GeoJSON & Leaflet Plugins

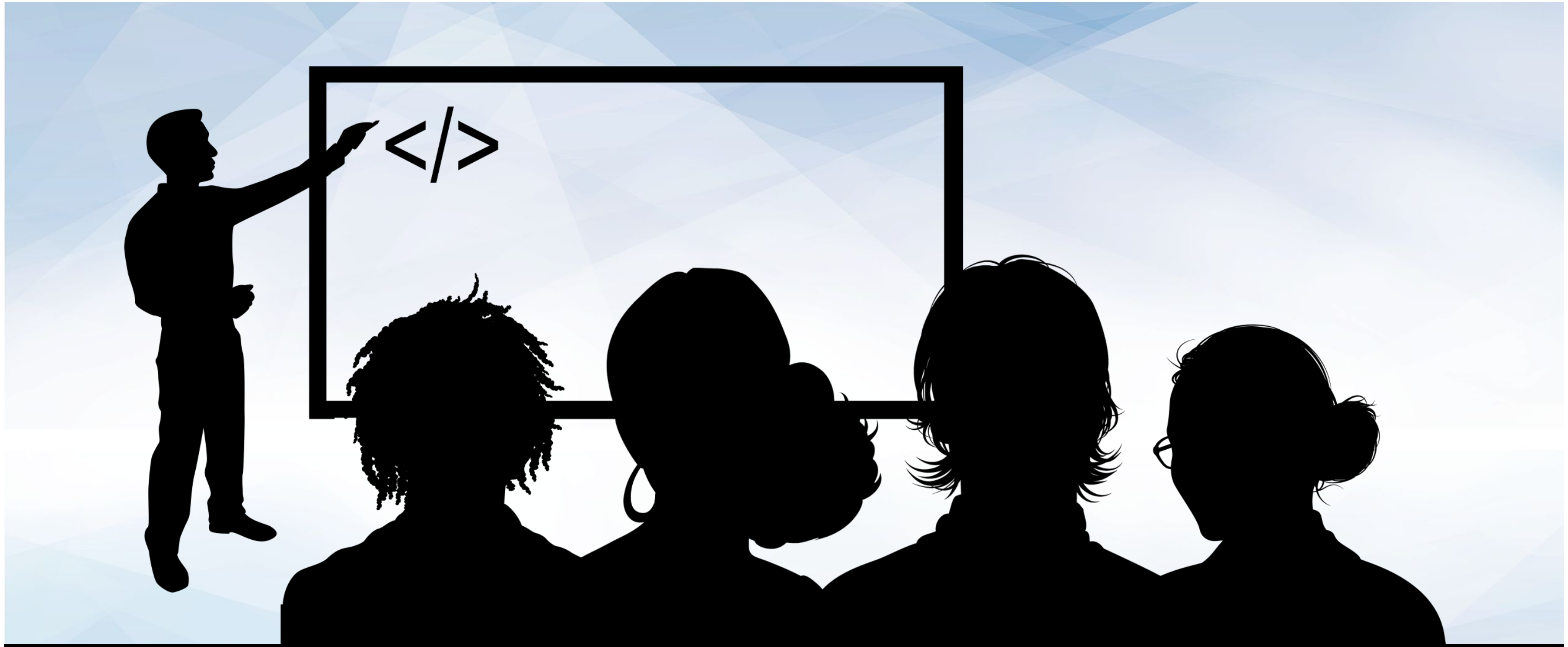
Data Boot Camp

Lesson 15.2

Class Objectives

By the end of this lesson, you will be able to:

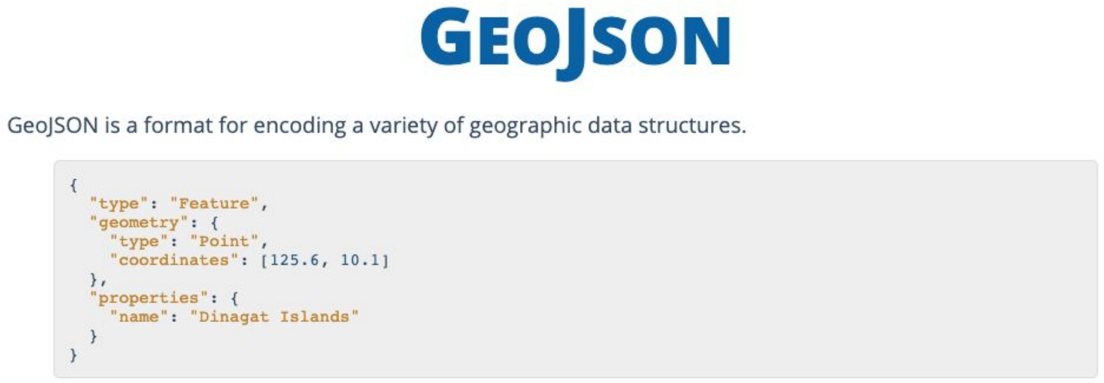
- Leverage Leaflet.js plugins and other third-party libraries.
- Differentiate between maps and map elements for visualizing different datasets.
- Create and deploy custom interactive dashboards.



