Tutorial: Publishing Interactive Web Map of Trees

Description

This tutorial is meant to walk you through the setup and creation of a simple interactive web map.

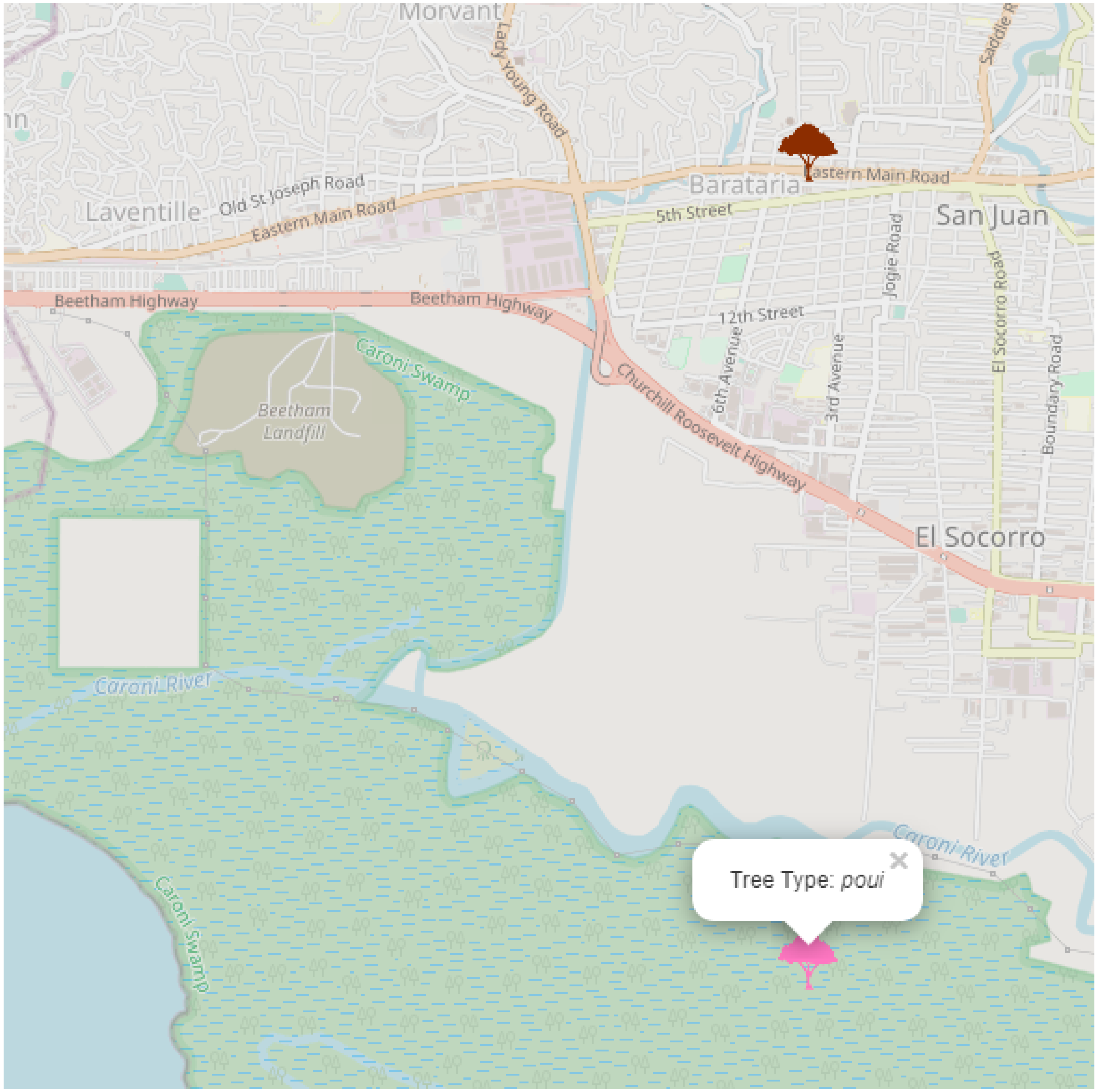
This tutorial will not teach you HTML/CSS/Javascript programming or website development.

However, following this walkthrough should allow you to produce a working interactive map based on freely available technologies and services.

Brief explanations, comments, references and links to further resources will accompany these steps.

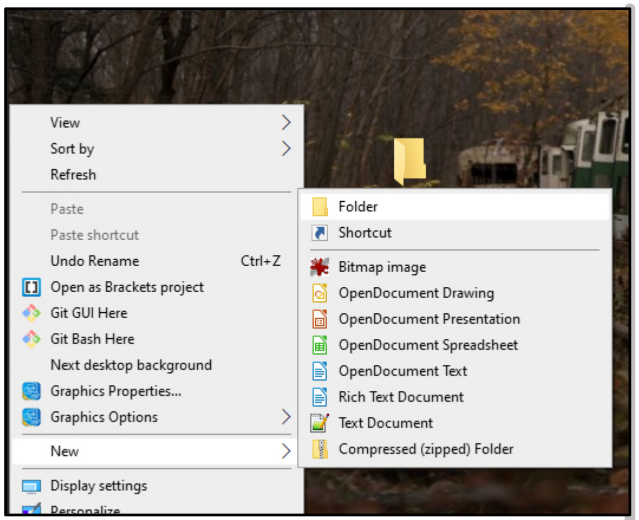
Requirements

- Text Editor (Notepad++)
- GIS Software (QGIS)
- Internet Browser (Google Chrome)
- Hosting Service (GitHub Pages)

