What is Sass?

Sass (Syntactically Awesome Style Sheets) is a CSS preprocessor.

How does it work?

- There are several ways you can compile Sass:
- The original Ruby Sass binary. Install it with gem install sass, and compile it by running sassc myfile.scss myfile.css.
- A GUI app such as Hammer, CodeKit, or Compass
- libsass, which is a fast Sass compiler written in C. You can also install libsass via NPM with node-sass (npm install node-sass).
- Sassmeister

.sass vs .scss

- When Sass first came out, the main syntax was noticably different from CSS. It used indentation instead of braces, didn't require semi-colons and had shorthand operators.
- In version 3 Sass changed it's main syntax to .scss. SCSS is a superset of CSS, and is basically written the exact same, but with new Sass features.

Sass syntax

```
$font-stack:    Helvetica, sans-serif
$primary-color: #333

body
  font: 100% $font-stack
  color: $primary-color
```

Scss syntax

```
$font-stack:    Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

Why would I use Sass?

- Sass makes writing maintainable CSS easier. You can get more done, in less code, more readably, in less time.
- Sass lets you use features that don't exist in CSS yet like variables, nesting, mixins, inheritance

Comments

```
/* this comment will be compiled and will
appear in the compiled css file */

// this comment will not appear in

// the compiled css file
```

Variables

- A way to store information that you want to reuse
- Acceptable values for variables include numbers, strings, colors, null, lists and maps.
- Variables in Sass are scoped using the $ symbol.

```
$primaryColor: #eeffcc;
```

Scope

- If you declare a variable within a selector, it is then scoped within that selector.

```
$primaryColor: #eeccff;
body {
  $primaryColor: #ccc;
  background: $primaryColor;
}
p {
  color: $primaryColor;
}
// When compiled, our paragraph selector's color is #eeccff
```

Global

- !global flag allows us to set a variable globally from within a declaration:

```
$primaryColor: #ccc !global;
```

- !default flag allows us to make sure there is a default value for a variable if not provided. If a value is provided, it is overwritten:

```
$firstValue: 62.5%;

$firstValue: 24px !default;
```

Nesting

HTML has a clear nested and visual hierarchy. CSS, on the other hand, doesn't.

Sass will let you nest your CSS selectors in a way that follows the same visual hierarchy of your HTML.

Generally too much nesting is considered bad practice – recommended not more than 3-4 levels.

Partials

- Partial Sass files that contain little snippets of CSS that you can include in other Sass files
- A partial is simply a Sass file named with a leading underscore.
  - _partial.scss
- Sass partials are used with the @import directive.

Import

- CSS has an import option lets you split your CSS into smaller, more maintainable portions
- Each time you use @import in CSS it creates another HTTP request
- Sass will take the file that you want to import and combine it with the file you're importing into so you can serve a single CSS file to the web browser

Partials and Imports

```
// _reset.scss


html, body, ul, ol {
    margin:  0;
    padding: 0;
}
```

Partials and Imports

```
// main.scss

@import 'reset';

body {
  font: 100% Helvetica, sans-serif;
  background-color: #efefef;
}
```

Mixins

- A mixin lets you make groups of CSS declarations that you want to reuse throughout your site.

```
@mixin flex {
    // write the css here
    display: -webkit-flex;
    display: flex;
}
```

Use a Mixin

To use a Mixin, we simply use @include followed by the name of the Mixin and a semi-colon.

```
.row {
    @include flex;
}
```

Mixin result in CSS

```
.row {
    display: -webkit-flex;
    display: flex;
}
```

Passing Arguments to Mixins

```
@mixin grid($flex) {
    @if $flex {
        @include flex;
    } @else {
        display: block;
    }
}
@include grid(true);
```

Extend/Inheritance

Using @extend lets you share a set of CSS properties from one selector to another.

```
%message-shared {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

.message {
  @extend %message-shared;
}
```

Maths/Operators

Doing math in your CSS is very helpful. Sass has a handful of standard math operators like +, -, *, /, and %.

```
article[role="main"] {
  float: left;
  width: 600px / 960px * 100%;
}
```

Exercises

- https://www.sassmeister.com/
- https://codepen.io/digital-career-institute/pen/mprPjV
- https://codepen.io/digital-career-institute/pen/LeRNKY
- https://www.codecademy.com/learn/learn-sass

- https://codepen.io/collection/zCrhE/

Sources

- https://marksheet.io/sass-mixins.html
- https://sass-lang.com/guide
-