WIKIPEDIA

Sass (stylesheet language)

Sass (short for ***syntactically awesome style sheets***) is a preprocessor scripting language that is interpreted or compiled into Cascading Style Sheets (CSS). SassScript is the scripting language itself.

Sass consists of two syntaxes. The original syntax, called "the indented syntax," uses a syntax similar to Haml.[2] It uses indentation to separate code blocks and newline characters to separate rules. The newer syntax, "SCSS" (Sassy CSS), uses block formatting like that of CSS. It uses braces to denote code blocks and semicolons to separate rules within a block. The indented syntax and SCSS files are traditionally given the extensions .sass and .scss, respectively.

CSS3 consists of a series of selectors and pseudo-selectors that group rules that apply to them. Sass (in the larger context of both syntaxes) extends CSS by providing several mechanisms available in more traditional programming languages, particularly object-oriented languages, but that are not available to CSS3 itself. When SassScript is interpreted, it creates blocks of CSS rules for various selectors as defined by the Sass file. The Sass interpreter translates SassScript into CSS. Alternatively, Sass can monitor the .sass or .scss file and translate it to an output .css file whenever the .sass or .scss file is saved.[3]

The indented syntax is a metalanguage. SCSS is a nested metalanguage, as valid CSS is valid SCSS with the same semantics.

SassScript provides the following mechanisms: variables, nesting, mixins, and selector inheritance.[2]

Sass

Designed by	Hampton Catlin
Developer	Natalie Weizenbaum, Chris Eppstein
First appeared	November 28, 2006
Stable release	3.5.6 / March 23, 2018[1]
Typing discipline	Dynamic
OS	Cross-platform
License	MIT License
Filename extensions	.sass, .scss
Website	sass-lang.com (https://sass-lang.com/)
Major implementations	
Dart, Ruby	
Influenced by	
CSS (both "indented" and SCSS) YAML and Haml (indented syntax) Less (SCSS)	
Influenced	
Less, Stylus, Tritium, Bootstrap (v4+)	

Contents

History
Major implementations
Features
Variables
Nesting
Loops
Arguments
In combination
Selector inheritance
libSass
IDE integration
See also

References
External links

History

Sass was initially designed by Hampton Catlin and developed by Natalie Weizenbaum.[4][5] After its initial versions, Weizenbaum and Chris Eppstein have continued to extend Sass with SassScript, a scripting language used in Sass files.

Major implementations

SassScript was implemented in multiple languages, the noteworthy implementations are:

- The original open-source Ruby implementation created in 2006,[6] since deprecated due to the lack of maintainers and reached End-of-Life in March 2019.[7][8]
- The official open-source Dart implementation.[6]
- libSass, the official open-source C++ implementation.
- the official JavaScript implementation, published as "sass" module on npm.
- JSass, an unofficial Java implementation.[9]
- phamlp, an unofficial SASS/SCSS implementation in PHP.[6]
- Vaadin has a Java implementation of Sass.[10]
- Firebug, a Firefox XUL ("legacy") extension for web development.[11] It has been since deprecated in favor of developer tools integrated into Firefox itself. It stopped working since Firefox 57 dropped support for XUL extensions.

Features

Variables

Sass allows variables to be defined. Variables begin with a dollar sign ($). Variable assignment is done with a colon (`:`).[11]

SassScript supports four data types:[11]

- Numbers (including units)
- Strings (with quotes or without)
- Colors (name, or names)
- Booleans

Variables can be arguments to or results from one of several available functions.[12] During translation, the values of the variables are inserted into the output CSS document.[2]

SCSS	Sass	Compiled CSS

$primary-color: #3bbfce;
$margin: 16px;
 	
.content-navigation {
  border-color: $primary-color;
  color: darken($primary-color, 10%);
}
 	
.border {
  padding: $margin / 2;
  margin: $margin / 2;
  border-color: $primary-color;
}
	$primary-color: #3bbfce
	$margin: 16px
	
	.content-navigation
	  border-color: $primary-color
	  color: darken($primary-color, 10%)
	
	.border
	  padding: $margin/2
	  margin: $margin/2
	  border-color: $primary-color
		.content-navigation {
		  border-color: #3bbfce;
		  color: #2b9eab;
		}
		
		.border {
		  padding: 8px;
		  margin: 8px;
		  border-color: #3bbfce;
		}

Nesting

CSS does support logical nesting, but the code blocks themselves are not nested. Sass allows the nested code to be inserted within each other.[2]

