

# **CSS Preprocessors**

DRY & KISS Stylesheets

Pavel Smolka

# Motivation: Corporate color

```
.headline {  
  color: #123456;  
}
```

```
#notification p {  
  background-color: #123456;  
}
```

Happy refactoring...

# CSS should contain variables

```
@color: #123456;
```

```
.headline {  
  color: @color;  
}
```

Unfortunately, it does not...

...yet: <http://dev.w3.org/csswg/css-variables/>

# We have just invented preprocessor

And we are not alone...

LESS: <http://lesscss.org/>

SASS/SCSS: <http://sass-lang.com/>

Stylus: <http://learnboost.github.io/stylus/>

# What do preprocessors provide?

Simplier syntax

Variables and arithmetics

Nesting rules

Inheritance/Mixins

Imports

# LESS - mixins

```
.rounded-corners (@radius: 5px) {  
  -webkit-border-radius: @radius;  
  -moz-border-radius: @radius;  
  -ms-border-radius: @radius;  
  -o-border-radius: @radius;  
  border-radius: @radius;  
}  
  
#header {  
  .rounded-corners;  
}  
#footer {  
  .rounded-corners(10px);  
}
```

# LESS - mixins (compiled)

```
#header {
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
  -ms-border-radius: 5px;
  -o-border-radius: 5px;
  border-radius: 5px;
}
#footer {
  -webkit-border-radius: 10px;
  -moz-border-radius: 10px;
  -ms-border-radius: 10px;
  -o-border-radius: 10px;
  border-radius: 10px;
}
```

# LESS - nested rules

```
#header {  
  h1 {  
    font-size: 26px;  
    font-weight: bold;  
  }  
  p { font-size: 12px;  
    a { text-decoration: none;  
      &:hover { border-width: 1px }  
    }  
  }  
}
```

Ampersand (&) stands for the parent selector.



# LESS - nested rules (compiled)

```
#header h1 {  
  font-size: 26px;  
  font-weight: bold;  
}  
#header p {  
  font-size: 12px;  
}  
#header p a {  
  text-decoration: none;  
}  
#header p a:hover {  
  border-width: 1px;  
}
```

# LESS - functions and operations

```
@the-border: 1px;
@base-color: #111;
@red:        #842210;

#header {
  color: (@base-color * 3);
  border-left: @the-border;
  border-right: (@the-border * 2);
}
#footer {
  color: (@base-color + #003300);
  border-color: desaturate(@red, 10%);
}
```

You can use any JavaScript function...

# LESS - functions and operations (c)

```
#header {  
  color: #333;  
  border-left: 1px;  
  border-right: 2px;  
}  
  
#footer {  
  color: #114411;  
  border-color: #7d2717;  
}
```

# Compilation in web browser (LESS)

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

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

Make sure you link the stylesheet before the script...

# Server-side compilation

```
$ npm install stylus
```

```
$ stylus < some.styl > some.css
```

```
$ npm install -g less
```

```
$ lessc styles.less styles.css
```

```
$ gem install sass # install with Ruby
```

```
$ npm install sass # install with Node
```

```
$ sass input.scss output.css
```

It's good idea to get a bit familiar with Node...

# Automatic compilation

```
sass --watch app/sass:public/stylesheets
```

```
stylus -w stylus -o css
```

Sprockets, ...

# So, what's the message?

~~All the markup is just crap with no good procedural processing tools~~

Make your stylesheets simpler and re-usable!

Twitter: [@pavelmolka](https://twitter.com/pavelmolka)

E-mail: [smolka@mail.muni.cz](mailto:smolka@mail.muni.cz)

GitHub: <https://github.com/pavelmolka>

WWW: I don't have any serious web :-{