# Less 的安装与使用

## Less 简介

#### 什么是 CSS 预处理

CSS 预处理是一种将 CSS 作为目标生成文件的，使用变量、函数及简单的逻辑实现更加简洁、适应性更强、可读性更好、更易于代码维护的兼容浏览器的页面样式文件。

#### 为什么要有 CSS 预处理

CSS 自诞生以来，基本语法和核心机制一直没有本质上的变化，它的发展几乎全是表现力层面上的提升。最开始 CSS 在网页中的作用只是辅助性的装饰，轻便易学是最大的需求；然而如今网站的复杂度已经不可同日而语，原生 CSS 已经让开发者力不从心。

而 CSS 预处理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编码工作。通俗的说，CSS 预处理器用一种专门的编程语言，进行 Web 页面样式设计，无需考虑浏览器的兼容性问题，然后再编译成正常的 CSS 文件，以供项目使用。

例如你可以在 CSS 中使用变量、简单的逻辑程序、函数等等在编程语言中的一些基本特性，可以让你的 CSS 更加简洁、适应性更强、可读性更佳，更易于代码的维护。

## Less 是什么

Less 是一门 CSS 预处理语言，它扩充了 CSS 语言，增加了诸如变量、混合、函数等功能，让 CSS 更易维护、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，这些规则最终会通过解析器，编译生成对应的 CSS 文件。Less 并没有裁剪 CSS 原有的特性，更不是用来取代 CSS 的，而是在现有 CSS 语法的基础上，为 CSS 加入程序式语言的特性。

## Less 安装和使用

Less 安装有以下两种方式。

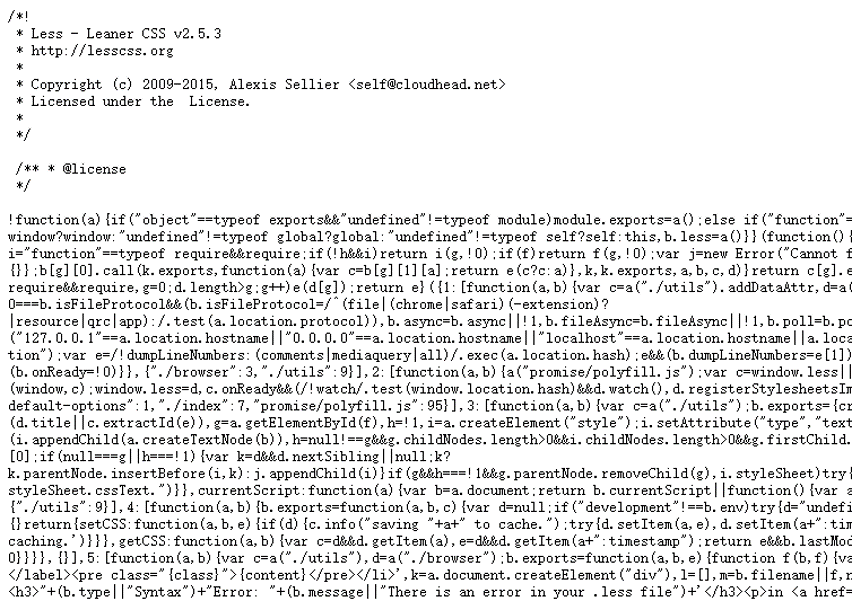
1. 第一种方式是直接在 html 页面引用 .less 文件，然后借助 less.js 去编译 less 文件动态生成 css 样式而存在于当前页面，这种方式适用于开发模式。
2. 第二种方式是我们首先写好 .less 文件的语法，然后借助工具（例如 Hbuilder , Koala , NodeJS ）生成对应的 .css 文件，然后客户端直接引用 .css 文件即可。比如我们常用的 bootstrap.css 就是通过工具编译而成，这种方式更适合运行环境。

