目录

[命令行 2](#_Toc498415133)

[命令行编译 2](#_Toc498415134)

[命令行其他配置选项 3](#_Toc498415135)

[四种style生成后的css 4](#_Toc498415136)

[gui编译 5](#_Toc498415137)

[编辑器编译 5](#_Toc498415138)

[在线编译 5](#_Toc498415139)

[基本语法 6](#_Toc498415140)

[变量 6](#_Toc498415141)

[less 6](#_Toc498415142)

[变量不定 6](#_Toc498415143)

[嵌套 6](#_Toc498415144)

[属性嵌套 6](#_Toc498415145)

[选择器嵌套 7](#_Toc498415146)

[同级嵌套& 7](#_Toc498415147)

[引入@import 8](#_Toc498415148)

[变量 8](#_Toc498415149)

[!default 8](#_Toc498415150)

[差值 (  #{}  ) 8](#_Toc498415151)

[多个变量nth($,num) 9](#_Toc498415152)

[less中 9](#_Toc498415153)

[Mixins 10](#_Toc498415154)

[基本语法 10](#_Toc498415155)

[less 10](#_Toc498415156)

[参数不定 11](#_Toc498415157)

[内容快 11](#_Toc498415158)

[继承(不是嵌套) 12](#_Toc498415159)

[基本语法 12](#_Toc498415160)

[继承带%的东西 13](#_Toc498415161)

[控制指令 14](#_Toc498415162)

[if each for while 14](#_Toc498415163)

[or not 15](#_Toc498415164)

[@function 15](#_Toc498415165)

[基本语法 15](#_Toc498415166)

[字符串函数 16](#_Toc498415167)

[unquote()函数 16](#_Toc498415168)

[quote()函数 16](#_Toc498415169)

[数字函数 16](#_Toc498415170)

[percentage() 16](#_Toc498415171)

[round()函数 17](#_Toc498415172)

[ceil()函数 17](#_Toc498415173)

[floor()函数 17](#_Toc498415174)

[abs()函数 17](#_Toc498415175)

[min()函数 17](#_Toc498415176)

[max()函数 17](#_Toc498415177)

[List函数 17](#_Toc498415178)

[@import 18](#_Toc498415179)

[css 18](#_Toc498415180)

[sass 18](#_Toc498415181)

[less 19](#_Toc498415182)

[常见错误 19](#_Toc498415183)

[路径问题 19](#_Toc498415184)

[运算符 19](#_Toc498415185)

# 命令行

## 命令行编译

单文件转换命令

sass style.scss style.css

sass <要编译的Sass文件路径>/style.scss:<要输出CSS文件路径>/style.css

单文件可以没有冒号

这是对一个单文件进行编译，如果想对整个项目里所有Sass文件编译成CSS文件，可以这样操作：

sass sass/:css/

单文件监听命令

sass --watch style.scss:style.css

文件夹监听命令

sass --watch sassFileDirectory:cssFileDirectory

css文件转成sass/scss文件（在线转换工具[css2sass](http://css2sass.heroku.com/)）

sass-convert style.css style.sass

sass-convert style.css style.scss

## 命令行其他配置选项

运行命令行帮助文档，可以获得所有的配置选项

sass -h

我们一般常用的有--style，--sourcemap，--debug-info等。

sass --watch style.scss:style.css --style compact

sass --watch style.scss:style.css --sourcemap

sass --watch style.scss:style.css --style expanded --sourcemap

sass --watch style.scss:style.css --debug-info

* --style表示解析后的css是什么格式，有四种取值分别为：nested(有对应缩进)，expanded（正常样式），compact（一个一行），compressed（完全紧促）。
* --sourcemap表示开启sourcemap调试。开启sourcemap调试后，会生成一个后缀名为.css.map文件。
* --debug-info表示开启debug信息，升级到3.3.0之后因为sourcemap更高级，这个debug-info就不太用了。

### 四种style生成后的css

// nested

#main {

color: #fff;

background-color: #000; }

#main p {

width: 10em; }

.huge {

font-size: 10em;

font-weight: bold;

text-decoration: underline; }

// expanded

#main {

color: #fff;

background-color: #000;

