
Today's agenda

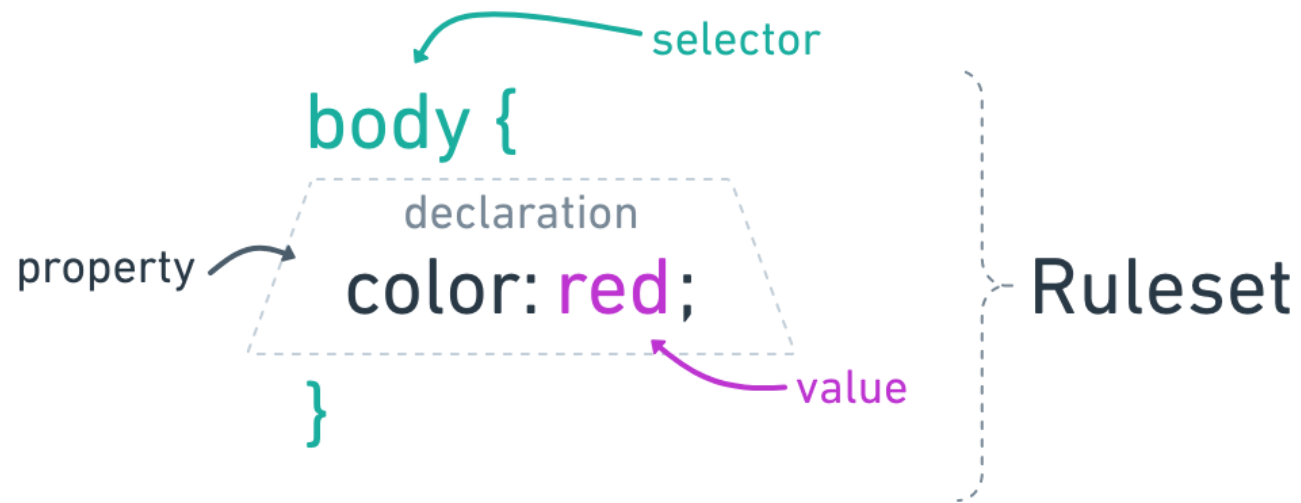
1. Slack checkin

2. CSS

- The cascade
- Defaults to be aware of
- Best practices
- Structuring your CSS
- Naming structures
- Some gotchas
- Sass: CSS superpowers

Deep dive into CSS

Talking about CSS



CSS defaults

Inline elements

Block-level elements

Browser defaults

Heading

Lorem

Another heading

- List item
- List item
- List item
- List item
- List item

The screenshot shows a web browser's developer tools interface. The left pane displays the HTML document structure, with the `<body>` element selected. The HTML code is as follows:

```
<!doctype html>
<html lang="en">
  <head>...</head>
  <body translate="no"> == $0
    <h1>Heading</h1>
    <p>Lorem</p>
    <h2>Another heading</h2>
    <ul>
      <li>List item</li>
      <li>List item</li>
      <li>List item</li>
      <li>List item</li>
      <li>List item</li>
    </ul>
  </body>
</html>
```

The right pane shows the 'Styles' tab, displaying the default user agent styles for the `body` element:

```
body {
  display: block;
  margin: 8px;
}
```

Below the styles, a box model diagram is shown, illustrating the default margins (8px) and padding (0px) for the `body` element. The diagram shows a blue box representing the content area (272 x 234.250) surrounded by a green padding area and an orange margin area (8px).

Best practices

Use classnames over bare elements or IDs

```
<h1 class="page-title cursive"></h1>
```

Write your CSS lowercase and kebab case

```
.page-title {  
  text-transform: uppercase;  
}
```

Transform case using CSS

Remember your semicolon

Best practices

Learn about methods of organizing CSS/Sass and use a system

```
div {  
  background-color:  
whitesmoke;  
  border: 3px dotted #f9c7dd;  
  color: #444;  
  display: inline-block;  
  font-family: "Montserrat",  
sans-serif;  
  font-size: 100%;  
  padding: 1em;  
}
```

```
div {  
  /* -- Box -- */  
  display: inline-block;  
  padding: 1em;  
  
  /* -- Border -- */  
  border: 3px dotted #f9c7dd;  
  
  /* -- Background -- */  
  background-color:  
whitesmoke;  
  
  /* -- Text -- */  
  color: #444;  
  font-family: "Montserrat",  
sans-serif;  
  font-size: 100%;  
}
```

Best practices

Use CSS comments

```
/*  
* === SITE HEADER ===  
*/
```

```
/*  
* --- Header nav ---  
*/
```

Best practices

Avoid !important

Best practices

Avoid using empty divs to achieve your styles

```
<div class="something-fancy"></div>
```

