2016.07.16-08.02

CSS预处理语言
Less

学信网 - 设计部 黄卉
huangh@chsi.com.cn

CSS预处理语言Less

- 1、什么是less
- 2、如何调用less
- 3、less语法详解
- 4、less和sass的对比

1、什么是Less

Less is More , Than CSS -少即是多，比CSS

1.1 Less介绍

- Less是一门css预处理语言，或是一种动态样式语言。扩展了css语言，增加了变量、继承、嵌套、运算、函数等特性，使css更易维护和扩展。（类似jquery）
- Less既可以在浏览器端上运行（支持IE8及其以上、chrome、ff等主流浏览器，不支持iPad），也可以在Node上运行。
- 现在流行的css预编译语言： Less和Sass

1.2 Less的优点

- 简单，易于维护，CSS管理更加容易
- 高效的进行开发
- 非常容易地实现配色和换肤
- 与CSS能够很好地融合使用
- 同时兼容CSS3

2、如何调用Less

2.1 Less调用方法

① 运用GUI工具Koala或者SimpLess等来自动解析成.css文件（已给大家讲过Koala的基本用法）

② 服务器端解析。引用less.js进行解析，在页面引入less.js对.less文件进行解析。

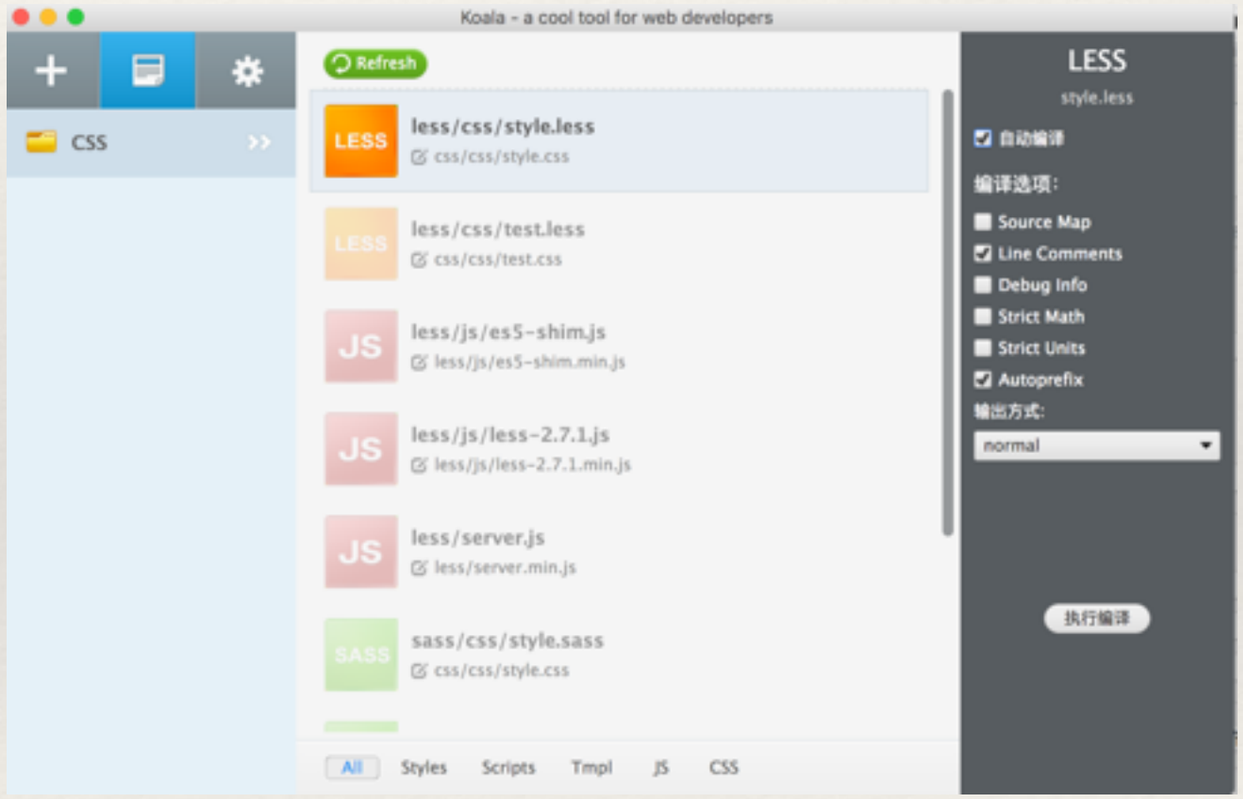
③ 运用node来解析成css

④ 利用gulp等自动化工具进行解析（研究中）

