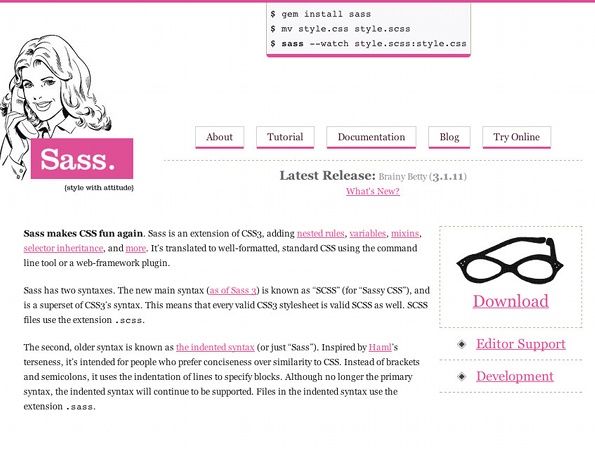
**Sass Css:-**



Sass is a meta-language on top of CSS that’s used to describe the style of a document cleanly and structurally, with more power than flat CSS allows. Sass both provides a simpler, more elegant syntax for CSS and implements various features that are useful for creating manageable stylesheets.

Sass makes CSS fun again. Sass is an extension of CSS3, adding nested rules, variables, mixins, selector inheritance, and more. It’s translated to well-formatted, standard CSS using the command line tool or a web-framework plugin.

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.

While the documentation on this site is mostly in the SCSS syntax, both syntaxes are fully supported – there is no functional difference between them. Use the syntax you prefer.

Your First Sass Stylesheet

We’ll start out by creating a very simple SCSS file. Since SCSS is an extension of CSS3, our first file will start out as plain CSS. Open up a new file called style.

/\* style.scss \*/

#navbar {

width: 80%;

height: 23px;

}

Install scoutApp:--the sass/compass Gui <http://mhs.github.com/scout-app/>

For Reading:-

Link:- <http://sass-lang.com>

**Sass cheet list:-**

**Variables:**

$red: #ff0000;

.myClass {

color: $red;

**!default:** If a variable isn't given a value,

it uses the one set for default

$color: #000000;

$color: #ff0000 !default;

.myClass {

color: $color;

}

**Mixin:**

@mixin myfont ($size, $weight, $family){

font: {

size: $size;

weight: $weight;

family: $family;

}

}

.myClass {

@include font(18px, "bold", "Arial,Helvetica,sans-serif");

}

**Nesting:**

nav ul {

margin: 0;

padding: 0;

li {

list-style: none;

display: inline;

a {

color: $red;

}

}}

**Selector Inheritance (Using @extend):**

.myClass {

color: #ff0000;

margin: 10px;

}

.anotherClass {

@extend .myClass;

border: 1px #000 solid;

}

**'&':** Referencing Parent Selectors

a {

color: #ff0000;

&:hover {

color: #666666;

}

}

**Operations:** +, -, \*, /, %

$width: 1000px;

.myClass {

width: $width - 400px;

}

**Interpolation:** #{}

$className: samcro;

$attr: background;

a.#{$className} { #{$attr}-color: #000000 }

**@if, @elseif, @else**

$type: monster;

p {

@if $type == ocean {

color: blue;

} @else if $type == matador {

color: red;

} @else if $type == monster {

color: green;

} @else {

color: black;

}

}

**@for** - (@while works similar)

@for $i from 1 through 3 {

.myClass-#{$i} { width: 2px \* $i; }

}

**Nested Properties**

.myFont {

font:{

family: Arial;

size: 14px;

weight: bold;

}

}

**Command Line Commands:**

compass compile

compass watch

compass help

compass validate

compass version

**Imports for Sass/SCSS**

(put in .scss file)

Cross-browser CSS3 mixins - includes all:

@import "compass/css3.scss";

Example of single case usage (In case you

are only using a few CSS3 properties):

@font-face mixin:

@import "compass/css3/font-face"

**Config File Properties You’ll Want To Know**

line\_comments = false

output\_style = :compact

(could be: :nested, :expanded, :compact, or

:compressed)

Note: You might have to turn off compass watch

in order for a change to the config to take place.

Go here for more:

http://compass-style.org/docs/tutorials/

configuration-reference/