目录

[什么是Sass和Less？ 2](#_Toc1130969)

[Less 2](#_Toc1130970)

[less是什么 2](#_Toc1130971)

[使用方法 2](#_Toc1130972)

[1）直接引入less.js 2](#_Toc1130973)

[2）koala编译使用 4](#_Toc1130974)

[3）在服务器环境编译node npm 4](#_Toc1130975)

[4）在编辑器中实时编译 4](#_Toc1130976)

[嵌套 5](#_Toc1130977)

[变量 6](#_Toc1130978)

[变量作用域 6](#_Toc1130979)

[注释 6](#_Toc1130980)

[混合 7](#_Toc1130981)

[带参数混合 8](#_Toc1130982)

[@arguments 变量 8](#_Toc1130983)

[模式匹配 9](#_Toc1130984)

[混合-命名空间 10](#_Toc1130985)

[Importing 11](#_Toc1130986)

[避免编译 11](#_Toc1130987)

[Sass 12](#_Toc1130988)

# 什么是 CSS 预处理器呢？

Sass和Less都属于**CSS预处理器**

CSS 预处理器：它是一种新的语言，用一种专门的编程语言，为 CSS 增加了一些编程的特性，将 CSS 作为目标生成文件，然后开发者就只要使用这种语言进行CSS的编码工作。

通俗易懂的来说就是“**用一种专门的编程语言，进行 Web 页面样式设计，再通过编译器转化为正常的 CSS 文件，以供项目使用**”。

# Less

## less是什么

一种 动态 样式 语言.

LESS 将 CSS 赋予了动态语言的特性，如 变量， 继承， 运算， 函数. LESS 既可以在 客户端 上运行 (支持IE 6+, Webkit, Firefox)，也可以借助Node.js或者Rhino在服务端运行。

总结：一个写CSS的工具 更灵活的统筹全局 更方便计算

## 使用方法

### 1）直接引入less.js

优点：能获取客户端数据，从而进行进一步计算

缺点：在客户端解析Less，造成性能浪费，不利于维护

**代码形式：**

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

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

**注意**

* 你的less样式文件一定要在引入less.js前先引入
* **rel="stylesheet/less"**（**rel 属性规定当前文档与被链接文档之间的关系**）
* 请在**服务器环境下使用**！本地直接打开可能会报错！

**举例说明优点：**

@height: `document.documentElement.clientHeight`;

body{

margin: 0;

background: #ccc;

height: @height\*1px;

}

### 2）koala编译使用

每次都要打开软件编译

### 3）在服务器环境编译node npm

对于初学者可能比较生疏

* 先安装node
* 再用npm**安装全局less** npm i less -g
* 到你要进行编译的less文件目录下启动命令行工具输入 如下命令进行编译

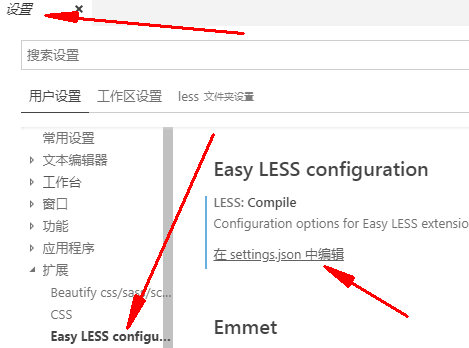
**lessc style.less > abc.css**

### 4）在编辑器中实时编译

配置稍微麻烦

**在vscode中配置**

* 首先下载Easy Less插件
* 其次找到settiings.json配置



* 在settings.json中添加如下代码

"less.compile":{

"out": "${workspaceRoot}\\"

}

## 嵌套

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 }

}

}

或者这样写:

#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.

## 变量

很容易理解:用来定义一些通用的样式，维护起来方便

@nice-blue: #5B83AD;

@light-blue: @nice-blue + #111;

#header { color: @light-blue; }

输出:

#header { color: #6c94be; }

甚至可以用变量名定义为变量:

@fnord: "I am fnord.";

@var: 'fnord';

content: @@var;

解析后:

content: "I am fnord.";

请注意 ：

LESS 中的变量为完全的 ‘常量’ ，所以只能定义一次.

