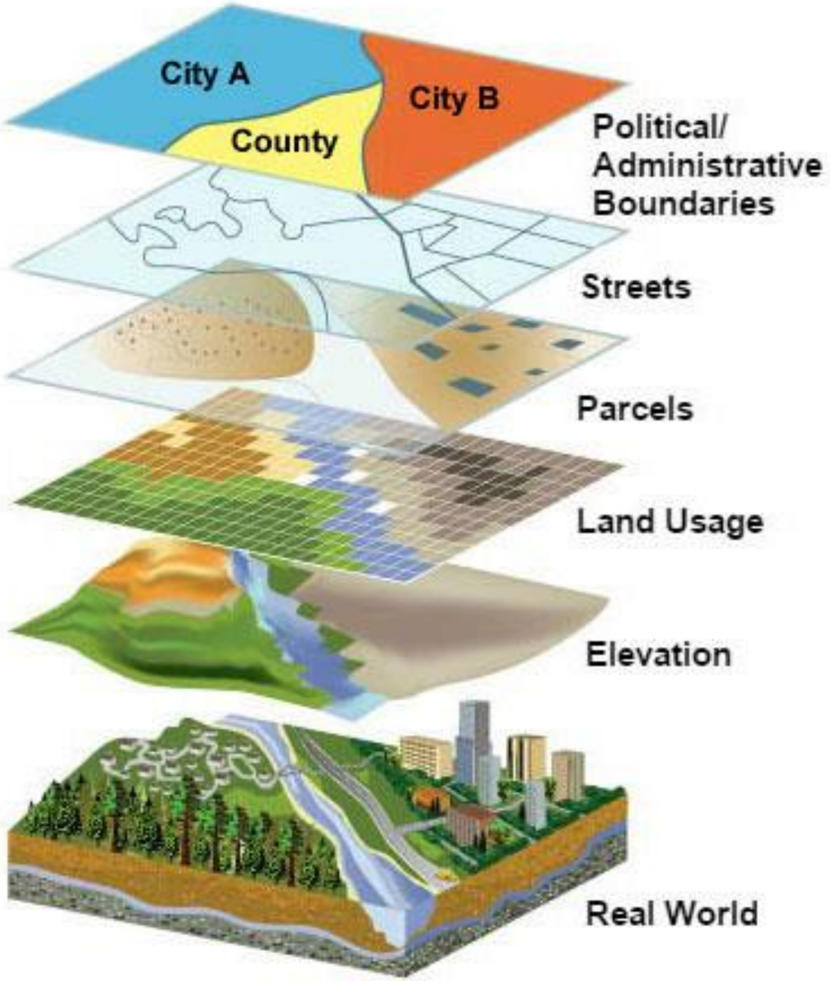
- **Exercise 4**: Can we extract more information about these buildings e.g. number of stories, year built, area...etc.
- **Exercise 5**: Let's do the same for Memphis, Tennessee

