Branch: master curriculum / 04-advanced-ui / 02_sass_basic.md

Find file Copy path

Xaviju Add install instruction ebbfa01 10 days ago

1 contributor

71 lines (45 sloc) 2.88 KB

Raw Blame History

Sass basics

Go into this website to test this lesson: https://www.sassmeister.com/

A CSS preprocessor is a program that lets you generate CSS from the preprocessor's own unique syntax. There are many CSS preprocessors to choose from, however most CSS preprocessors will add some features that don't exist in pure CSS, such as mixin, nesting selector, inheritance selector, and so on. These features make the CSS structure more readable and easier to maintain.

Preprocessing

Preprocessing CSS is becoming an standard in the workflow of web development. Preprocessing is a process in which the code we write in another language (Sass, LESS, Stylus, postCSS...etc) is converted automatically to standard CSS before being executed in our website. using a CSS preprocessor as Sass wil lgive you everything CSS has and many more things (mixins, conditionals, nesting...etc.). This features are intended to make CSS development faster and more effective

> Sass file ----> preprocessor tool ----> Standard CSS

We will be learning Sass since is the most common CSS preprocessor language but there are many:

- Sass: Sintactically awesome Style Sheets
- LESS
- Stylus
- PostCSS

Sass has two different syntaxes (sass and scss). We will use `scss` since is the most commonly used.

Variables

Sass implements variables. Think of it as similar to custom properties, although Sass variables are less powerful because we cannot change it dynamically through javascript.

WARNING: It is recommendable to use custom properties instead of Sass variables if our browser supports them

Think of variables as a way to store information that you want to reuse throughout your stylesheet. You can store things like colors, font stacks, or any CSS value you think you'll want to reuse. Sass uses the $ symbol to make something a variable. Here's an example:

```
$primary-color: #333;

body {
  color: $primary-color;
}

element {
  background: $primary-color;
}
```

> Execute this on `Sassmeister` to see what happens.

Nesting

When writing HTML you've probably noticed that it 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.

```
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }

  li { display: inline-block; }
}
```

Be aware that **overly nested rules will result in over-qualified CSS that could prove hard to maintain** and is generally considered bad practice.

> Execute this on `Sassmeister` to see what happens.

Hands on

Get a CSS file from a previous project and try to convert it (or part of it) to Sass using sassmeister. Remember to include custom properties, nesting selectors, sass variables.

Can you spot difference between custom properties and sass variables?