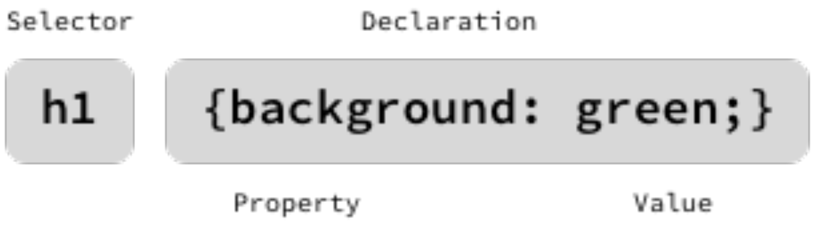
{WebCodeChicks}
</spartanburg>

Beginning Sass

sass-lang.com/guide

We will be using the website: codepen.io to write our code. A great resource to see how others write code. Sign up for a free account.



CSS OVERVIEW
A style sheet language used for describing the presentation of a document. The style definitions are normally saved in external .css files.

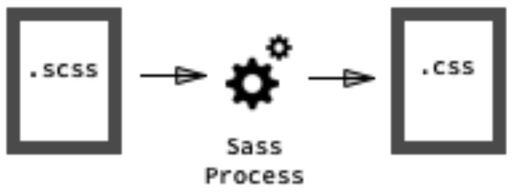
- The selector points to the HTML element you want to style.
- Each declaration includes a CSS property name and a value, separated by a colon.
- The declaration block contains one or more declarations separated by semicolons.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

```
p {
  color: #333;
  font-weight: bold;
}
```

REFERENCE
http://www.w3schools.com/cssref/

SASS
WHAT IS IT?

Syntactically **A**wesome **S**tyle **S**heets
a CSS preprocessor - a layer between the stylesheets you code and the .css files you serve to the browser.



SASS
THE "PRE" PROCESS

Browsers can't read SCSS, so an application needs to compile or convert SCSS into CSS - the styling language that the browser can read.

STAND ALONE APPS
Prepros - prepros.io - free/nag buy reminder $26
(Only stable app for Windows)
CodeKit - incident57.com/codekit - $32 Bower built in
LiveReload - livereload.com – $9.99 MacApp Store – NOT for
Use Chrome extension: LiveReload

CODE EDITORS:
Brackets, Atom and Sublime Text3 have plugins
NERD ALERT - You must go through the terminal installation - See page 7 for Sass installation instructions.

SASS
VARIABLES

Variables help us from repeating ourselves in our code. We set a value once and call upon it when we need it. It uses a name value pair and starts with a $:

```
$name: value;
```

It's important to come up with some good conventions for naming your variables so that they are easy to remember.

```
EXAMPLES:
// COLORS
$primary-color: teal;
$secondary-color: tan;

// TYPOGRAPHY
$body-font: ‘Open Sans’, Arial, sans-serif;
$header-font: ‘Roboto Slab’, serif;

// FONT WEIGHT
$thin-font-weight: 100;
$thick-font-weight: 700;
```

CENTRALIZED VARIABLES
Keep variables in a single file, called _vars.scss, that gets include in the primary stylesheet using the @import directive. This way, files that get included after the vars file will have access to those variables.
The “_” underscore before the name of the file lets the Sass preprocessor know to NOT compile the file to css - just import the data. Sass knows how to read the data and use it.

```
//INCLUDE VARIABLES - without the “_” - Yeah makes no sense!
@include “vars.scss”;
@include “other.scss”;
@include “another.scss”;
```

SASS
COLOR FUNCTIONS

RESOURCE: http://jackiebalzer.com/color
NERD ALERT EXAMPLE - http://codepen.io/yoksel/pen/yCHuJ
(click view compiled button)

So you have set a variable color:

```
$primary: red;
```

Then you want to change aspects of that color and let Sass do the math - YEAH!

```
section.secordary {
  background: lighten($primary, 30%);
}
```

Calling the darken function, passing the variable $primary-color and darkening the color by 30% and stuffing all of that in the background property will render:

```
section.secondary {
  background: #ff9999;
}
```

FUNCTION NAMES - SEE RESOURCE ABOUT FOR FULL LIST

```
darken($var, 1-100%);
lighten($var, 1-100%);
mix($var1, $var2, 1-100%);
rgba($var 0.1-1%); - transparency
```

MIXINS

A mixin is a small template of code that can be called upon at any time by including it by name. Parts of the mixin can be dynamic by passing through variable arguments. (Say Whaaat?! NERD ALERT!) This is COOL!

BASIC STATIC MIXIN

Declare a mixin by the @mixin prefix. Then give the mixin a name and add the styles.

```
@mixin article-title {
  font-family: $font-serif;
  font-size: 20px;
  font-weight: bold;
  text-transform: uppercase;
}
```

Use a selector and include it:

```
section.main h2 {
  @include article-title
}
```

If you want to add additional rules - go right ahead:

```
section.main h2 {
  @include article-title;
  color: orange;
}
```

DYNAMIC MIXIN

To add upon the previous static mixin and make it dynamic with passing through any color by setting a local variable. This variable is set to the mixin only and does not reference any other global variable you may have set any where else.

I usually set mixin variables by an abbreviation. In the case below, I would set the $color to $c as to not confuse myself thinking it is a variable somewhere else like the $font-serif already designated.

```
@mixin article-title($c) {
  font-family: $font-serif;
  font-size: 20px;
  font-weight: bold;
  text-transform: uppercase;
  color: $c;
}

section.main h2(orange) {
  @include article-title;
}
```

Yes you can use multiple variables. Be sure that they are in the correct order!

```
@mixin article-title($fs, $fw, $c) {
  font-family: $font-serif;
  font-size: $fs;
  font-weight: $fw;
  text-transform: uppercase;
  color: $c;
}
```

SASS
TERMINAL INSTALLATION

FOR WINDOWS (EXTRA WORK)
You must install Ruby:
http://rubyinstaller.org/ - Download and install

MAC - YOU ARE COVERED - RUBY ALREADY INSTALLED

INSTALLATION OF SASS

Windows:
Use Command Prompt and type:

```
gem install sass
```

Mac:
Use Terminal and type:

```
sudo gem install sass
```

To see installation version type:
```
sass -v
```

You should get something like:

```
Sass 3.4.21 (Selective Steve)
```

TERMINAL WATCH
For both Mac and Windows: Change into the root of the project folder:

sass --watch input:output (two dashes before watch)

EXAMPLE:

```
sass –-watch scss/style.scss:css/style.css
```