Instructor Demonstration

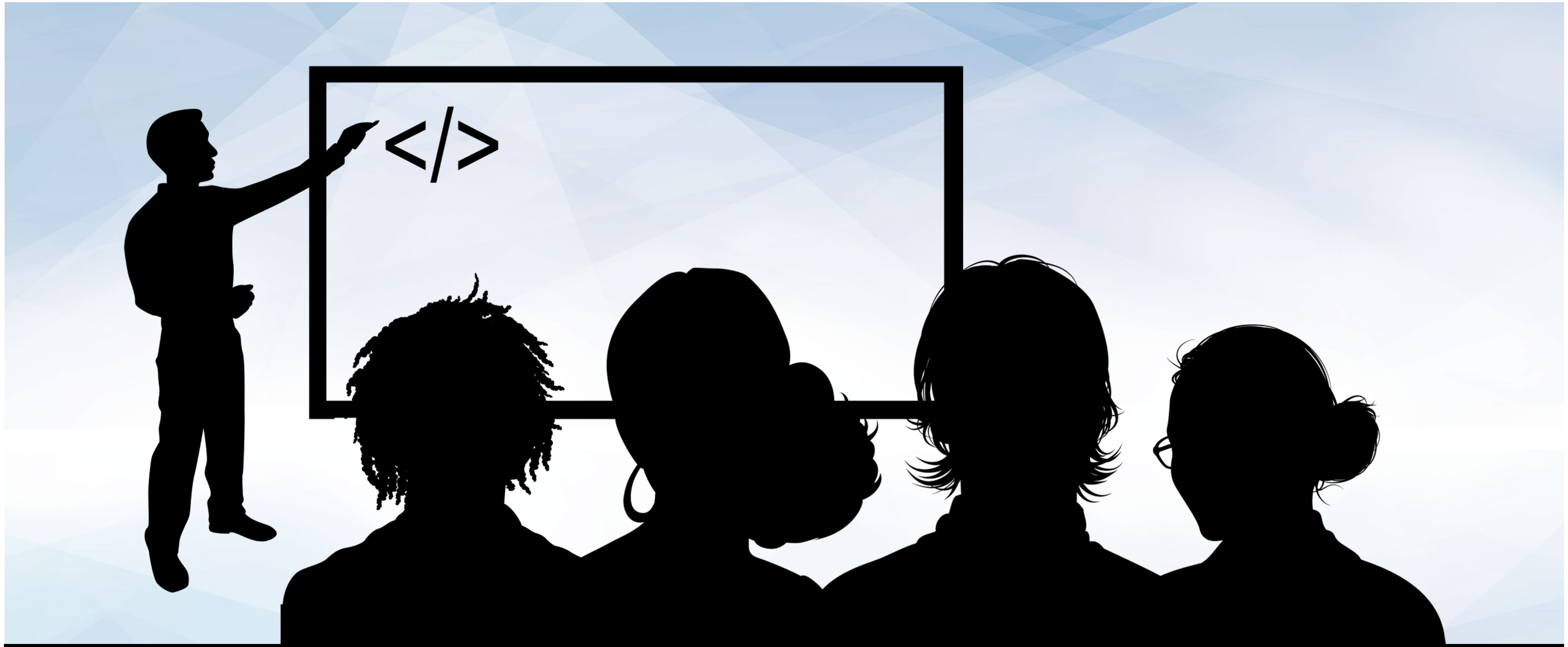
Welcome Students

Instructor Do: Welcome Students


GEOJSON

GeoJSON is a format for encoding a variety of geographic data structures.

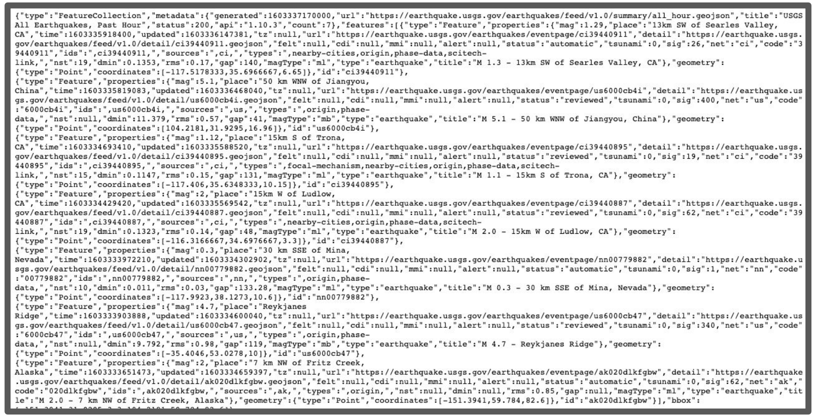
```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Dinagat Islands"
  }
}
```



