CSS Organization

We talked about the different approaches:

- Semantic
- BEM (incl OOCSS and similar)
- utility-first

But what about the rules and declarations in the CSS files themselves?

Organization is needed

You quickly start noticing the problem

- Have I styled this element already?
- Where?
- Have I declared this property already?
- Where?

CSS Organization

Some form of organization is needed

- How to order the selectors
- What files to put the selectors in
- How to order the properties in declarations

Multiple Options

We simply do not know what is best

We only know what is worst:

- No organization

Follow the style of your team, if you have one Otherwise, create a pattern

Preprocessors and Tooling

We will discuss preprocessors later

- They give additional options for organization
- Multiple files, nested declarations, etc

Files

- Can you break up rules into multiple files?
    - Without tools, this should be limited
    - B/C multiple files means more downloads for client
- Should only consider rules that are contained

Selectors

How are you organizing the different rules (by selector)

- expected page order?
- generic to specific?
- sections (globals, header, menu, etc)

Comments for boundaries help `/* comment */`

Properties

Within a CSS Rule, how are you ordering the list of declarations?

- alphabetical?
    - low-value
    - easy to automate in a large team
- by purpose (My recommendation)
    - display/position/box model/colors/etc
    - separate by blank lines

My Preferred Style: Selector order

- "sweeping" CSS first
    - `*` selectors, `body`, `html`
    - "resets"
    - Any common styling (Ex: `p`, etc)
- General Page order for next selectors
    - Often BEM style
    - Try to select based on class

My Preferred Style: Properties

Within each CSS Rule:

- "Paragraphs" for different categories
- Position, display, box model, colors, typography

```
.card__link {
  margin-top: auto;
  padding: 0.5rem;
  border: 1px solid black;
  border-radius: 10px;

  color: black;
  background-color: lime;

  font: inherit;
  text-decoration: none;
}
```

My preference is just a preference

- Different teams and projects make choices
    - System adds value through:
        - Consistency
        - Effectiveness of communication/finding
- No agreement on a "best"
- Course requires you have a consistent system
    - You pick what that system is

Enforcement

How do we make sure these conventions are followed?

- Automated tooling
- Code Reviews