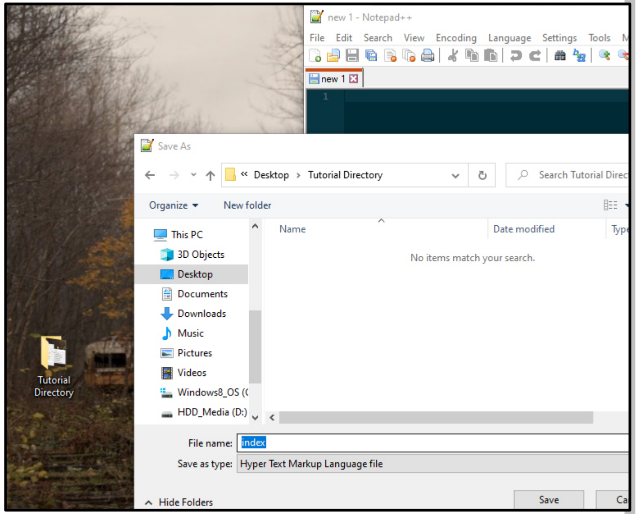


1) Setting up Working Folder

1.



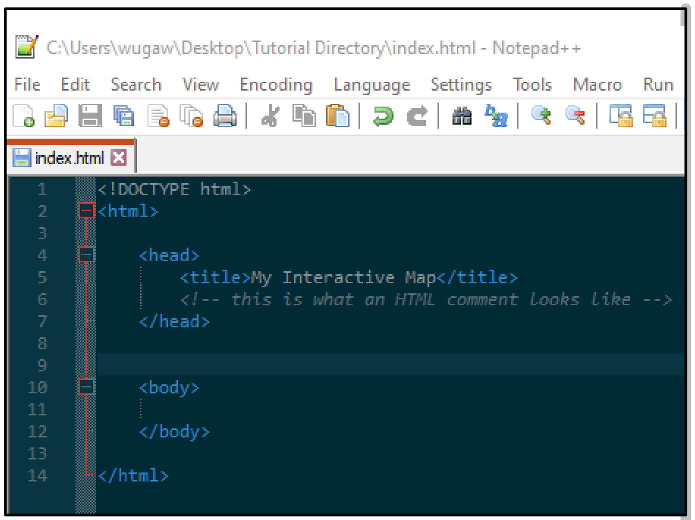
2.



File name: index

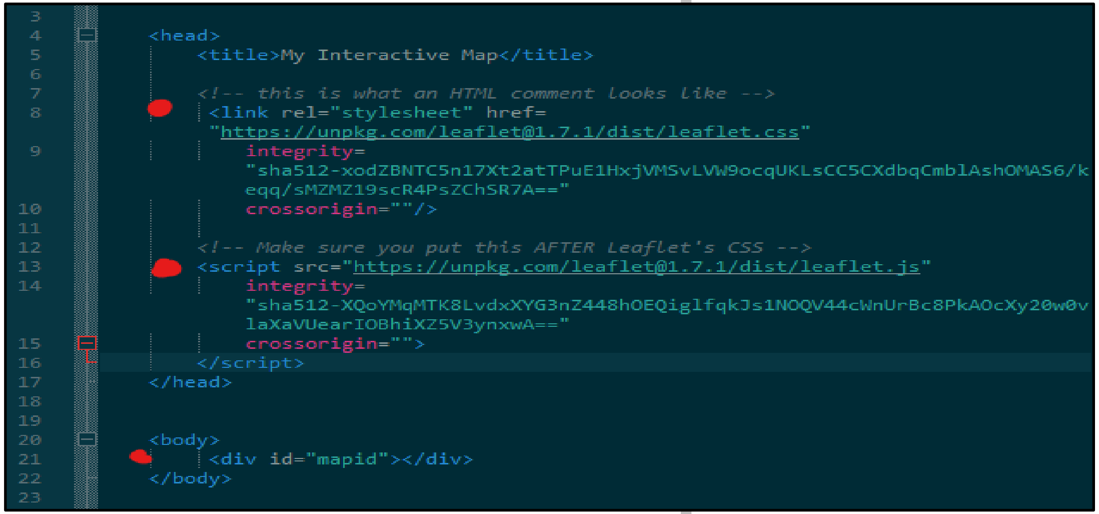
Save as type: Hyper Text Markup Language file

3.


