Before everyone starts to work on their stuff, I wanted to remind you of what I have to check for in all of your components as Product Manager.

According to Richard, I must check for:

- any plagiarized code (don't just copy and paste some other team's or framework's components). It's totally fine to copy Paul Irish's box-sizing reset, for example, but do not go to Bootstrap's site and copy their tab component's code.

- valid HTML and CSS without errors, and I will be running your JS (if any) through JSHint

I am also supposed to encourage you to follow the JavaScript best practices (link) and CSS best practices (from the chapter we were assigned) as much as possible

The ones which I especially encourage you to follow are:

JavaScript

- Variable declarations should be made a new line each, and at the top of their scope. (link). It's much easier to see at a glance what your variables are if they aren't all declared on one line.

- Use strict mode, especially being this will ideally end up in projects other people are working on (link)

- Comparisons - using strict equality (link)

- Naming your function expressions for debugging purposes (link)

CSS

- use relative units as much as possible. Try not setting things like width, padding, and margins in `px`. Use `em`, `rem`, and `%` as much as possible.

- whenever we decide on a naming convention, stick with it

- organize your CSS declarations in alphabetical order. This makes it easier to know where to look to change or add a new declaration. When they are randomly ordered, it's harder to work with. For example:

  GOOD

  ```
  .some-selector {
    color: red;
    line-height: 1.4;
    padding: 1em;
  }
  ```

  BAD

  ```
  .some-selector {
    padding: 1em;
    line-height: 1.4;
    color: red;
  }
  ```

- never use IDs for styling. Even if you just want to style that one element, give it a class and style that class. Using IDs for styling is a clear path to specificity hell if you ever need to override it.

  BAD

```
#my-el {
  color: blue;
}
```

GOOD

```
.my-el {
  color: blue;
}
```

- try to apply as little specificity as possible. It's not always possible to just use one selector when styling, but it should the the goal, and in general, you should be able to get away with it, only adding a bit of specificity where needed. Being overly specific should not be the default. For example, with the following markup:

  ```
  <nav class="main-nav">
    <ul>
      <li></li>
      <li></li>
      <li></li>
    </ul>
  </nav>
  ```

  if you want to style the `li` elements, try adding a class to them (something like `nav__item`) and styling that:

GOOD

```
.main-nav__item {
```

```
  padding: 1em;
}
```

instead of doing something like this, which adds specificity that we do not need:

BAD

```
.main-nav li {
  padding: 1em;
}
```