

# CSS & SASS

# Em,rem,px




# Bem



Example BEM class name





**SINGLE TOILET ROLL**  
POPULAR SELLER

£9.99

This is a luxury, quilted, 3-ply toilet roll with aloe vera for the smoothest wipe possible. No animals were harmed in the making or testing of this product. Due to high demand, limited to 2 per order.

LIKE

Block Element

`.product-card__image {`  
...  
`}`

Block Element

`.product-card__title {`  
...  
`}`

Block Element

`.product-card__price {`  
...  
`}`

Block Element

`.product-card__like-button {`  
...  
`}`





**SINGLE TOILET ROLL**  
POPULAR SELLER

£9.99

This is a luxury, quilted, 3-ply toilet roll with aloe vera for the smoothest wipe possible. No animals were harmed in the making or testing of this product. Due to high demand, limited to 2 per order.

LIKED ADD TO CART

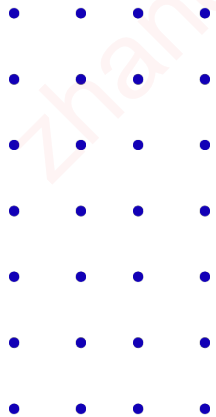
Block Modifier

.product-card--liked {  
...  
}

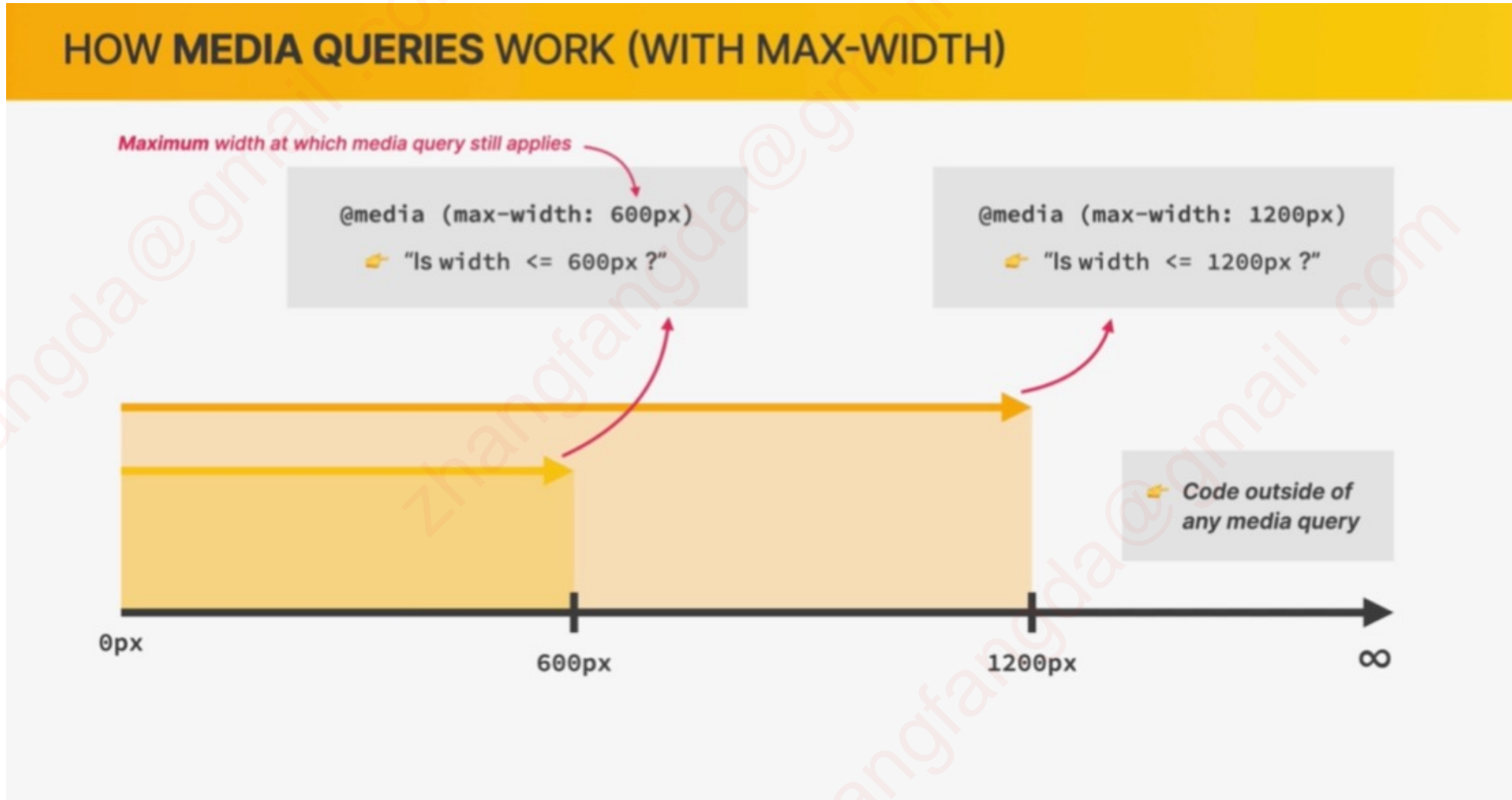
Block Element Modifier

.product-card\_\_like-button--liked {  
...  
}

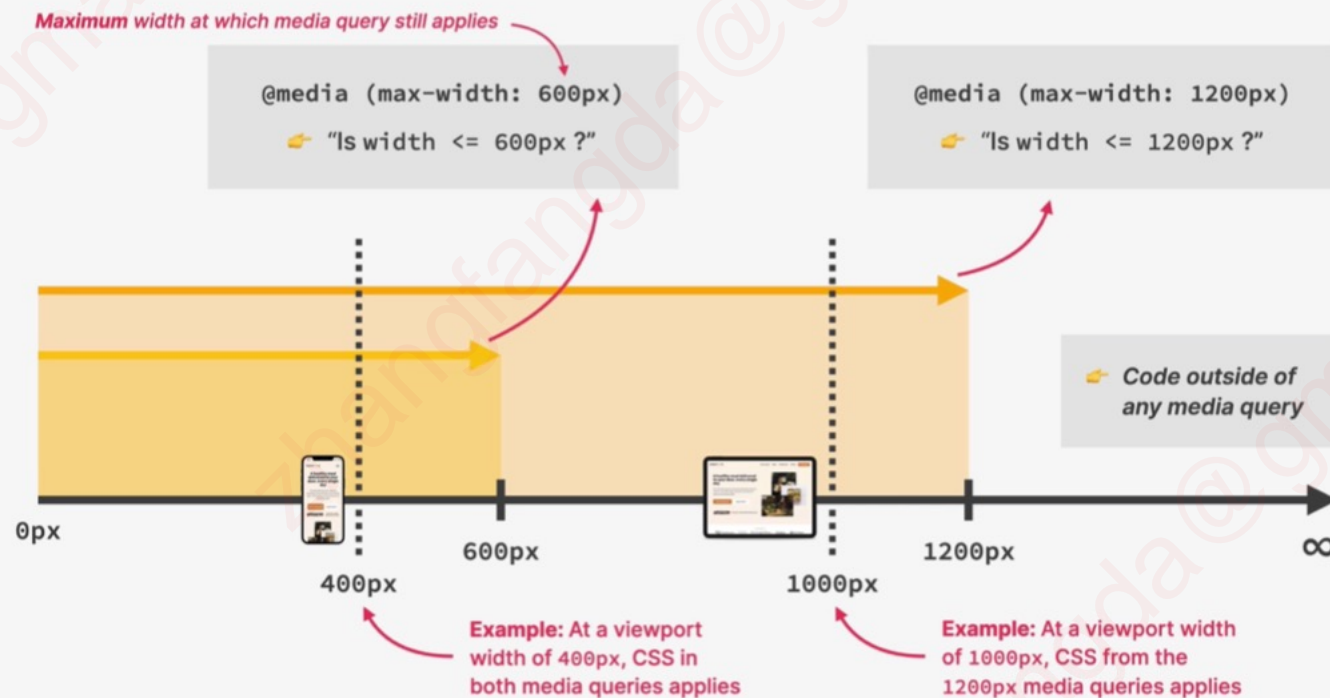
# What are the benefits of BEM?



