Variables

These are pretty self-explanatory:

@nice-blue: #5B83AD;

@light-blue: @nice-blue + #111;

#header {

color: @light-blue;

}

[**arn More About Variables**](http://lesscss.org/features/#variables-feature)

# Mixins

Mixins are a way of including ("mixing in") a bunch of properties from one rule-set into another rule-set. So say we have the following class:

.bordered {

border-top: dotted 1px black;

border-bottom: solid 2px black;

}

And we want to use these properties inside other rule-sets. Well, we just have to drop in the name of the class where we want the properties, like so:

#menu a {

color: #111;

.bordered;

}

.post a {

color: red;

.bordered;

}

# Nesting

Less gives you the ability to use nesting instead of, or in combination with cascading. Let's say we have the following CSS:

#header {

color: black;

}

#header .navigation {

font-size: 12px;

}

#header .logo {

width: 300px;

}

In Less, we can also write it this way:

#header {

color: black;

.navigation {

font-size: 12px;

}

.logo {

width: 300px;

}

}

## Nested At-Rules and Bubbling

At-rules such as @media or @supports can be nested in the same way as selectors. The at-rule is placed on top and relative order against other elements inside the same ruleset remains unchanged. This is called bubbling.

.component {

width: 300px;

@media (min-width: 768px) {

width: 600px;

@media (min-resolution: 192dpi) {

background-image: url(/img/retina2x.png);

}

}

@media (min-width: 1280px) {

width: 800px;

}

}

}

}

# Operations

Arithmetical operations +, -, \*, / can operate on any number, color or variable. If it is possible, mathematical operations take units into account and convert numbers before adding, subtracting or comparing them. The result has leftmost explicitly stated unit type. If the conversion is impossible or not meaningful, units are ignored. Example of impossible conversion: px to cm or rad to %.

*// numbers are converted into the same units*

@conversion-1: 5cm + 10mm; *// result is 6cm*

@conversion-2: 2 - 3cm - 5mm; *// result is -1.5cm*

*// conversion is impossible*

@incompatible-units: 2 + 5px - 3cm; *// result is 4px*

*// example with variables*

@base: 5%;

@filler: @base \* 2; *// result is 10%*

@other: @base + @filler; *// result is 15%*

Multiplication and division do not convert numbers. It would not be meaningful in most cases - a length multiplied by a length gives an area and css does not support specifying areas. Less will operate on numbers as they are and assign explicitly stated unit type to the result.

@base: 2cm \* 3mm; *// result is 6cm*

You can also do arithemtic on colors:

@color: #224488 / 2; *//results in #112244*

background-color: #112244 + #111; *// result is #223355*

However, you may find Less's [Color Functions](http://lesscss.org/functions/#color-operations) more useful.

# Functions

Less provides a variety of functions which transform colors, manipulate strings and do maths. They are documented fully in the [function reference](http://lesscss.org/functions/).

Using them is pretty straightforward. The following example uses percentage to convert 0.5 to 50%, increases the saturation of a base color by 5% and then sets the background color to one that is lightened by 25% and spun by 8 degrees:

@base: #f04615;

@width: 0.5;

.class {

width: percentage(@width); *// returns `50%`*

color: saturate(@base, 5%);

background-color: spin(lighten(@base, 25%), 8);

}

# Scope

Scope in Less is very similar to that of programming languages. Variables and mixins are first looked for locally, and if they aren't found, the compiler will look in the parent scope, and so on.

@var: red;

#page {

@var: white;

#header {

color: @var; *// white*

}

}

Mixin and variable definitions do not have to be placed before a line where they are referenced. So the following Less code is identical to the previous example:

@var: red;

#page {

#header {

color: @var; *// white*

}

@var: white;

}

<http://lesscss.org/features/#variables-feature-lazy-loading>

<https://tutorialzine.com/2015/07/learn-less-in-10-minutes-or-less>