JavaScript Exercises

These JavaScript exercises are intended to complement the JavaScript content on The Odin Project (TOP). They should only be done when instructed during the course of the curriculum.

Note: The `generator-exercise` file is not actually an exercise; it is a script that generates exercises. It was created to help efficiently write these exercises.

Contributing

If you have a suggestion to improve an exercise, an idea for a new exercise, or notice an issue with an exercise, please feel free to open an issue after thoroughly reading our contributing guide.

How To Use These Exercises

1. Fork and clone this repository. To learn how to fork a repository, see the GitHub documentation on how to fork a repo.

   - Copies of repositories on your machine are called clones. If you need help cloning to your local environment you can learn how from the GitHub documentation on cloning a repository.

2. Before you start working on any exercises, you should first ensure you have the following installed:

   - **NPM**. You should have installed NPM already in our Installing Node.js lesson. Just in case you need to check, type `npm --version` in your terminal. If you get back `Command 'npm' not found, but can be installed with:`, **do not follow the instructions in the terminal** to install with `apt-get` as this causes permission issues. Instead, go back to the installation lesson and install Node with NVM by following the instructions there.
   - **Jest**. After cloning this repository to your local machine and installing NPM, go into the newly created directory (`cd javascript-exercises`) and run `npm install`. This will install Jest and set up the testing platform based on our preconfigured settings.

3. Each exercise includes the following:

   - A markdown file with a description of the task, an empty (or mostly empty) JavaScript file, and a set of tests.
   - A `solutions` directory that contains a solution and the same test file with all of the tests unskipped.

   To complete an exercise, you'll need to go to the exercise directory with `cd exerciseName` in the terminal and run `npm test exerciseName.spec.js`.

This should run the test file and show you the output. When you first run a test, it will fail. This is by design! You must open the exercise file and write the code needed to get the test to pass.

4. Some of the exercises have test conditions defined in their spec file as `test.skip` compared to `test`. This is purposeful. After you pass one `test`, you will change the next `test.skip` to `test` and test your code again. You'll do this until all conditions are satisfied. **All tests must pass at the same time**, and you should not have any `test.skip` instances by the time you finish an exercise.
5. Once you successfully finish an exercise, check the `solutions` directory within each exercise to compare it with yours.
   - You should not be checking the solution for an exercise until you finish it!
   - Keep in mind that TOP's solution is not the only solution. Generally as long as all of the tests pass, your solution should be fine.
6. Do not submit your solutions to this repo, as any PRs that do so will be closed without merging.

Note: Due to the way Jest handles failed tests, it may return an exit code of 1 if any tests fail. NPM will interpret this as an error and you may see some `npm ERR!` messages after Jest runs. You can ignore these, or run your test with `npm test exerciseName.spec.js --silent` to supress the errors.

The first exercise, `helloWorld`, will walk you through the process in-depth.

Debugging

To debug functions, you can run the tests in the Visual Studio Code debugger terminal. You can open this by clicking the "Run and Debug" icon on the left or pressing Ctrl + Shift + D, then clicking JavaScript Debug Terminal. You will be able to set breakpoints as you would in the Chrome DevTools debugger. You can run `npm test exerciseName.spec.js` to then execute your code up until your breakpoint and step through your code as necessary. **NOTE**: To take advantage of the debugger, you **MUST** run the script in the debugger terminal, not the bash or zsh terminal.