Exercise | Monday

Intermediate JavaScript (/intermediate-javascript) / Test-Driven Development and Environments with JavaScript (/intermediate-javascript/test-driven-development-and-environments-with-javascript) / webpack and npm Practice

Text

Goal: Practice using npm and webpack to streamline your development process.

- Separate logic into at least two JS files (business logic and interface logic). Use `import` and `export` statements to get them communicating.
- Configure webpack to include all of the functionality covered in the section's weekend homework.
- Configure ESLint to lint your JavaScript files.

We recommend building every webpack environment from scratch today. Then, in the future, you can copy/paste your configuration files and run `$ npm install` in the root of your project to set it up..

Warm Up

- What is npm? Why is it useful for our projects?
- What is webpack and what is it used for?
- What is the difference between concatenation and minification?
- What is linting? Why is it useful in development?

Code

You are not expected to complete all of the project prompts on this page. Start by following along with the weekend homework. Then proceed to the journal project. If you have additional time, pick either Memory or Simon.

Feel free to expand on these problems to challenge yourself if you want, but only after your environment is fully set up.

Shape Tracker

Follow along with the homework to set up a basic environment. Make sure to start by creating your `.gitignore` file! Also, don't build out any business logic. In upcoming homework, we'll use test-driven development to write Shape Tracker's business logic. Just focus on getting familiar with the new tools we are using.

Journal

Create a journaling website where a user can write entries including at least a title and body. Create `Entry` objects that include a method to return the number of words in the entry. Then, add a separate method (or methods) to return the number of vowels and consonants in each entry. Finally, add a method called `getTeaser` to return the first sentence of the entry. If the sentence is over 8 words, only display those first 8 words.

Challenging: Memory

It's fine if you just work on Shape Tracker and Journal today, but if you are looking for more of a challenge, create a game based on the card game Memory.

You'll need to generate a grid of cards. Each card can have a word or number on it, and there should be two of each card. If you have a deck of ten cards there should be five different pairs of cards.

All cards should start face down. Each time the user takes a turn, they click on two cards. When a card is clicked, the pictures should be revealed. If the user gets a pair of cards with the same picture, the cards remain face-up. If the user's selected cards are different, they should return to face down. When all the cards are face up, the user has won. The object of the game is to find all the pairs in as few turns as possible.

If you'd like to try this prompt with images instead of numbers or words, you will need to use `image-loader`, which we haven't covered yet. See the lesson on Managing Images with webpack (https://www.learnhowtoprogram.com/intermediate-javascript/test-driven-development-and-environments-with-javascript/managing-images-with-webpack) for more information.

Most Challenging: Simon

This prompt takes the challenge level up another notch. Create a version of the game Simon (https://www.youtube.com/watch?v=4YhVyt4q5HI). In Simon, the game generates a sequence of colored lights for you to mirror. It starts out simple; first only one color, then if you get that right then you have to remember 2 colors. The sequence does not change every time, it just gets longer and longer.

For example, here is a sequence of turns:

```
["red"]
["red", "blue"]
["red", "blue", "yellow"]
["red", "blue", "yellow", "red"]
["red", "blue", "yellow", "red", "red"]
["red", "blue", "yellow", "red", "red", "yellow"]
...
```

This is just an example. You do not have to use the above format to generate sequences,

Hint: Since this game involves timed events (we have to time how fast it takes for the sequence to play on each turn) you may wish to investigate the `window.setInterval()` method (https://developer.mozilla.org/en-US/docs/Web/API/setInterval).

Peer Code Review

- Dependencies are managed with npm.
- webpack is used to lint, bundle, and process code.
- User interface and business logics are separated into different files.

Previous (/intermediate-javascript/test-driven-development-and-environments-with-javascript/journal-5-discussion)
Next (/intermediate-javascript/test-driven-development-and-environments-with-javascript/tdd-review)

disable dark mode

(http://www.epicodus.com)

© 2023 Epicodus (http://www.epicodus.com/), Inc.