Reviewing Responsive Grid

Overall for all students

- Reminder: PR comments in Github
- Class names both semantic and kebab-case
- Prefer class name selectors
- Pretty up CSS with spaces
- `<title>`
- Prefer span for grid-column
- commit messages

There are comments on your pull requests!

- Not in Canvas
- You may not have set to get emails
    - You may ignore emails
- *Valuable!* Go read these

Common issues with Class Names

Your class names need to be both:

- Semantic
- Kebab-case

Many are doing well, but still some common issues

Basic Semantic Class Names

Named for the what the content of the element represents

- Semantic: `profile`
- Semantic: `card`
- Semantic: `menu`

Semantic class names do not describe appearance!

Utility class names describe impact of styling

- Including appearance

We are using semantic class names, not utility class names

- In the future you may use both
- But learn difference now

Content or purpose is not a specific appearance

- Not Semantic: `left`
- Not Semantic: `grid`

Numbers in class names are not great

I see a lot of `.footer-link1` or `.card2`

Numbered class names:

- Make it hard to rearrange your contents
- Can't be used for common styling
    - "all cards"
- Usually unnecessary (`:nth-child`/`:nth-of-type`)

Semantics MAY match elements

SOME element names are semantic names:

- Semantic: `header`
- Semantic: `footer`
- Semantic: `main`

Not all semantic elements have semantic names:

- Not Semantic: `a`
- Not Semantic: `li`
- Not Semantic: `ul`

Semantic if element, Not semantic if appearance:

- `button`, `btn`

Compound Semantics

SOMETIMES a semantic name is multiple words

- SOMETIMES these are put together from elements
- Not always
- Only when the extra part better identifies the "thing" the element contains or represents
    - Or better distinguishes it from similar things
- Semantic: `primary-nav`
- Semantic: `menu-item`
- Probably Not Semantic: `list-item-link`

Avoid embedding structure into Semantic Classes

A `card` or `profile` are those things

- Regardless of where in the HTML they appear
- Selectors can "scope" styling based on structure
    - Example: `.header .menu` vs `.main .menu`
- Structure only in class name if part of identity
    - Example: `footer-link` probably too tied to structure
        - Just `link` or selector like `.footer a`

Kebab-case

- Separate concept from "semantic"
- All letters lowercase
- Single `-` between words
- No other special characters (no `_` until BEM)

I require kebab-case AND Semantic

- Note earlier comment about compound names

English is weird

- Some compound words are one word
- Sometimes inexact ("web site" vs "website")
- But `headerlink` is not Kebab-case

Prefer class name selectors

- This is not a "hard" rule
  - But not to be ignored, either
  - Not every element needs a class name
  - Goal is code that is easy to read and change
- Any element collecting descendants as a concept
  - Should be named
- Avoid selectors with a lot of parts
  - Poor: `nav ul li a`
  - Better: `.primary-nav a`
  - A bit changes when we get to BEM

CSS should be easy to skim

- Leave a space after selector:
    - Poor: `p{`
    - Better: `p {`
- Leave a space after colon:
    - Poor: `background-color:plum;`
    - Better: `background-color: plum;`

Very tiny effort, really makes a cumulative difference

- Will Skim more than Read, Read more than Write

Title Element

This surprised me

- Blank Title elements
- Meaningless Title elements
    - "Title"
    - "My Page"
    - "Something unrelated to assignment"

Always have a title element

- With a useful value
- May match `<h1>`, should be similar

Be laz...efficient. Use grid-column span

- `grid-column: 5 / 9;`
    - Works
    - Requires placement of every element
    - Requires changes if an element removed
    - Ideally only when you want empty cells
- `grid-column: span 4;`
    - Works
    - Automatically next cells/rows as needed
        - Handles changes automatically
    - Best unless you want empty cells

Commit messages

- Your commit message is part of ENTIRE repo
  - `Adds index.html and styles.css`
    - Meaningless, repo has MANY of these!
- One short sentence!
  - Team needs only enough to know if they care
- Poor: `Adds index.html and styles.css`
- Poor: `Requirement to generate a page where...`
- Fine: `Adds assignment 3`
- Fine: `Fixes missing files in assignment 3`