



CSS pre-processors, what are they?

Introduction to CSS pre-processors

CSS pre-processors takes code written in the pre-processed language and then converts that code into plain CSS.

Types of CSS pre-processors:

The logo for Sass, featuring the word "Sass" in a pink, cursive script font.The logo for LESS, featuring the word "less" in a blue, bold, sans-serif font with curly braces on either side.The logo for Stylus, featuring the word "stylus" in a black, cursive script font.

- **SASS** - SASS uses the `.sass` file extension. SASS also has two syntaxes `.sass` and `.scss`
- **LESS** - LESS uses the syntax with the `.less` file extension
- **STYLUS** - STYLUS uses the `.styl` file extension.

Benefits of a pre-processor

As CSS can be fun, stylesheets are getting larger, more complex, and harder to maintain. This is where a pre-processor can help.

They let you use features that doesn't exist in native CSS yet.





VARIABLES

Example of a variable used in SCSS

To declare a variable you use the dollar symbol.

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

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

SCSS

```
body {
  font: 100% Helvetica, sans-serif;
  color: #333;
}
```

CSS



NESTING



Example of a nesting used in SCSS

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  
  li { display: inline-block; }  
  
  a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
  }  
}
```

SCSS

```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
nav li {  
  display: inline-block;  
}  
nav a {  
  display: block;  
  padding: 6px 12px;  
  text-decoration: none;  
}
```

CSS



MIXINS



Example of a mix-in used in SCSS

Mixins are function that you can use to use your code many times without having to repeat your self.

```
@mixin transform($property) {  
  -webkit-transform: $property;  
  -ms-transform: $property;  
  transform: $property;  
}  
.box { @include transform(rotate(30deg)); }
```

SCSS

```
.box {  
  -webkit-transform: rotate(30deg);  
  -ms-transform: rotate(30deg);  
  transform: rotate(30deg);  
}
```

CSS



EXTEND/INHERITANCE



Example of a Extend used in SCSS

Extend allows us to take all the attributes from a specific CSS class and use them in another CSS class. Extend are similar to mixins , but the core main difference is that when you are using extend you can extend an actual CSS class. Meaning you can extend the functionality of a specific CSS class attributes.

```
.foo {  
  color: black;  
  border: 1px solid black;  
}  
  
.bar {  
  @extend .foo;  
  background-color: red;  
}
```

SCSS

```
.foo, .bar {  
  color: black;  
  border: 1px solid black;  
}  
  
.bar {  
  background-color: red;  
}
```

CSS

What are the differences?

In fact, all CSS pre-processors have similar functionalities such as variables, mixins, extending, and nesting and more.

SCSS SYNTAX

```
1 $font-color: #fff;
2 $bg-color: #00f;
3
4 #box{
5   color: $font-color;
6   background: $bg-color;
7 }
```

Sass SYNTAX

```
1 $font-color: #fff
2 $bg-color: #00f
3
4 #box
5   color: $font-color
6   background: $bg-color
```



Syntax Declaration: LESS

Uses .less extension. Syntax of LESS is quite similar to SCSS with the exception that for declaring variables, instead of \$ sign, LESS uses @.

```
1 @font-color: #fff;
2 @bg-color: #00f;
3
4 #box{
5     color: @font-color;
6     background: @bg-color;
7 }
```

Stylus

```
1 /* STYLUS SYNTAX WRITTEN LIKE NATIVE CSS */
2 font-color = #fff;
3 bg-color = #00f;
4
5 #box {
6     color: font-color;
7     background: bg-color;
8 }
9
```



How do we compile our pre-processor file to a CSS file?



The end

Any questions?

Resources

Relevant links and articles:

<https://www.lambdatest.com/blog/css-preprocessors-sass-vs-less-vs-stylus-with-examples/>
<https://sass-lang.com/guide>
<https://raygun.com/blog/css-preprocessors-examples/>
<http://lesscss.org/usage/>
<https://sass-lang.com/install>
<https://www.npmjs.com/package/stylus>
<https://webpack.js.org/loaders/sass-loader/>
<https://www.youtube.com/watch?v=YD91G8DdUsw>
https://www.youtube.com/watch?v=lziuNMk_8eQ
<https://www.youtube.com/watch?v=jftHT-INOno>
<https://github.com/webpack-contrib/sass-loader>

Compilers to use for CSS files:

<https://codekitapp.com/>
<https://prepros.io/>