# Media query



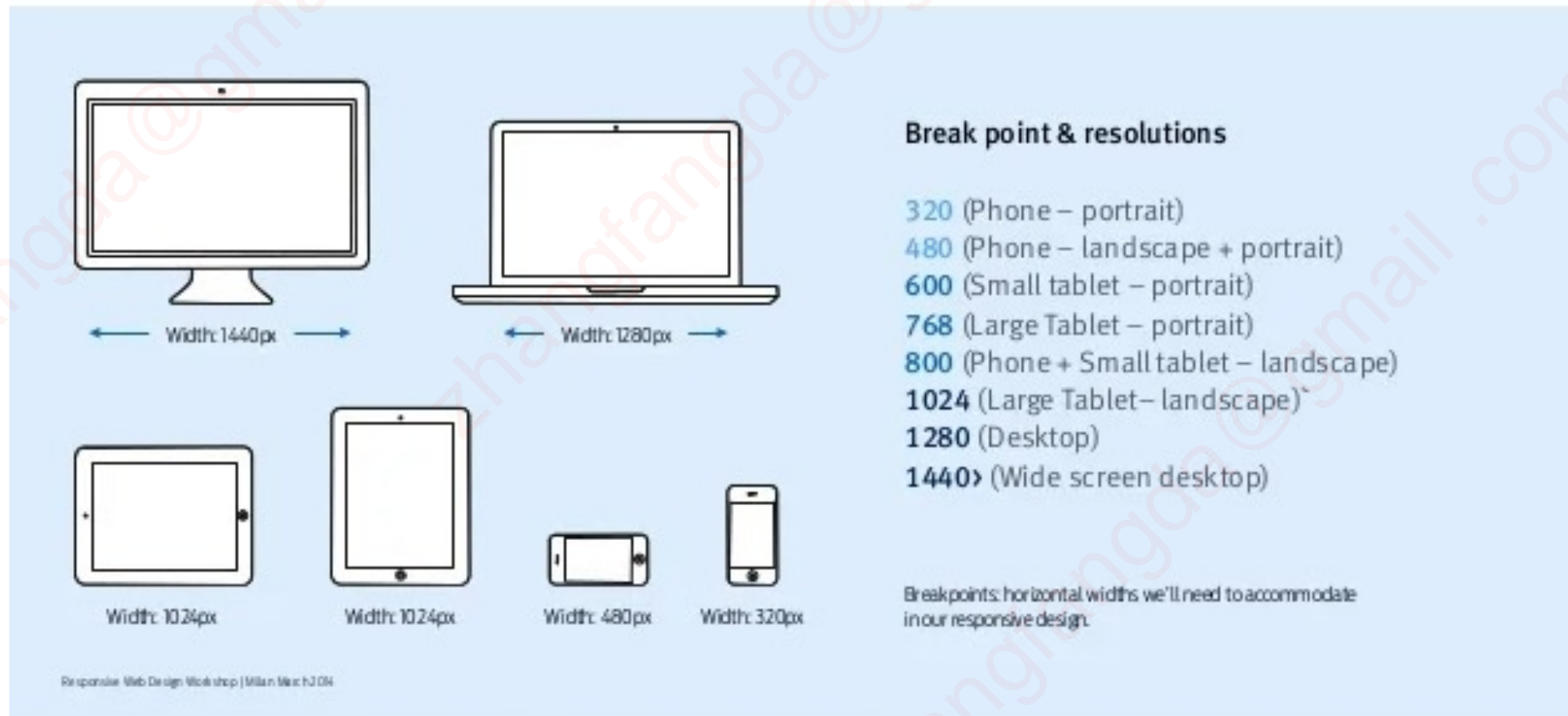
## HOW MEDIA QUERIES WORK (WITH MAX-WIDTH)





# Common device breakpoints

## Breakpoints & Media queries



# 静态定位

静态定位是元素的默认定位方式，无定位的意思

```
div {position: static}
```

很少用到

# 相对定位

相对定位是元素的是元素在移动的时候，相对于他原来的位置来说的

div {position: relative}

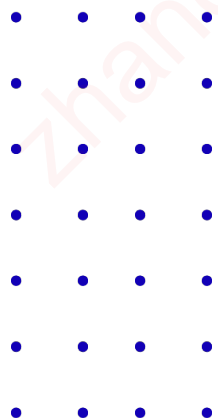
## 特点

- 1.参照自己原来的位置来移动
- 2.原来在标准流的位置继续占有，后面的盒子不会占有它原来位置

# 绝对定位

绝对定位是相对于父元素来移动的

```
div {position: absolute}
```



# 绝对定位

## 特点

- 1.如果**没有父元素**，或者**父元素没有定位**，则以浏览器为标准定位（document）。
- 2.如果祖先元素有定位（相对，绝对，固定），则参考最近一级的有定位的祖先元素进行移动。
- 3.绝对定位**不再占有原先的位置**。

