CSS 3

LESS CSS

Overview

- Dynamic stylesheet language (CSS preprocessor)

- Don’t confuse with LESS framework!
  - LessCSS extends CSS with dynamic behavior
  - LessFramework allows to design pages easily (bootstrap)

- Documentation findable here: http://lesscss.org/

Overview

- Open source

- Written first in Ruby
- Ruby replaced by JavaScript in later versions

- Syntax very close to classical CSS
  - CSS file is a valid Less file

Overview

- Can be used in a website in two ways:
  - By a local compiler who transcripts Less files in CSS files
  - By uploading Less files and using a JavaScript adapter who converts the code on the fly

- Similar projects:
  - Sass
  - Zuss

Advantages

- Less redundant and more readable code
  - Thanks to nesting selectors

- Advanced concepts:
  - Variables
  - Functions
  - Operations
  - …

Install

- Stylesheet inclusion:

```
<link rel="stylesheet/less" type="text/css"
          href="styles.less">
```

- JavaScript adapter inclusion:

```
<script src="less.js" type="text/javascript"></script>
```

Nesting

Less CSS

```
#header {
    h1 {
        font-size: 26px;
    }
    a {
        text-decoration: none;
        &:hover {
            color: black;
        }
     }
}
```

Generated CSS

```
#header h1 {
    font-size: 26px;
}

#header a {
    text-decoration: none;
}

#header a:hover {
    color: black;
}
```

Variables

Less CSS

```
@main-color: #FFDEAD;
#header {
    h1 {
        color: @main-color;
    }
}
#container {
    border-left: 1px solid
        @main-color;
    border-right: 1px solid
        @main-color;
}
```

Generated CSS

```
#header h1 {
    color: #FFDEAD;
}

#container {
    border-left: 1px solid
        #FFDEAD;
    border-right: 1px solid
        #FFDEAD;
}
```

Operations

Less CSS

```
@border-width: 1px;
@dark-gray: #111;
#header {
    color: @dark-gray * 3;
}
#container {
    border: @border-width * 2
        solid @dark-gray;
    color: (@dark-gray + #111)
        * 4;
}
```

Generated CSS

```
#header {
    color: #333;
}

#container {
    border: 2px solid #111;
    color: #888;
}
```

Mixins

Less CSS

```
.rounded-corners {
    border-radius: 5px;
    -moz-border-radius: 5px;
    -webkit-border-radius: 5px;
}
#header {
    .rounded-corners;
}
#container {
    .rounded-corners;
}
```

Generated CSS

```
#header, #container,
.rounded-corners {
    border-radius: 5px;
    -moz-border-radius: 5px;
    -webkit-border-radius: 5px;
}
```

Mixins with parameters

Less CSS

```
.rounded-corners(@size) {
    border-radius: @size;
    -moz-border-radius: @size;
    -webkit-border-radius: @size;
}
.border-base(@color: black, @size: 1px) {
    border: @size solid @color;
}
#container {
    .rounded-corners(5px);
    .border-base(#333);
}
```

Mixins with parameters

Generated CSS

```
#container {
    border-radius: 5px;
    -moz-border-radius: 5px;
    -webkit-border-radius: 5px;
    border: 1px solid #333;
}
```

Is it really like functions in other languages?

Mixins overloading

Less CSS

```
.grady(@a) {
    color: @a;
}
.grady(@a, @b) {
    color: fade(@a, @b);
}

#container { .grady(#555); }
#footer {
    .grady(black, 50%);
}
```

Generated CSS

```
#container {
    color: #555;
}

#footer {
    color: rgba(0, 0, 0, 0.5);
}
```

Built-in functions - Color

- Color changing functions (1 of 2):
  - Decline your entire layout by only two colors!

Example	Champ 2
lighten(@color, 10%);	Return a color 10% lighter than @color
darken(@color, 10%);	Return a color 10% darker than @color
saturate(@color, 10%);	Return a color 10% more saturated than @color
desaturate(@color, 10%);	Return a color 10% less saturated than @color
mix(@color1, @color2);	Return a mix of @color1 and @color2

Built-in functions - Color

- Color changing functions (2 of 2):
  - Decline your entire layout by only two colors!

Example	Champ 2
fadein(@color, 10%);	Return a color 10% less transparent than @color
fadeout(@color, 10%);	Return a color 10% more transparent than @color
fade(@color, 50%);	Return @color with 50% transparency
spin(@color, 10);	Return a color 10 degree larger in hue than @color
spin(@color, -10);	Return return a color 10 degree smaller hue than @color

Conditional statements

- Less CSS handles conditions!
  - Begins with « when »
  - Optionnaly followed by « **and** », « **,** » (logical OR), « **not** »

```
.fontcolor(@bg-opacity) when (@bg-opacity =< 0.5) {
    color: black;
}

.fontcolor(@bg-opacity) when (@bg-opacity > 0.5) {
    color: white;
}

#container { opacity: 0.5; .fontcolor(0.5) ; }
```

Conditional statements

- Five operators:
  - =>
  - >
  - =
  - <
  - =<

- Only dimension and keywords are comparable

Additional informations

- Less can be compiled and minified server side for a perfect integration with nodeJS and Rhino.
  - You’ll see it in the next semester!

- Note: At the time of this writing Chrome doesn’t support Less.js if the path to your page starts with "file:///".

Questions ?