⑤ Less官网在线解析：http://less2css.org

2.2 GUI工具Koala

- Source Map：编译过程中生成 css 对应的 map 文件。调试时使用，有了source map，浏览器里直接显示less，非常方便
- Line Comments：保留注释，勾选后编译中在注释上一行添加一条空行。
- Autoprefix：自动给 CSS3 元素加上浏览器前缀。



2.3 服务器端解析

- 先引入styls.less
  - 内联样式
  - 外联样式

```
<style type="text/less">
  // less 代码
</style>
```

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

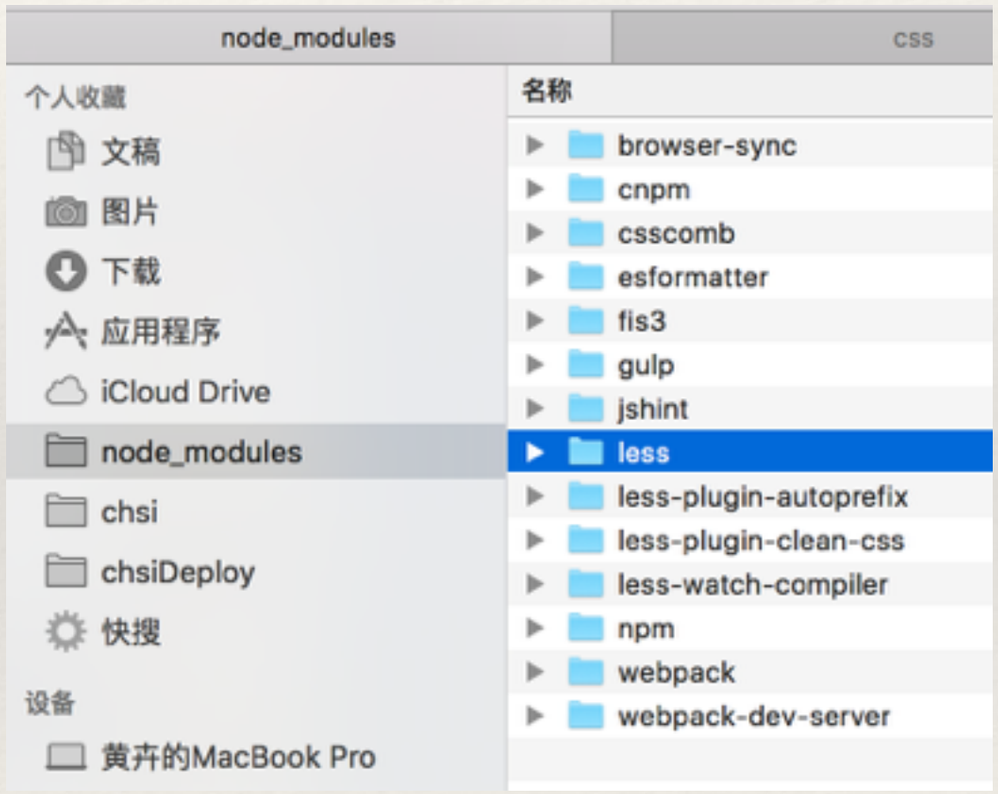
- 再引入less.js

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

- 注意
  - 如果加载多个less样式表文件，每个单独编译，一个文件中定义的任何变量都无法再其他文件中访问。不会存在变量重复问题。
  - 在服务器环境下使用，本地直接打开可能会报错！
  - Less兼容IE7+，如需兼容则需要先引入es5-shim.js即可。
  - 其中还有一些高级配置，也可通过浏览器直接调试，具体见参考文档。
  - 不建议使用，增加服务器压力，不利于调试。

