

Comparison Between LESS and SASS

Sep 22, 2013

For the CSS developers out there that have been hearing the buzz words of '[LESS](#)' and '[SASS](#)' and not knowing what they are, here's an article for you. After researching these languages, I've found that they are kind of like Javascript files that run on your computer and create CSS files for you. And the reason why you would use them is so that you don't have to do the repetitious tasks that CSS requires.

Both languages are pretty awesome and they both have pros and cons. That's why I wanted to write this article:

1. to compare the pros and cons of LESS and SASS
2. to write a cheat sheet for both of their syntax

With that goal, enjoy this resource:

	LESS	SASS
CSS Converting	Link page to your .less file and less javascript file or convert .less file into a .css file with a converter like CodeKit or SimpLESS .	Convert .scss or .sass file into file using Ruby on Rails or a converter like CodeKit .
Variable	<pre>@blue: #00214D; span.blue{ color: @blue; }</pre>	<pre>\$blue: #00214D; span.blue{ color: \$blue; }</pre>
Mixin	<pre>.small-serif{ font-size:11px; line-height:15px; font-family:Georgia, serif; } p.small-serif{ .small-serif;</pre>	<pre>@mixin small-serif{ font-size:11px; line-height:15px; font-family:Georgia, serif; } p.small-serif{ @include small-serif;</pre>

	<pre> }</pre>	<pre> }</pre>
Mixins with Parameters	<pre> .border-radius(@radius:5px){ -webkit-border-radius:@radius; -moz-border-radius:@radius; border-radius:@radius; } div.sqground{ .border-radius; } div.circle{ .border-radius(10px); }</pre>	<pre> @mixin border-radius(\$radius){ -webkit-border-radius:\$radius; -moz-border-radius:\$radius; border-radius:\$radius; } div.sqground{ @include border-radius; } div.circle{ @include border-radius(10px); }</pre>
Nested Rules	<pre> a{ text-decoration:underline; span{ color:#000; } &:hover{ color:#c00; } }</pre>	<pre> a{ text-decoration:underline; span{ color:#000; } &:hover{ color:#c00; } }</pre>
Operators	<pre> @color:#f00; @padding:10px; @border:1px; p.highlight{ color:@color + #00f; border:(@border+5)/2 solid #000; padding:@padding @padding +10; }</pre>	<pre> \$power: on; \$color: if(\$power=='on',#c00,#f00); \$padding:10px; \$border:1px; p.highlight{ color:\$color + #00f; border:(\$border+5)/2 solid #000; padding:\$padding \$padding +10; }</pre>
	<pre> .alert(error){ //attr specific color:#c00; } .alert(success){ //attr specific color:#069; } .alert(other, #0f0){ color:@color; } .alert(@_){ padding:10px; }</pre>	

Pattern Matching	<pre> border:1px solid #000; } .alert(@_,@_){ padding:10px; border:1px solid #000; } @type:error; p.message{ .alert(@type); } p.other-message{ .alert(other, #000); } </pre>	N/A
Looping	<pre> p.text(@color) when (darkness(@color) =< 50%){ color:lighten(@color, 20%); } p.text(@_){ color:#000; } .loop(@index) when (@index > 0){ .class{ z-index:@index; } //call itself .loopCount(@index - 1); } //stop loop .loopCount(0){} //output stuff .class(10); </pre>	<pre> \$boolean: true; =conditional-mixin @if \$boolean display:block; @else display:none; .img{ +conditional-mixin } @for \$i from 1 through 5{ div.col{ width:10px * \$i; } } </pre>
Functions	<pre> p.footnote{ lighten(@color, 50%); } </pre>	<pre> p.footnote{ lighten(@color, 50%); } </pre>

Hope this resource helps you as much as it does me!