

## Sass学习

### 1: 初认识

#### 1.1. 安装

```
gem install sass
```

#### 1.2. 使用

`sass test.scss`                      屏幕上显示.scss文件转化的css代码

`sass test.scss test.css`        显示结果保存成文件

#### 1.3. SASS提供了四个编译风格的选项

- \* nested: 嵌套缩进的css代码，它是默认值。
- \* expanded: 没有缩进的、扩展的css代码。
- \* compact: 简洁格式的css代码。
- \* compressed: 压缩后的css代码。

#### 1.4. SASS监听文件或目录

```
// watch a file
sass --watch input.scss:output.css
// watch a directory
sass --watch app/sass:public/stylesheets
```

## 2: 基本用法

### 2.1使用变量，以\$开头

```
$blue : #1875e7;

div {
  color : $blue;
}
```

变量若放在字符串中，需要写上#{ } 中

```
$side : left;

.rounded {
  border-#{ $side }-radius: 5px;
}
```

### 2.2计算功能

代码中可使用算式

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

## 2.3 嵌套

```
div h1 {
    color : red;
}
```

可以写成

```
div {
    hi {
        color:red;
    }
}
```

属性嵌套：属性后面需要加上冒号

```
p {
    border: {
        color: red;
    }
}
```

元素嵌套：使用&引用父元素

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

## 2.4 注释

- /\* \*/: 标准css注释，编译后保留
- //: 单行注释，编译省略
- /\*! 重要注释! \*/ :重要注释，即使是压缩模式编译，也会保留这行注释，通常可以用于声明版权信息。

## 3: 代码重用

### 3.1 继承

允许一个选择器，继承另一个选择器

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

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

## 2. Mixin

为了可以重用代码块。

使用@mixin命令，定义一个代码块。

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

使用@include命令，调用这个mixin。

```
div {  
    @include left;  
}
```

mixin强大之处，在于可以指定参数和缺省值

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

使用的时候，也需要加入参数

```
div {  
    @include left(20px);  
}
```

mixin实例，用来生成浏览器前缀

```
@mixin rounded($vert, $horz, $radius: 10px) {  
    border-#{ $vert }-#{ $horz }-radius: $radius;  
    -moz-border-radius-#{ $vert }#{ $horz }: $radius;  
    -webkit-border-#{ $vert }-#{ $horz }-radius: $radius;  
}
```

使用时，这样调用

```
#navbar li { @include rounded(top, left); }

#footer { @include rounded(top, left, 5px); }
```

### 3.3 颜色函数

Sass提供了一些内置颜色函数，以便生成系列颜色。

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

### 3.4 插入文件

@import命令，用来插入外部文件，等同于css的import命令。

sass局部文件的文件名以下划线开头，避免编译无用的文件。

```
@import "path/filename.scss";
```

### 3.5 ! default

类似! important，只不过这个是使用于变量的。

## 4: 高级用法

### 4.1 条件语句

@if可以用来判断

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

@else命令

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

### 4.2 循环语句

Sass支持的for循环

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

Sass支持的while循环

```
$i: 6;

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

Sass支持的循环，类似For循环

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

#### 4.3自定义函数

```
@function double($n) {
  @return $n * 2;
}

#sidebar {
  width: double(5px);
}
```

#### 5: Sass结构体系（7-1模式）

```
sass/
|
| - base/
|   | - _reset.scss      # Reset/normalize
|   | - _typography.scss # Typography rules
|   ...                  # Etc...
|
| - components/
|   | - _buttons.scss    # Buttons
|   | - _carousel.scss   # Carousel
|   | - _cover.scss      # Cover
|   | - _dropdown.scss   # Dropdown
|   | - _navigation.scss # Navigation
|   ...                  # Etc...
|
| - helpers/
|   | - _variables.scss  # Sass Variables
|   | - _functions.scss  # Sass Functions
|   | - _mixins.scss     # Sass Mixins
|   | - _helpers.scss    # Class & placeholders helpers
|   ...                  # Etc...
|
| - layout/
|   | - _grid.scss       # Grid system
|   | - _header.scss     # Header
|   | - _footer.scss     # Footer
|   | - _sidebar.scss    # Sidebar
|   | - _forms.scss      # Forms
|   ...                  # Etc...
|
| - pages/
|   | - _home.scss       # Home specific styles
|   | - _contact.scss    # Contact specific styles
|   ...                  # Etc...
|
| - themes/
|   | - _theme.scss      # Default theme
|   | - _admin.scss      # Admin theme
|   ...                  # Etc...
|
| - vendors/
|   | - _bootstrap.scss  # Bootstrap
|   | - _jquery-ui.scss  # jQuery UI
|   ...                  # Etc...
|
| - main.scss            # primary Sass file
```

- *base/*
- *\_reset.scss*
- *\_typography.scss*
- *\_color.scss*
- *components/*
- *\_buttons.scss*
- *\_navigation.scss*
- *\_gallery.scss*
- *layout/*
- *\_header.scss*
- *\_grid.scss*
- *\_sidebar.scss*
- *pages/*
- *\_home.scss*
- *\_about.scss*
- *\_contact.scss*
- *themes/*
- *\_theme.scss*
- *\_admin.scss*
- *helpers/ (or utils/)*
- *\_variables.scss*
- *\_functions.scss*
- *\_mixins.scss*
- *vendors/*
- *\_bootstrap.scss*
- *\_jquery-ui.scss*
- *main.scss*

`base` 目录存放引用内容。它存放着网站每个网页都应该接受的样式。

`components` 目录存放所有小的布局文件。样式所需要的按钮、导航和类似的文章组件。

大的布局文件则应该放在 `layouts` 目录下。主要部分的布局样式，比如说头部、尾部或者表格系统样式则应该属于这个目录。

如果网站中有些独立的页面有特殊样式，你可以将他们放在 `pages` 目录中。举个常见的例子，网站主页会引用而其他页面不接受的特定样式。

`themes` 目录存放项目特定主题的样式文件。比如，网站一部分会使用三原色的配色，而另一部分则使用的是基于非色彩的土色调。

`helper` 或者 `utils` 目录存放Sass工具，帮助文件，变量以及配置文件。这些文件将不会被编译。

最后 `vendors` 目录存放第三方代码，`main.css` 文件使用 `@import` 语句包含其他的文件。

对于7-1模式的补充资源

<https://sass-guidelin.es/#the-7-1-pattern>

<https://scotch.io/tutorials/aesthetic-sass-1-architecture-and-style-organization>