Sass css预处理器，我也不知道自己为什么选择sass,而不是less，其实都差不多，但不管怎么样既然选择了，就开始收集资料学习吧。

<http://www.ruanyifeng.com/blog/2012/06/sass.html> -- 阮一峰

<http://www.ruby-lang.org/zh_cn/downloads/> -- 安装Ruby

1. **安装。**

Sass是Ruby语言写的，所以要先安装Ruby,然后再安装sass.

如果是window系统就直接下载RubyInstaller ，安装，就可以直接用命令行安装sass了。

安装 sass: gem install sass.

这样就可以基本上可以使用sass了。

1. **使用。**

最简单的输出使用：

Sass test.scss

这个是直接在命令行中查看输出的css样式。

Sass test.scss test.css 这个是生成一个转换后的test.css文件

1. **sass用法**

**3.1变量：**

Sass使用变量，所有变量$开头。

　　$blue : #1875e7;

　　div {

　　　color : $blue;

}

如果要将变量镶嵌到字符串中，需要写在#{}之中。

$ml:margin-left;

$left:25px;

div {

#{$ml}:$left;

}

**3.2计算功能**

Sass允许在代码中进行运算操作。

body {  
　　　　margin: (14px/2);  
　　　　top: 50px + 100px;  
　　　　right: $var \* 10%;  
　　}

**3.3选择器嵌套**

Sass允许选择器嵌套。

普通嵌套：

div h1 {  
　　　　color : red;  
　　}

高一级嵌套：

div {  
　　　　h1 {  
　　　　　　color:red;  
　　　　}  
　　}

Css属性嵌套：如broder-color属性。

p {  
　　　　border: {  
　　　　　　color: red;  
　　　　}  
　　}

在代码块中引用父元素，如a:hover伪类。

a {  
　　　　&:hover { color: #ffb3ff; }  
　　}

1. **代码重用**

**4.1继承**

Sass允许一个选择器继承另一个选择器。使用@extend 命令如：

.class1 {  
　　　　border: 1px solid #ddd;  
　　}

　　.class2 {  
　　　　@extend .class1;  
　　　　font-size:120%;  
　　}

.class2最后编译成：

.class2{border:1px solid #ddd;font-size120%;}

**4.2Mixin**

使用@mixin命令，可以定义一个代码块。

@mixin left {  
　　　　float: left;  
　　　　margin-left: 10px;  
　　}

使用@include 命令，可以调用定义好的代码块mixin。

div {  
　　　　@include left;  
　　}

mixin在定义代码块的时候，可以指定参数和缺省值(一直不明白缺省值是什么)

@mixin left($value:10px;) {  
　　　　float: left;  
　　　　margin-left: $value;  
　　}

div {  
　　　　@include left(20px);  
　　}

**4.3颜色函数**

Sass提供了一些内置的颜色函数，以便生成系列颜色。

lighten(#cc3, 10%) // #d6d65c  
　　darken(#cc3, 10%) // #a3a329  
　　grayscale(#cc3) // #808080  
　　complement(#cc3) // #33c

**4.4外部引入文件**

Sass 提供@import命令，用来插入外部文件

@import "path/filename.scss";

@import "foo.css";

1. 高级用法

5.1条件语句

Sass 提供 @if...@else 命令。

用法一：

p {  
　　　　@if 1 + 1 == 2 { border: 1px solid; }  
　　　　@if 5 < 3 { border: 2px dotted; }  
　　}

用法二：

@if lightness($color) > 30% {  
　　　　background-color: #000;  
　　} @else {  
　　　　background-color: #fff;  
　　}

5.2循环语句

Sass提供了@for命令行，可以进行循环。

　　@for $i from 1 to 10 {  
　　　　.border-#{$i} {  
　　　　　　border: #{$i}px solid blue;  
　　　　}  
　　}

@while 循环

$i: 6;

@while $i > 0 {  
 .item-#{$i} { width: 2em \* $i; }  
 $i: $i - 2;

}

@each 循环

@each $member in a, b, c, d {  
　　　　.#{$member} {  
　　　　　　background-image: url("/image/#{$member}.jpg");  
　　　　}  
　　}

5.3自定义函数

Sass允许用户编写自己的函数。

定义函数：

@funtion d($n){

$b = $n+2;

@return $b;

}

调用函数：

#s {

Width: d(5px);

}