C:\Users\wugaw\Desktop\Tutorial Directory\index.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run
index.html

```html
<!DOCTYPE html>
<html>

    <head>
        <title>My Interactive Map</title>
        <!-- this is what an HTML comment looks like -->
    </head>


    <body>

    </body>

</html>
```

4



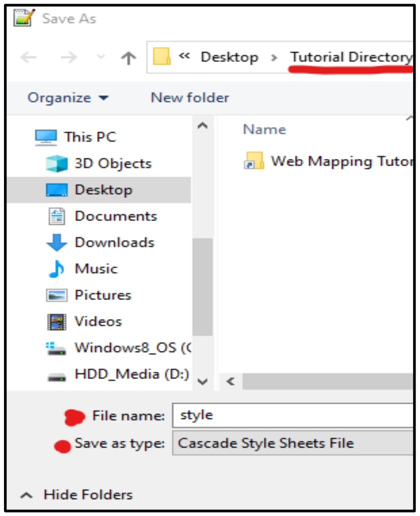
```html

    <head>
        <title>My Interactive Map</title>

        <!-- this is what an HTML comment looks like -->
         <link rel="stylesheet" href=
         "https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
            integrity=
            "sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/k
            eqq/sMZMZ19scR4PsZChSR7A=="
            crossorigin=""/>

        <!-- Make sure you put this AFTER Leaflet's CSS -->
        <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"
            integrity=
            "sha512-XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0v
            laXaVUearIOBhiXZ5V3ynxwA=="
            crossorigin="">
        </script>
    </head>


    <body>
         <div id="mapid"></div>
    </body>

```

<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/
sMZMZ19scR4PsZChSR7A=="
crossorigin=""/>

<!-- Make sure you put this AFTER Leaflet's CSS -->
<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"
integrity="sha512-
XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vlaXaVUearIOBhiXZ5V3ynxwA=="
crossorigin="">

2) Adding and Styling the Map Element

5.

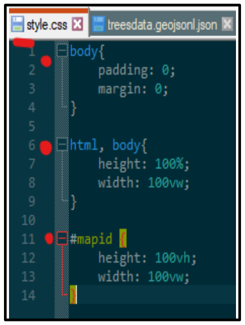


« Desktop › Tutorial Directory

File name: style

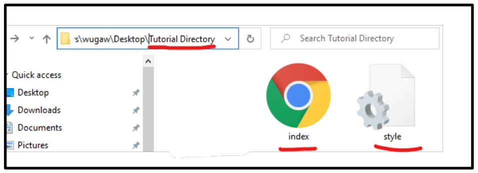
Save as type: Cascade Style Sheets File

6.



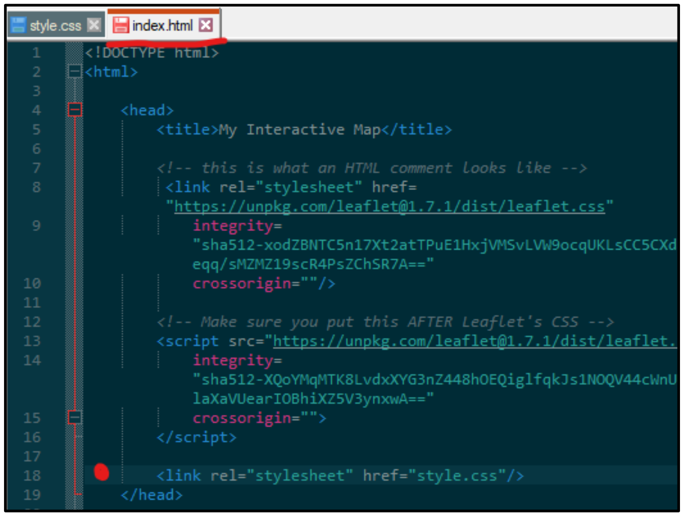
```
body{
    padding: 0;
    margin: 0;
}

html, body{
    height: 100%;
    width: 100vw;
}

#mapid {
    height: 100vh;
    width: 100vw;
```

```
body{
        padding: 0;
        margin: 0;
}
html, body{
        height: 100%;
        width: 100vw;
}
#mapid {
        height: 100vh;
        width: 100vw;
}
```