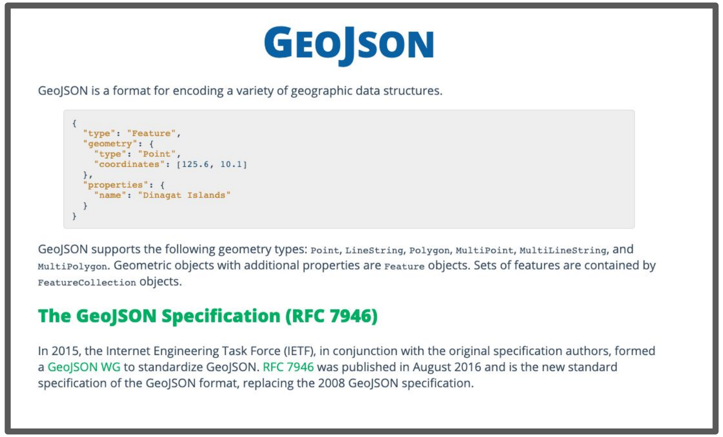
Instructor Demonstration

Review GeoJSON

Instructor Do: Review GeoJSON



```
{"type":"FeatureCollection","metadata":{"generated":1603337170000,"url":"https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_hour.geojson","title":"USGS
All Earthquakes, Past Hour","status":200,"api":"1.10.3","count":7},"features":[{"type":"Feature","properties":{"mag":1.29,"place":"13km SW of Searles Valley,
CA","time":1603335918400,"updated":1603336147381,"tz":null,"url":"https://earthquake.usgs.gov/earthquakes/eventpage/ci39440911","detail":"https://earthquake.usgs.
gov/earthquakes/feed/v1.0/detail/ci39440911.geojson","felt":null,"cdi":null,"mmi":null,"alert":null,"status":"automatic","tsunami":0,"sig":26,"net":"ci","code":"3
9440911","ids":",ci39440911,","sources":",ci,","types":",nearby-cities,origin,phase-data,scitech-
link,","nst":19,"dmin":0.1353,"rms":0.17,"gap":140,"magType":"ml","type":"earthquake","title":"M 1.3 - 13km SW of Searles Valley, CA"},"geometry":
{"type":"Point","coordinates":[-117.5178333,35.6966667,6.65]},"id":"ci39440911"},
{"type":"Feature","properties":{"mag":5.1,"place":"50 km WNW of Jiangyou,
China","time":1603335819083,"updated":1603336468040,"tz":null,"url":"https://earthquake.usgs.gov/earthquakes/eventpage/us6000cb4i","detail":"https://earthquake.us
gs.gov/earthquakes/feed/v1.0/detail/us6000cb4i.geojson","felt":null,"cdi":null,"mmi":null,"alert":null,"status":"reviewed","tsunami":0,"sig":400,"net":"us","code"
:"6000cb4i","ids":",us6000cb4i,","sources":",us,","types":",origin,phase-
data,","nst":null,"dmin":11.379,"rms":0.57,"gap":41,"magType":"mb","type":"earthquake","title":"M 5.1 - 50 km WNW of Jiangyou, China"},"geometry":
{"type":"Point","coordinates":[104.2181,31.9295,16.96]},"id":"us6000cb4i"},
{"type":"Feature","properties":{"mag":1.12,"place":"15km S of Trona,
CA","time":1603334693410,"updated":1603335588520,"tz":null,"url":"https://earthquake.usgs.gov/earthquakes/eventpage/ci39440895","detail":"https://earthquake.usgs.
gov/earthquakes/feed/v1.0/detail/ci39440895.geojson","felt":null,"cdi":null,"mmi":null,"alert":null,"status":"reviewed","tsunami":0,"sig":19,"net":"ci","code":"39
440895","ids":",ci39440895,","sources":",ci,","types":",focal-mechanism,nearby-cities,origin,phase-data,scitech-
link,","nst":15,"dmin":0.1147,"rms":0.15,"gap":131,"magType":"ml","type":"earthquake","title":"M 1.1 - 15km S of Trona, CA"},"geometry":
{"type":"Point","coordinates":[-117.406,35.6348333,10.15]},"id":"ci39440895"},
{"type":"Feature","properties":{"mag":2,"place":"15km W of Ludlow,
CA","time":1603334429420,"updated":1603335569542,"tz":null,"url":"https://earthquake.usgs.gov/earthquakes/eventpage/ci39440887","detail":"https://earthquake.usgs.
gov/earthquakes/feed/v1.0/detail/ci39440887.geojson","felt":null,"cdi":null,"mmi":null,"alert":null,"status":"reviewed","tsunami":0,"sig":62,"net":"ci","code":"39
440887","ids":",ci39440887,","sources":",ci,","types":",nearby-cities,origin,phase-data,scitech-
link,","nst":19,"dmin":0.1323,"rms":0.14,"gap":48,"magType":"ml","type":"earthquake","title":"M 2.0 - 15km W of Ludlow, CA"},"geometry":
{"type":"Point","coordinates":[-116.3166667,34.6976667,3.3]},"id":"ci39440887"},
{"type":"Feature","properties":{"mag":0.3,"place":"30 km SSE of Mina,
Nevada","time":1603333972210,"updated":1603334302902,"tz":null,"url":"https://earthquake.usgs.gov/earthquakes/eventpage/nn00779882","detail":"https://earthquake.u
sgs.gov/earthquakes/feed/v1.0/detail/nn00779882.geojson","felt":null,"cdi":null,"mmi":null,"alert":null,"status":"automatic","tsunami":0,"sig":1,"net":"nn","code"
:"00779882","ids":",nn00779882,","sources":",nn,","types":",origin,phase-
data,","nst":10,"dmin":0.011,"rms":0.03,"gap":133.28,"magType":"ml","type":"earthquake","title":"M 0.3 - 30 km SSE of Mina, Nevada"},"geometry":
{"type":"Point","coordinates":[-117.9923,38.1273,10.6]},"id":"nn00779882"},
{"type":"Feature","properties":{"mag":4.7,"place":"Reykjanes
Ridge","time":1603333903888,"updated":1603334600040,"tz":null,"url":"https://earthquake.usgs.gov/earthquakes/eventpage/us6000cb47","detail":"https://earthquake.us
gs.gov/earthquakes/feed/v1.0/detail/us6000cb47.geojson","felt":null,"cdi":null,"mmi":null,"alert":null,"status":"reviewed","tsunami":0,"sig":340,"net":"us","code"
:"6000cb47","ids":",us6000cb47,","sources":",us,","types":",origin,phase-
data,","nst":null,"dmin":9.792,"rms":0.98,"gap":119,"magType":"mb","type":"earthquake","title":"M 4.7 - Reykjanes Ridge"},"geometry":
{"type":"Point","coordinates":[-35.4046,53.0278,10]},"id":"us6000cb47"},
{"type":"Feature","properties":{"mag":2,"place":"7 km NW of Fritz Creek,
Alaska","time":1603333651473,"updated":1603334659397,"tz":null,"url":"https://earthquake.usgs.gov/earthquakes/eventpage/ak020dlkfgbw","detail":"https://earthquake
.usgs.gov/earthquakes/feed/v1.0/detail/ak020dlkfgbw.geojson","felt":null,"cdi":null,"mmi":null,"alert":null,"status":"automatic","tsunami":0,"sig":62,"net":"ak","
code":"020dlkfgbw","ids":",ak020dlkfgbw,","sources":",ak,","types":",origin,","nst":null,"dmin":null,"rms":0.85,"gap":null,"magType":"ml","type":"earthquake","tit
le":"M 2.0 - 7 km NW of Fritz Creek, Alaska"},"geometry":{"type":"Point","coordinates":[-151.3941,59.784,82.6]},"id":"ak020dlkfgbw"}],"bbox":
```



