F R O N T   E N D

S.A.S.S

LEONARDO MENDIETA

HERNAN ARISMENDI

S.A.S.S

Sass is a very powerful and popular CSS preprocessor that adds advanced features to standard CSS.

INSTALLATION

GLOBAL:

```
npm install -g sass
```

LOCAL:

```
npm install --save-dev node-sass
```

COMPILE:

```
sass file.sass file.css
```

COMPILE WITH NPM:

```
node-sass file.sass file.css
```

COMPILE WATCHING CHANGES:

```
node-sass --watch file.sass:file.css
```

COMPILE CHANGES WITH SCRIPT:

```
"scripts": {"compile-sass": "node-sass file.sass
file.css" }
```

Advantages of using S.A.S.S

CSS Compatible
Sass is completely compatible with all versions of CSS. We take this compatibility seriously, so that you can seamlessly use any available CSS libraries.

Feature Rich
Sass boasts more features and abilities than any other CSS extension language out there. The Sass Core Team has worked endlessly to not only keep up, but stay ahead.

Mature
Sass has been actively supported for about 16 years by its loving Core Team.

Industry Approved
Over and over again, the industry is choosing Sass as the premier CSS extension language.

Large Community
Sass is actively supported and developed by a consortium of several tech companies and hundreds of developers.

Frameworks
There are an endless number of frameworks built with Sass. Bootstrap, Bourbon, and Susy just to name a few.

Some disavantages

Additional complexity
Sass requires upfront configuration and a specific tool to use, and may require additional processing in your workflow.

Learning curve
Learning to use the features of Sass can take some getting used to.

Performance
The Sass build process can be slower than other preprocessors.

Syntax

Sass Syntax

This syntax is a bit different from the standard CSS syntax. It doesn't differ much. For example, it prevents you from placing semicolons at the end of property values. Also, the keys are not used and are indented instead.

SCSS syntax

It is a syntax quite similar to the syntax of CSS itself. In fact, the CSS code is valid SCSS code. We could say that SCSS is CSS code with some extra things.

Variables

In Sass, variables are declared with the symbol "$"

```
$primary-color: blue;
$font-size: 16px;

p {
 color: $primary-color;
 font-size: $font-size;
}
```

Nested Rules

Sass allows you to nest selectors inside others to reflect the hierarchy of elements in HTML

```
nav {
    ul {
       list-style: none;
       li {
         display: inline-block;
        }
    }
}
```

Mixins

Mixins are sets of CSS declarations that can be reused across multiple selectors.

```
@mixin rounded-corners {
     border-radius: 10px;
}

.button {
    @include rounded-corners;
     background-color: blue;
}
```

Functions and operations

Sass provides a number of functions and mathematical operations that can be used in style sheets.

```
$base-font-size: 16px;
p {
   font-size: floor($base-font-size * 1.2);
   line-height: 1.5;
}
```

Imports

Sass allows you to import other Sass files into a main file.

```
@import "variables";
@import "mixins";

body {
    font-size: $base-font-size;
    @include rounded-corners;
}
```

Interpolation

Sass allows interpolation of variables within selectors and attributes.

```
$my-class: border;
.#{$my-class} {
    border: 1px solid blue;
}
```

Conditional Expresions

Sass allows you to use if/else and loop conditional expresions to control styles.

```
$button-type: large;

.btn{
    @if $button-type == large {
        padding: 20px;
        font-size: 18px;
    } @else {
        padding: 10px;
        font-size: 14px;
    }
}
```

Thank you