

Sass

Responsive Web Design

CSS pre-processors: SASS

CSS pre-processors...



- ❑ **CSS pre-processors:** helps you write maintainable, future-proof code and it will seriously reduce the amount of CSS you have to write. Where these tools shine best are in large-scale user interfaces that require huge stylesheets and many style rules.
- ❑ CSS preprocessor is based on CSS and compiles to CSS.
- ❑ Its needed to keep CSS organized, and make it easy to maintain and quicker to build
- ❑ It gives extra functionality over CSS e.g. variables, functions, mixins..etc.
- ❑ Most popular CSS Preprocessor:
 - Sass (Syntactically Awesome Style Sheets):
 - <http://sass-lang.com>
 - Less
 - <http://lesscss.org>
- ❑ Bootstrap supports both LESS and Sass.



SASS vs. Less



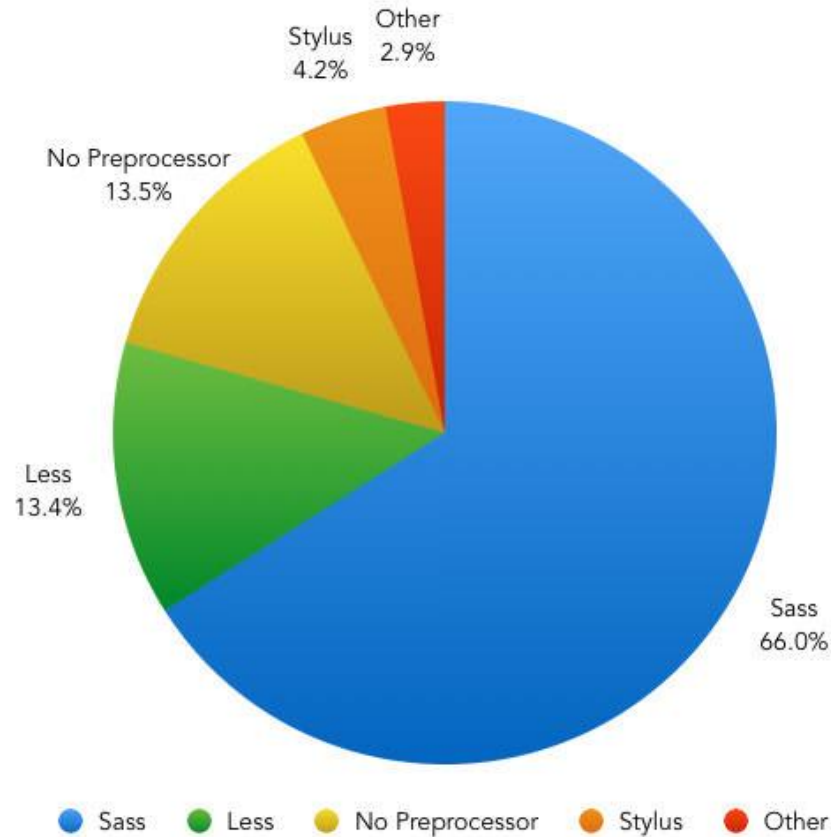
{ Comparison }

	LESS (@)	SASS (\$)
Variables	<pre>@plainRed: #ff0000; @softBlue: #bce7f3;</pre>	<pre>\$plainRed: #ff0000; \$softBlue: #bce7f3;</pre>
Mixins	<pre>.fluidBox{ width: 50%; box-sizing: border-box; }</pre>	<pre>@mixin fluidBox{ width: 50%; box-sizing: border-box; }</pre>
Parametric Mixins	<pre>.rounded(@radius: 5px){ -webkit-border-radius: @radius; -moz-border-radius: @radius; border-radius: @radius; }</pre>	<pre>@mixin rounded(\$radius: 5px){ -webkit-border-radius: \$radius; -moz-border-radius: \$radius; border-radius: \$radius; }</pre>
Functions	<pre>lighten(#ff0000, 10%); Saturate(#ff0000, 20%);</pre>	<pre>lighten(#ff0000, 10%); Saturate(#ff0000, 20%);</pre>
Operators	<pre>#header{ width: (@headerW - 50) * 2; }</pre>	<pre>#header{ width: (\$headerW - 50) * 2; }</pre>
Frameworks	LESSHat, LESS ELEMENTS	COMPASS
Language Base	Javascript (originally Ruby)	Ruby