GIS Introduction

- **GIS stands for Geographical Information System**

- **Information is represented in a set of layers**

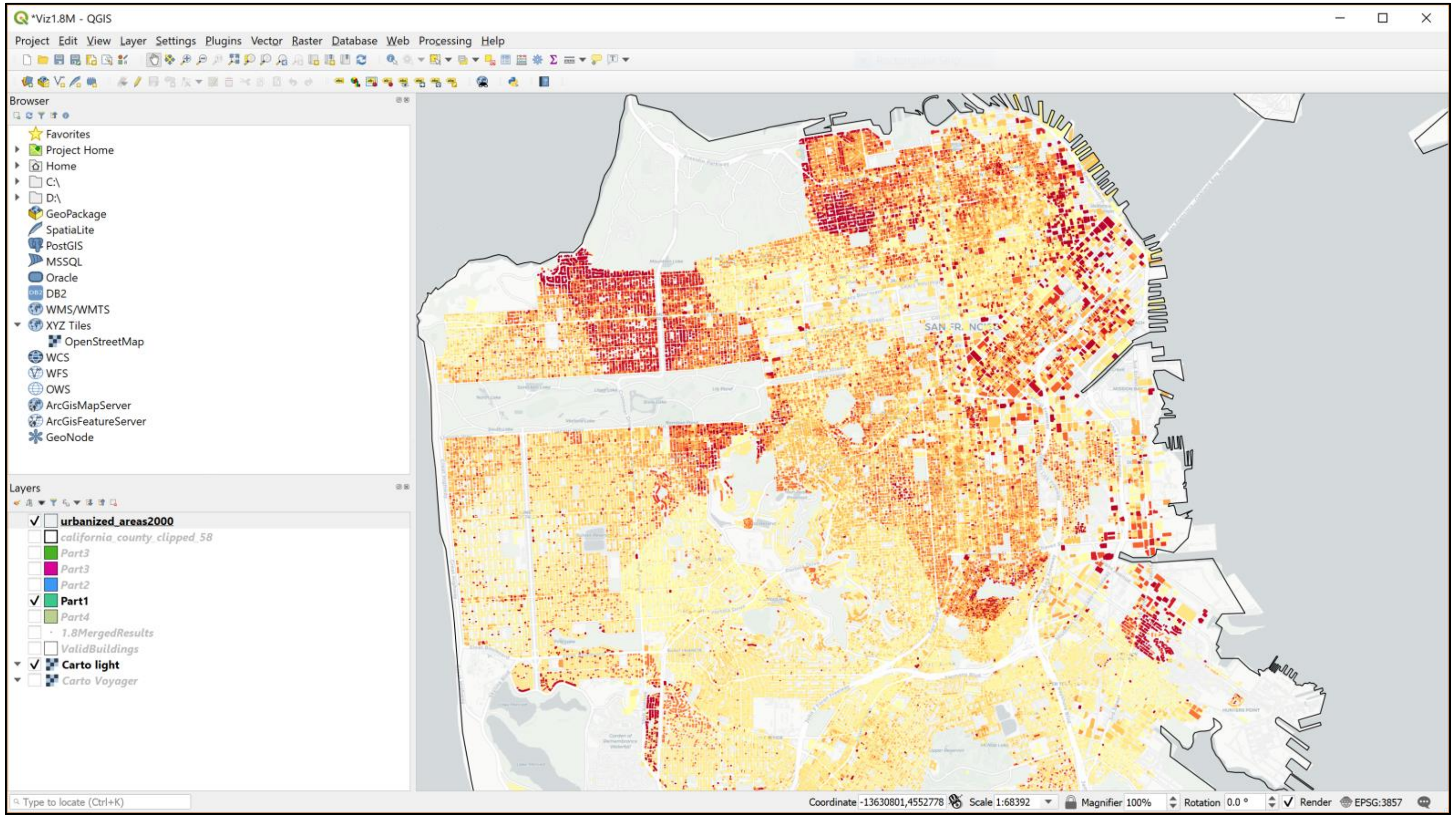
- **GIS platforms can help you:**
  - Generate maps & visualize geospatial data
  - Transform and edit data
  - Perform spatial analysis on the data (e.g. spatial joins)



