

WEB222 Assignment 4 – DOM Manipulation

Grade value: 6% of your final course grade

Objectives

This assignment will help you learn and practice interactive DOM programming. Please do this assignment on your own and submit only your own work. Gaining experience with JavaScript and the web takes a lot of personal practice and working on these problems yourself will help you build your skills.

We will be creating a web application for exploring iNaturalist (<https://www.inaturalist.org/>) wildlife observation data. This will build on many of the skills and techniques you've used in previous assignments, as well as allow you to work with dynamic web pages using real data.

In addition, you will have a chance to practice mixing code you write with code someone else has written, reading code comments and specifications, as well as debugging in the browser.

Setup

Begin by downloading the assignment Zip file and setting up the necessary dependencies:

```
cd <directory where you unzipped the file>
npm install
```

This will download and save all the necessary files to a folder named **node_modules** in the current directory. It will also allow you to run a number of scripts, including:

1. `npm start` – this will start a local webserver on <http://localhost:8080> which you can use to run and debug your web app. It will automatically update when the code changes (i.e., when you save files in VSCode).
2. `npm run prettier` – this will run prettier.io on your source code and automatically reformat it correctly.
3. `npm run prepare-submission` – this will create your final submission.zip file to be handed in, see more details below on submitting.

Instructions

Imagine that you have been hired by a company to work on a web site that was previously finished, but the original source code accidentally lost due to hard drive corruption—only an early version of the code remains, and it needs to be finished. See the video of the finished version to get a sense of how things should work.

The HTML and CSS need minor tweaks, but the JavaScript needs a lot of work. The original developer left many comments explaining the work she was going to do, but the original implementation is now lost.

You are asked to complete the code for the web app in the **src/** folder. The program is broken up into multiple files. Each file is focused on a particular aspect of the program. Start by looking through all the code, in order to familiarize yourself with what there is:

1. **src/data.js** has the iNaturalist data we need to work with. This code is complete, you won't need to modify it.
2. **src/observations.js** has functions for working with the iNaturalist data to process and filter it. You will need to complete this.
3. **src/simple-map.js** has code to create and update an interactive map. This code is complete, you won't need to modify it.
4. **src/ui.js** has functions to create and manipulate DOM elements based on data. You will need to complete this.
5. **src/app.js** is the main part of the program, and connects all the other parts, managing the data, creating the interface, and interacting with the user. You will need to complete this.

Read the comments above each function and write your code in the empty function bodies to implement the behaviours described in the comments. Your goal is to recreate the finished version.

TODO Checklist

Here is a list of the things you need to fix or implement:

1. **src/index.html** needs all of the following third-party CSS files added to the `<head>`
 - *Normalize CSS across browsers:*
<https://unpkg.com/normalize.css@8.0.1/normalize.css>
 - *Use the Water CSS theme (see <https://watercss.kognise.dev/>):*
<https://cdn.jsdelivr.net/npm/water.css@2/out/light.min.css>
 - *Use the Leaflet Map Stylesheet (see <https://leafletjs.com/examples/quick-start/>):*
<https://unpkg.com/leaflet@1.7.1/dist/leaflet.css>

- *Use the Leaflet Marker Cluster Plugin (see <https://github.com/Leaflet/Leaflet.markercluster>):*

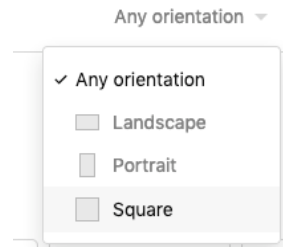
<https://unpkg.com/leaflet.markercluster@1.4.1/dist/MarkerCluster.css>
- 2. **src/index.html** needs our app's CSS file (src/css/style.css) added to the <head> **after** all the third-party CSS files you added in step 1.
- 3. **src/index.html** needs all of the following third-party JavaScript added to the **bottom** of the <body> (i.e., put it at the very end before the closing body tag):
 - *Use the Leaflet Map JavaScript (see <https://leafletjs.com/examples/quick-start>)*

