302、Less

目标

大纲

# 熟悉Less

## Less概述

LESSCSS是一种动态样式语言，属于CSS预处理语言的一种，它使用类似CSS的语法，为CSS的赋予了动态语言的特性，如变量、继承、运算、函数等，更方便CSS的编写和维护。

LESSCSS可以在多种语言、环境中使用，包括浏览器端、桌面客户端、服务端。

## 浏览器端使用

LESSCSS也可以不经编译，直接在浏览器端使用。

使用方法：

下载LESSCSS的.js文件，例如lesscss-1.4.0.min.js。

在页面中引入.less文件

<link rel="stylesheet/less" href="less/main.less" />

需要注意rel属性的值是stylesheet/less，而不是stylesheet。

1. 引入第1步下载的.js文件

<script src="lib/less.js"></script>

## Node编译

LESSCSS官方有一款基于Node.js的库，用于编译.less文件。

使用时，首先全局安装less

cnpm install less -g

lessc less/main.less css/main.css

# Less语法

## 变量

很容易理解：

**@nice-blue**: #5B83AD;  
**@light-blue**: **@nice-blue** + #111;  
  
#header { color: **@light-blue**; }

输出

#header {  
 color: #6c94be;  
}

甚至可以在定义变量值时使用其它的变量：

**@fnord**: "I am fnord.";  
**@var**: 'fnord';  
header{  
 content: **@@var**;  
}

解析后：

header {  
 content: "I am fnord.";  
}

如果对同一个变量定义两次的话，在当前作用域中最后一次定义的将被使用。这与CSS的机制类似，最后一次定义的值会成为这个属性的值。

比如：

**@len**: 0px;  
header {  
 **@var**: 100px;  
 h2 {  
 **@var**: 200px;  
 height: **@var**;  
 **@var**: 300px;  
 }  
 height: **@var**;  
}

会编译成：

header {  
 height: 100px;  
}  
header h2 {  
 height: 300px;  
}

变量是“按需加载”（lazy loaded）的，因此不必强制在使用之前声明。

下面是一个有效的LESS代码片段：

header {  
 width: **@var**;  
}  
**@var**: **@a**;  
**@a**: 9%;

会编译成：

header {  
 width: 9%;  
}

## 混合

在 LESS 中我们可以定义一些通用的属性集为一个 class，然后在另一个 class 中去调用这些属性，下面有这样一个 class：

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

那如果我们现在需要在其他 class 中引入那些通用的属性集，那么我们只需要在任何 class 中像下面这样调用就可以了：

#menu a {  
 color: #111;  
 .bordered;  
}  
.post a {  
 color: red;  
 .bordered;  
}

.bordered class 里面的属性样式都会在 #menu a 和 .post a 中体现出来：

.bordered {  
 border-top: dotted 1px black;  
 border-bottom: solid 2px black;  
}  
#menu a {  
 color: #111;  
 border-top: dotted 1px black;  
 border-bottom: solid 2px black;  
}  
.post a {  
 color: red;  
 border-top: dotted 1px black;  
 border-bottom: solid 2px black;  
}

任何 CSS class, id 属性集都可以以同样的方式引入。

## 带参数混合

在 LESS 中，你还可以像函数一样定义一个带参数的属性集合：

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

然后在其他 class 中像这样调用它：

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

编译后

#header {  
 border-radius: 4px;  
 -moz-border-radius: 4px;  
 -webkit-border-radius: 4px;  
}  
.button {  
 border-radius: 6px;  
 -moz-border-radius: 6px;  
 -webkit-border-radius: 6px;  
}

我们还可以像这样给参数设置默认值：

.border-radius (**@radius**: 5px) {  
 border-radius: **@radius**;  
 -moz-border-radius: **@radius**;  
 -webkit-border-radius: **@radius**;  
}

所以现在如果我们像这样调用它的话：

#header {  
 .border-radius;  
}

radius 的值就会是 5px。

你也可以定义不带参数属性集合，如果你想隐藏这个属性集合，不让它暴露到 CSS 中去，但是你还想在其他的属性集合中引用，你会发现这个方法非常的好用：

.wrap () {  
 text-wrap: normal;  
 white-space: pre-wrap;  
 white-space: -moz-pre-wrap;  
 word-wrap: break-word;  
}  
  
pre { .wrap }

输出

pre {  
 text-wrap: normal;  
 white-space: pre-wrap;  
 white-space: -moz-pre-wrap;  
 word-wrap: break-word;  
}

## 多参数混合

多个参数可以使用分号分隔

使用同样的名字和同样数量的参数定义多个混合是合法的。在被调用时，LESS会应用到所有可以应用的混合上。比如你调用混合时只传了一个参数.mixin(green)，那么所有只强制要求一个参数的混合都会被调用：

.mixin(**@color**) {  
 color: **@color**;  
}  
.mixin(**@color**; **@padding**:2px) {  
 color: **@color**;  
 padding: **@padding**;  
}  
.mixin(**@color**; **@padding**; **@margin**: 2px) {  
 color: **@color**;  
 padding: **@padding**;  
 margin: **@margin @margin @margin @margin**;  
}  
.some .selector div {  
 .mixin(#008000);  
}

编译后

.some .selector div {  
 color: #008000;  
 padding: 2px;  
}

## 嵌套规则

LESS 可以让我们以嵌套的方式编写层叠样式。 让我们先看下下面这段 CSS：

#header { color: black; }  
  
#header .navigation {  
 font-size: 12px;  
}  
  
#header .logo {  
 width: 300px;  
}  
  
#header .logo:hover {  
 text-decoration: none;  
}

在 LESS 中, 我们就可以这样写：

#header {  
 color: black;  
  
 .navigation {  
 font-size: 12px;  
 }  
  
 .logo {  
 width: 300px;  
 &:hover { text-decoration: none }  
 }  
}

代码更简洁了，而且感觉跟 DOM 结构格式有点像。

注意 & 符号的使用 — 如果你想写串联选择器，而不是写后代选择器，就可以用到 & 了。这点对伪类尤其有用如 :hover 和 :focus。

例如：

.bordered {  
 &.float {  
 float: left;  
 }  
 .top {  
 margin: 5px;  
 }  
}

编译成

.bordered.float {  
 float: left;  
}  
.bordered .top {  
 margin: 5px;  
}

## 运算

任何数字、颜色或者变量都可以参与运算，运算应该被包裹在括号中。来看一组例子：

**@base**: 5%;  
**@base-color**:green;  
**@filler**: (**@base** \* 2);  
**@other**: (**@base** + **@filler**);  
  
header {  
 color: (#888 / 4);  
 background-color: (**@base-color** + #111);  
 height: (100% / 2 + **@filler**);  
}

编译成

header {  
 color: #222222;  
 background-color: #119111;  
 height: 60%;  
}