GIS Software

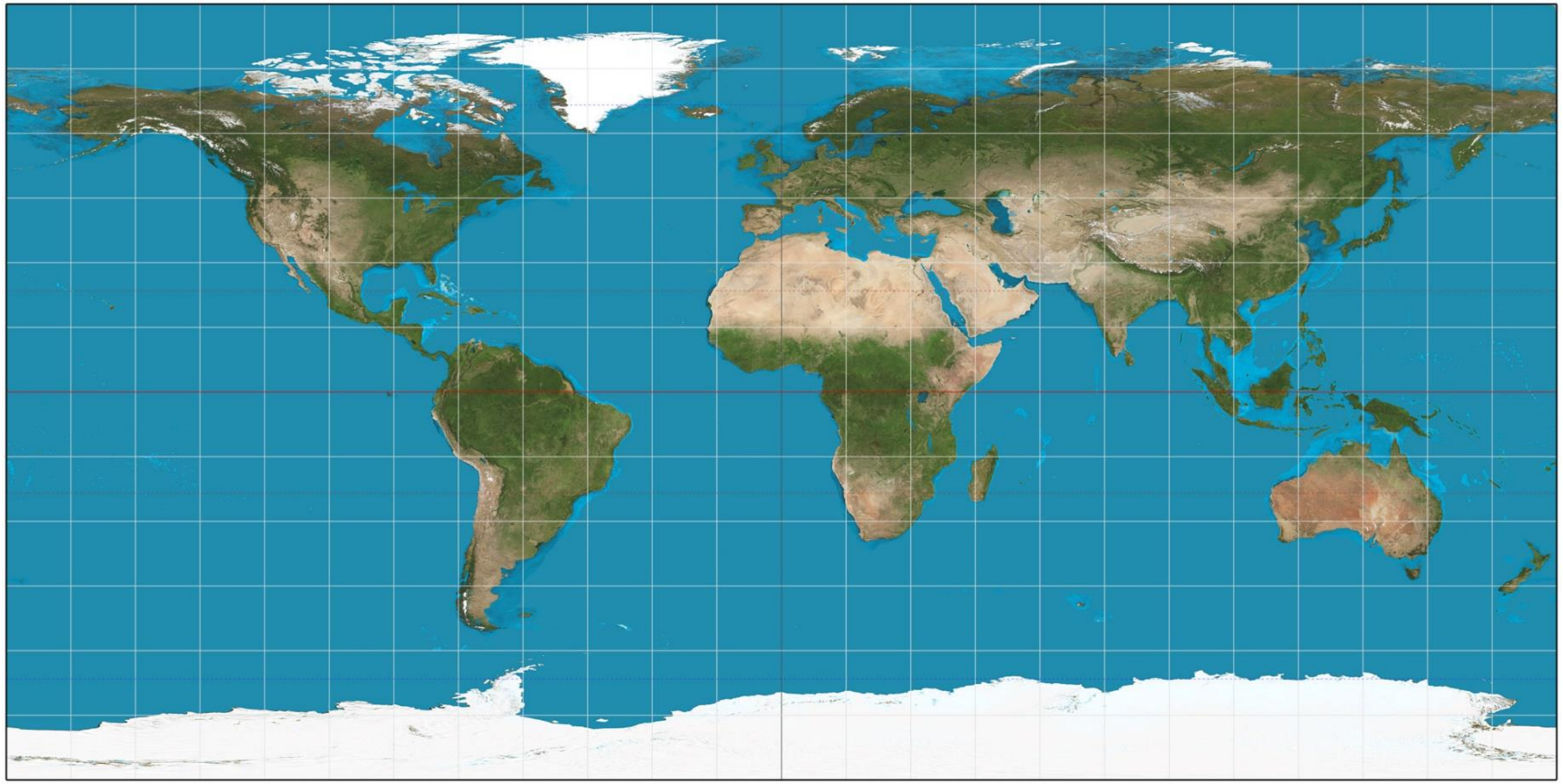
- **ArcGIS (Commercial)**
  - Desktop & Online (cloud/web-based)
  - Many universities provide access to student, staff and faculty

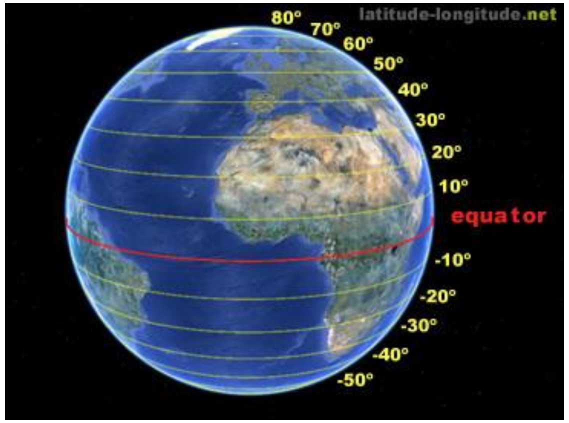
- **QGIS (Free & Open-Source)**
  - Desktop only
  - Easy to use
  - Extensible using Python