2.4 Node解析

- 先安装node.js和npm：https://nodejs.org/en/
- 下载less的管理工具包less
  - $ sudo npm install -g less（mac全局）
  - $ npm install less（局部）
- 执行方法解析成css文件
  - $ lessc style.less style.css
- 解析成min.css文件
  - 安装clean-css：$ npm install clean-css
  - $ lessc --clean-css style.less style.min.css



- node上安装即时编译的插件
  - $ (sudo) npm install -g less-watch-compiler
  - $ less-watch-compiler *path_input path_out*
  - $ less-watch-compiler *path_input path_out fileName.less*
- 添加厂商前缀
  - $(sudo) npm install -g less-plugin-autoprefix
  - $ less *inputFile.less outFile.css* —autoprefix=“*browsers*”
  - $ lessc test.less test.css --autoprefix="ie >= 8, last 3 versions, > 2%”
- less命令行用法

2.5 常用IDE对应的插件

- SublimeText：Less-sublime、Sublime-Less-to-CSS、Less-build-sublime、SublimeOnSaveBuild
- Dreamweaver：DMXzone Less CSS Compiler
- Notepad++ 6.x：less.js语法高亮
- Brackets：内置支持 语法高亮，还有一些扩展 BracketLESS

3、Less语法详解

- 3.1 注释
- 3.2 变量@
- 3.3 嵌套
- 3.4 运算
- 3.5 继承
- 3.6 匹配模式
- 3.7 @arguments 变量
- 3.8 字符串插入@{name}
- 3.9 避免编译~
- 3.10 ！important
- 3.11 @import
- 3.12 颜色函数

3.1 注释

- /** 可解析 **/
- //不可解析

```less
@charset 'utf-8';
/**
 * less语法详解
 * huangh@chsi.com.cn
 * 2016.07.16
 */
//less语法详解, 2016.07.23
body{
  text-align: center;
  margin:0 auto;
  font-size:16px;
  h1,h2{
    text-align: center;
    border-bottom: 1px solid #ccc;
  }
}
```

```less
@charset 'utf-8';
/**
 * less语法详解
 * huangh@chsi.com.cn
 * 2016.07.16
 */
body {
  text-align: center;
  margin: 0 auto;
  font-size: 16px;
}
body h1,
body h2 {
  text-align: center;
  border-bottom: 1px solid #ccc;
}
```

3.2 变量@

- 变量允许单独定义一系列通用的样式，然后在需要的时候进行调用。用@进行定义。

```less
//less
@cGray:#ccc;
@cRed:red;
@width:200px;
@height:400px;
#div1{
  color:@cGray;
  border:1px solid @cGray;
  width:@width;
  height:@height;
}
#div2{
  color:@cRed;
  background:@cGray;
  width:@height;
  height:@width;
}
```

```css
/*生成的css*/
#div1{
  color:#ccc;
  border:1px solid #ccc;
  width:200px;
  height:400px;
}
#div2{
  color:red;
  background:#ccc;
  width:200px;
  height:400px;
}
```

3.3 嵌套

- 在一个选择其中嵌套另一个选择器来实现继承，减少代码量，代码更清晰。代码优化考虑，最多嵌套三层。
- 有`&`时解析的是同一个元素或此元素的伪类，没有&解析是后代元素

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

```css
/* 生成的 CSS */
#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;
}
```

3.4 运算

- 运算提供了加减乘除操作，可以对任何数字、颜色、变量进行运算，可以实现属性值之间的复杂关系。和Javascript代码一样。

```less
//less
@border: 1px;
@baseColor: #111;
@blue: #09c;
#header {
  color: @baseColor * 3;
  border-left: @border;
  border-right: @border * 2;
}
#footer {
  color: @baseColor + #003300;
  border-color: desaturate(@blue, 10%);
}
```

```css
/* 生成的 CSS */
#header {
  color: #333;
  border-left: 1px;
  border-right: 2px;
}
#footer {
  color: #114411;
  border-color: #7d2717;
}
```