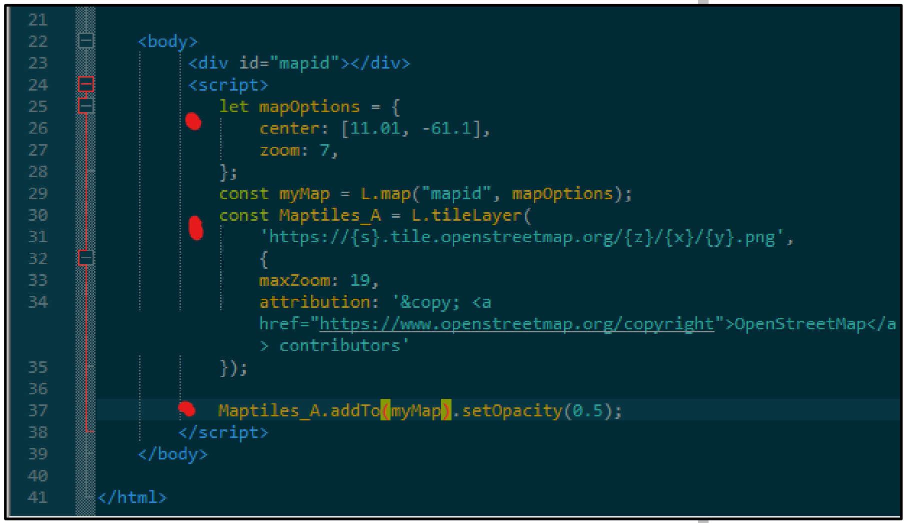
s\wugaw\Desktop\Tutorial Directory

7.



```
<!DOCTYPE html>
<html>

    <head>
        <title>My Interactive Map</title>

        <!-- this is what an HTML comment looks like -->
         <link rel="stylesheet" href=
         "https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
            integrity=
            "sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXd
            eqq/sMZMZ19scR4PsZChSR7A=="
            crossorigin=""/>

        <!-- Make sure you put this AFTER Leaflet's CSS -->
        <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.
            integrity=
            "sha512-XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnU
            laXaVUearIOBhiXZ5V3ynxwA=="
            crossorigin="">
        </script>

        <link rel="stylesheet" href="style.css"/>
    </head>
```

8.

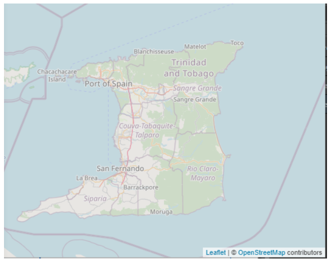


```

    <body>
        <div id="mapid"></div>
        <script>
            let mapOptions = {
                center: [11.01, -61.1],
                zoom: 7,
            };
            const myMap = L.map("mapid", mapOptions);
            const Maptiles_A = L.tileLayer(
                'https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',
                {
                maxZoom: 19,
                attribution: '&copy; <a
                href="https://www.openstreetmap.org/copyright">OpenStreetMap</a
                > contributors'
            });

            Maptiles_A.addTo(myMap).setOpacity(0.5);
        </script>
    </body>

</html>
```

```
<script>

    let mapOptions = {center: [11.01, -61.1], zoom: 7,
    };

    const myMap = L.map("mapid", mapOptions);

    const Maptiles_A = L.tileLayer( 'https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
maxZoom: 19, attribution: '&copy; <a href="https://www.openstreetmap.org/
copyright">OpenStreetMap</a> contributors'
    });

    Maptiles_A.addTo(myMap).setOpacity(0.5);

</script>
```

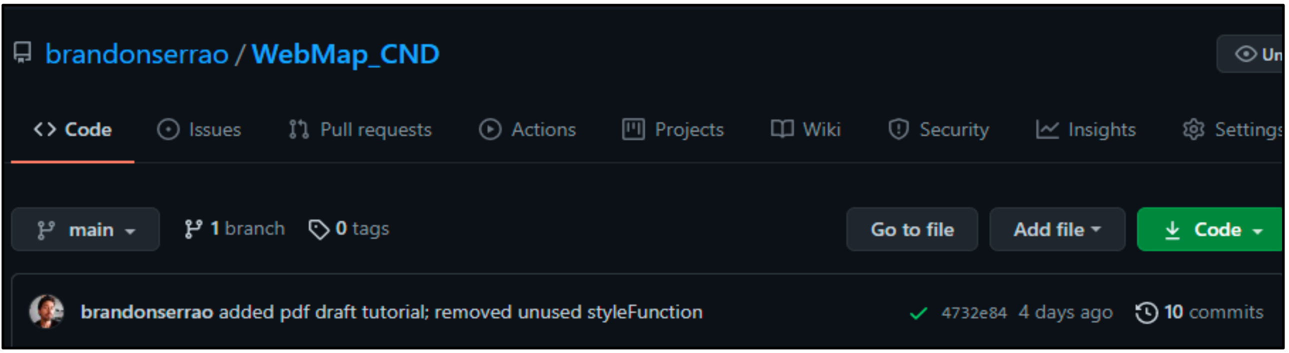
9. Save changes to index.html and style.css.

Open index.html with an internet browser and the interactive basemap should be functional.

