

Introduction to Sass



What is Sass?

- Sass stands for Systematically Awesome Stylesheets
- Sass is a preprocessor / CSS extension
- Sass adds many frequently used and necessary CSS tools such as variables, nested rules and imports.
- Since Sass is compiled to CSS, it doesn't require any special browser support

Variables

- Used for storing information that you want to reuse throughout your stylesheet.
- Sass uses the **\$** symbol to make something a variable.

For example: **\$mainBlue: #4285f4;**

Nesting

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

Example:

```
.contact {  
  color: black;  
  &-info {  
    color: $mainBlue;  
  }  
  &:hover {  
    color: yellow;  
  }  
}
```

Mixins

A mixin is a chunk of reusable CSS code that we can inject into any element that we want. A mixin, just like a function in JS, can have variables with default values.

Example:

```
@mixin blockBtn($bgColor: gray, $color: white) {  
    display: inline-block;  
    border: 1px solid $mainBlue;  
    padding: 16px 32px;  
    background-color: $bgColor;  
    color: $color;  
}
```

Partials

A partial is simply a Sass file named with a leading underscore.

The underscore lets Sass know that the file is only a partial file and that it should not be generated into a CSS file.

Sass partials are used with the `@import` directive.

When using `@import`, full file name is not required. Example: `@import 'vars'`

Built-in Functions

Sass has some useful built-in functions. Also, declaring your own functions is possible.

Example: `darken($color, [$darken-amount])`,

`rgba($color, $alpha-value)`

[List of Sass functions](#)

Loops & Interpolation

Interpolation: You can use Sass variables in selectors and property names using `#{} interpolation syntax`

Loops: Writing similar classes with minor variations, like utility classes, can be a pain to write and update, this is where `@for` loops become handy:

```
@for $i from 1 through 10{  
  .col-#{ $i } {  
    background-color: black;  
    border: 1px solid white;  
    height: 100px;  
    width: 100% / $i;  
  }  
}
```