Less

Devpoint Labs

CSS

CSS is great for styling a website but after awhile of working with CSS you start to realize that there is a lot of code reuse and some basic missing programming principles. For example where are the variables? Couldn’t we reduce a lot of reusable code by passing variables around?

Couldn’t we change the theme of our website in one spot instead of all over the place?

Less

Less is a dynamic stylesheet language.

Less uses JavaScript to enhance its purpose.

Less is only used for development. At the end of the day it is compiled into CSS.

JavaScript is used to give functionality to a stylesheet beyond just setting properties.

Less gives you access to many programming principals inside of your stylesheet.

When is less compiled?

- On the fly by using less.js
- Vla the command line (Please compile my less to css thank you)
- Whenever a change is made by using a library to watch for changes.

For the purpose of this tutorial we will be manually compiling less in the command line. If your changes do not appear make sure to run the command to compile the less files before refreshing your page.

To install a less compiler...

npm install less -g

To compile run... lessc style.less > style.css
Replace style with the name of your stylesheet

index.html

```
<!doctype html>
<html>
<head>
  <link rel='stylesheet' href='style.css'>
</head>
<body>
  <header></header>
  <main>
    <article>
      <div class='big-button' id='button1'></div>
      <div class='big-button' id='button2'></div>
      <div class='big-button' id='button3'></div>
    </article>
    <aside>
      <h1>Hello World</h1>
    </aside>
  </main>
</body>
</html>
```

style.less

```
.big-button {
  display: inline-block;
  width: 300px;
  height: 300px;
  background: #9B30FF;
  margin: 30px;
}
```

What if we want to use #9B30FF as a theme throughout the app? It would be nice if it only had to change in one place.

```
@mainColor: #9B30FF;

.big-button {
  ...
  background: @mainColor;
  ...
}
```

Mixins

```
.YellowBorder { border-color: #fff000; }

.BigBorder { border-width: 10px; }

.SolidBorder { border-style: solid; }

.DashedBorder { border-style: dashed; }
```

```
#button1 {
  .YellowBorder;
  .DashedBorder;
}
#button2 {
  .YellowBorder;
  .SolidBorder;
  .BigBorder;
}
#button3 {
  .YellowBorder;
  .SolidBorder;
}
```

Parametric Mixins

```
.RoundedCorners( @radius: 0 ) {
  border-radius: @radius;
  -webkit-border-radius: @radius;
  -moz-border-radius: @radius;
}
```

```
#button1 {
  .RoundedCorners;
  ...
}
#button2 {
  .RoundedCorners(25px);
  ...
}
#button3 {
  .RoundedCorners(9999px);
  ...
}
```

Math

```
#button1 {
  ...
  background-color: @mainColor;
}
#button2 {
  ...
  backgroundColor: @mainColor - #222;
}
#button3 {
  ...
  backgroundColor: @mainColor + #222;
}

NOTE:  You can do math on any number
eg. @pixelSize * 2
```

What we have done with the buttons is darkened button2 by 2 hexadecimal units to RGB

On button3 we lightened by 2 hexadecimal units to RGB

#000 has no effect and #fff has max effect

Color functions

Less has built in color functions:

darken() & lightent(): Adds some black and white

saturate() & desaturate(): To be more vibrant or grayscale

fadein() & fadeout(): To change transparency

spin(): Modifies the hue

```
@grey: #d3d3d3;

@borderColor:desaturate(@grey, 100%)
```

Nesting

```
aside {
  h1 {
    color: @mainColor;
    &:hover {
      background-color: @defaultThemeColor;
    }
  }
}
```

Importing

We can import files to keep things like our theme separated. Let's move all of our colors to theme.less

@mainColor: #9B30FF;

@defaultThemeColor: #d3d3d3;

@borderColor:desaturate(@defaultThemeColor, 100%);

Now at the type of style.less

@import ‘theme’;

Making our own grid

grid.less

```
@base_px: 16;  //sets base px to most common browser px size

@toRem: (1 / @base_px) + 0rem;  //set values of default px size converted to rem

@toEm: ( 1 / @base_px) + 0em;  // same as above converted to em

@default-width: 1200 * @toRems;

@default-colspan: 1;

@default-total_cols: 4;
```

grid.less (cont)

```
.Row ( @width : @default-width ) {
  max-width: @width;
  width: 100%;
  margin: 0 auto;
  &:before,
  &:after {
    content: “”;
    display: table;
  }
  &:after{
    clear: both;
  }
}
```

```
.Cols ( @colspan:@default-colspan; @total_cols :@default-total_cols) {
  width: (@colspan * (100 / @total_cols) ) + 0%;
  float: left;
}
```

style.less

```
@import ‘grid’;
…
aside {
  .Cols(1);
  min-height: 500 * @toRems;
  background-color: #eee;
  ….
}
article {
  .Cols(3);
  min-height: 500 * @toRems;
  background-color: #ddd;
}
```

Less vs Sass

Which is better? That is debateable. Sass and Less have a lot to offer and it is worth knowing both of them.

lesscss.org

sass-lang.com