

一、什么是Sass?

Sass 是一种 CSS 的预编译语言。它提供了 变量、嵌套、混合 (mixins)、函数 等功能，并且完全兼容 CSS 语法。Sass 能够帮助复杂的样式表更有条理，并且易于在项目内部或跨项目共享设计。

文件后缀为 .scss 或者 .sass，一般使用 .scss。

类似的css预处理有less。

二、安装

1. vscode下载插件“easy sass”
2. 在用户设置中加入一下代码：`"easysass.targetDir": "./css"`

三、基本语法

1. 变量与插值与法

- a. SCSS允许使用“\$”声明变量

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

- b. 如果变量需要镶嵌在字符串之中，就必须需要写在插值语句“#{ }”之中

```
1 $side: left;
2 .rounded {
3   border-#{ $side }-radius: 5px;
4 }
```

2. 计算功能：SCSS允许在代码中使用算式

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

3. 嵌套：SCSS允许选择器嵌套

```
1 div {
2   h1 {
3     color : red;
4   }
5 }
6 // 属性也可以嵌套
7 p{
8   border:{
9     color:red;
10  }
11  //注意 : 属性嵌套后面需要加上冒号。
12 }
13 // 在嵌套的代码块内，可以使用 & 引用父元素
14 a{
15   &:hover{
16     color:#ffb3ff;
17   }
18 }
```

4. 注释

- a. /* 多行注释 */，会保留到编译后的文件。
- b. // 单行注释，只保留在SASS源文件中，编译后被省略。
- c. 在/*后面加一个感叹号，表示这是“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。

```
  /*!  
    重要注释!  
  */
```

四、代码的重用

1. 占位符%

```
1 // SCSS允许通过 % 来声明一个选择器，但必须通过 @extend 指令调用  
2 %wh{  
3     width:100px;  
4     height:100px;  
5 }  
6  
7 div{  
8     @extend wh;  
9 }
```

2. 继承 @extend

```
1 // SCSS允许一个选择器，继承另一个选择器  
2 .class1 {  
3     border: 1px solid #ddd;  
4 }  
5 // class2要继承class1，就要使用 @extend 命令  
6 .class2 {  
7     @extend .class1;  
8 }
```

3. 混合宏 @mixin

```
1 // 使用 @mixin 命令，定义一个可以重用的代码块。  
2 @mixin left{  
3     float: left;  
4     margin-left:10px;  
5 }  
6 // 使用 @include 命令，调用这个混合宏。  
7 div{  
8     @include left;  
9 }  
10 // @mixin的强大之处，在于可以指定参数和默认值。  
11 @mixin common($w,$h,$bg:pink) {  
12     width:$w;  
13     height:$h;  
14     background:$bg;  
15 }  
16 // 使用 的时候，根据需要加入参数：  
17 div{  
18     @include common(100px,100px,red)  
19 }
```

4. 插入文件

```
1 // @import== 命令，用来插入外部文件。  
2 @import "reset.scss";
```

五、高级用法

1. 循环语句

```
1 // @for $var from through  含头含尾
```

```
2 @for $i from 1 through 10{
3   .border-#{ $i }{
4     border:#{ $i }px solid blue;
5   }
6 }
7 // @for $var from to 含头不含尾
8 @for $i from 1 to 4 {
9   .list li {
10     &:nth-child(#{ $i }) {
11       font-size: #{ $i }0px;
12     }
13   }
14 }
```

2. 自定义函数

```
1 // @function 用于声明函数 @return 用于返回值
2 // 先声明后使用
3 @function double($n) {
4   @return $n * 2;
5 }
6
7 #sidebar {
8   width: double(5px);
9 }
```