## 下载 Less

1. 在开发模式下要成功的使用 Less，需要一个脚本的支持，这个脚本我们把它叫做 less.js。在 Less 的[中文官方网站](http://lesscss.cn/)即可下载 less 脚本。其网站界面如下图：



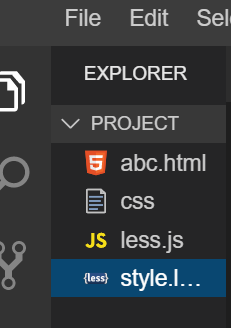
1. 当前最新版本为 v2.5.3 版本，点击下载按钮，界面跳转至代码界面，按下 Ctrl+A 全选代码，将代码 copy 至自己新建的 less.js 脚本文件中即可。如下图：

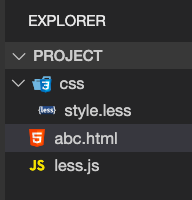


## 使用

引入 less 文件和 less.js 文件-开发模式：

步骤 1：在开发模式下，项目要成功使用 less，首先需要新建一个 less 文件，假设在 css 文件夹下创建了 style.less，如下图：





那么在 abc.html 页面的 <head></head> 引用该 less 文件。注意：设置  rel  属性值为 “stylesheet/less”。

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>less 安装和使用</title>

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

</head>

步骤 2：引入 less 文件后，需要使用 <script> 文件导入标记，将 less.js 脚本文件导入页面中。假设该文件下载后保存在项目文件夹 js 中，那么在页面的 <head></head> 中引入该文件。

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>less 安装和使用</title>

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

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

</head>

### 注意

如果不想下载 less.js 文件，也可以直接使用 CDN 的方式引用 less.js，并且 less 样式文件一定要在引入 less.js 前先引入。

<script src="//cdnjs.cloudflare.com/ajax/libs/less.js/2.5.3/less.min.js"></script>

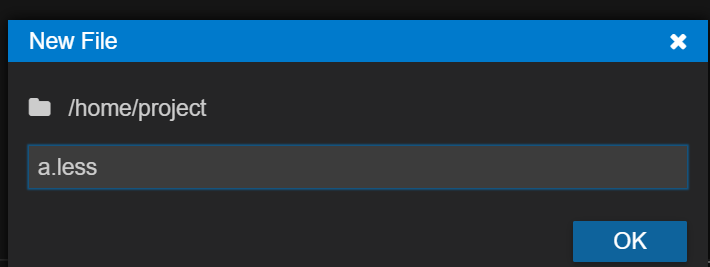
务必确保在 less.js 脚本文件之前加载 less 文件。

请在服务器环境下使用！本地直接打开可能会报错。

引入 less 文件和 less.js 文件-运行模式:

运行环境下，最好是将 .less 文件直接编译成 .css 文件，然后在 html 文件中直接引用生成的 .css 文件即可。将 .less 文件编译成 .css 文件需要借助工具（例如 Hbuilder , Koala , NodeJS），NodeJS 我们会在以后的课程中详细讲解，因此此处给大家讲解下实验楼环境下编译 less 文件的做法。

步骤 1：打开实验环境，点击 File -> New File 新建一个名为 less 的文件，如下图。



步骤 2：在 a.less 文件上写入 less 代码。

@color: #253636;

@highlight: "color";

#header {

color: @@highlight;