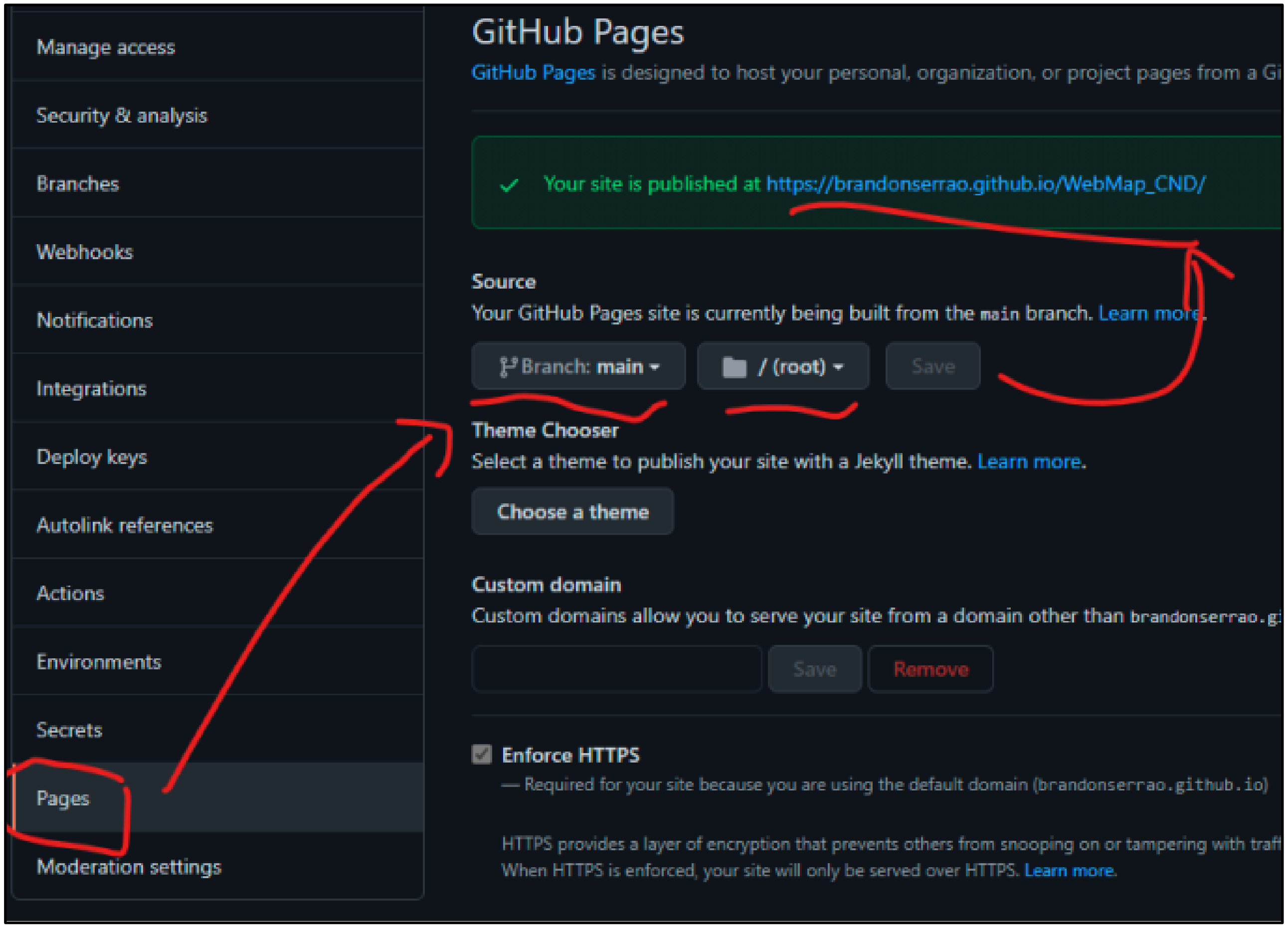


3) Hosting with GitHub Pages

- Make an account and sign into GitHub
- Create a new Repository (top left + symbol on page). Give it a name and description. Choose Public and Add a Readme file. Finish by clicking “Create Repository”.
- Go to the Repository’s page. Click “Add File” and upload all the files from your directory.
- Go back to the repository’s page (the address will look like: https://github.com/brandonserrao/WebMap_CND). Go to Settings > Pages. (scroll down to find it.)
- In Source, make the following changes:
  Branch: main, /(root),
  and save.
- Return again to Settings>Pages. There should be a message saying “site ready to be published at https://yourname.github.io/repositoryname/”.
  This is the link to your Github Pages-hosted static website.
  It may take a few minutes for the changes to you repository to register online and so for your website to update.
- Click your site link and it should open up your webpage with your map. Save/bookmark this link to your map in a document somewhere to make it easy to return to.



brandonserrao added pdf draft tutorial; removed unused styleFunction ✓ 4732e84 4 days ago 10 commits



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a Gi

✓ Your site is published at https://brandonserrao.github.io/WebMap_CND/

Source

Your GitHub Pages site is currently being built from the `main` branch. Learn more.

Theme Chooser

Select a theme to publish your site with a Jekyll theme. Learn more.

Custom domain

Custom domains allow you to serve your site from a domain other than `brandonserrao.gi`

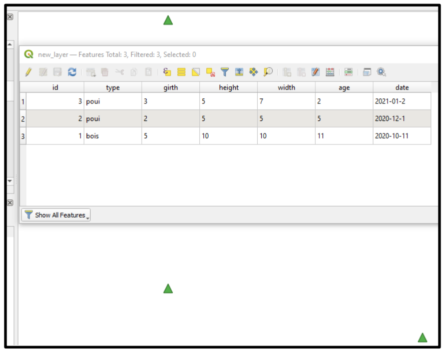
☑ Enforce HTTPS

— Required for your site because you are using the default domain (`brandonserrao.github.io`)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traf
When HTTPS is enforced, your site will only be served over HTTPS. Learn more.