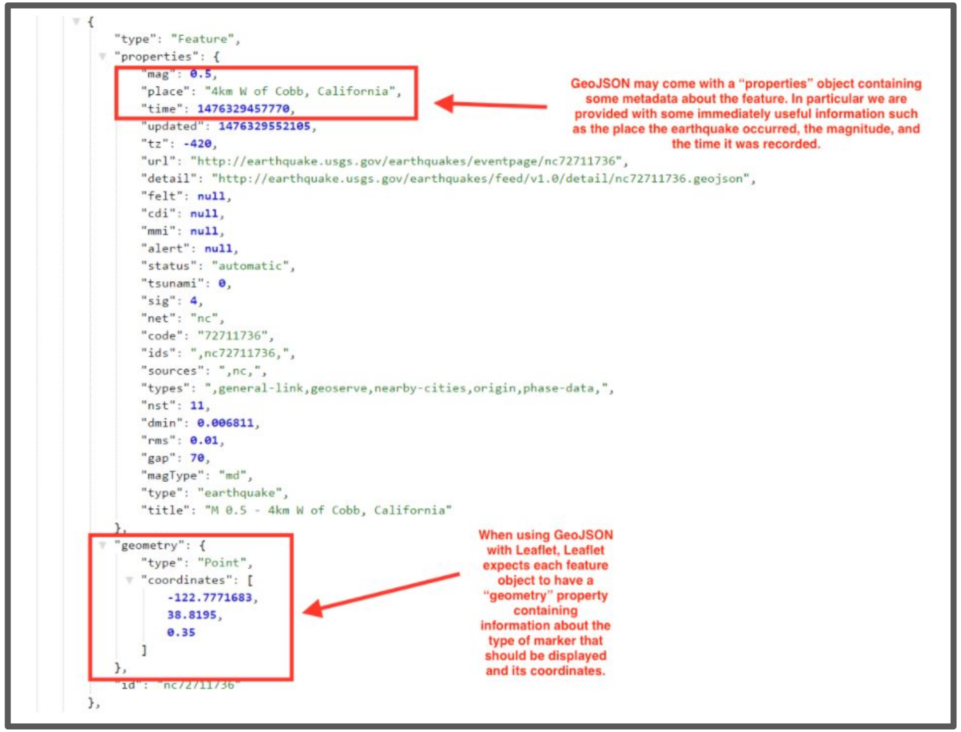
GeoJSON is a format for encoding a variety of geographic data structures.

