Using LESS and SASS: Module 3

SASS

Shawn Wildermuth
AgiliTrain
agilitrain.com

pluralsight
hardcore developer training

Agenda

- **SASS**
  - What is SASS
  - Using SASS on the Server
  - Variables
  - Rules
  - Importing
  - Extending
  - Mixins
  - Functions
  - Control Directives

What is SASS?

- **Dynamic Style Sheet Language**
  - Syntactically Awesome StyleSheets
  - “Compiles” to CSS
  - Introduces programming features to CSS

What is SASS? (2)

- **SASS has two syntaxes**
  - SASS and SCSS

```
/* SCSS */
$baseFontSize: 14px;

#main
{
  h1
  {
    font-size: $baseFontSize;
  }
}
```

SASS on the Server

- **Support for Server-Side**
  - Node.js
  - ASP.NET
  - Other ways for Rails, JSP, etc.

Node.js

```
$ npm install sass

var less = require('sass');

sass.render(sassContents,
            function (e, css) {
              console.log(css);
            });
```

ASP.NET via Plugin

- **Chirpy**
  - Design-time Compilation
  - Must name files "foo.chirpy.sass"
  - http://chirpy.codeplex.com

ASP.NET via Nuget

```
install-package SassAndCoffee
```

Using SASS in ASP.NET

Demo

Variables

```
$myColor: #ffeedd;

$a: Black;                  // Color
$b: 4px;                    // Units
$c: 1.0em;                  // Units
$d: Helvetica, sans-serif;  // Lists
$e: 1px #000 Solid 0 0;     // Also Lists
```

Operations

```
// Operations Just Work
font-size: 4px + 4;      // 8px
font-size: 20px * .8;    // 16px;
color: #FFF / 4;         // #404040;
width: (100% / 2) + 25%; // 75%
```

Color Functions

```
color: lighten($color, 10%);
color: darken($color, 10%);

color: saturate($color, 10%);
color: desaturate($color, 10%);

color: fade_in($color, .1);
color: fade_out($color, .1);

color: invert($color);
color: complement($color);
```

More Functions

```
$quoted: quote($sometext);
$unquoted: unquote($sometext);

$value: if(true, $color1, $color2);

$rnd: round(3.14);
$top: ceil(3.14);
$bot: floor(3.14);
$per: percentage(.14);
```

String Interpolation

```
// Can use Ruby/PHP style string insertion
$root: "/images/";

#form {
  background: url("#{$root}background.jpg");
  // Becomes url("/images/background.jpg")
}

// Also
$name: "my-class";
.#{$name} {
  color: Blue;
}
```

Using Variables and Operations

Demo

Rules

- **Allows you to structure rules in a logical way**
  - Hierarchies imply the cascading/specificity
  - SASS then deconstructs it into CSS for you

Rules

```
// SCSS
nav {
  font-size: 14px;
  font-weight: bold;
  float: right;
  ul {               // Makes "nav ul {...}"
    list-style-type: none;
    li {             // Makes "nav ul li {...}"
      float: left;
      margin: 2px;
    }
  }
}
```

Rules

```
// Use Parent Selector (&) to mix with parent:
a {
  text-decoration: none;
  &:hover {
    text-decoration: underline;
  }
}

// Results in
a { text-decoration: none; }
a:hover { text-decoration: underline; }
```

Rules

```
// Nested Properties too
.button {
  font: {
    family: Verdana, Helvetica, sans-serif;
    size: 14px;
  }
}

// Results in
.button {
  font-family: Verdana, Helvetica, sans-serif;
  font-size: 14px;
}
```

Using Rules

Demo

Directives

- **Operations on the CSS**
  - @import
  - @extend
  - @mixin
  - @function

@import

```
@import "foo.css";  // Emits CSS Import
                    // @import url(foo.css);

@import "foo.scss"; // Embeds in result
@import "foo";      // Also Embeds

// Nested Import too
#main {
 @import "colors";
}
```

Using @import

Demo

@extend

```
// Inherits Styles from another
.button {
  color: Black;
}
.submit-button {
  @extend .button;
  border: 1px Black solid;
}
// Emits
.submit-button {border: 1px solid Black; }
.button, .submit-button {
  color: Black;
}
```

@extend

```
// multiple inheritance too
.submit-button {
  @extend a:hover;  // inherit from any rule
  @extend .button;  // multiple rules
  border: 1px Black solid;
}
```

Using @extend

Demo

@mixin

- **Repeatable sections**
  - Feel like functions
  - Used insert one or more than one name/value pair
  - Can accept parameters, defaults and overloads

@mixin

```
@mixin font-large {
  font: {
    size: 14px;
    family: san-serif;
    weight: bold;
  }
}

#form {
  @include font-large;
}
```

@mixin

```
// Parameters
@mixin rounded-corners-all($size) {
  border-radius: $size;
  -webkit-border-radius: $size;
  -moz-border-radius: $size;
}

#form {
  @include rounded-corners-all(5px);
}
```

@mixin

```
// Default Parameter Value
@mixin rounded-corners-all($size: 5px) {
  border-radius: $size;
  -webkit-border-radius: $size;
  -moz-border-radius: $size;
}

#form {
  @include rounded-corners-all;  // Optional
}
```

Using @mixin

Demo

@function

```
// Value calculations
$app-width: 900px;
@function column-width($cols) {
  @return ($app-width / $cols) - ($cols * 5px);
}

.col2 {
  width: column-width(2);
}

.col3 {
  width: column-width(3);
}
```

Using @function

Demo

Control Directives

- **For control flow:**
  - @if
  - @for
  - @each
  - @while

@if

```
h1 {
  @if $size > 14px {
    color: Blue;
  }
  @else if $size < 14px {
    color: Red;
  }
  @else {
   color: Green;
  }
}
```

@for

```
$page-width: 1000px;

@for $col from 1 through 4 {
  .col#{$col} {
    width: $page-width / $col;
  }
}
```

@each and @while

```
@each $item in first, second, third, fourth {
  .#{$item} {
    background-url: url(/images/#{$item}.jpg);
  }
}

$i: 1;
@while $i < 5 {
  h#{$i} {
    font-size: $i * 4px;
    $i: $i + 1;
  }
}
```

Using Control Directives

Demo

Summary

- **Staying SASSy**
  - SASS allows you to bring your developer head to design
  - SASS improves reuse and readability
  - SASS allows you to structure and modularize your designs
  - SASS enables control flow, templating and more to your styling.