



Less

Trainer: Alex Drimbe



What is LESS?

Less is a CSS pre-processor, meaning that it extends the CSS language, adding features that allow variables, mixins, functions and many other techniques that allow you to make CSS that is more maintainable, themable and extendable.



Client-side usage

Using less.js in the browser is great for development, but it's not recommended for production

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

```
<script src="//cdnjs.cloudflare.com/ajax/libs/less.js/2.5.1/less.min.js"></script>
```



Grunt The JavaScript Task Runner

<http://gruntjs.com/>

- Less plugin - <https://www.npmjs.com/package/grunt-contrib-less>
- Watch plugin - <https://www.npmjs.com/package/grunt-contrib-watch>



- Install nodejs - <https://nodejs.org/download/>
- Install Grunt's command line interface - `npm install grunt-cli`
- Install Grunt – `npm install grunt --save`
- Install Less compile plugin – `npm install grunt-contrib-less --save`
- Install Watch plugin – `npm install grunt-contrib-watch --save`
- Configure the Grunt file – `Gruntfile.js`



These are pretty self-explanatory:

```
@nice-blue: #5B83AD;  
@light-blue: @nice-blue + #111;
```

```
#header {  
  color: @light-blue;  
}
```

Outputs:

```
#header {  
  color: #6c94be;  
}
```



Classic Css

```
<a class="animal bear">Bear</a>
```

```
.animal {  
  background-color: black;  
  color: white;  
}  
.bear {  
  background-color: brown;  
}
```

Less

```
<a class="bear">Bear</a>
```

```
.animal {  
  background-color: black;  
  color: white;  
}  
.bear {  
  &:extend(.animal);  
  background-color: brown;  
}
```



```
.bordered {  
  border-top: dotted 1px black;  
  border-bottom: solid 2px black;  
}
```

```
#menu a {  
  color: #111;  
  .bordered;  
}
```

```
.post a {  
  color: red;  
  .bordered;  
}
```




```
.border-radius(@radius) {  
  -webkit-border-radius: @radius;  
  -moz-border-radius: @radius;  
  border-radius: @radius;  
}
```

```
#header {  
  .border-radius(4px);  
}  
.button {  
  .border-radius(6px);  
}
```



Less gives you the ability to use nesting instead of, or in combination with cascading.

```
#header {  
  color: black;  
  .navigation {  
    font-size: 12px;  
  }  
  .logo {  
    width: 300px;  
  }  
}
```



The & operator represents the parent selectors of a nested rule and is most commonly used when applying a modifying class or pseudo-class to an existing selector:

```
a {  
  color: blue;  
  &:hover {  
    color: green;  
  }  
}
```



Import styles from other style sheets

```
@import "foo";    // foo.less is imported  
@import "foo.less"; // foo.less is imported  
@import "foo.php"; // foo.php imported as a less file  
@import "foo.css"; // statement left in place, as-is
```



```
.mixin (@a) when (lightness(@a) >= 50%) {  
  background-color: black;  
}  
.mixin (@a) when (lightness(@a) < 50%) {  
  background-color: white;  
}  
.mixin (@a) {  
  color: @a;  
}
```

```
.class1 { .mixin(#ddd) }  
.class2 { .mixin(#555) }
```

```
//Output  
.class1 {  
  background-color: black;  
  color: #ddd;  
}  
.class2 {  
  background-color: white;  
  color: #555;  
}
```



```
loop(@counter) when (@counter > 0) {  
  .loop((@counter - 1)); // next iteration  
  width: (10px * @counter); // code for each iteration  
}
```

```
div {  
  .loop(5); // launch the loop  
}
```

```
//Output  
div {  
  width: 10px;  
  width: 20px;  
  width: 30px;  
  width: 40px;  
  width: 50px;  
}
```



```
.generate-columns(@n, @i: 1) when (@i =< @n) {  
  .column-@{i} {  
    width: (@i * 100% / @n);  
  }  
  .generate-columns(@n, (@i + 1));  
}
```

```
.generate-columns(4);
```

//Output

```
.column-1 {  
  width: 25%;  
}  
.column-2 {  
  width: 50%;  
}  
.column-3 {  
  width: 75%;  
}  
.column-4 {  
  width: 100%;  
}
```