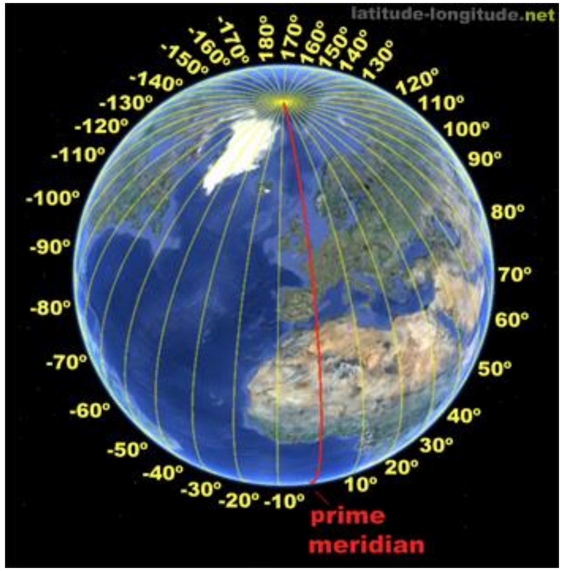


GIS Basics

- **Coordinate Systems (CRS)**
  - Map Projection
  - There are many systems (e.g. Local CRS)
  - Latitude and Longitude (**WGS84 EPSG:4326**)







GIS Basics

- **Two Types of Data Layers**

  - **Vector Data**

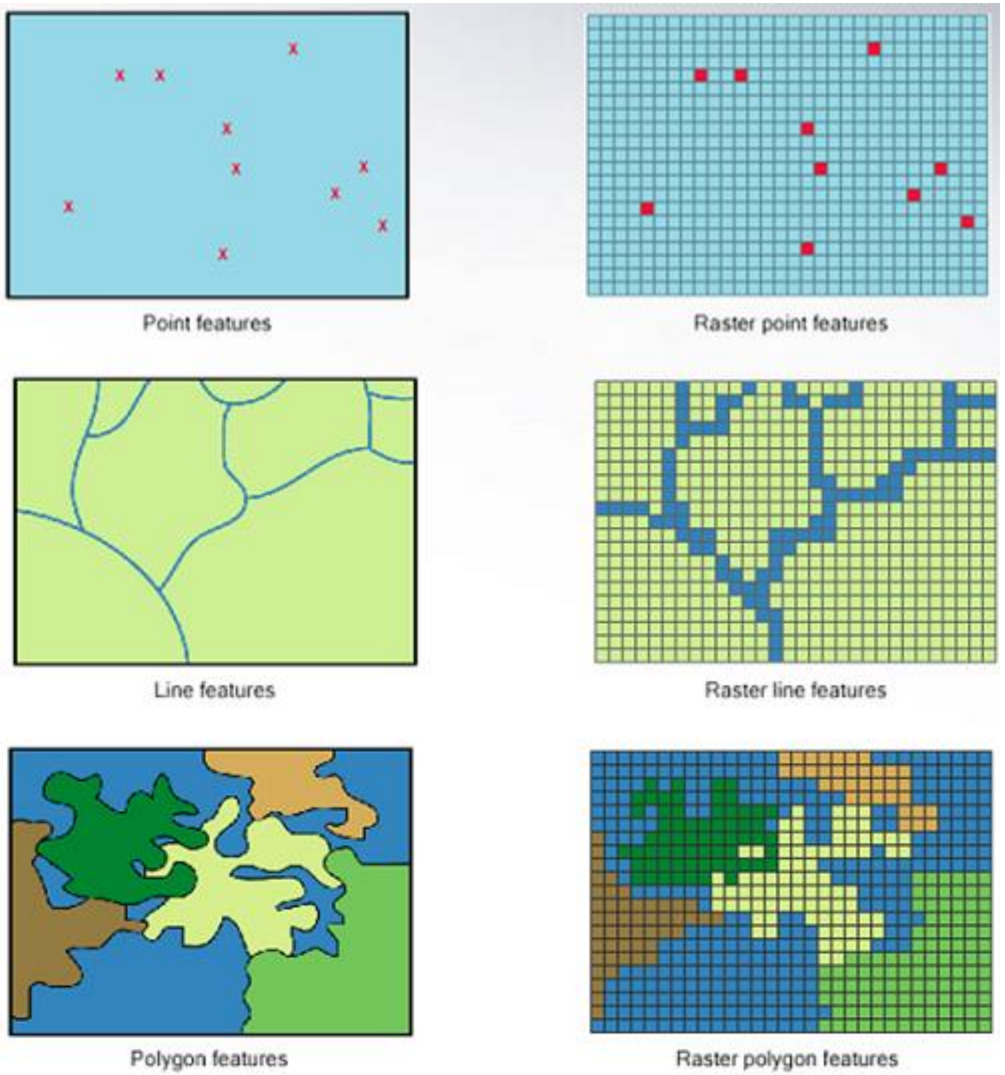
    Suitable for discrete and distinct feature