# 固定定位 fixed

固定定位是固定于浏览器可视区的位置。

div {position: fixed}

## 特点

1. 以浏览器的可视窗口为参照点移动元素。
2. 跟父元素没有任何关系。
3. 不随滚动条滚动。
4. 不再占有原来位置。

## 练习

# 粘性定位 sticky

粘性定位是相对定位和固定定位的混合。

```
div {position: sticky;top:10px}
```

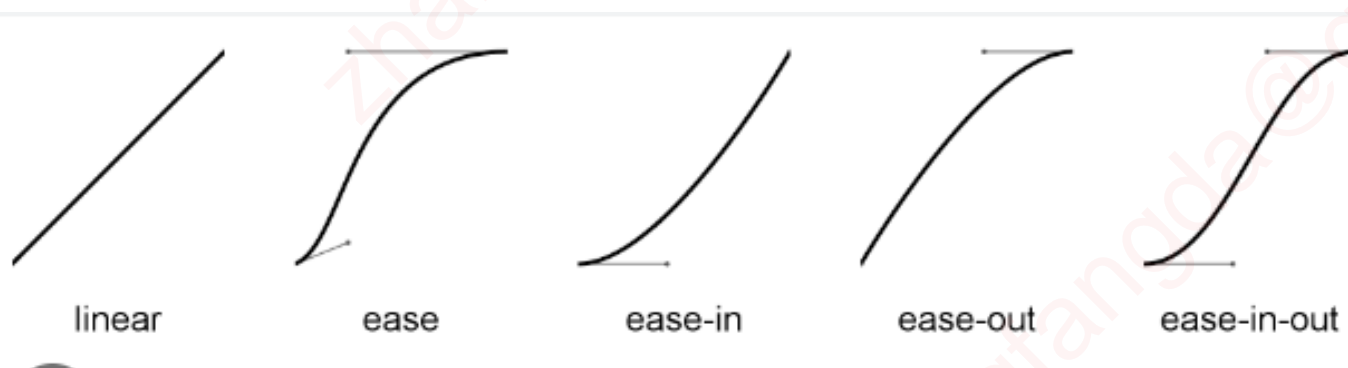
## 特点

- 1.以浏览器的可视窗口为参照移动元素（fixed特点）。
- 2.粘性定位占有原先的位置（relative特点）。
- 3.必须添加top, left,right,bottom其中一个才有效。

## 练习

# Transition

transition: property opacity duration 1s timing function ease delay 2s;



加速

速度慢下来

加速

减速

先加速后减速



# 为什么要使用scss

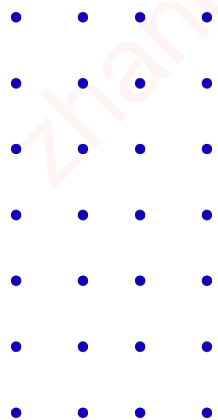
- 1.Scss完全兼容所有版本的CSS。
- 2.特性丰富， Scss拥有比其他任何CSS扩展语言更多的功能和特性。
- 3.技术成熟， 功能强大。
- 4.行业认可， 越来越多的人使用Scss。
- 5.社区庞大， 大多数科技企业和成百上千名开发者为Sass提供支持。
- 6.有无数框架使用Scss构建， 如Compass、 Bootstrap、 Bourbon和Susy。

# Scss是css的一种预处理语言

css的预处理语言有less,sass,scss

.less,.scss,.sass预处理语言只有在开发阶段使用

html无法识别css预处理语言,实际html引入的解析的都是css



## SASS has 5 primary features

1. Variables
2. Nesting
3. Mixin
4. Partials
5. Extent/Inheritance

# Variables

当样式表里需要换一套颜色，要找到好几处地方，并且替换好几次颜色值。如果我们能够在只改变某一处的属性值，其他地方的属性也会随之相应改变？那么我们可以使用Sass！

```
a {  
  color: #08c;  
}  
.nav {  
  background-color: #08c;  
}  
h1{  
  color: #08c;  
}  
.foo{  
  color: #08c;  
}  
.head{  
  color: #08c;  
}
```

