



做真实的自己，用良心做教育

张骏
H5教学部

JavaScript



CSS预处理语言

--- Less

课程概要：

CSS预处理语言
LESS的安装和运行
变量和作用域
变量的使用和连接
嵌套
混合
外部导入
运算
循环
条件
函数

CSS预处理语言

LESS是一门CSS预处理语言，它扩展了CSS语言。增加了诸如变量、混合(mixin)、函数等功能。让CSS更容易维护、方便制作主题、扩充。

LESS可以运行在Node或浏览器端。

本质上，LESS包含一套自定义的语法以及一个解析器。用户根据这些语法定义自己的样式规则，这些规则最终通过解析器，编译生成对应的CSS文件。

LESS没有裁减CSS原有的特性，更不是用来取代CSS的。而是在现有的CSS语法基础上，为CSS加入程序式的语言特性。

LESS的安装和运行

服务端

传统的安装方式是通过node的包管理工具npm进行less安装

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

编写好less文件后使用lessc命令进行文件编译

```
$ lessc input.less output.css
```

也可以在项目构建工具(grunt、gulp、webpack等)配置文件监听自动编译less

客户端

引入less的js文件

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

使用link元素引入less文件 设置 rel="stylesheet/less"

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

推荐使用方式

使用vscode插件 easy less

安装插件后会自动监听less文件，当文件发生改变时自动将其编译成css文件

输出目录路径配置

在vscode的用户设置配置 settings.json 中 添加

```
"less.compile": {  
  "out": "..\\css\\"  
}
```

变量和作用域

使用变量可以使代码变得更易于维护

在less中申明变量 使用 @ 符号

在less中变量的作用域分为

全局作用域 -- 在顶级作用域中申明

块级作用域 -- 在代码块中进行申明

例如:

```
@color : red;
```

```
@baseurl : "../img";
```

变量的使用 and 连接

```
@link-color: #428bca; // sea blue
@link-color-hover: darken(@link-color,10%);
```

```
a.link{
    color:@link-color;
}
```

```
a:hover{
    color:@link-color-hover;
}
```

```
@my-selector: banner;
@images-url:"../images";
```

```
 @{my-selector}{
    line-height:40px;
    margin:0 auto;
}
```

```
.box{
    background:url("@{images-url}/bg.png");
}
```


嵌套

LESS支持类似于HTML结构的样式嵌套写法，支持所有的css选择器

```
.baneer{  
  width:1200px;  
  margin:0 auto;  
  .list{  
    list-style:none;  
  }  
}
```

层级嵌套时默认为 后代选择器 如需子代选择器需要在选择前进行添加

```
.baneer{  
  width:1200px;  
  margin:0 auto;  
  >.list{  
    list-style:none;  
  }  
}
```

混合

LESS提供了类似于函数调用的混合(混入)功能

Input

```
.mixin-a{
    color:red;
}

.banner{
    width:1200px;
    mrgin:0 auto;
    a{
        .mixin-a;
        font-size:16px;
    }
}
```

Output

```
.mixin-a{
    color:red;
}

.banner{
    width:1200px;
    margin:0 auto;
}

.banner a{
    color:red;
    font-size:16px;
}
```

混合(带参数)

当使用参数功能时, `.border-radius` 将不再被编译到CSS中

Input

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

```
#head{  
    .border-radius(5px);  
}
```

```
.button{  
    .border-radius(20%);  
}
```

Output

```
#head{  
    -webkit-border-radius:5px;  
    -moz-border-radius:5px;  
    border-radius:5px;  
}
```

```
.button{  
    -webkit-border-radius:20%;  
    -moz-border-radius:20%;  
    border-radius:20%;  
}
```

混合(!important)

我们可以为mixin添加 **!important** 关键字
也可以为参数设置默认值

Input

```
.foo(@bg:#f5f5f5,@color:#900){  
  background:@bg;  
  color:@color;  
}
```

```
.unimportant{  
  .foo();  
}
```

```
.important{  
  .foo()!important;  
}
```

Output

```
.unimportant{  
  background:#f5f5f5;  
  color:#999;  
}
```

```
.important{  
  background:#f5f5f5!important;  
  color:#999!important;  
}
```

外部导入

推荐将样式文件分模块书写
使用 **@import** 导入多个模块
可以提升样式代码的可维护性

```
@import "public.less";
```

运算

LESS支持运算功能

Input

```
.list{  
  list-style:none;  
  width:1200px;  
  li{  
    width:1200px/5;  
  }  
}
```

Output

```
.list{  
  list-style:none;  
  width:1200px;  
}  
  
.list li{  
  width:240px;  
}
```

循环

LESS支持循环遍历

太长写不下

参考博客：

<https://blog.csdn.net/seiEight/article/details/95035129>

条件

使用 when 表达式可以进行条件判断

```
.bgWidth(@width) when(@width<150){  
    width:100px;  
    height:50px;  
}
```

```
.bgWidth(@width) when(@width>=150){  
    width:200px;  
    height:100px;  
}
```

```
.bg{  
    .bgWidth(300px);  
}
```


函数

常用函数介绍

ceil()	向上取整
floor()	向下取整
percentage()	百分数
round()	四舍五入
sqrt()	平方根
abs()	绝对值
sin()	正弦
cos()	余弦
asin()	反正弦
acos()	反余弦
tan()	正切
atan()	反正切
pi()	圆周率
min()	最小数
max()	最大数

THANK YOU



做真实的自己，用良心做教育