



# Sass总结笔记 基础入门 (超级直观细节)

原创

北极光之夜。

2021-01-27 17:01:38

5493

★ 收藏 37

版权

分类专栏:

前端大杂烩

前端核心技术总结

文章标签:

Sass笔记

sass总结

前端

css



前端大杂烩

同时被 2 个专栏收录 ▾

14 订阅

103 篇文章

订阅专栏

## 目录

一.什么是sass:

二. 安装:

三.编译.scss文件为.css文件:

四.基础用法与功能:

1. **\*\*变量\*\*** (定义变量后, 在选择器里可以直接引用) :
2. **\*\*嵌套\*\*** (父选择器里可以嵌套子选择器)
3. **\*\*mixin 混合\*\*** (相当于预先写好了一组样式, 其它地方直接引用) :
4. **\*\*继承/扩展\*\*** (一个选择器可以继承另一个选择器的全部样式)
5. **\*\*@import\*\*** 引入一个.scss后缀的文件作为自己的一部分, 被引入的那个文件并不会转换成.css格式的, 所以此文件命名要注意以下划线开头, 如
6. **\*\*计算\*\***



北极光之夜。

关注

12

17

37



专

7. **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。)
8. **Interpolation** 把一个值插入到另一个值, 具体用法如下 #{变量} 如:
10. **for循环**
11. **列表循环**, 能循环一遍一个列表的值, 列表相当于数组;
12. **while循环**, 有判断条件更灵活。
13. **自定义函数 function**, 自己定义的函数可以调用;

总结:

## 一.什么是sass:

1. Sass 扩展了 CSS3, 增加了规则、变量、混入、选择器、继承等等特性。Sass 生成良好格式化的 CSS 代码, 易于组织和维护。
2. SASS是对CSS3(层叠样式表)的语法的一种扩充, 它可以使用巢状、混入、选择子继承等功能, 可以更有效有弹性的写出Stylesheet。Sass最后还是会编译出合法的CSS让浏览可以使用, 也就是说它本身的语法并不太容易让浏览器识别(虽然它和CSS的语法非常的像, 几乎一样), 因为它不是标准的CSS格式, 在它的语法内部可以使用动态变量等, 所以它更像一种极简单的动态语言。
3. 工程越大, css文件越大, 重复代码越来越多, 会变得难以维护, 借助Sass可以提高写css的效率。

## 二. 安装:

cmd打开本地命令控制窗口, 输入下面字符串然后回车就装好了。

```
1 | npm install -g sass
```



北极光之夜。

关注

👍 12

💬 17

★ 37



```
命令提示符
Microsoft Windows [版本 10.0.18363.752]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\夜>npm install -g sass
C:\Users\夜\AppData\Roaming\npm\sass -> C:\Users\夜\AppData\Roaming\npm\sass
```

