Assignment 3

Contents

While writing HTML and CSS is necessary for making any website, the core that drives these technologies forward is JavaScript. It is the backbone of every modern web framework and is required if you want to make interactive web pages. This assignment, however, wont be using a web browser at all. When we want to execute JavaScript outside of the browser environment we use Node.js. This makes writing JavaScript like writing any other programming language.

1. Setup

When you start the assignment open the a03 folder in your terminal and run the `npm install` command as you have done in past assignments. There are a few reasons why this is required for this assignment.

1.1 Why can't we just run the files normally?

JavaScript that runs in the browser follows a slightly different standard than JavaScript on your computer. This has been a point of conflict in the past, but now most things are shifting towards the browser standard.

One of the main differences is how importing and exporting is handled.

> ES6 modules and the `import` and `export` keywords are the standard. The specification is written by the ECMA TC39 (technical committee in charge of the language standards).
>
> However, it's (in relative terms) recent and since JavaScript didn't support modules prior to it, workarounds were developed. These include CommonJS (NodeJS modules) and AMD (RequireJS modules).
>
> On Node, CommonJS is going strong and there's a bit of a conflict regarding the "migration" to the standard, so there you likely don't get to choose.
>
> However, if you use JavaScript modules on browsers, you probably have a build step that bundles up your modules with something like Webpack or Browserify (at least until module loading is handled across the all relevant browsers). There you should definitely go for the standard ES6 modules and it's considered best practice.
>
> reddit post about this topic

Because of this difference and the fact that this is a web development course, we are going to be using es6 import export syntax. To use es6 style import and exports we require the use of the esm package. Here is a stackoverflow post about the topic. Feel free to google around.

1.2 How to run your code!

Visual Studio Code: Navigate to your `package.json`. Right click on the script you want to run and press 'Run Script'

Webstorm: Navigate to your `package.json`. Press the green play button next to the script you would like to execute.

All of the starting files where you will place your code have been provided for you inside the a03 folder. If you wish to separate your code into separate files or make helper functions feel free to do so but it should not be necessary.

> Normally when you want to run a `.js` file you just use the command line and call `node FileName.js`. But because the es6 import and export syntax is experimental in node, we need to use a package called ESM. For us, the user, the only difference is that we now run our node command as follows: `node -r esm FileName.js`. ESM will not be required in later versions of node.

1.3 Import / Export Example

- Import Documentation
- Export Documentation

hello_world.js

```
export function helloWorld(message) {
  console.log('Hello World,', message);
}
```

functions.js

```
import {helloWorld} from "./hello_world";

helloWorld('and exit.');
```

output after running the functions script inside `package.json`

```
Hello World, and exit.
```

2. Time to Code

Here are the MDN section on functions, arrays, and objects. Reading this isn't required but will give you the tools and knowledge to complete the assignment. The later section on advanced functions will have problems that are already solved by the language, so knowing what JavaScript can do is very helpful.

Below are two ways to write functions that output 'Basic Functions' as a string when called. *Note the export keyword is used here so they can be imported into a different file. This is how the autograder will check your work.*

```
// Defined with the function keyword
export function hello1() {
  return "Basic Functions";
}

// Defined with an arrow function
export const hello2 = () => {
  return "Basic Functions";
```

```
};

// Calling both functions
hello1();
hello2();
```

Before starting make sure you have the latest package.json from the CLI.

The following files need to have their function definitions filled out:

1. `mild_1.js`
2. `mild_2.js`
3. `medium_1.js`
4. `medium_2.js`
5. `medium_3.js`
6. `spicy_9.js`

If you find yourself writing a lot of code on `medium_*.js`:

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/reduce
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Map