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

GeoJSON supports the following geometry types: `Point`, `LineString`, `Polygon`, `MultiPoint`, `MultiLineString`, and `MultiPolygon`. Geometric objects with additional properties are `Feature` objects. Sets of features are contained by `FeatureCollection` objects.

The GeoJSON Specification (RFC 7946)

In 2015, the Internet Engineering Task Force (IETF), in conjunction with the original specification authors, formed a GeoJSON WG to standardize GeoJSON. RFC 7946 was published in August 2016 and is the new standard specification of the GeoJSON format, replacing the 2008 GeoJSON specification.



```
{
  "type": "Feature",
  "properties": {
    "mag": 0.5,
    "place": "4km W of Cobb, California",
    "time": 1476329457770,
    "updated": 1476329552105,
    "tz": -420,
    "url": "http://earthquake.usgs.gov/earthquakes/eventpage/nc72711736",
    "detail": "http://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/nc72711736.geojson",
    "felt": null,
    "cdi": null,
    "mmi": null,
    "alert": null,
    "status": "automatic",
    "tsunami": 0,
    "sig": 4,
    "net": "nc",
    "code": "72711736",
    "ids": ",nc72711736,",
    "sources": ",nc,",
    "types": ",general-link,geoserve,nearby-cities,origin,phase-data,",
    "nst": 11,
    "dmin": 0.006811,
    "rms": 0.01,
    "gap": 70,
    "magType": "md",
    "type": "earthquake",
    "title": "M 0.5 - 4km W of Cobb, California"
  },
  "geometry": {
    "type": "Point",
    "coordinates": [
      -122.7771683,
      38.8195,
      0.35
    ]
  },
  "id": "nc72711736"
},
```

GeoJSON may come with a "properties" object containing some metadata about the feature. In particular we are provided with some immediately useful information such as the place the earthquake occurred, the magnitude, and the time it was recorded.
When using GeoJSON with Leaflet, Leaflet expects each feature object to have a "geometry" property containing information about the type of marker that should be displayed and its coordinates.

Everyone Do: Map the NY Neighborhoods

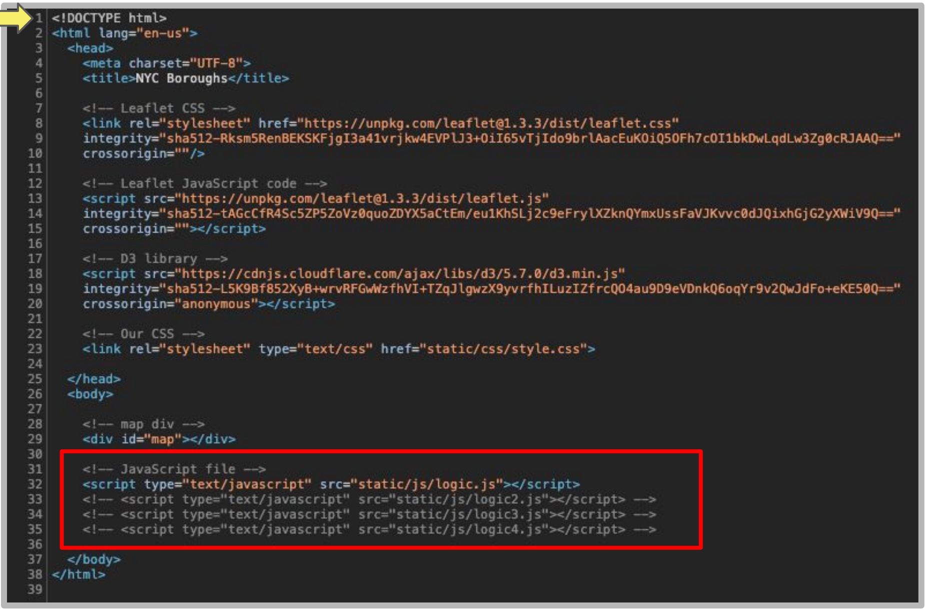
In this activity, we will be diving into some advance Leaflet/GeoJSON functionality by building a map of New York City broken down by boroughs and neighborhoods.

