

Random Useful Bits

I haven't found a good place for all of these yet

- You should dive into MDN and ask questions

Emmet

- Editor may have "snippets"
 - define expansions of known content
- Emmet is a generic standard
 - for HTML and CSS (and lorem ipsum text)

<https://docs.emmet.io/>

Lorem Ipsum

Fake text

- taken randomly from an old latin speech
- see how a layout looks with "text-like" content
- Real content is always better
 - But rarely available at design time
- Many tools to generate "lorem text"
 - in-editor or websites to cut/paste

CSS Units

- `vh` and `vw`
 - "viewport"
- `rem` vs `em`
 - "em" causes inheritance problem
 - sizes based off of "root" font "em" width
 - "root" is `<html>` element
 - **<https://css-tricks.com/html-vs-body-in-css/>**
 - `rem` useful with browser text settings

CSS Variables

Often we have values that we want to reuse

- height/widths of elements interacted with (nav?)
- colors (background, accent, highlight, etc)

Technically these are "custom properties", and follow the normal cascading/precedence rules

Outside CSS

CSS took a long time to add variables

We will talk about SASS and LESS later, they have their own solutions

- But SASS/LESS aren't actual CSS

This is the pure (but limited) CSS solution

Using a CSS Variable

Assign:

```
.some-selector {  
  --my-var: black;  
  --another: 5rem;  
}
```

Use:

```
p {  
  color: var(--my-var);  
}
```

"Global" assign:

```
:root {  
  --main-bg-color: #BADA55;  
}
```

CSS Properties

- `filter`
 - `filter: brightness()`
- `opacity`
- `font-family`
- `visibility`
 - hides without removing from layout
 - can be good/bad for accessibility

Pseudo-classes

Added to a selector to indicate a state

- `:hover`
- `:focus` and `:focus-within`
- `:active`
- `:not()`
- `:first-child`
- `:nth-child()`

Pseudo-elements

Not elements, but allow you to style them like one

- `::selection`
- `::before` and `::after`
 - `content` property

CSS Functions

- `calc()`
- `max()` and `min()`
- `clamp()`
 - 3 args, preferred should be a value that changes

Media Queries

- Wraps CSS Rules
- Rules applied or not based on query
- Says if the rules are matched

Screen Width

If CONDITION, apply CSS rules

```
@media (min-width: 1000px) {  
  body {  
    background-color: red;  
  }  
}
```

Reduced Motion

- options are `no-preference` or `reduce`
- which involves less work?

```
@media (prefers-reduced-motion: no-preference) {  
  .my-element {  
    animation: flashy-zoom-in-out 1s;  
  }  
}
```

Orientation

- If you care past width...

```
@media (orientation: portrait) {  
  body {  
    display: flex;  
    flex-direction: column;  
  }  
}
```

Printing

A deep rabbithole

- Alternative to generating PDFs
- Not always the best alternative

```
@media print {  
  h3 {  
    page-break-before: always;  
  }  
}
```