- 运算有顺序，和平时的四则运算一样

```
@var: 20px;
#header {
  width: @var + 5 * 2;/* 先计算了5 * 2 = 10 然后在计算了 @var + 10 = 30px,其实就是"@var+(5*2)"*/
  height: (@var + 5 ) * 2;/*先计算了(@var + 5) = 25px,然后在计算了25*2=50px, 因为括号更具有优先权，小学数学题*/
}
```

3.5 继承

- 继承可以将一个定义好的class A轻松引入到另个class B里面，从而简单实现class B继承class A中的所有属性。还可以带参数调用，和函数一样。
- 有默认值，也可以不加默认值，或者是不加参数、多个参数都可行。

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

```css
/* 生成的 CSS */
#header {
  border-radius: 5px;
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
}
#footer {
  border-radius: 10px;
  -webkit-border-radius: 10px;
  -moz-border-radius: 10px;
}
```

3.6 匹配模式

- 匹配模式，即使同一个函数用不同的参数时调用不同的方法。例如写一个三角：

```
.triangleBase{
  width:0;
  height:0;
  overflow:hidden;
}
.triangle(top,@w:5px,@c:#ccc){
  .triangleBase;
  border-width:@w;
  border-color:@c transparent transparent transparent;
  border-style:solid dashed dashed dashed;
}
.triangle(bottom,@w:5px,@c:#ccc){
  .triangleBase;
  border-width:@w;
  border-color:transparent transparent @c transparent;
  border-style: dashed dashed solid dashed;
}
.div1{
  .triangle(top);
}
.div2{
  .triangle(bottom);
}
```

```
.div1 {
  width: 0;
  height: 0;
  overflow: hidden;
  border-width: 5px;
  border-color: #cccccc transparent transparent transparent;
  border-style: solid dashed dashed dashed;
}
.div2 {
  width: 0;
  height: 0;
  overflow: hidden;
  border-width: 5px;
  border-color: transparent transparent #cccccc transparent;
  border-style: dashed dashed solid dashed;
}
```

- 简化上面的示例，@_默认都调用

```
.triangleBase{
  width:0;
  height:0;
  overflow:hidden;
}
.triangle(top,@w:5px,@c:#ccc){
  .triangleBase;
  border-width:@w;
  border-color:@c transparent transparent transparent;
  border-style:solid dashed dashed dashed;
}
.triangle(bottom,@w:5px,@c:#ccc){
  .triangleBase;
  border-width:@w;
  border-color:transparent transparent @c transparent;
  border-style: dashed dashed solid dashed;
}
```

```
//@_默认都调用
.triangle(@_){
  width:0;
  height:0;
  overflow:hidden;
}
.triangle(top,@w:5px,@c:#ccc){
  border-width:@w;
  border-color:@c transparent transparent transparent;
  border-style:solid dashed dashed dashed;
}
.triangle(bottom,@w:5px,@c:#ccc){
  border-width:@w;
  border-color:transparent transparent @c transparent;
  border-style: dashed dashed solid dashed;
}
```

3.7 @arguments 变量

- @arguments 变量包含所有传递进来的参数

```less
.border(@w:5px,@c:red,@s:solid){
  border:@arguments;
}
#demo6{
  div{
    margin:0 auto;
    margin-bottom:20px;
  }
  .div1{
    .border(10px);
  }
}
```

```less
#demo6 div {
  margin: 0 auto;
  margin-bottom: 20px;
}
#demo6 .div1 {
  border: 10px red solid;
}
```

3.8 字符串插入

- 字符串插入，字符串也可以用于变量中，然后通过@{name}来调用

```
@base_url : 'http://www.t1.chei.com.cn/common';
background-image: url("@{base_url}/images/background.png");
```