Suggested Time:
20 Minutes

<Time to Code>

Everyone Do: Map the NY Neighborhoods

index.html

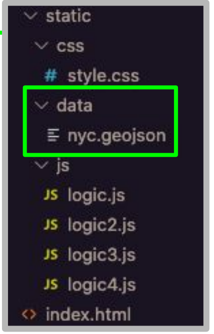


```html
<!DOCTYPE html>
<html lang="en-us">
  <head>
    <meta charset="UTF-8">
    <title>NYC Boroughs</title>

    <!-- Leaflet CSS -->
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.3.3/dist/leaflet.css"
    integrity="sha512-Rksm5RenBEKSKFjgI3a41vrjkw4EVPlJ3+OiI65vTjIdo9brlAacEuKOiQ5OFh7cOI1bkDwLqdLw3Zg0cRJAAQ=="
    crossorigin=""/>

    <!-- Leaflet JavaScript code -->
    <script src="https://unpkg.com/leaflet@1.3.3/dist/leaflet.js"
    integrity="sha512-tAGcCfR4Sc5ZP5ZoVz0quoZDYX5aCtEm/eu1KhSLj2c9eFrylXZknQYmxUssFaVJKvvc0dJQixhGjG2yXWiV9Q=="
    crossorigin=""></script>

    <!-- D3 library -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/d3/5.7.0/d3.min.js"
    integrity="sha512-L5K9Bf852XyB+wrvRFGwWzfhVI+TZqJlgwzX9yvrfhILuzIZfrcQO4au9D9eVDnkQ6oqYr9v2QwJdFo+eKE50Q=="
    crossorigin="anonymous"></script>

    <!-- Our CSS -->
    <link rel="stylesheet" type="text/css" href="static/css/style.css">

  </head>
  <body>

    <!-- map div -->
    <div id="map"></div>

    <!-- JavaScript file -->
    <script type="text/javascript" src="static/js/logic.js"></script>
    <!-- <script type="text/javascript" src="static/js/logic2.js"></script> -->
    <!-- <script type="text/javascript" src="static/js/logic3.js"></script> -->
    <!-- <script type="text/javascript" src="static/js/logic4.js"></script> -->

  </body>
</html>
```

- **Note** that we will work with four JavaScript files. To access the different steps, modify your HTML file to use a different `logic.js` file.

- We will be using a static version of the data originally from BetaNYC.

File Structure



```
static
  css
    style.css
  data
    nyc.geojson
  js
    logic.js
    logic2.js
    logic3.js
    logic4.js
index.html
```

Happy Coding!

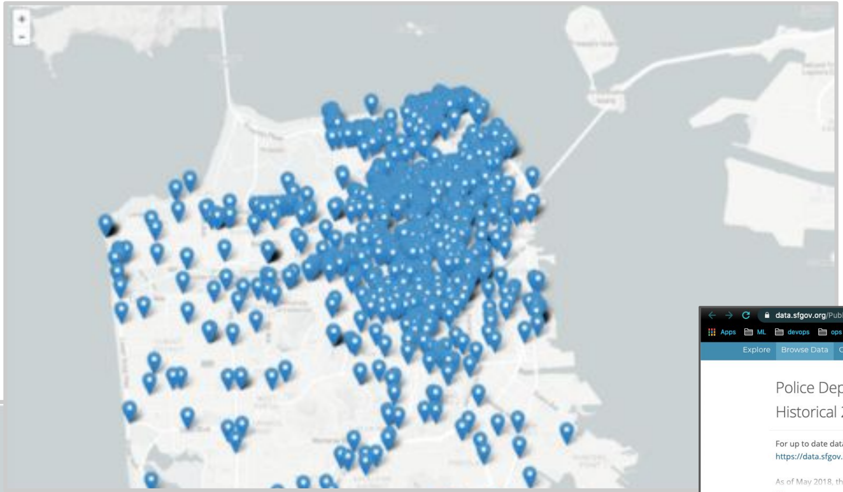
Everyone Do: Create a Heat Map of Crime in San Francisco

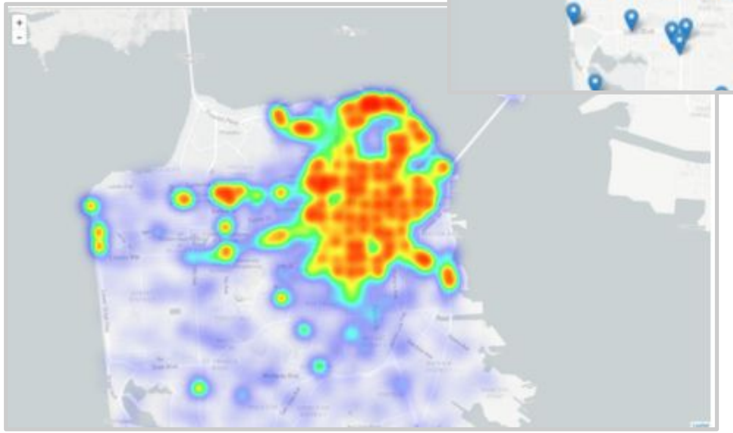
In this activity, we will focus on plotting some basic data with vanilla Leaflet and then adding a third party plugin to make a really cool map.

Suggested Time:
20 Minutes

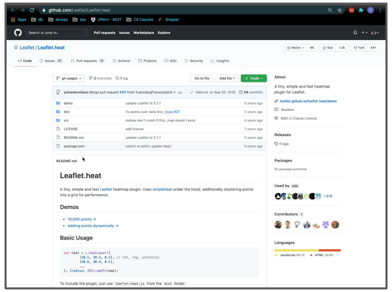
Everyone Do: Create a Heat Map of Crime in San Francisco

`<Time to Code>`





Happy Coding!



github.com/Leaflet/Leaflet.heat

Leaflet / Leaflet.heat

gh-pages · 2 branches · 1 tag

yohanboniface Merge pull request #69 from YuanxiangFranck/patch-1 · 5885ec0 on Sep 30, 2016 · 56 commits

demo	update Leaflet to 0.7.7	5 years ago
dist	fix points over date line, close #27	5 years ago
src	redraw don't crash if this._map doesn't exist	4 years ago
LICENSE	add license	7 years ago
README.md	update Leaflet to 0.7.7	5 years ago
package.json	switch to eslint, update deps	5 years ago

README.md

Leaflet.heat

A tiny, simple and fast Leaflet heatmap plugin. Uses simpleheat under the hood, additionally clustering points into a grid for performance.