SCSS	Sass	Compiled CSS

table.hl {
  margin: 2em 0;
  td.ln {
    text-align: right;
  }
}
 	
li {
  font: {
    family: serif;
    weight: bold;
    size: 1.3em;
  }
}
	table.hl
	  margin: 2em 0
	  td.ln
	    text-align: right
	
	li
	  font:
	    family: serif
	    weight: bold
	    size: 1.3em
		table.hl {
		  margin: 2em 0;
		}
		table.hl td.ln {
		  text-align: right;
		}
		
		li {
		  font-family: serif;
		  font-weight: bold;
		  font-size: 1.3em;
		}

More complicated types of nesting including namespace nesting and parent references are discussed in the Sass documentation.[11]

SCSS	Sass	Compiled CSS

@mixin table-base {
  th {
    text-align: center;
    font-weight: bold;
  }
  td, th {
    padding: 2px;
  }
}
 	
#data {
  @include table-base;
}
	=table-base
	  th
	    text-align: center
	    font-weight: bold
	  td, th
	    padding: 2px
	
	#data
	  +table-base
		#data th {
		  text-align: center;
		  font-weight: bold;
		}
		#data td, #data th {
		  padding: 2px;
		}

Loops

Sass allows for iterating over variables using `@for`, `@each` and `@while`, which can be used to apply different styles to elements with similar classes or ids.

Sass	Compiled CSS

$squareCount: 4
@for $i from 1 through $squareCount
  #square-#{$i}
   background-color: red
   width: 50px * $i
   height: 120px / $i
	#square-1 {
	  background-color: red;
	  width: 50px;
	  height: 120px;
	}
	
	#square-2 {
	  background-color: red;
	  width: 100px;
	  height: 60px;
	}
	
	#square-3 {
	  background-color: red;
	  width: 150px;
	  height: 40px;
	}

Arguments

Mixins also support arguments.[2]

Sass	Compiled CSS

=left($dist)
  float: left
  margin-left: $dist
 	
#data
  +left(10px)
	#data {
	  float: left;
	  margin-left: 10px;
	}

In combination

Sass	Compiled CSS

=table-base
  th
    text-align: center
    font-weight: bold
  td, th
    padding: 2px
 	
=left($dist)
  float: left
  margin-left: $dist
 	
#data
  +left(10px)
  +table-base
	#data {
	  float: left;
	  margin-left: 10px;
	}
	#data th {
	  text-align: center;
	  font-weight: bold;
	}
	#data td, #data th {
	  padding: 2px;
	}

Selector inheritance

While CSS3 supports the Document Object Model (DOM) hierarchy, it does not allow selector inheritance. In Sass, inheritance is achieved by inserting a line inside of a code block that uses the @extend keyword and references another selector. The extended selector's attributes are applied to the calling selector.[2]

Sass	Compiled CSS

`.error`
`  border: 1px #f00`
`  background: #fdd`
 	
`.error.intrusion`
`  font-size: 1.3em`
`  font-weight: bold`
 	
`.badError`
`  @extend .error`
`  border-width: 3px`
	`.error, .badError {`
	`  border: 1px #f00;`
	`  background: #fdd;`
	`}`
	
	`.error.intrusion,`
	`.badError.intrusion {`
	`  font-size: 1.3em;`
	`  font-weight: bold;`
	`}`
	
	`.badError {`
	`  border-width: 3px;`
	`}`

Sass supports multiple inheritance.[11]

libSass

At the 2012 HTML5 Developer Conference, Hampton Catlin, the creator of Sass, announced version 1.0 of libSass, an open source C++ implementation of Sass developed by Catlin, Aaron Leung, and the engineering team at Moovweb.[13][14] Current Sass maintainer, Chris Eppstein, has expressed intent to contribute as well.[15]

According to Catlin, libSass can be "drop[ped] into anything and it will have Sass in it...You could drop it right into Firefox today and build Firefox and it will compile in there. We wrote our own parser from scratch to make sure that would be possible."[16]

The design goals of libSass are:

- Performance – Developers have reported 10x speed up increases over the Ruby implementation of Sass.[17]
- Easier integration – libSass makes it easier to integrate Sass into more software. Before libSass, tightly integrating Sass into a language or software product required bundling the entire Ruby interpreter. By contrast, libSass is a statically linkable library with zero external dependencies and C-like interface, making it easy to wrap Sass directly into other programming languages and tools. For example, open source libSass bindings now exist for Node, Go, and Ruby.[14]
- Compatibility – libSass's goal is full compatibility with the official Ruby implementation of Sass. This goal has been achieved on libsass 3.3.[18]