}

#main p {

width: 10em;

}

.huge {

font-size: 10em;

font-weight: bold;

text-decoration: underline;

}

**// compact**

#main { color: #fff; background-color: #000; }

#main p { width: 10em; }

.huge { font-size: 10em; font-weight: bold; text-decoration: underline; }

**// compressed**

#main{color:#fff;background-color:#000}#main p{width:10em}.huge{font-size:10em;font-weight:bold;text-decoration:underline}

## gui编译

这里推荐[koala](http://www.w3cplus.com/sassguide/koala-app.com),它是一个优秀的免费编译器，界面清晰简洁，操作起来也非常简单，详细的大家可以移歩到：[Less/Sass编译工具，koala使用指南](http://www.w3cplus.com/blog/777.html)，简单操作如下图：



## 编辑器编译

某些高上大的编辑器本身就内置了sass的编译，如webstorm等，而对于sublime text也有相应的插件可以使用：[编译](https://github.com/jaumefontal/SASS-Build-SublimeText2" \t "_blank)，[保存即编译](https://github.com/alexnj/SublimeOnSaveBuild)。如果不清楚你的编辑器是否拥有自动编译的功能，可谷歌百度。

## 在线编译

[sassmeister](http://sassmeister.com/)提供了在线编译。

# 基本语法

## 变量

**$**linkColor : green;

.variate a {color:**$**linkColor;}

## less

1. @width : 300px;
2. .meng {width: @width;  }

### 变量不定

 **$myArguments : red, 14px, 400px, 1000px;**



 @mixin fiveMixin ($color , $size, $width, $height) {

     color:$color;

     font-size:$size;

     width:$width;

     height:$height;

 }



 div {

     @include fiveMixin(**$myArguments...**);

 }

## 嵌套

### 属性嵌套

 .AttributesNested {

     font **:**{

         size : 20px;

         family : "宋体";

         weight : bold;

     }

 }

### 选择器嵌套

 .chooserNested {

     font-size:100px;

     span {

         font-size:30px;

     }

 }

### 同级嵌套&

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。

.main {

float: left; width: 45em;

@media (max-width: 480px) {

& { float: none; width: 100%; }

} }

另一种写法

.main { float: left; width: 45em;

& { @media (max-width: 480px) {

float: none; width: 100%; } } }

高级用法

 .meng, .long {

     **div &** {

         color: black;

     }

     & + & {

         color: red;

     }

     & ~ & {

         color: red;

     }

 }

## 引入@import

@import "modules/setup";

# 变量

## !default

 $width : 10px !default;



 .meng {

     width : $width;

 }



 .long {

     $width :20px;

     width : $width;

 }

# 差值 (  #{}  )

变量不仅可以用在属性值，还可以用在选择器或者是属性上面

1. $class : meng;
3. **.#{$class}** {
4. width : 20px;
5. }

属性

 $class : height;

 $lineHeight :30px;

 $fontSize : 20px;



 .meng {

     **#{$class}**: 20px;

     font:bold #{$lineHeight}/$fontSize "Arial";

 }

## 多个变量nth($,num)

 $linkSize: 30px 20px !default;



 a{

    font-size:nth($linkSize,1);



     &:hover{

         font-size:**nth($linkSize,2)**;

     }

 }

## less中

选择器插值

 @Myname: meng1314;

 .**@**{Myname} {

     width:1000000000000px;

 }

字符串插值

 @myUrl: "http://blog.csdn.net/lee\_magnum";

 .meng {

     background-image: url("@{myUrl}/images/bg.png");

 }

用一个变量的变量值

 meng : 5201314px;



 @loveDay : meng;



 div {

     width:@@loveDay;

 }

# Mixins

## 基本语法

 **@mixin** setBorder(**$color, $width**) {

   border: {

     color: $color;

     width: $width;

     style: dashed;

   }

 }

 p {

     **@include** setBorder(blue, 1px);

 }

less的几种情况sass都可以用

### less

不带参数

 .width {

     width:1232px;

 }

 #height {

     height:2980px;

 }

 .meng {

         #height;

         .width;

