

Assignment 4 Instructions

Introduction

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 build your skill.

Submission

Please read and follow all instructions below carefully. If you have problems or questions, talk to your classmates or professor.

To hand in your work, see the "Submitting your Assignment" section below.

Instructions

Complete the code in `src/js/index.js` to make the table and menus interactive:

- add functionality to all of the stubbed-out functions/methods based on comments in the code.
- make it possible to click on any of the menu items and have the table's contents update to show the requested data. The table takes the following form:

Flag – an `` with the flag for the country. All flag images are located in `src/flags/` and named according to the country code.

Code – a string with the country's code

Country/Dep. Name – a string with the name of the country. This will either be in English, or whatever language the user requests.

Continent – a string with the continent for this country

Area (Km2) – a number with the area in square km
















Population – a number with the population of the country

Capital – a string with the name of the country's capital

All country data comes from `src/js/countries-data.js` (see this file for format details). The complete array of countries is available via `countries.all` in `src/js/index.js`.

Here is an example of what the table might look like:

List of Countries and Dependencies – Country/Dep. Name in Chinese (中文)

Flag	Code	Country/Dep. Name	Continent	Area (Km ²)	Population	Capital
	AF	阿富汗	Asia	652230	35530081	Kabul
	AL	阿尔巴尼亚	Europe	28748	2930187	Tirana
	DZ	阿尔及利亚	Africa	2381741	41318142	Algiers
	AD	安道尔	Europe	468	76965	Andorra la Vella
	AO	安哥拉	Africa	1246700	29784193	Luanda
	AI	安圭拉	Americas	91	14909	The Valley
	AQ	南极洲	Antarctica	14000000		Antarctica
	AG	安提瓜和巴布达	Americas	442	102012	St. John's
	AR	阿根廷	Americas	2780400	44271041	Buenos Aires
	AM	亚美尼亚	Asia	29743	2930450	Yerevan
	AW	阿鲁巴	Americas	180	105264	Oranjestad
	AU	澳大利亚	Oceania	7692024	24450561	Canberra
	AT	奥地利	Europe	83871	8735453	Vienna
	AZ	阿塞拜疆	Asia	86600	9827589	Baku
	BS	巴哈马	Americas	13943	395361	Nassau

Finally, make sure you update `src/index.html` and `src/honesty.html` with your name and information.

Setup

This assignment relies on a number of dependencies, which must be installed on your computer.

First, install Node.js on your computer. See installation instructions at:

<https://nodejs.org/en/>

You can install the LTS (Long Term Support) version of node.js.

Install Dependencies

Open a command line terminal and navigate (i.e., "cd") to the directory where you have unzipped the assignment files. When you type "dir" (Windows) or "ls" (Linux/macOS) you should see this README.md file, package.json, assignment4.js, etc.

In this directory, install the assignment dependencies using the Node.js Package Manager (npm), which is installed along with node.js. In your terminal, type the following:

```
npm install
```

This will download and save all the necessary files to a folder named `node_modules/` in the current directory.

If you have trouble getting "npm install" to work:

- Did you install node.js?
- If you type "node --version" in your terminal, does it print the version?
- Are you in the right directory (you must cd into the correct directory)

If you need help, ask your classmates and/or talk to your professor.

Learn how to Run the Assignment Web Server

This assignment includes multiple HTML, JS, and CSS files. The correct way to open these files in your browser is using a web server. To run a web server for the assignment files, use the npm command:

```
npm run server

Did not detect a `bs-config.json` or `bs-config.js` override file. Using lite-
server defaults...
** browser-sync config **
{ injectChanges: false,
  files: [ './**/*.html,css,js' ],
  watchOptions: { ignored: 'node_modules' },
  server: { baseDir: 'src', middleware: [ [Function], [Function] ] } }
[Browsersync] Access URLs:
-----
    Local: http://localhost:3000
  External: http://192.168.1.110:3000
-----
    UI: http://localhost:3001
  UI External: http://localhost:3001
-----
[Browsersync] Serving files from: src
[Browsersync] Watching files...
19.11.12 18:58:57 200 GET /index.html
19.11.12 18:58:57 200 GET /css/style.css
19.11.12 18:58:58 200 GET /js/index.js
19.11.12 18:58:58 200 GET /js/countries-data.js
19.11.12 18:58:58 404 GET /favicon.ico]
```

This will start a web server at `http://127.0.0.1:3000`, which points to the `src/` directory. Open your browser to point to this URL. When you save your files in `src/` the web server will automatically reload the browser.

NOTE: you should disable your browser's network cache when developing locally. See <https://nicholasbering.ca/tools/2016/10/09/devtools-disable-caching/>

Learn how to Lint your Code

Linting helps to find and remove common errors in code, for example, missing a semi-colon, or forgetting to declare a variable.

To run eslint, use the npm command:

```
npm run eslint
```

If there are no problems, there will be no output. If there is any output, pay attention to what it says, so you can make corrections. For example:

```
assignment4/assignment4.js
18:9  error  'x' is defined but never used  no-unused-vars
```

Here, we see a lint error, which has various information:

1. The filename is listed first, assignment4/assignment4.js
2. The line number is listed next: 18
3. The column number on line 18 is listed next: 9
4. The actual error or warning comes next: error 'x' is defined but never used
5. The rule name comes last: no-unused-vars. You can lookup how to fix these errors using the rule name, for example: <https://eslint.org/docs/rules/no-unused-vars>

Your code should have no lint errors when you submit it.

Learn how to Properly Format your Code

Source code needs to be properly structured, use consistent indenting, semi-colons, etc. Doing so makes it easier to understand, read, and debug your code.

Consider the following two functions:

```
// Improperly formatted and indented
function BaD(x){
if(      x> 10 ){
    return x;}
    return 0
}

// Properly formatted and indented
function good(x) {
    if (x > 10) {
        return x;
    }
    return 0;
}
```

Your code must be properly and consistently formatted. You can do it by hand, or, you can use Prettier (<https://prettier.io/>) to do it automatically.

To use Prettier, use the npm command:

```
npm run prettier
```

This will rewrite your assignment4.js file to use proper formatting. NOTE: running this command will overwrite your file, so make sure you have saved your work before you run it.

Submitting your Assignment

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, assignment4/
3. Copy your code, tests, and the package.json file into assignment4/
4. Run eslint on your code and save the result to assignment4/eslint.log
5. Run the unit tests and save the result to assignment4/test.log
6. Create assignment4.zip from the contents of assignment4/*

You can upload and submit the assignment4.zip to Blackboard.

Discussion of Other Assignment Files

You may be wondering about some of the other files in this project. While you don't need to modify them, or use them directly, here is what each one does:

node_modules/ – the installed dependencies necessary to run prettier, eslint, etc., installed when you run `npm install`.

.eslintrc.js – configuration for eslint, see <https://eslint.org/docs/user-guide/configuring>

.npmrc – configuration settings for npm, see <https://docs.npmjs.com/files/npmrc>

.prettierrc.js – configuration settings for prettier, see <https://prettier.io/docs/en/configuration.html>

package.json – node.js package info, dependencies, build scripts, see <https://docs.npmjs.com/files/package.json>

package-lock.json – a generated file with dependency version information, see <https://docs.npmjs.com/files/package-lock.json>