

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
 - Scss (Sassy CSS) syntax examined in this course
 - Stylus
- All are compiled into traditional CSS syntax automatically before use in a web page
- We will examine Less and Sass briefly
- Bootstrap 3 used Less, but Bootstrap 4 migrated to use Sass

Typical CSS Preprocessors Features

- Variables
- Nesting selectors
- Mixins
- Functions & Expressions

Variables

- Less

```
@lt-gray: #ddd;
```

```
@background-dark: #512DA8;
```

```
@carousel-item-height: 300px;
```

```
.navbar-inverse {  
    background:@background-dark;  
}
```

```
.carousel .carousel-item {  
    height: @carousel-item-height;  
}
```

variables can also have scope

- Scss

```
$lt-gray: #ddd;
```

```
$background-dark: #512DA8;
```

```
$carousel-item-height: 300px;
```

```
.navbar-inverse {  
    background:$background-dark;  
}
```

```
.carousel .item {  
    height: $carousel-item-height;  
}
```

Nesting

- Less

```
@background-dark: #512DA8;  
@carousel-item-height: 300px;
```

```
.carousel {  
  background:@background-dark;  
  .carousel-item {  
    height: @carousel-item-height;  
    img {  
      position: absolute;  
      top: 0;  
      left: 0;  
      min-height: 300px;  
    }  
  }  
}
```

- Scss

```
$background-dark: #512DA8;  
$carousel-item-height: 300px;
```

```
.carousel {  
  background:$background-dark;  
  .carousel-item {  
    height: $carousel-item-height;  
    img {  
      position: absolute;  
      top: 0;  
      left: 0;  
      min-height: 300px;  
    }  
  }  
}
```

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;  
}
```

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

- Scss

```
@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;  
}
```

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;  
}
```

- Scss

```
@ 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-item-height: 300px;

.carousel-item {
    height: @carousel-item-height;
}

.carousel-item .item-large{
    height: @carousel-item-height *2;
}
```

- Scss

```
$carousel-item-height: 300px;

.carousel-item {
    height: $carousel-item-height;
}

.carousel-item .item-large{
    height: $carousel-item-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