4) Loading Your GIS Data

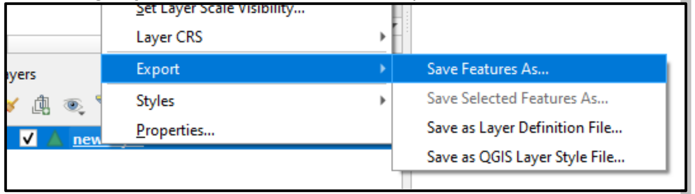
The GIS data you are using is stored as a shapefile and will look like the attribute table below:



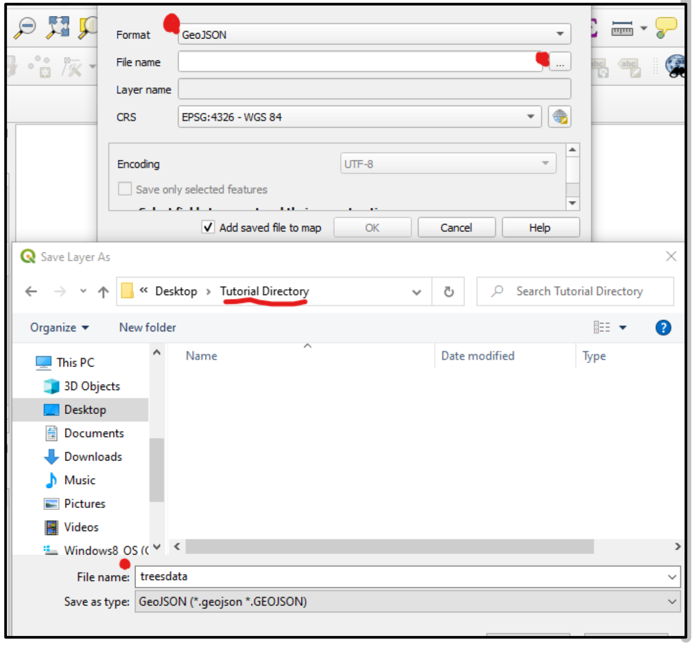
new_layer — Features Total: 3, Filtered: 3, Selected: 0

	id	type	girth	height	width	age	date
1	3	poui	3	5	7	2	2021-01-2
2	2	poui	2	5	5	5	2020-12-1
3	1	bois	5	10	10	11	2020-10-11

10. Using QGIS, export your shapefile point data to your Tutorial Directory as a geojson file. (!!Not geojson newline delimited!!)



Set Layer Scale Visibility...
Layer CRS
Export
Styles
Properties...
Save Features As...
Save Selected Features As...
Save as Layer Definition File...
Save as QGIS Layer Style File...



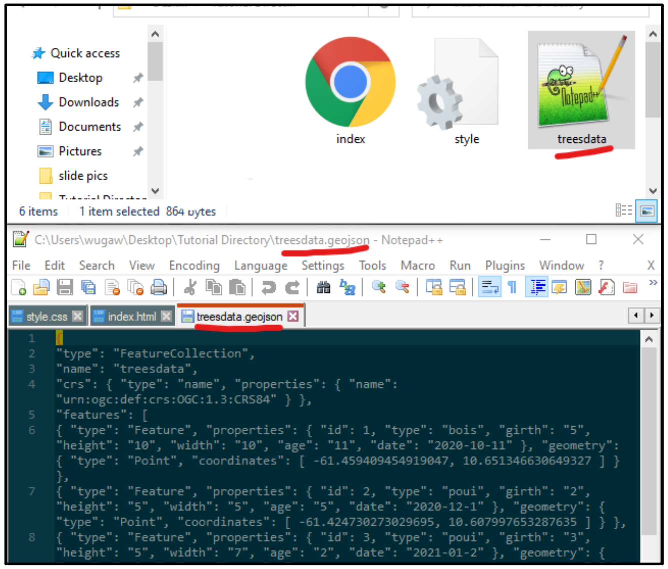
Format: GeoJSON
File name:
Layer name:
CRS: EPSG:4326 - WGS 84
Encoding: UTF-8
☐ Save only selected features
☑ Add saved file to map

Save Layer As

« Desktop › Tutorial Directory

File name: treesdata
Save as type: GeoJSON (*.geojson *.GEOJSON)

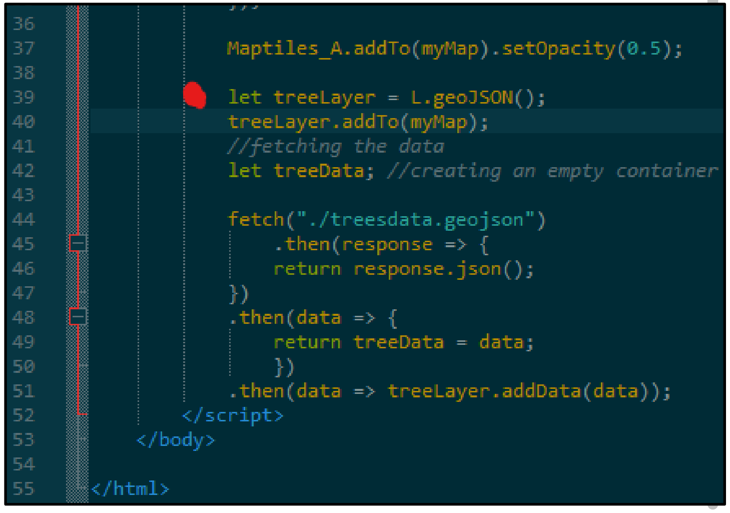
The exported geojson should look like below if opened in your text editor:



