Interacting with the DOM using Events

- Due No Due Date
- Points 1
- Submitting a website url

(https://github.com/learn-co-curriculum/phase-1-practice-interacting-with-the-dom) (https://github.com/learn-co-curriculum/phase-1-practice-interacting-with-the-dom/issues/new)

Learning Goals

- Create event listeners on DOM nodes using `addEventListener()`
- Manipulate the DOM in response to events

Instructions

If you open up index.html in the browser, you'll find a small application. It has a few features:

- A Counter that increases by 1 each second
- Plus and Minus buttons that increment or decrement the counter
- A "like" button (❤️) that adds a "like" for the number that is currently displayed by the timer
- A comment box that adds comments when submitted

First, take a few minutes to explore and test out the behavior of the page. Think about how each of the features must work. What must be happening underneath each of these features?

The challenge is to reverse engineer these features and build them yourself, using what you've learned about JavaScript and DOM Manipulation.

Note that there are **no tests** for this lab. To confirm if the application is working, you'll have to manually test the application in the browser.

Once you're done, be sure to commit and push your code up to GitHub, then submit the assignment using CodeGrade. Even though this practice lab does not have tests, it must still be submitted through CodeGrade in order to be marked as complete in Canvas.

Getting Started

To begin, comment out the script tag linking to `js/index.min.js` in the html file. This will "shut off" the existing behaviors so you can begin recreating them, using the `js/challenge.js` file to write your code. You can see the `<script>` tag that loads this file just after the one for `index.min.js`.

Deliverables

Add the following features one by one, refreshing the page to see your functionality working as you build.

> Note: These deliverables are written in the form of *User Stories*. They describe the features from the perspective of a user when they visit the page.

As a user, I can:

- See the timer increment every second once the page has loaded.
- Manually increment and decrement the counter using the plus and minus buttons.
- "Like" an individual number of the counter. I should see the count of the number of "likes" associated with that number displayed.
- Pause the counter, which should:
  - pause the counter
  - disable all buttons except the pause button
  - switch the label on the button from "pause" to "resume"
- Click the "resume" button to restart the counter and re-enable the buttons.
- Leave comments on my gameplay, such as: "Wow, what a fun game this is."

Timer Hints

If you're not sure how to create or pause a timer, look into:

- `setTimeout` (https://developer.mozilla.org/en-US/docs/Web/API/WindowOrWorkerGlobalScope/setTimeout)
- `setInterval` (https://developer.mozilla.org/en-US/docs/Web/API/WindowOrWorkerGlobalScope/setInterval)
- `clearinterval` (https://developer.mozilla.org/en-US/docs/Web/API/WindowOrWorkerGlobalScope/clearInterval)