     }

另一种写法

1. .width () {
2. width:1234px;
3. }
4. #height () {
5. height:5678px;
6. }
7. .long {
8. .width();
9. .meng {
10. #height();
11. .width();
12. }
13. }

**带参数**

 .width (@width) {

     width:@width;

 }

 .meng {

         #height(30em);

         .width(50%);

     }

带默认值

 .width (@width : 2345px) {

     width:@width;

 }

## 参数不定

1. @mixin boxShadow(**$shadows...**) {
2. -moz-box-shadow: $shadows;
3. -webkit-box-shadow: $shadows;
4. box-shadow: $shadows;
5. }
7. div {
8. @include boxShadow(0px 4px 5px #666, 2px 6px 10px #999);
9. }

less中使用，和；区分，；表示不同参数分开，，号可表示同一参数多个值

## 内容快

*SASS代码*

1. @mixin apply-to-ie6-only {
2. div {
3. width:300px;
4. **@content;**
5. }
6. }
7. @include apply-to-ie6-only {
8. **.meng {**
9. **color:red;**
10. **}**
11. }
12. @include apply-to-ie6-only {
13. **height:500px;**
14. }

*编译后的CSS代码*

1. div {
2. width: 300px;
3. }
4. div .meng {
5. color: red;
6. }
8. div {
9. width: 300px;
10. **height: 500px;**
11. }

# 继承(不是嵌套)

## 基本语法

1. .one {
2. width:100px;height:100px;
3. }
4. .two {
5. @extend .one;
6. background:red;border:5px solid #000;
7. }

css

 .one, .two {

   width: 100px;

   height: 100px;

 }



 .two {

   background: red;

   border: 5px solid #000;

 }

想要继承的选择器(如“.two”)和其引用的选择器(如“.one”)组成群组选择器中间用逗号隔开，组成群组选择器。

可以继承多个

可以链式继承

如果两者没有同一父类，会进行交叉，出现混乱

继承在指令中是有作用域问题的，继承是无法使在指令如(@media)之外的选择器继承的，要是继承就只能写在指令中。

1. .meng a {
2. font-weight:bold;
3. }
5. .long .link {
6. @extend a;
7. }
8. .meng a, **.meng .long .link, .long .meng .link**{
9. font-weight: bold;
10. }

## 继承带%的东西

有时候你想继承一个类名，但是并不想再在HTML中使用，就单单想写一个能够继承的类名。尤其是不想让它出现在css样式中。我们可以用占位符来写一些继承的样式(如“%one”)，然后再通过@extend继承，这样就可以防止被渲染到CSS的规则集中

1. #meng a%long {
2. color: blue;
3. font-weight: bold;
4. font-size: 2em;
5. }
7. .notice {
8. @extend %long;
9. }
10. #meng a.notice {
11. color: blue;
12. font-weight: bold;
13. font-size: 2em;
14. }

## less

1. .parentClass{
2. color:red;
3. }
4. .subClassOne{
5. &:extend(.parentClass);
6. }
7. .subClassTwo:extend(.parentClass){
9. }

# 控制指令

## if each for while

if

 p {

     @if 1 + 1 == 2 {

         width:30px;

     }@else {

         width:100px;

     }

 }

each

 @each $meng in meng1, meng2, meng3, meng4 {

   .#{$meng}-long {

     background-image: url('/images/#{$meng}.png');

   }

 }

@for $var from through 和@for $var from to 。

1. @for $i from 1 through 5 {
2. .meng#{$i} {
3. width : 2px \* $i;
4. }
5. }

while

1. $i: 6;
2. $end : -2;
4. @while $i > $end {  //或者@while ($i>$end) { 小括号可加可不加
5. .item-#{$i} { width: 2px \* $i; }
6. $i: $i - 2;
7. }

## or not

  @if $a or $b {

         width:300px;

         }

# @function

## 基本语法

1. $oneWidth: 10px;
2. $twoWidth: 40px;
4. **@function**widthFn($n) {
5. **@return**$n \* $twoWidth + ($n - 1) \* $oneWidth;
6. }
8.  .leng {
9. width: **widthFn($n : 5);**  一个变量时可以简写width: widthFn(5);
10.  }