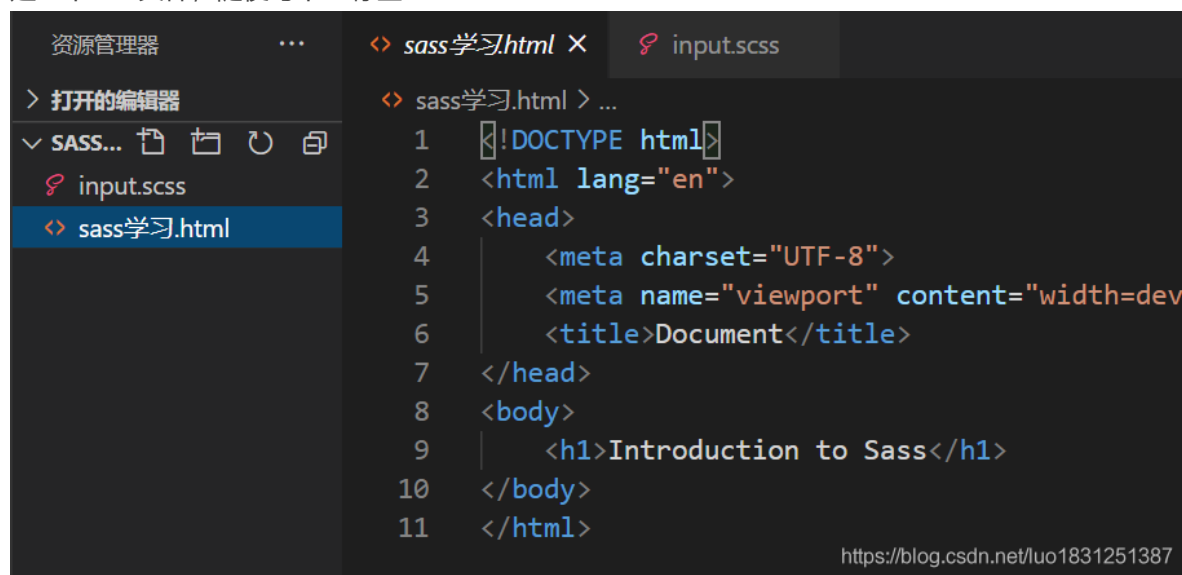
更多安装方法可以点我Sass去官网。

### 三.编译.scss文件为.css文件:

Sass使用.scss作为文件后缀名, 不能直接在< link >标签里使用, 需要编译为 .css文件。

演示:

1. 建一个html文件, 随便写个h1标签:



```
资源管理器  ...
> 打开的编辑器
SASS...  新建  打开  保存  撤销  重做
input.scss
sass学习.html

sass学习.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title>Document</title>
7  </head>
8  <body>
9      <h1>Introduction to Sass</h1>
10 </body>
11 </html>
```

2. 建一个.scss后缀的文件, 如input.scss, 写点基本样式sass的语法:



北极光之夜。

关注

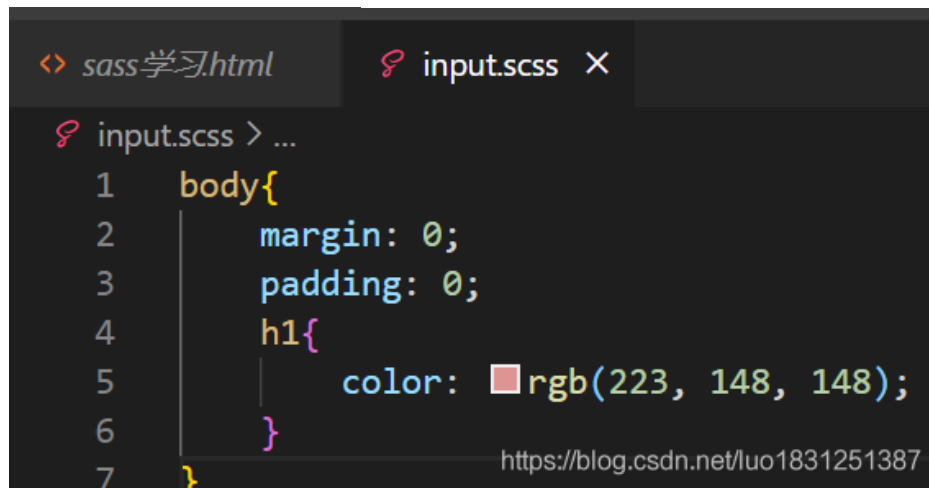
12

17

37



专

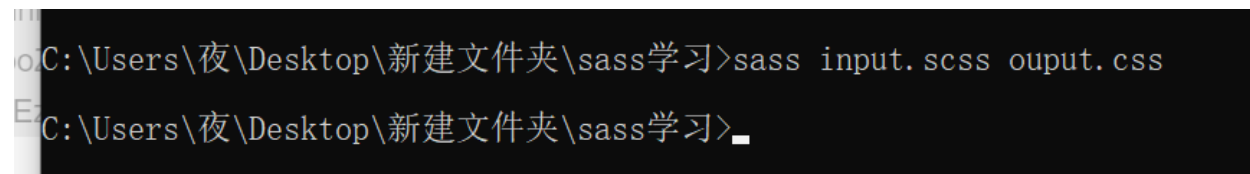


```
input.scss > ...
1  body{
2      margin: 0;
3      padding: 0;
4      h1{
5          color: rgb(223, 148, 148);
6      }
7  }
```