Best practices

Measurements of zero don't need units

Avoid

```
div {  
  margin: 0px;  
  padding: 0px;  
  width: 0px;  
}
```

Do this

```
div {  
  margin: 0;  
  padding: 0;  
  width: 0;  
}
```

Best practices

Avoid overdoing your selectors and specificity

```
<header class="site-header">
  <nav class="nav-bar">
    <ul class="nav-menu">
      <li class="nav-
menu__item"></li>
      <li class="nav-
menu__item"></li>
      <li class="nav-
menu__item"></li>
    </ul>
  </nav>
</header>
```

Avoid

```
header.site-header nav.nav-
bar ul.nav-menu li.nav-
menu__item {}
```

Do this

```
.nav-menu__item {}
```

OR

```
.site-header .nav-menu__item
{}
```

Best practices

Avoid making unnecessary assumptions about the markup

Avoid

```
<span class="highlight"><a  
href='#>Highlighted link</a></span>
```

```
.highlight a { background: yellow; }
```

Do this

```
<a href='#' class="highlight">Highlighted  
link</a>
```

```
.highlight { background: yellow; }
```

Best practices

Avoid writing CSS you're immediately going to override

Avoid

```
li {  
  display: inline-block;  
  margin-left: 14px;  
}  
li:first-of-type { margin-left: 0; }
```

Do this

```
li { display: inline-block; }  
li + li { margin-left: 14px; }
```

Best practices

Avoid making an HTTP request for a resource when you can easily replicate the element with CSS

Avoid

```
li::before {  
  content: url(green-  
circle.svg);  
}
```

Do this

```
li::before {  
  background: green;  
  border-radius: 50%;  
  content: "";  
  display: block;  
  height: 20px;  
  width: 20px;  
}
```

View on CodePen: <https://codepen.io/angeliquejw/pen/jvYvpy>

Best practices Q&A

Structuring your CSS

1. Reset
2. Utilities/helpers
3. Base
4. Custom styles

Learn about resets

Reset.css (group 1)

- Read about it: <https://meyerweb.com/eric/tools/css/reset/>
- View the code: <https://codepen.io/angeliquejw/pen/BOmXNa?editors=0100>

Normalize.css (group 2)

- Read about it: <http://nicolasgallagher.com/about-normalize-css/>
- View the code: <https://codepen.io/angeliquejw/pen/rZYXOz?editors=0100>

Extra time? <https://css-tricks.com/reboot-resets-reasoning/>

Learn about resets

1. Join your group
2. Establish roles:
 - Timekeeper
 - Presenter
3. As a group, spend 10 mins discussing:
 - 3-5 main things to know
 - 2 pros and 2 cons

Reset

<https://s.codepen.io/angeliquejw/debug/ZMvqBj>

Utility classes

<https://codepen.io/angeliquejw/pen/wEpvqN?editors=0100>

Base styles

```
body {  
  background: whitesmoke;  
  color: #1b203a;  
  font-family: "Fira Sans", Sans-Serif;  
  margin: 0;  
}  
  
h1, h2, h3, h4, h5, h6, p, li {  
  line-height: 1.5625em;  
  margin: 0;  
}
```

Custom CSS

Structuring your CSS Q&A

CSS Naming Conventions

BEM

BEM
Block

BEM

Block

```
.block {}
```

BEM

Block

```
.block {}
```

Element

BEM

Block

```
.block {}
```

Element

```
.block__element {}
```

BEM

Block

```
.block {}
```

Element

```
.block__element {}
```

Modifier

BEM

Block

```
.block {}
```

Element

```
.block__element {}
```

Modifier

```
.block--modifier {}  
.block__element--modifier {}
```

BEM examples

```
<figure class="photo">  
    
  <figcaption class="photo__caption">C'est  
moi!</figcaption>  
</figure>
```

```
.photo {}  
.photo__img {}  
.photo__caption {}
```

BEM examples

```
<figure class="photo photo--featured">  
    
  <figcaption class="photo__caption">C'est  
moi!</figcaption>  
</figure>
```

```
.photo {}  
.photo__img {}  
.photo__caption {}  
  
.photo--featured {}  
.photo--featured .photo__img {}  
.photo--featured .photo__caption {}
```

BEM examples

```
<form action="#" class="search-form">  
  <input class="search-form__input"  
type="search">  
  <button class="button search-form__button"  
type="submit"></button>  
</form>
```

BEM variations

By the book

```
<button class="button button--disabled"  
type="submit"></button>
```

```
.button {}  
.button--disabled {}
```

Functional variation

```
<button class="button" disabled type="submit">  
</button>
```

```
.button {}  
.button[disabled] {}
```

BEM variations

By the book

```
<section class="plans plans--hidden">  
</section>
```

```
.plans {}  
.plans--hidden {}
```

Functional variation

```
<section class="plans" hidden></section>
```

```
.plans {}  
.plans[hidden] {}
```

Criticisms of BEM

- Ugly
- More typing

Naming things well

- Understandable
- Obvious
- Functional
- Consistent

Naming things Q&A