CSS Preprocessors: Less and Sass

Jogesh K. Muppala

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

CSS

- CSS is great for defining styles and repeatedly applying them to various HTML elements
- CSS is limited when it comes to features like defining variables, nesting selectors, expressions, functions
  - Maintainability suffers

CSS Preprocessors

- Several popular:
  - Less
  - Sass (Syntactically Awesome Style Sheets), and
  - Stylus
- All are compiled into traditional CSS syntax automatically before use in a web page
- We will examine Less and Sass briefly
- Bootstrap was originally available in Less, now also available as Sass

Typical CSS Preprocessors Features

- Variables
- Nesting selectors
- Mixins
- Functions & Expressions

Variables

- Less

```
@indigo: #303f9f;
@carousel-height: 300px;

.navbar-inverse {
        background:@indigo;
}
.carousel .item {
        height:  @carousel-height;
}
```

variables can also have scope

- Sass

```
$indigo: #303f9f;
$carousel-height: 300px;

.navbar-inverse {
        background:$indigo;
}
.carousel .item {
        height:  $carousel-height;
}
```

Nesting

- Less

```
@dark-indigo: #1a237e;

// Height variables
@carousel-height: 300px;

.carousel {
        background:@dark-indigo;
        .item {
                height: @carousel-height;
                img {
                        left: 0;
                        min-height: @carousel-height;
                        position: absolute;
                        top: 0;
                }
        }
}
```

- Sass

```
$dark-indigo: #1a237e;

// Height variables
$carousel-height: 300px;

.carousel {
        background:$dark-indigo;
        .item {
                height: $carousel-height;
                img {
                        left: 0;
                        min-height: $carousel-height;
                        position: absolute;
                        top: 0;
                }
        }
}
```

Mixins

- Less

```
.zero-margin {
        margin:0px auto;
        background:@white;
}

.row-header {
        .zero-margin;
        padding:0px auto;
}

.row-content {
        .zero-margin;
        border-bottom: 1px ridge;
        min-height:400px;
        padding: 50px 0px 50px 0px;
}
```

- Sass

```
@mixin zero-margin {
        margin:0px auto;
        background:@white;
}

.row-header {
        @include zero-margin;
        padding:0px auto;
}

.row-content {
        @include zero-margin;
        border-bottom: 1px ridge;
        min-height:400px;
        padding: 50px 0px 50px 0px;
}
```

Note: use .zero-margin() { … } if you don't want .zero-margin itself to be included in the compiled CSS

Mixins can hold multiple CSS declarations unlike variables

Mixins with Parameters

- Less

```
.zero-margin (@pad-up-dn: 0px, @pad-left-right: 0px) {
        margin:0px auto;
        padding: @pad-up-dn @pad-left-right;
}

.row-header {
        .zero-margin();
        padding:0px auto;
}

.row-content {
        .zero-margin (50px, 0px);
        border-bottom: 1px ridge;
        min-height:400px;
}
```

- Sass

```
@ mixin zero-margin ($pad-up-dn, $pad-left-right) {
        margin:0px auto;
        padding: $pad-up-dn $pad-left-right;
}

.row-header {
        @include zero-margin(0px, 0px);
        padding:0px auto;
}

.row-content {
        @include zero-margin (50px, 0px);
        border-bottom: 1px ridge;
        min-height:400px;
}
```

Mathematical Operations

- Less

```
@carousel-height: 300px;
.carousel .item {
        height:  @carousel-height;
}
.carousel .item .item-small{
        height:  (@carousel-height / 4);
}
.carousel .item .item-large{
        height:  @carousel-height *2;
}
```

- Sass

```
$carousel-height: 300px;
.carousel .item {
        height:  $carousel-height;
}
.carousel .item .item-small{
        height:  ($carousel-height / 4);
}
.carousel .item .item-large{
        height:  $carousel-height *2;
}
```

Other Features

- Functions: math, list, string, color, color operations, color blending etc.
  - See the documentation
- Imports
  - e.g., in both Less and Sass: @import “foo”

Exercise: Less

- Simple exercise to write a Less file
- Generate the corresponding CSS file
  - Use lessc compiler