3. 在html文件所在的路径下打开cmd命令控制符，输入：

```
1 | sass input.scss ouput.css
```

表示把名字为 input.scss 转换成名字为 ouput.css 的文件（名字自己随意起）。



```
C:\Users\夜\Desktop\新建文件夹\sass学习>sass input.scss ouput.css
C:\Users\夜\Desktop\新建文件夹\sass学习>_
```

回车后，接下来发现就得到了css的文件。



北极光之夜。





关注

12

17

37



新建文件夹 > sass学习		^	↺
名称		修改日期	
 input.scss		2021/1/26 1	
 ouput.css		2021/1/26 1	
 ouput.css.map		2021/1/26 1	
 sass学习.html		2021/1/26 1	

<https://blog.csdn.net/luo1831251387>

css文件内容如下，可以看出转换好了：

```
文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) 终端(T) 帮助(H) ouput.css - sa
资源管理器 ... input.scss # ouput.css X
> 打开的编辑器
SASS学习
input.scss
# ouput.css
# ouput.css.map
sass学习.html
# ouput.css > body
1 body {
2   margin: 0;
3   padding: 0;
4 }
5 body h1 {
6   color: #df9494;
7 }
8
9 /*# sourceMappingURL=ouput.css.map */
10
```

<https://blog.csdn.net/luo1831251387>

接下来就是老操作了，在HTML里用 < link >标签把css文件引入就行。

4. 但是不可能说写一句.scss语句就转换一次，太麻烦，所以可以自动转换，当我在.scss里写一句，.css就自动生成一句。在cmd输入以下：

```
1 | sass --watch input.scss:ouput.css
```



北极光之夜。

关注

12

17

37



专

表示监视变化，当input.scss一变化，output.css就变化:

```
C:\Users\夜\Desktop\新建文件夹\sass学习>sass --watch input.scss:output.css
```

5. 如果一个html文件里有多css文件呢？那么可以直接监视文件夹变化：  
如：

```
1 | sass --watch yuan:bian
```

表示：当名字为yuan这个文件夹里任意一个.scss后缀的文件变化时，就将其编译到名字bian这个文件夹里面  
(会自动生成相应的.css文件)

## 四.基础用法与功能：

目录：

- 1.变量；
- 2.嵌套；
- 3.mixin；
- 4.继承；
5. import；
6. 计算；
7. 颜色函数；
8. Interpolation；
9. if 判断；
10. for循环；
11. 列表循环；
12. while循环；
- 13.function自定义函数；
- ...待更新...

详细用法：

1. 变量（定义变量后，在选择器里可以直接引用）

如：我定义一个变量名为 \$yanse，值为颜色rgb



极光之夜。

关注

👍 12

💬 17

★ 37



```
1 | $yanse: rgb(223, 148, 148);
```

直接在选择器引用：

```
1 | h1{
2 |     color: $yanse;
3 | }
```

当然变量也可以套娃，比如我定义一个变量名为 \$kuang，里面引用了 \$yanse

```
1 | $yanse: rgb(223, 148, 148);
2 | $kuang: 1px solid $yanse;
```

然后也直接用：

```
1 | h1{
2 |     border: $kuang;
3 | }
```

## 2. 嵌套（父选择器里可以嵌套子选择器）

如：有以下标签

```
1 | <div>
2 |     <ul>
3 |         <li></li>
4 |     </ul>
5 | </div>
```

可以直接这样写：

```
1 | div{
2 |     height: 100px;
3 |     ul{
4 |         height: 80px;
5 |         li{
```



北极光之夜。

关注

👍 12

💬 17

★ 37



专

```
6 |         height: 50px;
7 |     }
8 | }
9 | }
```

相当于:

```
1 | div {
2 |     height: 100px;
3 | }
4 | div ul {
5 |     height: 80px;
6 | }
7 | div ul li {
8 |     height: 50px;
9 | }
```

