

## SASS 和 SCSS

SASS 是 CSS3 的一个扩展，增加了规则嵌套、变量、混合、选择器继承等等。

SCSS 是 SASS 的新语法，是 Sassy CSS 的简写，是 CSS3 语法的超集，也就是说所有有效的 CSS3 样式也同样适合于 SASS。

## 创建和编译

### HBuilder

#### 1. 打开 HBuilder ，新建 scss 文件

1.scss

```
1. @charset "UTF-8";
2. $blue: blue;
3. $c: #fff;
4.
5. html,body {
6.     background: $blue;
7.     color: $c;
8. }
```

#### 2. 鼠标在该文件上右键，编译，生成 css 文件

```
1. <link rel="stylesheet" href="1.css" type="text/css" />
```

## 语法

### 1. 基本用法

#### 1 变量

SASS 允许使用变量，所有变量以 \$ 开头。

```
1. $blue : #1875e7;
2. div {
3.     color : $blue;
4. }
```

如果变量需要镶嵌在字符串之中，就必须需要写在#{ }之中。

```
1. $side : left;
2. .rounded {
3.     border-top-#{ $side }-radius: 50px;
4. }
```

## 2 计算功能

SASS允许在代码中使用算式：

```
1. $var:20;
2. body {
3.     margin: (14px/2);
4.     top: 50px + 100px;
5.     right: $var * 10%;
6. }
```

1. + 还可以做字符链接
- 2.
3. - 减法注意空格
- 4.
5. \* 如果进行乘法运算时，两个值单位相同时，只需要为一个数值提供单位即可(多个乘数中只需要一个乘数提供单位，否则报错)。
- 6.
7. / 众所周知“/”符号在 CSS 中已做为一种符号使用。因此在 Sass 中做除法运算时，直接使用“/”符号做为除号时，将不会生效，编译时既得不到我们需要的效果，也不会报错。
8. 需要给运算的外面添加一个小括号()才能执行除法运算

## 3 嵌套

SASS允许选择器嵌套。比如，下面的CSS代码：

```
1. div h1 {
2.     color : red;
3. }
```

可以写成：

```
1. div {
2.     h1 {
3.         color:red;
4.     }
5. }
```

子节点

```
1. .div1{
2.     >ul{
3.         border: 10px solid green;
4.     }
5. }
```

属性也可以嵌套：

```
1. p {
2.     border-color: red;
3. }
```

可以写成：

```
1. p {
2.     border: {
3.         color: red;
4.         width: 10px;
5.         style: dashed;
6.     }
7. }
```

注意，border后面必须加上冒号。

## 4 注释

SASS共有两种注释风格。

标准的CSS注释 `/* comment */`，会保留到编译后的文件。

单行注释 `// comment`，只保留在SASS源文件中，编译后被省略。

## 2. 代码的重用

### 1 继承

SASS允许一个选择器，继承另一个选择器。比如，现有class1：

```
1. .class1 {
2.     border: 1px solid #ddd;
3. }
```

class2要继承class1，就要使用@extend命令：

```
1. .class2 {
2.     @extend .class1;
3.     font-size:120%;
4. }
```

## 2 Mixin

Mixin有点像C语言的宏（macro），是可以重用的代码块。

使用@mixin命令，定义一个代码块。

```
1. @mixin left {  
2.     float: left;  
3.     margin-left: 10px;  
4. }
```

使用@include命令，调用这个mixin。

```
1. div {  
2.     @include left;  
3. }
```

mixin的强大之处，在于可以指定参数和缺省值。

```
1. @mixin left($value: 10px) {  
2.     float: left;  
3.     margin-right: $value;  
4. }
```

使用的时候，根据需要加入参数：

```
1. div {  
2.     @include left(20px);  
3. }
```

## 3. 高级用法

### 1 条件语句

@if可以用来判断：

```
1. p {  
2.     @if 1 + 1 == 2 { border: 1px solid red; }  
3.     @if 5 < 3 { border: 2px dotted blue; }  
4. }
```

配套的还有@else命令：

```

1.  @if 1 > 2 {
2.      .r2:before{
3.          content: 'aaa';
4.      }
5.  } @else {
6.      .r2:before{
7.          content: 'bbb';
8.      }
9.  }

```

## 2 循环语句

SASS支持for循环：

```

1.  @for $i from 1 to 10 {
2.      .border-#{ $i } {
3.          border: #{ $i }px solid blue;
4.      }
5.  }

```

也支持while循环：

```

1.  $i: 6;
2.  @while $i > 0 {
3.      .item-#{ $i } { width: 2em * $i; }
4.      $i: $i - 2;
5.  }

```

each命令，作用与for类似：

```

1.  @each $member in a, b, c, d {
2.      .#{ $member } {
3.          background-image: url("/image/#{ $member }.jpg");
4.      }
5.  }

```

## 3 自定义函数

SASS允许用户编写自己的函数。

```

1.  @function double($n) {
2.      @return $n * 2;
3.  }
4.  #sidebar {
5.      width: double(5px);
6.  }

```

(完)