3.9 避免编译

- 避免编译"~"，输入一些不正确的css或者是less不认识的专有语法，例如ie的一些特殊属性。

```
.test{
  filter: ~"progid:DXImageTransform.Microsoft.Alpha(opacity=20)";
}
```

```
.test{
  filter: progid:DXImageTransform.Microsoft.Alpha(opacity=20);
}
```

3.10 !important

- !important为所有样式加上!important。

```
#demo5{
  div{
    margin:0 auto;
    margin-bottom:20px;
  }
  .div1{
    .triangle(top)!important;
  }
  .div2{
    .triangle(bottom);
  }
}
```

```
#demo5 div {
  margin: 0 auto;
  margin-bottom: 20px;
}
#demo5 .div1 {
  width: 0 !important;
  height: 0 !important;
  overflow: hidden !important;
  border-width: 5px !important;
  border-color: #ccc transparent transparent transparent !important;
  border-style: solid dashed dashed dashed !important;
}
#demo5 .div2 {
  width: 0;
  height: 0;
  overflow: hidden;
  border-width: 5px;
  border-color: transparent transparent #ccc transparent;
  border-style: dashed dashed solid dashed;
}
```

3.11 @import

- 用@import导入css或者less文件，减少服务器资源请求
  - less文件可以省略后缀名，同时可以在文件任意地方引入
  - css文件需要加入后缀名
- 用@import时，Koala编译会让软件崩溃，用node吧！

3.12 颜色函数

- 明暗度，lighten(param1,param2)浅一点，darken(param1,param2)深一点

```
@ahover:lighten(@colorBase,10%);
@afocus:darken(@colorBase,10%);
```

- 饱和度，saturate(param1,param2)和desaturate(param1,param2)。饱和度定义了一种颜色的深度。饱和度越大，颜色越亮丽，最低饱和度则会使颜色趋于灰色。

```
@btnHover:saturate(@colorBase,10%);
@btnFocus:desaturate(@colorBase,10%);
@btnDisable:lightness(desaturate(@colorBase,100%),30%;
```

总结

- 注释，可解析注释和不可解析的注释
- 变量@，通过改变一个值改变多处样式
- 嵌套，符合dom结构，最多三层
- 运算，加减乘除四则运算规律
- 继承，和js函数一样
- 匹配模式，类似函数参数
- @arguments匹配所有参数
- 字符串插入@{name}
- 避免编译”~”
- !important，为所有样式加上!important。
- @import导入less或者css文件
- 颜色函数

4、Less和Sass的对比

4.1 相同点

- 相同点
  - 两者都是css预编译
  - 很多语法类似，思想一样

4.2 不同点

- Less环境以及使用方面比Sass简单
- 条件语句与控制，less不支持。Sass可以使用if{}else{}条件语句，以及for{}循环。它甚至支持 and、 or和 not，以及 <、 >、 <=、 >= 和 == 等操作符。
- 输出格式，Less：normal（正常）、compress（压缩为一行）。而Sass提供4中输出选项：nested（正常缩进）、expanded（括号不单独占一行）、 compact（一个类一行显示）和compressed （压缩为一行）。
- less基于纯JavaScript，通过服务器端来处理的；Sass是基于Ruby的，在服务器端处理。

4.3 用哪种

- 如果你是Ruby的粉丝，那么Sass会是你的好助手。对我来说，一个前端开发人员，我倾向于LESS，因为它便于引入和能够使用JavaScript的表达式以及文档属性。
- 对于刚接触的css预处理语言的前端，我推荐用Less
- 等学会了Less，并熟练运用后，可再学习Sass

参考资料

- less相关介绍的网址
  - http://www.lesscss.net
  - http://lesscss.cn/usage/
  - http://less.bootcss.com/usage/
- 如何在浏览器中调试less & sass：http://www.cr173.com/html/20939_1.html
- color颜色函数：
  - http://www.wzsky.net/html/Website/Color/125373.html

The end , Thanks!