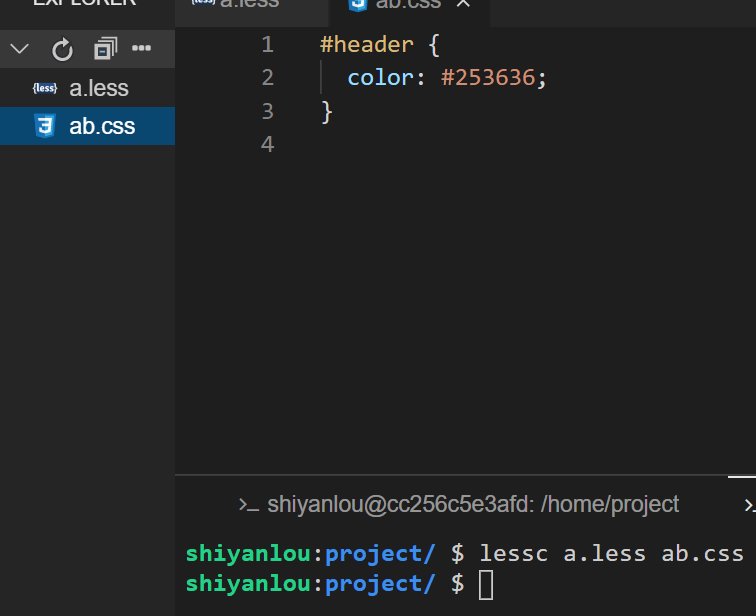
}



步骤 3：在终端输入以下命令。

lessc a.less ab.css

按下 Enter 之后会生成一个 ab.css 文件，效果如下图：



步骤 4：将编译成功的 CSS 文件引入 html 文件中即可调用 css 样式。

监视模式

* 简介：less 自带监视模式，监视模式是客户端的一个功能，这个功能允许当你改变样式的时候，客户端将自动刷新。
* 用法：可以通过在终端运行 less.watch() 来启动监视模式。

<script src="js/less-1.7.3.min.js" type="text/Javascript"></script>

<script>

less.watch();

</script>

# Less 语法

## 变量

Less 中的变量允许你在样式中的某个地方对常用的值进行定义，然后应用到样式中，这样只要改变你定义的变量参数值就可以达到改变全局的效果。

示例 1 ：打开实验环境，点击 File -> New File 新建一个名为 index1.less 的文件，在其中写入以下代码。

Less 代码：

/\*======== 定义变量===========\*/

@color: #4d926f;

/\*======== 应用到元素中 ========\*/

#header {

color: @color;

}

h2 {

color: @color;

}

在命令行输入以下命令并按下 Enter 键。

lessc index1.less index1.css

编译后会生成一个 index1.css 文件：

#header {

color: #4d926f;

}

h2 {

color: #4d926f;

}

示例 2 点击 File -> New File 新建一个名为 index2.less 的文件，在其中写入以下代码。

Less 代码：

@color: #253636;

@highlight: "color";

#header {

color: @@highlight;

}

在命令行输入以下命令并按下 Enter 键。

lessc index2.less index2.css

编译后会生成一个 index2.css 文件:

#header {

color: #253636;

}

注意：在 Less 中的变量实际上就是一个 “常量”，因为它们只能被定义一次。

## 混合用法

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。

示例 1 点击 File -> New File 新建一个名为 index3.less 的文件，在其中写入以下代码。

Less 代码：

/\*======== 有这样一个 class 类 .bordered======== \*/

.bordered {

border-top: dotted 1px black;

border-bottom: solid 2px black;

}

/\*========如果我们现在需要在其他 class 中引入. bordered 类，

那么我们只需要在任何 class 中像下面这样调用即可。======== \*/

#menu a {

color: #111;

.bordered;

}

.post a {

color: red;

.bordered;

}

在命令行输入以下命令并按下 Enter 键。

lessc index3.less index3.css

编译后会生成一个 index3.css 文件:

/\*========.bordered 类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;

}

注意：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。

示例 1 点击 File -> New File 新建一个名为 index4.less 的文件，在其中写入以下代码。

Less 代码：

/\*========定义一个带参数的.border-radius类======== \*/

.border-radius (@radius) {

border-radius: @radius;

-moz-border-radius: @radius;

-webkit-border-radius: @radius;

}

/\*========在其他class中调用======== \*/

#header {

.border-radius(4px);

}

.button {

.border-radius(6px);