    e.g. Buildings, Roads…etc

  - **Raster Data**:

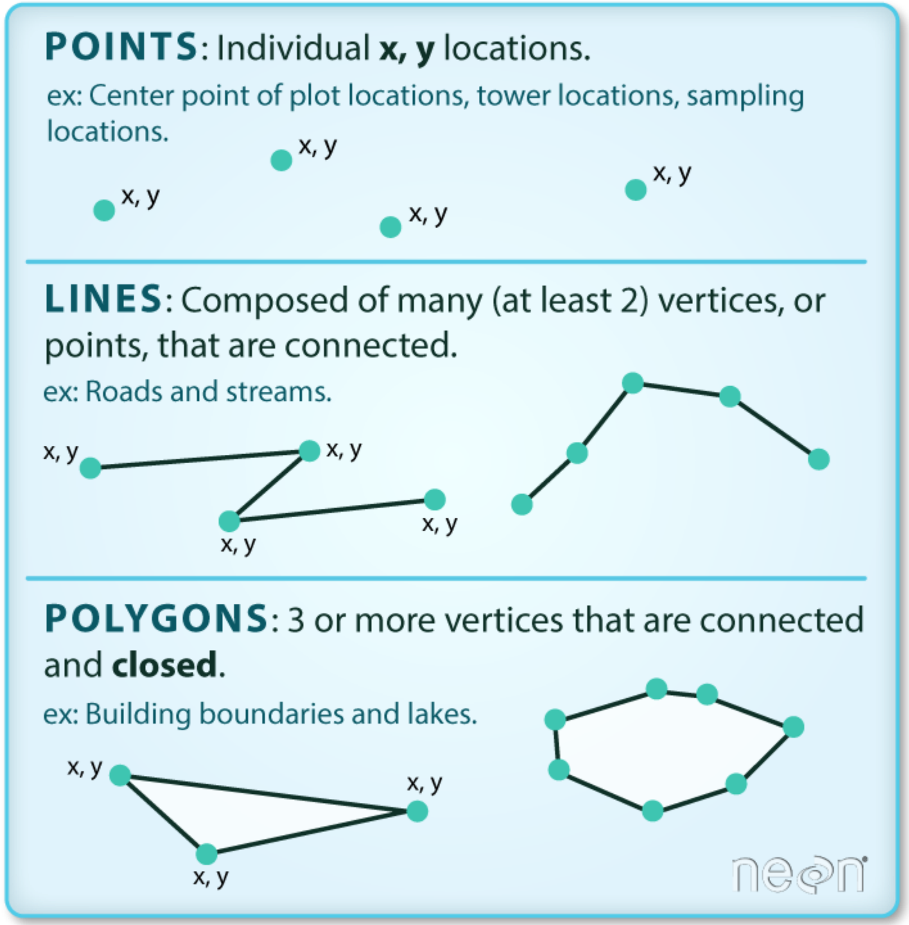
    Suitable for continuous features

    e.g. elevation, temperature, soil properties….etc



GIS Basics

- **Vector Data:** Geometry and Attributes



POINTS: Individual **x, y** locations.