## 字符串函数

## unquote()函数

unquote()函数主要是用来删除一个字符串中的引号，如果这个字符串没有带有引号，将返回原始的字符串。简单的使用终端来**[测试](http://lib.csdn.net/base/softwaretest" \t "_blank" \o "软件测试知识库)**这个函数的运行结果。unquote()函数只能删除字符串最前和最后的成对引号(双引号或单引号)，而无法删除字符串中间的引号。如果字符没有带引号，返回的将是字符串本身。

 .test1 {

     content:  unquote('梦龙小站!') ;

 }

 .test1 {

   content: 梦龙小站!;

 }

## quote()函数

如果字符串，自身带有引号会统一换成双引号""。使用quote()函数增加只能给字符串增加双引号，而且字符串中间有单引号或者空格时，需要用单引号或双引号括起，否则编译的时候将会报错。quote()碰到特殊符号，比如说!、?、>等，除中折号-和下划线\_都需要使用双引号括起，否则编译器在进行编译的时候同样会报错。

 .test1 {

     content:  quote('梦龙小站!');

 }

 .test1 {

   content: "梦龙小站!";

 }

# 数字函数

## percentage()

将一个不带单位的数字转换成百分比形式，如果您转换的值是一个带有单位的值，那么在编译的时候会报错误信息。

 div {

     width:percentage(.2);

 }

 div {

   width: 20%;

 }

## round()函数

一个数四舍五入为一个最接近的整数，在round()函数中可以携带单位的任何数值。

## ceil()函数

向上取整

## floor()函数

向下取整

## abs()函数

返回一个数的绝对值。

## min()函数

min()函数功能主要是在多个数之中找到最小的一个，这个函数可以设置任意多个参数。不过在min()函数中同时出现两种不同类型的单位将会报错（0.2和20%不会）

## max()函数

# List函数

length($list)、nth($list, $n)、join($list1, $list2, [$separator])、append($list1, $val, [$separator])、zip($lists…)和index($list, $value)。

# @import

## css

@import url(...)

## sass

@import "modules/setup";

css有一个不太常用的特性，即[@import](http://my.oschina.net/import" \t "_blank) 导入功能，它允许在一个css文件中导入其他css文件。然而，结果是只有执行到[@import](http://my.oschina.net/import" \t "_blank) 规则时，浏览器才会去下载其他css文件，这会导致页面样式加载特别慢，从而容易出现页面闪的问题。

ass也有[@import](http://my.oschina.net/import" \t "_blank) 导入规则，与css不同的是，sass中的[@import](http://my.oschina.net/import" \t "_blank) 规则会在生成css文件时，把相关的文件导入合并成一个文件，而无需浏览器去下载其他的文件。另外在被导入文件中定义的变量等也可以在导入文件中正常使用。

在使用[@import](http://my.oschina.net/import" \t "_blank) 导入sass文件时，可以省略sass文件的后缀名.sass或.scss，

如果你是编译整个sass目录的话，会发现一个问题，在生成style.css 的同时也生成了一个a.css ，这个结果并不是我们想要的，a.scss 作为一个中间文件，一般情况下是不需要在生成css的。sass开发者也考虑到了这点，我们**只需要在文件名前加上下划线\_ ，sass编译的时候就会忽略这个文件，@import 引用的时候可以加下划线引用，也可以不加。**还是上面的例子，

导入css文件

**当然，如果你需要像原生css那样去导入其他的css文件，也是可以的，如果符合以下三条中的任意一种情况，sass就会认为你想用css原生的@import:**

#### 被导入的文件名以.css结尾

#### 被导入的文件是一个在线的url地址

#### 以@import url(...)方式去导入文件

## less

@import './xxx' 或者 @import url('');

@import 可以根据文件扩展名不同而用不同的方式处理

如果文件是.css的扩展名，将处理为CSS和@import语句保持原样。

如果为其他的扩展名将处理为less导入。

如果没有扩展名，将会为他添加.less扩展名，作为less导入。

# 常见错误

## 路径问题

资源路径（即编译以后依然会用到的资源），一定要对应编译以后的输出文件。

## 运算符

使用运算符前后一定要有空格，不然会被认为是一个变量。比如+， - ，（，）