若 li 有伪类选择器: hover可以这样写 (里面添加&: hover{}) :

```
1 | div{
2 |     height: 100px;
3 |     ul{
4 |         height: 80px;
5 |         li{
6 |             height: 50px;
7 |             &:hover {
8 |                 color: #000;
9 |             }
10 |     }
11 | }
12 | }
```

相当于又写了:

```
1 | div ul li:hover {
2 |     color: #000;
3 | }
```



北极光之夜。

关注

👍 12

💬 17

★ 37





不止选择器，属性里也可以嵌套，如这样写：

```
1  div{
2      height: 100px;
3
4      font: {
5          family: 'fangsong';
6          size: 20px;
7          weight: 700;
8      }
9      border: 1px solid red {
10         left: 0;
11         top: 0;
12     }
13 }
```

相当于：

```
1  div {
2      height: 100px;
3      font-family: "fangsong";
4      font-size: 20px;
5      font-weight: 700;
6      border: 1px solid red;
7      border-left: 0;
8      border-top: 0;
9  }
```

### 3. mixin 混合（相当于预先写好了一组样式，其它地方直接引用）：

基本语法：

```
1  @mixin 名字（参数1，参数2，...）
2  {
3      .....样式.....
4
5  }
```



北极光之夜。

关注

👍 12

💬 17

★ 37



如（无参数的，里面也可以嵌套，下面定义了一个名字为hunhe的mixin，然后在div这个选择器里通过（@include 名字）调用）：

```
1 |
2 | @mixin hunhe {
3 |     color: red;
4 |     a {
5 |         font-size: 12px;
6 |     }
7 | }
8 |
9 | div{
10 |     @include hunhe;
11 | }
```

相当于：

```
1 | div {
2 |     color: red;
3 | }
4 | div a {
5 |     font-size: 12px;
6 | }
```

有参数的（更灵活，参数相当于你要的数值，参数名前面要写\$，调用时值的位置要对）：

如：

```
1 | @mixin hunhe($one,$two) {
2 |     color: $one;
3 |     a {
4 |         color: $one;
5 |         font-size: $two;
6 |     }
7 | }
8 |
9 | div{
10 | }
```



北极光之夜。

关注

👍 12

💬 17

★ 37



```
11 | @include hunhe(red,15px);  
    | }
```

\*div也可以这样写,指定参数名, 参数位置就可以随意变换:

```
1 | div{  
2 |   @include hunhe($two:15px,$one:red);  
3 | }
```

上面相当于:

```
1 | div {  
2 |   color: red;  
3 | }  
4 | div a {  
5 |   color: red;  
6 |   font-size: 15px;  
7 | }
```

#### 4. 继承/扩展 (一个选择器可以继承另一个选择器的全部样式)

如: .two类里继承了.one类的全部样式 (@extend 名字); 还不止.one的, 跟.one相关的都继承了, 具体如下:

```
1 | .one{  
2 |   color: #000;  
3 | }  
4 | .one a{  
5 |   font-size: 10px;  
6 | }  
7 | .two{  
8 |   @extend .one;  
9 |   background-color: #fff;  
10 | }
```

相当于:



北极光之夜。

关注



12



17



37



专

```
1 | .one, .two {
2 |     color: #000;
3 | }
4 |
5 | .one a, .two a {
6 |     font-size: 10px;
7 | }
8 |
9 | .two {
10 |     background-color: #fff;
11 | }
```

5. @import 引入一个.scss后缀的文件作为自己的一部分，被引入的那个文件并不会转换成.css格式的，所以此文件命名要注意以下划线开头，如：\_base.scss，引入它的时候不用写下划线。

语法：@import: "...路径";

如：

建立一个文件叫 \_base.scss，里面写上一些选择器和样式，然后在一个正常的如one.scss文件里引入它,假如目录是同等级的：

```
1 | @import: "base.scss";
```

这样.one.scss就有了\_base.scss里的全部内容。

6. 计算功能（SASS允许在代码中使用算式）如：

```
1 | $chang: 20px;
2 | body{
3 |     margin: (10px*2);
4 |     left: 20px + $chang;
5 | }
```