IDE integration

IDE integration of Sass

IDE	Software	website
Adobe Dreamweaver CC 2017		https://blogs.adobe.com/creativecloud/getting-started-with-css-preprocessors-less-and-sass/
Eclipse		
Emacs	SCSS Mode	https://github.com/antonj/scss-mode/
JetBrains IntelliJ IDEA (Ultimate Edition)		https://www.jetbrains.com/idea/
JetBrains PhpStorm		http://www.jetbrains.com/phpstorm/
JetBrains RubyMine		http://www.jetbrains.com/ruby/
JetBrains WebStorm		http://www.jetbrains.com/webstorm/
Microsoft Visual Studio	Mindscape	http://www.mindscapehq.com/products/web-workbench
Microsoft Visual Studio	SassyStudio	http://visualstudiogallery.msdn.microsoft.com/85fa99a6-e4c6-4a1c-9f00-e6a8129b6f4d
Microsoft WebMatrix		http://www.microsoft.com/web/
NetBeans		http://plugins.netbeans.org/plugin/34929/scss-support
Vim	haml.zip	http://www.vim.org/scripts/script.php?script_id=1433
Atom		https://github.com/atom/language-sass
Visual Studio Code		https://code.visualstudio.com/Docs/languages/css
Sublime		https://github.com/P233/Syntax-highlighting-for-Sass
Edit+		https://www.editplus.com/others.html

See also

- Less (stylesheet language)
- Stylus (stylesheet language)

References

1. Latest releases (https://github.com/sass/sass/releases)
2. Media Mark (3.2.12). "Sass - Syntactically Awesome Stylesheets" (http://sass-lang.com/). Sass-lang.com. Retrieved 2014-02-23.
3. Sass - Syntactically Awesome Stylesheets (http://sass-lang.com/tutorial.html) Tutorial
4. "Sass: Syntactically Awesome Style Sheets" (https://web.archive.org/web/20130901145805/http://sass-lang.com/about.html). *sass-lang.com*. Archived from the original (http://sass-lang.com/about.html) on 2013-09-01.
5. "Natalie Weizenbaum's blog" (https://web.archive.org/web/20071011121541/http://nex-3.com/). Archived from the original (http://nex-3.com/) on 2007-10-11.
6. "Sass / Scss" (https://drupal.org/project/sass). Drupal.org. 2009-10-21. Retrieved 2014-02-23.
7. Weizenbaum, Natalie. "Ruby Sass Has Reached End-Of-Life « Sass Blog" (http://sass.logdown.com/posts/7828841-ruby-sass-is-unsupported). *sass.logdown.com*. Retrieved 2019-04-21.
8. "Sass: Ruby Sass" (https://sass-lang.com/ruby-sass). *sass-lang.com*. Retrieved 2019-04-21.
9. "jsass - A Java implementation of the Sass compiler (and some other goodies). - Google Project Hosting" (https://code.google.com/p/jsass/). Retrieved 2014-02-23.

10. "SassCompiler (Vaadin 7.0.7 API)" (https://vaadin.com/api/7.0.7/com/vaadin/sass/SassCompiler.html). Vaadin.com. 2013-06-06. Retrieved 2014-02-23.
11. Sass (Syntactically Awesome StyleSheets) (http://sass-lang.com/docs/yardoc/file.SASS_REFERENCE.html) SASS_REFERENCE
12. Module: Sass::Script::Functions (http://sass-lang.com/docs/yardoc/Sass/Script/Functions.html) Sass Functions
13. H. Catlin (2012-10-15). "Hampton's 6 Rules of Mobile Design" (https://www.youtube.com/watch?v=j75-SslllvY&feature=player_detailpage#t=1831s). HTML5 Developer Conference. Retrieved 2013-07-11.
14. M. Catlin (2012-04-30). "libsass" (https://web.archive.org/web/20130508080521/http://blog.moovweb.com/2012/04/libsass/). Moovweb Blog. Archived from the original (http://blog.moovweb.com/2012/04/libsass/) on 2013-05-08. Retrieved 2013-07-11.
15. C. Eppstein (2012-04-15). "Tweet" (https://twitter.com/chriseppstein/status/323845132417900544). Retrieved 2013-07-11.
16. A. Stacoviak & A. Thorp (2013-06-26). "Sass, libsass, Haml and more with Hampton Catlin" (http://5by5.tv/changelog/94). Retrieved 2013-07-30.
17. D. Le Nouaille (2013-06-07). "Sassc and Bourbon" (http://www.damln.com/log/sassc-and-bourbon-it-works/). Retrieved 2013-07-11.
18. "Sass Compatibility" (http://sass-compatibility.github.io/). *sass-compatibility.github.io*. Retrieved 2019-11-29.

External links

- Official website (https://sass-lang.com/)
- pyScss, a Python Scss library and client (https://github.com/Kronuz/pyScss)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Sass_(stylesheet_language)&oldid=995389963"

This page was last edited on 20 December 2020, at 19:47 (UTC).