Demos

- 10,000 points →
- Adding points dynamically →

Basic Usage

```
var heat = L.heatLayer([
        [50.5, 30.5, 0.2], // lat, lng, intensity
        [50.6, 30.4, 0.5],
        ...
], {radius: 25}).addTo(map);
```

To include the plugin, just use `leaflet-heat.js` from the `dist` folder:

About

A tiny, simple and fast heatmap plugin for Leaflet.

leaflet.github.io/leaflet.heat/demo

Readme

BSD-2-Clause License

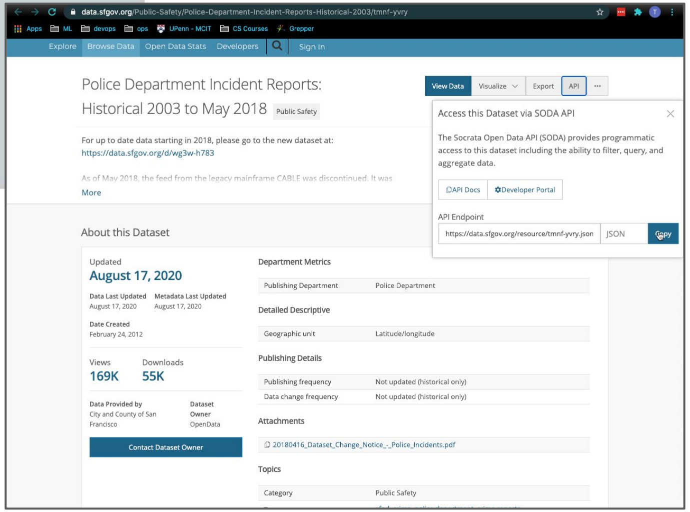
Releases: 1 tags

Packages: No packages published

Used by 686

Contributors 7

Languages: JavaScript 66.1% · HTML 33.9%



data.sfgov.org/Public-Safety/Police-Department-Incident-Reports-Historical-2003/tmnf-yvry

Police Department Incident Reports: Historical 2003 to May 2018 · Public Safety

For up to date data starting in 2018, please go to the new dataset at: https://data.sfgov.org/d/wg3w-h783

As of May 2018, the feed from the legacy mainframe CABLE was discontinued. It was

More

Access this Dataset via SODA API

The Socrata Open Data API (SODA) provides programmatic access to this dataset including the ability to filter, query, and aggregate data.

API Endpoint: https://data.sfgov.org/resource/tmnf-yvry.json · JSON

About this Dataset

Updated: August 17, 2020

Data Last Updated: August 17, 2020 · Metadata Last Updated: August 17, 2020

Date Created: February 24, 2012

Views: 169K · Downloads: 55K

Data Provided by: City and County of San Francisco · Dataset Owner: OpenData

Department Metrics

Publishing Department	Police Department

Detailed Descriptive

Geographic unit	Latitude/longitude

Publishing Details

Publishing frequency	Not updated (historical only)
Data change frequency	Not updated (historical only)

Attachments

20180416_Dataset_Change_Notice_-_Police_Incidents.pdf

Topics

Category	Public Safety

Activity: Map Rodent Clusters

In this activity, we will access data from the NYC Open Data website and plot it using Leaflet plugin.

Suggested Time:
30 Minutes

Instructions:

Activity: Map Rodent Clusters

1. Checkout the data for all 311 (police non emergency) service requests in NYC.
2. Build a query URL for the data so that only rodent complaints are returned.

   Note: To start, limit the data that's returned to 100 data points.

3. After you successfully plot your rodent data, incorporate the Leaflet.markercluster plugin.

   Note: Cluster plugins can help to declutter a map that has tons of data on it.

- **Hint:**
  - You can increase the data limit to 10,000 after you get the cluster plugin working. But, be aware that plotting 10,000 normal markers on a map might slow down your computer quite a bit.

- **Bonus:**
  - If you finish plotting the rodent-sighting data on the map, use the 311 service requests data to plot a similar graph for a different type of data.

