

Assignment 3 Instructions

Introduction

This assignment will help introduce DOM programming with HTML and JavaScript.

Please follow instructions in the WEB222 Assignment 3.pdf file.

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.

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, src/, 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.

Install a Proper Editor

You will need a proper editor to write code for the web. The most popular choice is Microsoft Visual Studio Code. VSCode runs on Windows, Linux, and macOS. You can download and install it for free from:

<https://code.visualstudio.com/>

A number of extensions are available that will work automatically with the scripts and configuration in this project, and are recommended:

- eslint <https://marketplace.visualstudio.com/items?itemName=dbaumer.vscode-eslint>
- prettier <https://marketplace.visualstudio.com/items?itemName=esbenp.prettier-vscode>
- stylelint <https://marketplace.visualstudio.com/items?itemName=shinnn.stylelint>

Once these are installed VSCode will automatically show you errors as you type.

If you want to use a different editor, that's fine. Confirm with your professor that it will work for our needs. You may not use Notepad, for example.

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...
19.10.16 10:33:14 200 GET /index.html
19.10.16 10:33:14 200 GET /css/normalize.css
19.10.16 10:33:14 200 GET /css/lab3-theme.css
19.10.16 10:33:14 404 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.

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 Run the Assignment Tests

This assignment does not have unit tests, but does provide automated HTML validation. To run the tests, use the npm command:

```
npm test
```

Your goal is to get all of the tests to pass correctly. If a test fails, pay attention to the error messages that get produced, and make corrections to your code..

Learn how to Lint your Code

In addition to running unit tests, you can also run a linter called `eslint`. 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:

```
src/js/audio.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, `src/js/audio.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 all files in `src/` 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
```

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