

<http://www.w3cplus.com/sassguide/>

在common.scss里定义一些函数，其它文件直接调用

css编译预处理：less，sass

less:偏前端：慕课网有视频，不多

```
用这两个东西 而已换个方式写css  
less视频  
http://www.imooc.com/learn/102  
http://www.imooc.com/learn/61
```

sass偏后端：需要编译环境

1:koala(停止更新);

2:命令方式：ruby

```
的方式  
ruby http://rubyinstaller.org/downloads
```

sass有两种后缀名文件：一种后缀名为sass，不使用大括号和分号；另一种就是我们这里使用的scss文件，这种和我们平时写的css文件格式差不多，使用大括号和分号。而本教程中所说的所有sass文件都指后缀名为**SCSS**的文件。

编译：//不会被编译

/* */会被编译

sass --watch style.scss:style.css --style expanded 一般用这个命令编译

命令行编译

```
--style 编译后的模式  
nested 默认 expanded css纵向写法 compact 横向 compressed 压缩(所有都压缩  
(码))
```

单文件监听

```
sass --watch common.scss:xx.css
```

文件夹监听

```
sass --wath 文件名名称 (要监听的文件夹); 编译到哪儿 (编译后的位置)
```

自行搜索 (windows cmd 命令大全)

dir 查看当前目录文件

cd 打开文件 cd a cd a.txt cd .. 上一层

cls 清除当前屏幕 I

ctrl + c 是否终止操作 y / n

按上下箭头 使用你最近使用的几个命令

语法

1 sass可以设置多址变量：取值用nth(变量，第几个)

```
$bgc:yellowgreen pink;
```

```
//background-color: nth(那个变量,第几个); I
```

```
background-color: nth($bgc,1);
```

会出现yellowgreen

2 sass定义的变量，局部变量优先级大于全局变量

3 sass可以导入新文件，如果导入sass文件，会编译后合并，如果导入css文件，不会合并

4 嵌套

```
1
//嵌套
.list{
  $lh:40px;
  li{
    height: $lh;
    line-height: $lh;
  }
  a{
    float: left;
  }
}
```

跳出嵌套 @at-root

```
.news{
  @at-root .news_title{
    width: 10px;
  }
  .news_content{
    width: 20px;
  }
}
```

则生成class

.news_title , .news .news_content

4 属性嵌套

```
//属性嵌套 I
.box{
  //复合样式
  font:{
    style:normal;
    weight:bold;
    size:14px;
    family:$ms;
  }
}
```

要用：隔开

```
25
26 .box {
27   font-style: normal;
28   font-weight: bold;
29   font-size: 14px;
30   font-family: punctuation, PingFangSC-Regu
31
32   /*# sourceMappingURL=common.css.map */
```

5

@at-root 与 & 配合使用

```
//sass style
//-----
.child{
  @at-root .parent &{
    color:#f00;
  }
}

//css style
//-----
.parent .child {
  color: #f00;
}
```

```
.box{
  width: 200px;
  &_cont{
    width: 400px;
    &_abc{
      width: 200px;
    }
  }
}
```

生成class.box_cont

6 调用用@inlude

混合声明@mixin

@mixin opa() 不写参数 直接可以调用

@mixin opa(\$opa) 调用时候 必须传参

@mixin opa(\$opa:40) 调用的时候, 如果传了就用传了的, 没传就用默认的

```
//混合 @mixin 声明
@mixin opa($opa:50){
  opacity: $opa / 100;
  filter:alpha(opacity=$opa);
}

//@inlude 是调用 @mixin 所声明的东西
.demo{
  @inlude opa(80);
}
```

.demo里可传参，也可不传，默认50

形参...表示参数个数

不确定

```
@mixin shadow($shadow...){
  box-shadow: $shadow;
}

.demo{
  @include opa(80);
  @include shadow(0 0 10px red,0 0 20px yellow);
}
```

给参数一个初始值，然后根据传入值变更样式

```
@mixin sj($fx:top,$size:10px,$color:red){
  @if ($fx == top) {
    border-color: transparent transparent $color transparent;
    border-style: dashed dashed solid dashed;
  }
  @else if($fx == right){
    border-color: transparent transparent transparent $color;
    border-style: dashed dashed dashed;
  }
  @else if($fx == bottom){
```

```
.demo{
  @include sj(bottom);
  //@include opa(80);
  //@include shadow(0 0 10px r
}
```

如果记住变量名，则可以跳过中间传参

```
.demo{
  @include sj($fx:bottom,$color:yellowgreen);
  //@include opa(80);
  //@include shadow(0 0 10px red,0 0 20px yellow);
}
```

7 继承 : extend

```
//-----  
.message {  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
  
.success {  
  @extend .message;  
  border-color: green;  
}
```

占位符%

```
//占位选择器 (在用到的时候 才会被编译出来)  
%clearfix{  
  *zoom:1;  
  &:after,  
  &:before{  
    content: '';  
    display: table;  
  }  
  &:after{  
    clear: both;  
  }  
}  
  
.abc{  
  width: 300px;  
  @extend %clearfix;  
}
```

clearfix不显示,直接出现在class:after

```
.box2, .abc { *zoom: 1; }  
.box2:after, .abc:after, .box2:before, .a  
.box2:after, .abc:after { clear: both; }
```

8:字符串插值

```

//字符串插值
$icon: '../images/icon/';
$img: '../images/';

// #[变量名]

.content{
  background: url(#{ $icon }a.jpg) left top no-repeat;
}

```

编译为：

```

.content { background: url(../images/icon/a.jpg) left top no-repeat; }

```

9 : sass运算

运算符前后要有空格

```

.y{
  color: #F00 - 100;
}

```

#f00是16进制

```

.y { color: #9b0000; }

```

避免运算，用字符串插值

```

$fs: 12px;
$lh: 24px;

font: #{ $fs } / #{ $lh } "Microsoft YaHei", "微软雅黑";

```

10 : for循环

背景图片位置案例：

\$i是变量，for a through b : <= b

for a to b : < b

```

//js的 for(var i=0;i<aaa.length;i++){
$max:10;
$min:5;
//sass
// <=
@for $i from $min through $max{
  .item_#{$i}{
    background-position: -($i - 1) * 20px 0;
  }
}
// <
@for $i from 1 to 10{
  .demo_#{$i}{
    background-position: -($i - 1) * 20px 0;
  }
}
}

```

while

出现5,4,3,2

```

$a:5;
@while($a > 1){
  .aaa_#{$a}{
    width: 20px * $a;
  }
  $a:$a - 1;
}

```

each: icon图标名

```

//each I
$icon_name:user,pass,checked,select;
@each $i in $icon_name{
  .icon_#{$i}{
    width: 10px;
  }
}

```

```

.icon_user { width: 10px; }
.icon_pass { width: 10px; }
.icon_checked { width: 10px; }
.icon_select { width: 10px; }

```

三目运算符

Sass中的其他内容

- `if($condition, $if true, $if false)`
 - sass中的三目判断
 - `if(true, 1px, 2px) => 1px`
 - `if(false, 1px, 2px) => 2px`

案例：C:\exerciseProject\xiaomi