}

在命令行输入以下命令并按下 Enter 键。

lessc index4.less index4.css

编译后会生成一个 index4.css 文件:

#header {

border-radius: 4px;

-moz-border-radius: 4px;

-webkit-border-radius: 4px;

}

.button {

border-radius: 6px;

-moz-border-radius: 6px;

-webkit-border-radius: 6px;

}

示例 2 点击 File -> New File 新建一个名为 index5.less 的文件，在其中写入以下代码。

Less 代码：

/\*========给参数设置默认值======== \*/

.border-radius (@radius: 4px) {

border-radius: @radius;

-moz-border-radius: @radius;

-webkit-border-radius: @radius;

}

/\*========在class中调用======== \*/

#header {

.border-radius;

}

在命令行输入以下命令并按下 Enter 键。

lessc index5.less index5.css

编译后会生成一个 index5.css 文件：

#header {

border-radius: 4px;

-moz-border-radius: 4px;

-webkit-border-radius: 4px;

}

示例 3 点击 File -> New File 新建一个名为 index6.less 的文件，在其中写入以下代码。

Less 代码：

.wrap () {

text-wrap: wrap;

white-space: pre-wrap;

white-space: -moz-pre-wrap;

word-wrap: break-word;

}

pre {

.wrap;

}

在命令行输入以下命令并按下 Enter 键。

lessc index6.less index6.css

编译后会生成一个 index6.css 文件：

pre {

text-wrap: wrap;

white-space: pre-wrap;

white-space: -moz-pre-wrap;

word-wrap: break-word;

}

示例 4 点击 File -> New File 新建一个名为 index7.less 的文件，在其中写入以下代码。

Less 代码：

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

box-shadow: @arguments;

-moz-box-shadow: @arguments;

-webkit-box-shadow: @arguments;

}

#header {

.box-shadow(2px, 5px);

}

在命令行输入以下命令并按下 Enter 键。

lessc index7.less index7.css

编译后会生成一个 index7.css 文件：

#header {

box-shadow: 2px 5px 1px #000;

-moz-box-shadow: 2px 5px 1px #000;

-webkit-box-shadow: 2px 5px 1px #000;

}

## 嵌套规则

嵌套规则主要是针对多层元素的样式规则写法，以前我们在多层元素中写样式，要么从头选下来，要么另外给这个元素加上类名或 id 名，但在 Less 中我们不需要这样操作，Less 可以让我们以嵌套的方式编写层叠样式。

示例 1 点击 File -> New File 新建一个名为 index8.less 的文件，在其中写入以下代码。

Less 代码：

#header {

color: black;

.navigation {

font-size: 12px;

}

.logo {

width: 200px;

}

}

在命令行输入以下命令并按下 Enter 键。

lessc index8.less index8.css

编译后会生成一个 index8.css 文件：

#header {

color: black;

}

#header .navigation {

font-size: 12px;

}

#header .logo {

width: 200px;

}

使用 Less 嵌套规则写的层叠样式相比传统 CSS 样式代码来说，减少了选择器的层级关系，代码更简洁了，更具有阅读性，因为他的写法就是模仿 HTML 的 DOM 结构来写的。

如果想写串联选择器，而不是写后代选择器，可以用到 “&”。这点对伪类尤其有用如 :hover 和 :focus。

示例 2 点击 File -> New File 新建一个名为 index9.less 的文件，在其中写入以下代码。

Less 代码：

.bordered {

&:hover {

text-decoration: underline;

}

.top {

background: red;

color: white;

}

}

在命令行输入以下命令并按下 Enter 键。

lessc index9.less index9.css

编译后会生成一个 index9.css 文件：

.bordered:hover {

text-decoration: underline;

}

.bordered .top {

background: red;

color: white;

}

## 运算

运算可以对元素的属性值，颜色进行四则运算：加、减、乘、除。

示例 1 点击 File -> New File 新建一个名为 index10.less 的文件，在其中写入以下代码。

Less 代码：

@the-border: 2px;

