LESS Extensions - http://lesscss.org/#

**What is LESS?**

LESS is a [JavaScript library](http://www.hongkiat.com/blog/javascripts-resources-20-scripts-tutorials-and-frameworks/). With similar syntax to CSS, LESS is very convenient when one tries to learn it because in LESS all CSS code is valid but not vice versa. This is great because your existing CSS is already valid LESS code—a very small learning curve!

**Why Use LESS?**

There are no great advantages except for the fact you could rapidly create / edit / push a bunch of LESS files to your server and see the changes before doing a compile. You don’t have to compile whenever a small program change is made.

**How to Use LESS?**

Here are a few LESS extensions that you might need to include in your project:

* **CSS3 Mixins**: [LESS Elements](http://lesselements.com/), [Preboot](http://markdotto.com/bootstrap/), [LESS Mixins](https://github.com/tophermade/LESS-Mixins).
* **Grid**: [960.gs](https://github.com/adrianlee44/960gs.less), [Frameless](https://github.com/jonikorpi/Frameless), [Semantic.gs](http://semantic.gs/)
* **Layout**: [Even.less](https://github.com/bisrael/even.less)
* **Misc**: [Twitter Bootstrap](http://twitter.github.com/bootstrap/)

There are Various (pro) mechanism of LESS:

* **Variables**. Set a constant variable named @green, add your favorite hex green color and use it in your stylesheet, that you can access throughout your program. No “Find and replace” when you can just change one line!
* **Mixins** allow you to embed all the properties of a class into another class by simply including the class name as one of its properties. It’s just like variables, but for whole classes.
  + **Example:**
  + LESS has an ability to do "guarded mixins." These are mixins that only take affect when a certain condition is true. Perhaps you want to set a background color based on the current text color in a module. If the text color is "pretty light" you'll probably want a dark background. If it's "pretty dark" you'll want a light background. So you have a single mixin broke into two parts with these guards that ensure that only one of them takes effect.
  + .set-bg-color (@text-color) when (lightness(@text-color) >= 50%) { background: black; } .set-bg-color (@text-color) when (lightness(@text-color) < 50%) { background: #ccc; }
  + So then when you use it, you'll get the correct background:
* **Nested  rules**. You can simply nest selectors inside other selectors. These can save you a lot of coding, and they make your code clearer.
  + .g {
    - &.posted { }

}

The & character has the function of a this keyword

**1. Cleaner Structure With Nesting**[**Link**](https://www.smashingmagazine.com/2010/12/using-the-less-css-preprocessor-for-smarter-style-sheets/#1-cleaner-structure-with-nesting)

In CSS, we write out every rule set separately. Here’s a typical example:

<code>#header {} #header #nav {} #header #nav ul {} #header #nav ul li {} #header #nav ul li a {}</code>

LESS allows us to nest rule sets inside other rule sets, as a way to show hierarchy. Let’s rewrite the above example with nesting:

# header { #nav { ul { li { a {} } } } }

keep code organized because it groups related items together visually.

* **Namespaces for grouping variables**. This is very useful if you want to avoid naming clashes
* **Operators and functions**, which let you create CSS properties mathematically.

LESS can also do self-referencing recursion where a mixin can call itself with an updated value creating a loop.

LESS give you programmatic constructs which help you in making your code modular and reusable.

Pros:

1. You can reuse many of the redundant parts of the code ex. you can write a function for giving rounded edges to elements, etc.

Create a stylesheet with .less extension and link it in your document using the rel="stylesheet/less" attribute.

|  |  |
| --- | --- |
| 1 | <link rel="stylesheet/less" type="text/css" href="main.less" /> |

**Final Thought**

Ultimately, we hope this post can give you a basic understanding on how we can write CSS in a better way using LESS. You maybe feel a little awkward at first, but as you try it more often, it will surely become a lot easier.