C:\Users\wugaw\Desktop\Tutorial Directory\treesdata.geojson - Notepad++

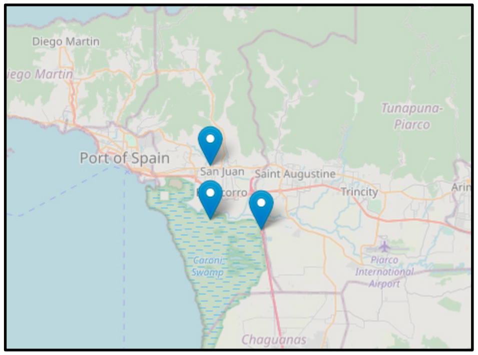
```
{
"type": "FeatureCollection",
"name": "treesdata",
"crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },
"features": [
{ "type": "Feature", "properties": { "id": 1, "type": "bois", "girth": "5", "height": "10", "width": "10", "age": "11", "date": "2020-10-11" }, "geometry": { "type": "Point", "coordinates": [ -61.459409454919047, 10.651346630649327 ] } },
{ "type": "Feature", "properties": { "id": 2, "type": "poui", "girth": "2", "height": "5", "width": "5", "age": "5", "date": "2020-12-1" }, "geometry": { "type": "Point", "coordinates": [ -61.424730273029695, 10.607997653287635 ] } },
{ "type": "Feature", "properties": { "id": 3, "type": "poui", "girth": "3", "height": "5", "width": "7", "age": "2", "date": "2021-01-2" }, "geometry": {
```

11. Edit your index.html file to add the code below:



```
        Maptiles_A.addTo(myMap).setOpacity(0.5);

        let treeLayer = L.geoJSON();
        treeLayer.addTo(myMap);
        //fetching the data
        let treeData; //creating an empty container

        fetch("./treesdata.geojson")
            .then(response => {
            return response.json();
        })
        .then(data => {
            return treeData = data;
            })
        .then(data => treeLayer.addData(data));
      </script>
    </body>

</html>
```

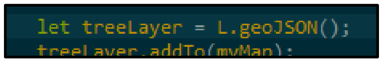
12. Save all your changes and reupload the files from your Tutorial Directory to your GitHub repository like before. After a few minutes the changes will take effect, and when you revisit the link to your webmap page you should see the GIS data now visible:



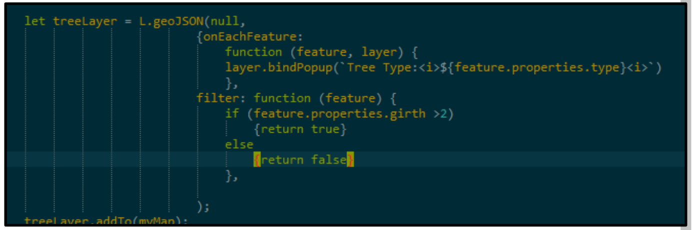
```
let treeLayer = L.geoJSON();

treeLayer.addTo(myMap);

let treeData; //creating an empty container for treedata

fetch("./treesdata.geojson")

.then(response => {
return response.json();
})
.then(data => {
return treeData = data;
})
.then(data => treeLayer.addData(data));
```

5) Functionality and Symbology

13. Edit your index.html file to add the code below:



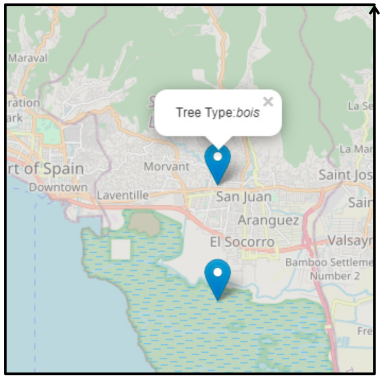
```
let treeLayer = L.geoJSON();
treeLayer.addTo(myMap);
```



```
let treeLayer = L.geoJSON(null,
                        {onEachFeature:
                            function (feature, layer) {
                            layer.bindPopup(`Tree Type:<i>${feature.properties.type}<i>`)
                            },
                        filter: function (feature) {
                            if (feature.properties.girth >2)
                                {return true}
                            else
                                {return false}
                            },

                        );
treeLayer.addTo(myMap);
```

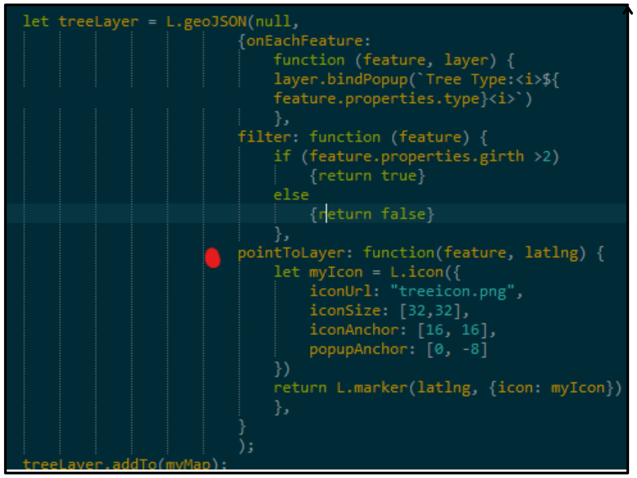
```
let treeLayer = L.geoJSON(null,
        {onEachFeature:
                function (feature, layer) {
                layer.bindPopup(`Tree Type:<i>${feature.properties.type}<i>`)
                },
        filter: function (feature) {
                if (feature.properties.girth >2)
                        {return true}
                else
                        {return false}
                },
        }
        );
```

14. Save and reupload to your GitHub repository. Wait a bit and refresh your webpage. Now popups should be visible on click, and only data records with Girth > 2 should be added to your map.