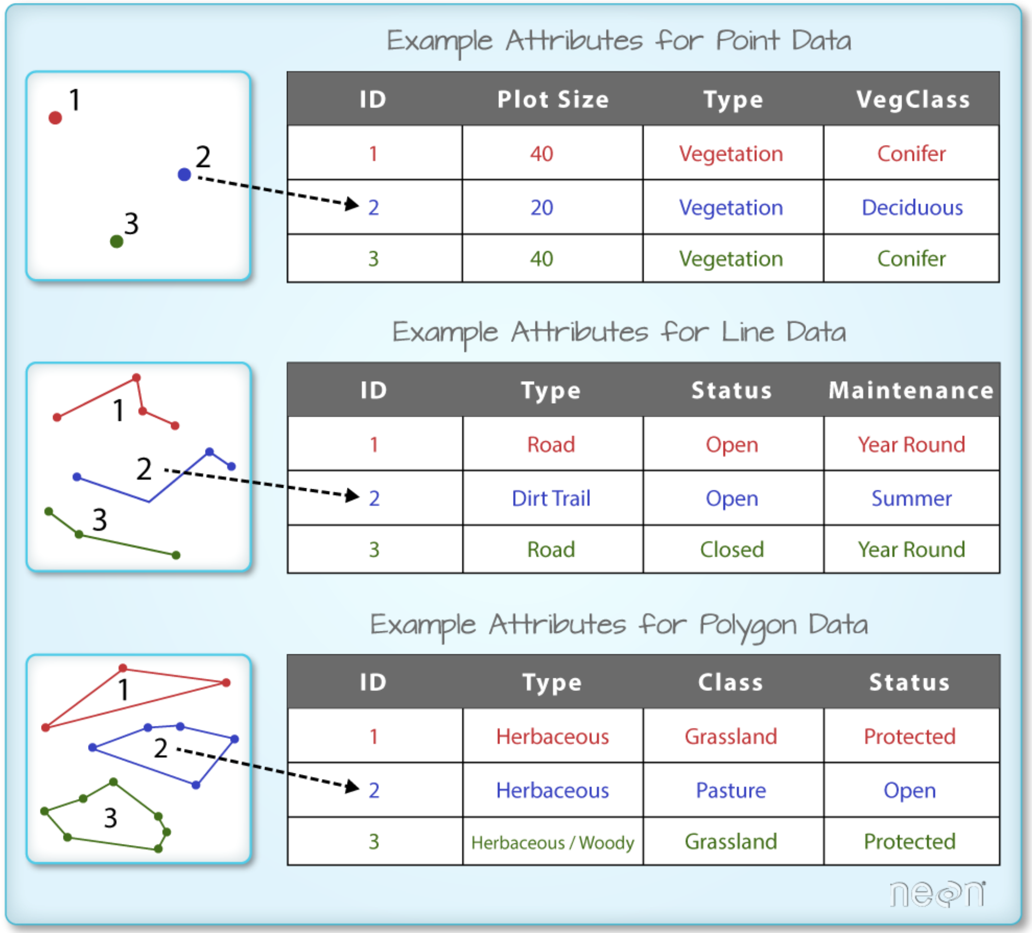
ex: Center point of plot locations, tower locations, sampling locations.

LINES: Composed of many (at least 2) vertices, or points, that are connected.

ex: Roads and streams.

POLYGONS: 3 or more vertices that are connected and **closed**.

ex: Building boundaries and lakes.



Example Attributes for Point Data

ID	Plot Size	Type	VegClass
1	40	Vegetation	Conifer
2	20	Vegetation	Deciduous
3	40	Vegetation	Conifer

Example Attributes for Line Data

ID	Type	Status	Maintenance
1	Road	Open	Year Round
2	Dirt Trail	Open	Summer
3	Road	Closed	Year Round

Example Attributes for Polygon Data

ID	Type	Class	Status
1	Herbaceous	Grassland	Protected
2	Herbaceous	Pasture	Open
3	Herbaceous / Woody	Grassland	Protected