相当于：

```
1 | body {
2 |     margin: 20px;
3 | }
```



北极光之夜。

关注

👍 12

💬 17

★ 37



```
4 | left: 40px;
   | }
```

## 7. 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。)

hsl (色相, 饱和度, 明度) ;

```
1 | body{
2 |   background-color: hsl(5, 20%, 20%);
3 | }
```

相当于:

```
1 | body {
2 |   background-color: #3d2b29;
3 | }
```

hsl (色相, 饱和度, 明度, 不透明度) ;

```
1 | body{
2 |   background-color: hsl(5, 20%, 20%,0.5);
3 | }
```

相当于:

```
1 | body {
2 |   background-color: rgba(61, 43, 41, 0.5);
3 | }
```

用来调整色相: adjust-hue(颜色, 调整的度数);

```
1 | body{
2 |   background-color: adjust-hue(hsl(0,100,50%), 100deg);
3 | }
```

相当于:



北极光之夜。

关注

👍 12

💬 17

★ 37



```
1 | body {
2 |   background-color: #55ff00;
3 | }
```

用来调整颜色明度: **lighten**让颜色更亮, **darken**让颜色更暗

如: lighten (颜色, 增加亮度的百分比) ;

```
1 | body{
2 |   background-color: lighten(rgb(228, 145, 145),50%);
3 |   color: darken(rgb(228, 145, 145),50%);
4 | }
```

相当于:

```
1 | body {
2 |   background-color: white;
3 |   color: #5f1717;
4 | }
```

用来调整颜色纯度 **saturate**让颜色更纯, **desaturate**让颜色不纯

saturate (颜色, 百分比) ;

**8. Interpolation** 把一个值插入到另一个值, 具体用法如下 **#{变量}** 如:

```
1 | $yanse: color;
2 | body{
3 |
4 |   #{$yanse}:red;
5 |
6 | }
```

相当于:

```
1 | body {
2 |   color: red;
3 | }
```



北极光之夜。

关注

👍 12

💬 17

★ 37



## 9. if 判断 (逻辑跟C语言差不多) :

语法:

```
1  @if 判断条件 {
2  .....执行语句...
3  } @else {
4  ...else有就写没就不写....
5  }
6
```

如:

```
1  $jia: false;
2  body{
3
4      @if false{
5          color: red;
6      }
7      @if 2>3 {
8          background-color: white;
9      }@else{
10         background-color: black;
11     }
12
13 }
```

相当于:

```
1  body {
2      background-color: black;
3  }
```

## 10. for循环

语法:



北极光之夜。

关注

👍 12

💬 17

★ 37



```
1 | 结束值不执行:
2 | @for 变量 from 开始值 through 结束值 {
3 |     .....
4 | }
5 | 结束值也执行:
6 | @for 变量 from 开始值 to 结束值 {
7 |     .....
8 | }
```

例子:

```
1 | @for $i from 1 to 3 {
2 |
3 |     .div#{$i}{
4 |         height: $i*20px;
5 |     }
6 |
7 | }
```

相当于:

```
1 | .div1 {
2 |     height: 20px;
3 | }
4 |
5 | .div2 {
6 |     height: 40px;
7 | }
```

**11. 列表循环，能循环一遍一个列表的值，列表相当于数组；**

语法:

```
1 | @each 变量 in 列表{
2 | ...
3 | }
```



北极光之夜。

关注

👍 12

💬 17

★ 37





例子:

```
1 $yanse: red blue black;
2 @each $i in $yanse {
3
4     .div#{$i}{
5         color: $i;
6     }
7
8 }
```

相当于:

```
1 .divred {
2     color: red;
3 }
4
5 .divblue {
6     color: blue;
7 }
8
9 .divblack {
10    color: black;
11 }
```