<https://unpkg.com/leaflet@1.7.1/dist/leaflet.js>
 - *Use the Leaflet Marker Cluster Plugin (see <https://github.com/Leaflet/Leaflet.markercluster>):*

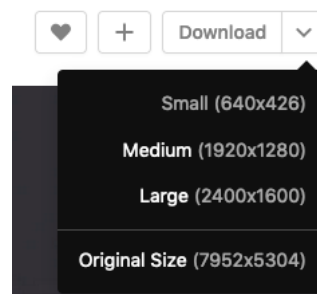
<https://unpkg.com/leaflet.markercluster@1.4.1/dist/leaflet.markercluster.js>
- 4. **src/index.html** needs all of the JavaScript files for our app (i.e, all files in **src/js**) added to the **bottom** of the <body> **after** the previous third-party scripts. Add them in this order: simple-map.js, data.js, observations.js, ui.js, app.js. The **order matters**, since some scripts use functions from previous scripts.
- 5. **src/js/observations.js** needs all of the functions with a **// TODO** comment to be completed. See the code comments to understand what you have to do. Most functions will require you to receive data, process it, and return something back again.
- 6. **src/js/ui.js** needs all of the functions with a **// TODO** comment to be completed. See the code comments to understand what you have to do. NOTE: you are required to use **DOM methods** to create and manipulate HTML Elements. In almost every case, **using innerHTML and HTML as strings is the wrong approach**.
- 7. **src/js/app.js** needs all of the portions of the code with a **// TODO** comment to be completed. See the code comments to understand what you have to do.
- 8. **src/js/app.js** has a lot of code duplication. The **showAll**, **showOnlyNative**, and **showOnlyIntroduced** functions are all nearly identical. Try to rewrite the code so that all of the common aspects are done in a fourth function, and the parts that differ are passed to it as arguments.

9. **src/index.html** uses a free image from <https://unsplash.com/> and an inspirational quote about nature by Steven Wright. To make this site your own personal creation, replace both with an image and quote of your choosing.

TIP: Use Unsplash.com to find your photo, and try to pick one that is square:



Also, when you download your image, choose the smallest file size possible (e.g., 640px wide) so that the image isn't huge to use on the web (you can further reduce its size using <https://squoosh.app/>). Place it in the **src/images** directory



The CSS that controls how the image and quote look is in **src/css/style.css** lines 1-22. You are free to modify this if you like or leave it as it is if it looks OK with your image and quote.

Debugging

No matter how experienced you get at programming, you will always deal with difficult bugs in your code. The difference between a beginner and an expert isn't so much how much they know about a language, as it is how well they know how to use their development tools. Learning to use the browser's built-in developer tools is a super power that you need to acquire.

To learn how to use these tools, see <https://developers.google.com/web/tools/chrome-devtools> for Chrome, and <https://developer.mozilla.org/en-US/docs/Tools> for Firefox..

This video is a great introduction of various ways to use the Dev Tools in Chrome: <https://www.youtube.com/watch?v=H0XScE08hy8>.

For example, you can call all of your functions directly in the Console to test that they are working as you expect, set breakpoints in the Sources/Debugger tab to inspect and step through code, etc. If you have questions on how to do any of this, ask your professor.

NOTE: when developing web sites and apps, it is highly advisable to disable your browser cache while the dev tools are open. See <https://nicholasbering.ca/tools/2016/10/09/devtools-disable-caching/> for instructions on doing this.

Submission

When you have completed your assignment, you need to prepare your submission. To do so, use the npm command:

```
npm run prepare-submission
```

This will do a number of things automatically for you:

1. Run prettier on your assignment code, formatting it
2. Create a directory called submission/
4. Create submission.zip from the contents of submission/

You can upload and submit the submission.zip to Blackboard.