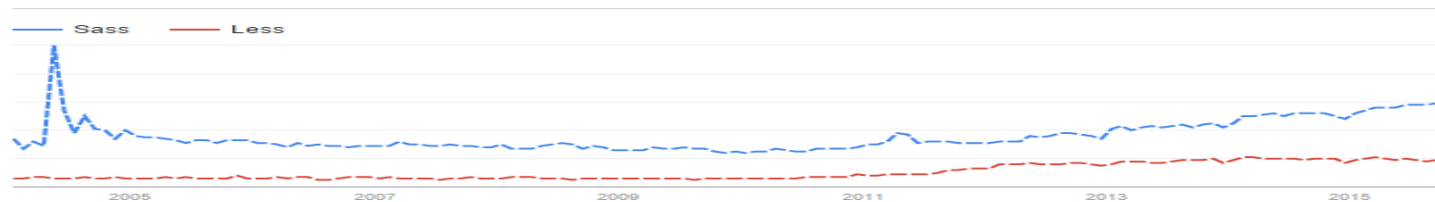
❑ SASS Vs. Less in detail:

- ❑ <https://www.keycdn.com/blog/sass-vs-less/>
- ❑ <https://css-tricks.com/sass-vs-less/>
- ❑ https://www.slant.co/versus/763/764/~less_vs_sass

SASS vs. Less (Cont.)



Interest over time. Web Search. Worldwide, 2004 - present.



What's SASS?



❑ What is SASS?

- SASS (Syntactically Awesome Stylesheet) is a CSS pre-processor, which helps to reduce repetition with CSS and saves time. It is more stable and powerful CSS extension language that describes the style of a document cleanly and structurally.

❑ History:

- It was initially designed by **Hampton Catlin** and developed by **Natalie Weizenbaum in 2006**. Later, Weizenbaum and Chris Eppstein used its initial version to extend the Sass with SassScript.

What's SASS? (Cont.)



❑ Features and advantages of SASS:

- It is more stable, powerful, and compatible with versions of CSS.
- It is a super set of CSS and is based on JavaScript.
- It uses its own syntax and compiles to readable CSS.
- You can easily write CSS in less code within less time.
- It is an open source pre-processor, which is interpreted into CSS.
- It allows writing clean CSS in a programming construct.
- It helps in writing CSS quickly.
- It is possible to use nested syntax and useful functions such as color manipulation, mathematics and other values.

❑ Disadvantages of SASS:

- It takes time to learn new features present in this pre-processor.
- If many people are working on the same site, then should use the same preprocessor.
- There are chances of losing benefits of browser's built-in element inspector.

Sass - Installation



❑ Installing SASS:

- Since SASS is built on Ruby, you need to install Ruby:
<https://rubyinstaller.org>
- Install SASS using Ruby Command line:

```
gem install sass
```

- To check if SASS was installed successfully, in your command line run the following command (It should return the version of SASS):

```
sass -v
```

Congratulations! You've successfully installed Sass.

Sass – Installation (Cont.)



❑ Different Options to install and compile SASS:

- The original **Ruby Sass binary**. Install it with `gem install sass`.
- To compile SASS files to CSS files, you can use the following command:

The most direct way to make this happen is in your terminal. Once Sass is installed, you can run `sass input.scss output.css` from your terminal. You can watch either individual files or entire directories. In addition, you can watch folders or directories with the `--watch` flag. An example of running Sass while watching an entire directory is the following:

```
sass --watch style.scss
```

- A GUI app such as **Hammer, CodeKit, or Compass**
- **libsass**, which is a blazing fast Sass compiler written in C. You can also install libsass via NPM with `node-sass` (`npm install node-sass`).
- The simplest solution is a browser tool for writing and compiling Sass right on the spot - **SassMeister** (<https://www.sassmeister.com>).
- Another online converter: <http://sass.js.org>

Sass – Getting Started



❑ SASS supports two syntaxes namely SCSS and Indented syntax:

- The **SCSS** (Sassy CSS) is **an extension of CSS syntax**. This means every valid CSS is a valid SCSS as well. SCSS makes much easier to maintain large stylesheets and can recognize vendor specific syntax, Many CSS and SCSS files use the extension **.scss**.
- **Indented** – This is **older syntax** and sometimes just called as SASS. Using this form of syntax, CSS can be written concisely. SASS files use the extension **.sass**.

“Sass has two syntaxes. The most commonly used syntax is known as “SCSS” (for “Sassy CSS”), and is a superset of CSS3’s syntax. This means that every valid CSS3 stylesheet is valid SCSS as well. SCSS files use the extension .scss.

The second, older syntax is known as the indented syntax (or just “.sass”). Inspired by Haml’s terseness, it’s intended for people who prefer conciseness over similarity to CSS. Instead of brackets and semicolons, it uses the indentation of lines to specify blocks. Files in the indented syntax use the extension .sass.”

Sass – Syntax



- ❑ <http://sass-lang.com/guide>
- ❑ <https://scotch.io/tutorials/getting-started-with-sass>
- ❑ <https://tutorialzine.com/2016/01/learn-sass-in-15-minutes>