## 12. while循环, 有判断条件更灵活。

语法:

```
1 @while 条件 {
2     ...
3 }
```

例子:

```
1 $gao: 1;
2 @while $gao<4 {
```



北极光之夜。

关注

👍 12

💬 17

★ 37



```
3 | .div#{$gao}{  
4 |     height: $gao*10px;  
5 | }  
6 | $gao : $gao+1;  
7 | }
```

相当于:

```
1 | .div1 {  
2 |     height: 10px;  
3 | }  
4 |  
5 | .div2 {  
6 |     height: 20px;  
7 | }  
8 |  
9 | .div3 {  
10 |     height: 30px;  
11 | }
```

### 13.自定义函数 function,自己定义的函数可以调用;

语法:

```
1 | @function 名字(参数1, 参数2, ..){  
2 | ....  
3 | }
```

例子:

```
1 | @function ziji ($bian)  
2 | {  
3 |     @return $bian+10px;  
4 | }  
5 |  
6 | div{  
7 |     width: ziji(5px);  
8 | }
```



北极光之夜。

关注

👍 12

💬 17

★ 37



专

相当于：

```
1 | div {  
2 |   width: 15px;  
3 | }
```

总结：

注意：这篇文章我还会一直补充和更新内容~

有帮助的话就点个赞吧~

其它文章：

- 响应式卡片悬停效果 html+css
- 水波加载动画 html+css
- 导航栏滚动渐变效果 html+css+js
- 书本翻页 html+css
- 3D立体相册 html+css
- 炫彩流光按钮 html+css
- 等等等...

javascript笔记 超级详细 入门 12-16

学习笔记，真的是好东西啊，初学者必备，不看回后悔的哦。

jquery学习总结（超级详细） 10-25


本文仅针对jquery的部分知识点做总结，更为全面的可以去官网看中文文档。可以更为详细的了解jquery及其特性。



优质评论可以帮助作者获得更高权重



评论

 strive\_day: 文章很棒，谢谢分享，期待下一期继续 9月前 回复 ...

👍 1

 北极光之夜。 博主 回复： 感谢 9月前 回复 ...

👍 1

 前端·卡卡西: 偷偷学前端，然后惊艳所有人，哈哈

 北极光之夜。 关注

-  极光之夜。 博主 回复： 哈哈~大佬666 9月前 回复 ...

 1
-  Co\_Co\_爸： 大佬出品，必属精品~ 9月前 回复 ...

 1
-  极光之夜。 博主 回复： 🙄 9月前 回复 ...

 1
-  水巷石子： 大佬呀~ 9月前 回复 ...

 1
-  极光之夜。 博主 回复： 哈哈~ 9月前 回复 ...

 1
-  退出人间： 膜拜大佬 9月前 回复 ...

 1
-  极光之夜。 博主 回复： 哈哈，不敢当不敢当 9月前 回复 ...

 1
-  极光之夜。 博主： 可 9月前 回复 ...

 1
-  hhhmj666： 看着总有种醍醐灌顶的感觉 膜拜巨佬 2月前 回复 ...

 1
-  极光之夜。 博主 回复： 😏😏😏 2月前 回复 ...

 1
-  不正经的kimol君： 写的不错，学习了，学习的道路上一同进步，也期待你的关注与支持！ 9月前 回复 ...

 1
-  极光之夜。 博主 回复： 好 9月前 回复 ...

 1
-  盼盼编程： 写的不错！要是能受到大佬的回访那就太好了！ 9月前 回复 ...

 1
-  极光之夜。 博主 回复： 好 9月前 回复 ...

 1

Sass-学习笔记【基础篇】\_xing.org1^ 9-27


Sass和SCSS其实是同一种东西,平时都称之为Sass。二者其实有不同 两个区别点如下 (1).文件扩展不同。 Sass用“.sass”...

sass入门总结\_weixin\_34104341的博客 9-23


sass入门总结 变量\$ \$width:5em; #main{ width:\$width; } 嵌套引用 嵌套引用在其他编程语言中即是字符串插值,需要用#{}


HTML基础总结 (自己的笔记) 02-23


对HTML基础学习，同时做了笔记总结的东东，希望对


 极光之夜。


关注

 12

 17

 37





专