15. Go to my repository and download treeicon.png, and save it to your Tutorial Directory folder.

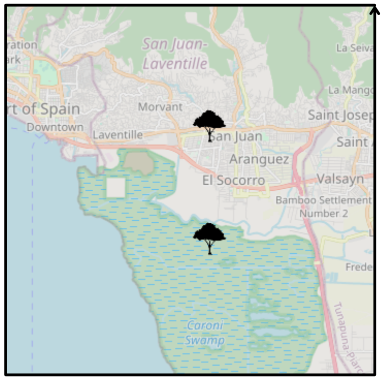
16. Make the following changes code to your index.html:



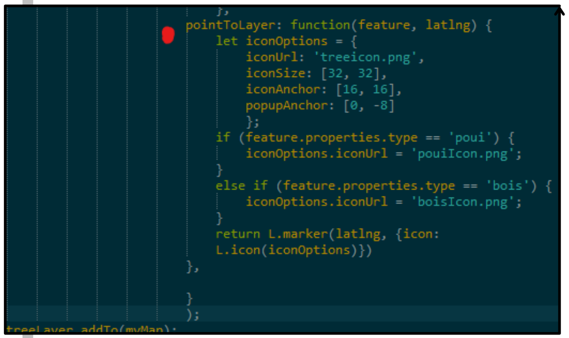
```
let treeLayer = L.geoJSON(null,
                        {onEachFeature:
                            function (feature, layer) {
                            layer.bindPopup(`Tree Type:<i>${
                            feature.properties.type}<i>`)
                            },
                        filter: function (feature) {
                            if (feature.properties.girth >2)
                                {return true}
                            else
                                {return false}
                            },
                        pointToLayer: function(feature, latlng) {
                            let myIcon = L.icon({
                                iconUrl: "treeicon.png",
                                iconSize: [32,32],
                                iconAnchor: [16, 16],
                                popupAnchor: [0, -8]
                            })
                            return L.marker(latlng, {icon: myIcon})
                            },
                        }
                        );
treeLayer.addTo(myMap);
```

```
pointToLayer: function(feature, latlng) {

        let myIcon = L.icon({
                iconUrl: "treeicon.png",
                iconSize: [32,32],
                iconAnchor: [16, 16],
                popupAnchor: [0, -8]
                })

        return L.marker(latlng, {icon: myIcon})

        },
```

Save changes, reupload all the files from your Tutorial Directory to your GitHub repository again. After a while your map should look like the following:



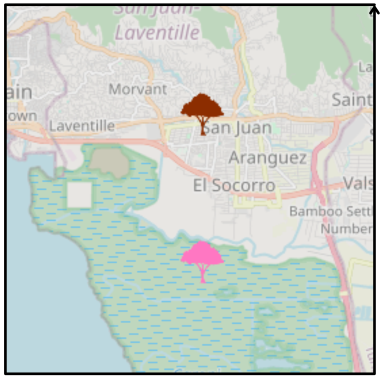
16. Make the following code changes to index.html:



```
                            },
                        pointToLayer: function(feature, latlng) {
                            let iconOptions = {
                                iconUrl: 'treeicon.png',
                                iconSize: [32, 32],
                                iconAnchor: [16, 16],
                                popupAnchor: [0, -8]
                                };
                            if (feature.properties.type == 'poui') {
                                iconOptions.iconUrl = 'pouiIcon.png';
                            }
                            else if (feature.properties.type == 'bois') {
                                iconOptions.iconUrl = 'boisIcon.png';
                            }
                            return L.marker(latlng, {icon:
                            L.icon(iconOptions)})
                        },

                        }
                        );
treeLayer.addTo(myMap);
```

```
pointToLayer: function(feature, latlng) {

        let iconOptions = {

                iconUrl: 'treeicon.png',
                iconSize: [32, 32],
                iconAnchor: [16, 16],
                popupAnchor: [0, -8]
        };

        if (feature.properties.type == 'poui') {

                iconOptions.iconUrl = 'pouiIcon.png';
        }
        else if (feature.properties.type == 'bois') {

                iconOptions.iconUrl = 'boisIcon.png';
        }

        return L.marker(latlng, {icon: L.icon(iconOptions)})

},
```

Save changes, reupload all the files from your Tutorial Directory to your GitHub repository again. Your map should finally look like below:


San Juan-Laventille
Morvant
Laventille
San Juan
Aranguez
El Socorro
Valsayn
Bamboo Settlement Number