@base-color: #333;

@red: #e0171f;

#header {

color: @base-color \* 3;

border-width: @the-border @the-border\*2 @the-border\*3 @the-border;

}

在命令行输入以下命令并按下 Enter 键。

lessc index10.less index10.css

编译后会生成一个 index10.css 文件：

#header {

color: #999999;

border-width: 2px 4px 6px 2px;

}

示例 2 点击 File -> New File 新建一个名为 index11.less 的文件，在其中写入以下代码。

Less 代码：

@the-border: 2px;

@red: #e0171f;

#header {

border: (@the-border+2) \* 2 solid black;

border-color: @red;

}

在命令行输入以下命令并按下 Enter 键。

lessc index11.less index11.css

编译后会生成一个 index11.css 文件。

#header {

border: 8px solid black;

border-color: #e0171f;

}

## 作用域

Less 中的作用域跟其他编程语言非常类似，只是在 Less 中采取的是就近原则，首先会从本地查找变量或者混合模块，如果没找到的话会去父级作用域中查找，直到找到为止。

示例 1 点击 File -> New File 新建一个名为 index12.less 的文件，在其中写入以下代码。

Less 代码：

@var: black;

#page {

@var: red;

#header {

/\*========red======== \*/

color: @var;

}

}

#footer {

/\*========black======== \*/

color: @var;

}

在命令行输入以下命令并按下 Enter 键。

lessc index12.less index12.css

编译后会生成一个 index12.css 文件。

#page #header {

color: red;

}

#footer {

color: black;

}

## 函数

Color 函数：Less 提供了一系列的颜色运算函数。颜色会先被转化成 HSL 色彩空间， 然后在通道级别操作，如下图：



示例 1 点击 File -> New File 新建一个名为 index13.less 的文件，在其中写入以下代码。

Less 代码：

@base: #f04615;

#header {

color: saturate(@base, 5%);

background-color: lighten(spin(@base, 8), 25%);

}

在命令行输入以下命令并按下 Enter 键。

lessc index13.less index13.css

编译后会生成一个 index13.css 文件。

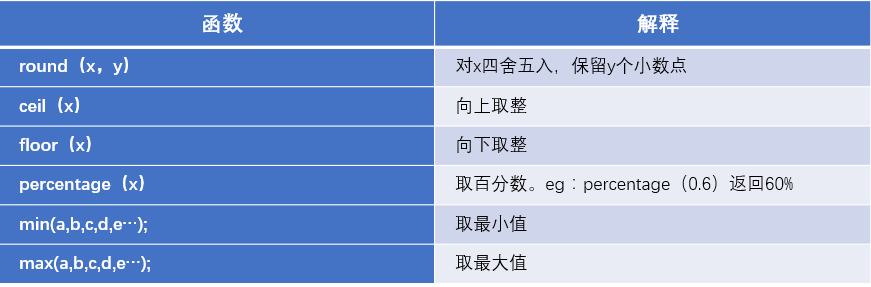
#header {

color: #f6430f;

background-color: #f8b38d;

}

Math 函数：Less 提供了一组方便的数学函数，可以使用它们处理一些数字类型的值。如下图:



## 避免编译

有时候我们需要输出一些不正确的 CSS 语法或者使用一些 Less 不认识的专有语法，要输出这样的值我们可以将要避免编译的值用 “ ” 包含起来，然后在字符串前加上一个 “ ~ ”。

示例 1 点击 File -> New File 新建一个名为 index14.less 的文件，在其中写入以下代码。

Less 代码：

@base-color: #cccccc;

.box {

width: ~"calc(300px - 30px)";

color: @base-color;

}

在命令行输入以下命令并按下 Enter 键。

lessc index14.less index14.css

编译后会生成一个 index14.css 文件。

.box {

width: calc(300px - 30px);

color: #cccccc;

}