Let's Review

Countdown timer

15:00

(with alarm)

Partners Do: Create a Choropleth Map

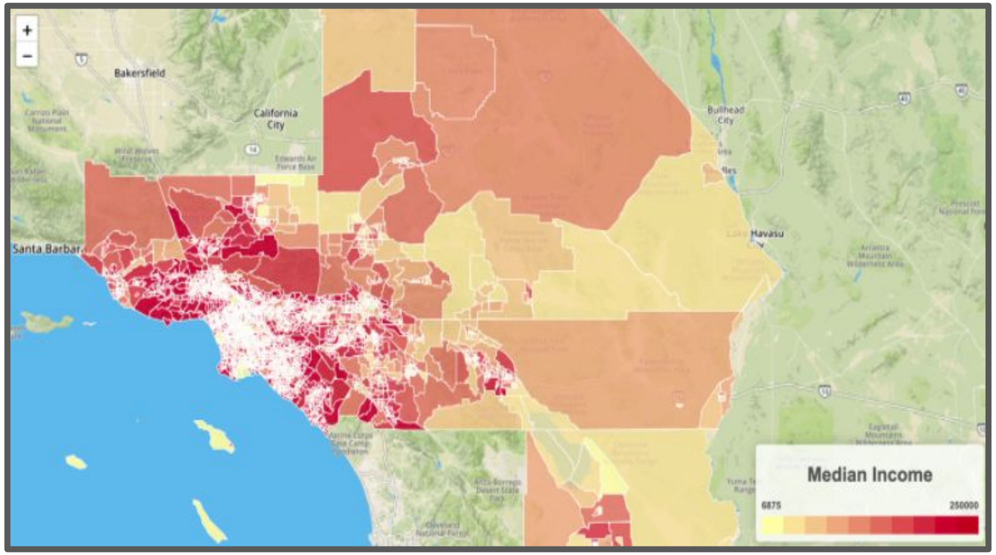
In this activity, you and your partner will work together to create a choropleth map that visualizes the median household income of Los Angeles and the surrounding counties.

Suggested Time:
30 Minutes

Instructions:

Partners Do: Create a Choropleth Map

- A choropleth map has areas that are shaded or patterned in proportion to the statistical variable that the map is representing. It also provides a way to easily visualize how a measurement varies across a geographical area. It does this by showing the level of variability within a region.
- You and your partner will use a new plugin named leaflet-choropleth to create this map. You can find the plugin in the dist folder of the repository.
- You'll work your way through this activity step by step with your partner. The class will reconvene after each step to review what everyone did together before moving on to the next step.



Instructions:

Partners Do: Create a Choropleth Map

- **Step 1:**
  - Get all the data with **D3**, and log it to the console. In Google Chrome, open `index.html`. In Chrome, open **Developer tools**, and then go to the Console tab. Explore the data by using the console, and find where the data stores the Median Household Income (`MHI2016`) for each feature. Note that the amount of data is large, so it might take up to 30 seconds for it to load.

- **Hint:**
  - As you examine the GeoJSON data, look for `MHI2016` or `Median Household Income 2016`.
  - Check out the colorbrewer2 website, which supplies color schemes (in hex values) that you can use to customize a choropleth map.

Let’s Review

Instructions:

Partners Do: Create a Choropleth Map

- **Step 2:**
  - Download `choropleth.js` from the `leaflet-choropleth` repository, place it in your `js` folder, and then in your `index.html` file, uncomment the following line:

    ```
    <script type="text/javascript" src="static/js/choropleth.js"></script>
    ```

- **Hint:**
  - As you examine the GeoJSON data, look for `MHI2016` or `Median Household Income 2016`.
  - Check out the colorbrewer2 website, which supplies color schemes (in hex values) that you can use to customize a choropleth map.

Let’s Review

Instructions:

Partners Do: Create a Choropleth Map

- **Step 3:**
  - Using the leaflet-choropleth documentation, create a new choropleth layer. Make sure to do the following:
  - Change the `valueProperty` to the property that you want to base the map on.
  - Define an `onEachFeature` method that binds a popup containing the value of the feature to the layer. Display the zip code and the median household income.

- **Hint:**
  - As you examine the GeoJSON data, look for `MHI2016` or `Median Household Income 2016`.
  - Check out the colorbrewer2 website, which supplies color schemes (in hex values) that you can use to customize a choropleth map.

Let's Review

Instructions:

Partners Do: Create a Choropleth Map

- **Step 4:**
  - Consult the leaflet-choropleth examples for how to add a legend. Make sure to do the following:
    - Use `L.control` to add a control (and to choose its position).
    - Use `L.DomUtil.create(‘div’, ‘info legend’)` to create a `div` with the `info` and `legend` classes.
    - Loop through the colors and values of your choropleth data, and add them with `div.innerHTML`.
    - Return `div` when done.

- **Hint:**
  - As you examine the GeoJSON data, look for `MHI2016` or `Median Household Income 2016`.
  - Check out the colorbrewer2 website, which supplies color schemes (in hex values) that you can use to customize a choropleth map.

Let's Review

Groups Do: Create a Map of Your Own

In this activity, you and your group will create a map from scratch.

Suggested Time:
30 Minutes

Instructions:

Groups Do: Create a Map of Your Own

- Get into small groups.
- Create a map of your own from scratch.
- Find a dataset.
- Use a new plugin to visualize the data in an interesting way.
- You and your group will share your map with the class and give a brief presentation on it.
- **Note**: This assignment focuses on getting a working map. Therefore, the presentations are secondary to making a splendid map.

Everyone Do: Mini-Presentations on Maps

In this activity, we will be presenting our maps!

Suggested Time:
15 Minutes