Common issues with the use of CSS classes

- When to have classes
- What name to use
- BEM or not?

When to have a class (Generally)

- To define the default behavior of an element
    - Use element selector
        - `<ul>` will look like this by default
        - `<button>` will look like this by default
- If you are styling HTML that you don't control
    - Use "compound" selectors of whatever
- Otherwise, use a class
    - Mostly a single class
    - Compound only when dependent on a state
        - Ex: menus having open/active ancestor

Rule of thumb for classes

- Will this element be styled?
    - Give it a class
- Will this element be interactive?
    - Give it a class

BEM Style

- BEM style is an option for this course
    - A classname is either BEM or it is not
- BEM style:
    - BLOCK, ELEMENT, MODIFIER each semantic
        - Each a meaningful name
    - `.BLOCK`
    - `.BLOCK--MODIFIER`
    - `.BLOCK__ELEMENT`
    - `.BLOCK__ELEMENT--MODIFIER`
- BEM is for humans
    - "It works" is talking about the computer

BEM Block

- `.BLOCK`
- A "chunk" of the page
- Just a semantic label
    - Including being `kebab-case`
    - Multiple words connected by single hyphen
    - All lowercase
- No different than non-BEM Semantic labels
    - Benefit comes when we add elements
- Blocks can appear inside other blocks
    - Use when you will use/style parts inside

BEM Element

- `.BLOCK__ELEMENT`
    - Exactly two underscores!
    - Only two parts
        - One Block
        - One Element
- ELEMENT is subblock
    - Might be actual semantic element type
    - Might be a semantic class name
        - Same `kebab-case` naming rules

BEM Modifier

- `.BLOCK__ELEMENT--MODIFIER`
- Will be `.BLOCK__ELEMENT` as well!
    - Have both class names
- Exactly two underscores
- Exactly two hyphens
- MODIFIER is a semantic *state*
    - `open`, `active`, etc
    - same `kebab-case` rule
    - Comes up a lot more with JS
        - We add/remove states

Not Using BEM? Not a problem!

- Course requires for EVERY class name:
    - Semantic
        - Name describes block/contents
    - `kebab-case`
        - No underscores
- Outside of course
    - Lots of different rules sets (still rules!)
- Either way, want to use classes
    - Styled elements
    - Elements targeted by JS
    - Elements modified by JS

Why do we want classes?

- All selectors using single classes
  - Same specificity
- Makes CSS more readable
- CSS breaks less often when HTML changes

```
main div div a {
  /* styling */
}
```

vs

```
.card__link {
  /* styling */
}
```

Dropdown with JS - The Hint

- *Toggle a class on the element that you were looking at :hover/:focus-within on when the button is clicked*
- *Have your CSS apply the same changes you did on :hover/:focus-within, but only when that extra class is present*
- *Make sure to use a class name that is descriptive and semantic, such as "open"*

Toggle on the element you were looking at

> - *Toggle a class on the element that you were looking at :hover/:focus-within on when the button is clicked*

- Was like: `.menu__item .menu__button:hover ul {`
- Now like: `.menu__item .menu__button.open ul {`

Notice how the logic is very similar

BEM Style would be:

- `class="menu__button menu__button--open"`