格式上是  **@变量名：值；** 不能写成赋值语句

## 变量作用域

{}是一个作用域，和js中一样理解

div{

@a:10px;

p{

width:@a;

}

}

footer{

width: @a;//这里会出错

}

## 注释

在less文件中可以向在JS中添加注释一样添加注释

单行//

多行/\*\*/

注意：单行注释不会被编译到css文件中，多行注释会被编译到css文件中

## 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:

#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;

}

**备注：任何 CSS class, id 或者 元素 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; }

然后在其他class中像这样调用它:

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

.button { .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; }

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: wrap; white-space: pre-wrap; word-wrap: break-word; }

pre { .wrap }

输出:

pre {

text-wrap: wrap; white-space: pre-wrap; word-wrap: break-word;

}

### @arguments 变量

@arguments包含了所有传递进来的参数. 如果你不想单独处理每一个参数的话就可以像这样写:

.box-shadow (@x: 0, @y: 0, @blur: 1px, @color: #000) {

box-shadow: @arguments;

}

div{ .box-shadow(2px, 5px); }

将会输出:

div{ box-shadow: 2px 5px 1px #000; }

### 模式匹配

.border(top,@width){

border-top:@width solid red;

}

.border(right,@width){

border-top:@width solid red;

}

.border(bottom,@width){

border-top:@width solid red;

}

.border(left,@width){

border-top:@width solid red;

}

.border(@\_,@width){

width: 100px;

height: 200px;

}

.box1{

.border(top,1px);

}

.box2{

.border(right,2px);

}

.box3{

.border(bottom,3px);

}

.box4{

.border(left,4px);

}

第一个参数，top/right/bottom/left 为引导参数

如果第一个参数设置为@\_，将代表设置他们的共同属性

.box1 {

border-top: 1px solid red;

width: 100px;

height: 200px;

}

.box2 {

border-top: 2px solid red;

width: 100px;

height: 200px;

}

.box3 {

border-top: 3px solid red;

width: 100px;

height: 200px;

}

.box4 {

border-top: 4px solid red;

width: 100px;

height: 200px;

}

## 混合-命名空间

.header{

.button{

color:red;

}

}

.footer{

.button{

color:blue;

}

}

.main{

.header>.button

}

编译后

.header .button {

color: red;

}

.footer .button {

color: blue;

}

.main {

color: red;

}

## Importing

你可以在main文件中通过下面的形势引入 .less 文件, .less 后缀可带可不带:

@import "lib.less";

@import "lib";

如果你想导入一个CSS文件而且不想LESS对它进行处理，只需要使用.css后缀就可以:

@import "lib.css";

这样LESS就会跳过它不去处理它.

## 避免编译

@rem:30rem;

.box{

width:@rem;

font:(12/@rem)~'/'(20/@rem) '宋体';

}

这里的/符号是不要被编译的，所以 用~开头 在把/放在引号里面即可

.box {

width: 30rem;

font: 0.4rem / 0.66666667rem '宋体';

}

# Sass

## Sass和Less的比较

**不同之处：**

1、Less环境较Sass简单

Less基于JavaScript，是需要引入Less.js来处理代码输出css到浏览器

**Sass的安装需要安装Ruby环境**

2、Less使用较Sass简单

3、从功能出发，Sass较Less略强大一些

4、Less与Sass处理机制不一样

前者是通过客户端处理的，后者是通过服务端处理，相比较之下前者解析会比后

者慢一点

5、关于变量在Less和Sass中的唯一区别就是Less用@，Sass用$

**相同之处：**

1、混入(Mixins)——class中的class；

2、参数混入——可以传递参数的class，就像函数一样；

3、嵌套规则——Class中嵌套class，从而减少重复的代码；

4、运算——CSS中用上数学；

5、颜色功能——可以编辑颜色；

6、名字空间(namespace)——分组样式，从而可以被调用；

7、作用域——局部修改样式；

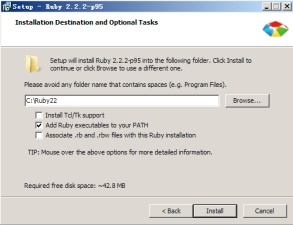
8、JavaScript 赋值——在CSS中使用JavaScript表达式赋值。

https://www.sass.hk/install/

## 安装Sass和Compass

sass基于Ruby语言开发而成，因此安装sass前需要安装Ruby。（注:mac下自带Ruby无需在安装Ruby!）

先从官网下载Ruby并安装。安装过程中请注意勾选Add Ruby executables to your PATH**添加到系统环境变量**。如下图：



**测试ruby安装有没有成功**,运行CMD输入以下命令：

ruby -v

//如安装成功会打印

ruby 2.2.2p95 (2015-04-13 revision 50295) [i386-mingw32]

**更换gem源（因为国内网络的问题导致gem源间歇性中断）**

gem sources --remove https://rubygems.org/ --add https://gems.ruby-china.com/

安装Sass和Compass。要安装最新版本的Sass和Compass，你需要输入下面的命令

gem install sass

gem install compass

**命令行编译;**

//单文件转换命令

sass input.scss output.css

//单文件监听命令

sass --watch input.scss:output.css

//如果你有很多的sass文件的目录，你也可以告诉sass监听整个目录：

sass --watch app/sass:public/stylesheets

Sass脚本有两种格式，分别是.sass和.scss，这两种格式的区别在于：.sass不使用大括号和分号。由于.scss的写法与.css更加相似，所以通常我们都使用.scss格式