编译

```
$linkColor: #08c;  
a {  
  color: $linkColor;  
}  
.nav {  
  background-color: $linkColor;  
}  
h1{  
  color: $linkColor;  
}  
.foo{  
  color: $linkColor;  
}  
.head{  
  color: $linkColor;  
}
```

# CSS

```
a {  
  color: #f00;  
  text-decoration: none;  
}  
  
h2 {  
  color: #f00;  
  font-size: 130%;  
}  
  
.button {  
  background: #f00;  
  color: #fff;  
}
```

# SASS

```
$highlightColor: #f00;

a {
  color: $highlightColor;
  text-decoration: none;
}

h2 {
  color: $highlightColor;
  font-size: 130%;
}

.button {
  background: $highlightColor;
  color: #fff;
}
```

# Nesting

## CSS

```
.carousel {  
  overflow: hidden; width: 250px;  
}  
.carousel ul.items {  
  position: relative;  
}  
.carousel ul.items li {  
  display: block; position: absolute;  
  width: 250px; left: -250px;  
}  
.carousel ul.items li a:link, .carousel ul.items li a:visited {  
  display: block; height: 50px;  
}  
.carousel ul.items li a:hover, .carousel ul.items li a:active {  
  opacity: 0.8;  
}
```

# SASS

```
.carousel {  
  overflow: hidden; width: 250px;  
  
  ul.items {  
    position: relative;  
  
    li {  
      display: block; position: absolute;  
      width: 250px; left: -250px;  
      a {  
        &:link, &:visited { display: block; height: 50px; }  
        &:hover, &:active { display: block; height: 50px; }  
      }  
    }  
  }  
}
```



# Mixin

在Sass中，可以为公用的CSS样式定义一个mixin，然后在需要使用这些样式的地方直接调用定义好的mixin。声明一个mixins时需要使用**@mixin**，然后后面紧跟mixins的名。

```
content {  
  font-size: 20px;  
  color:black;  
  line-height: 1.5;  
  margin:20px;  
}  
header {  
  font-size: 25px;  
  color:blue;  
  line-height: 1.5;  
  padding: 10px;  
}  
footer {  
  font-size: 30px;  
  color:red;  
  line-height: 1.5;  
  padding: 15px;  
}
```

编译后：

```
@mixin style($size: 15px, $color:#999) {  
  font-size: $size;  
  color:$color;  
  line-height: 1.5;  
}  
content {  
  @include style(20px,black);  
  margin:20px;  
}  
header {  
  @include style(25px,blue);  
}  
footer {  
  @include style(30px,red);  
}
```

# CSS

```
#about-us ul.options li {  
  background-color: #fff;  
  border-radius: 5px;  
}  
#search-results ul.products li {  
  background-color: #fff;  
  border-radius: 5px;  
  background-image: url('/images/product/list/results.top.png');  
  background-repeat: repeat-x;  
}  
/* ... */  
  
#navigation {  
  background-color: #fff;  
  border-radius: 5px;  
}
```

# SASS

```
@mixin redballoon-outline-box {  
  background-color: #fff;  
  border-radius: 5px;  
}  
  
/* ... */  
  
#about-us ul.options li { @include redballoon-outline-box; }  
#navigation { @include redballoon-outline-box; }  
  
#search-results ul.products li {  
  @include redballoon-outline-box;  
  background-image: url('/images/product/list/results.top.png');  
  background-repeat: repeat-x;  
}
```

# Extend

Sass中，选择器继承可以让选择器继承另一个选择器的所有样式。使用选择器的继承，要使用关键词**@extend**，后面紧跟需要继承的选择器。

```
#header,  
#footer,  
.ir{  
  text-shadow: none;  
  background-color: blue;  
  border: 0;  
}  
#header{  
  width:300px;  
}  
#footer{  
  width:300px;  
}
```

编译后:

```
%ir{  
  text-shadow: none;  
  background-color: blue;  
  border: 0;  
}  
#header{  
  @extend %ir;  
  height:100px;  
}  
#footer{  
  @extend %ir;  
  height:150px;  
}  
.ir{  
  @extend %ir;  
}
```

# 函数

Sass定义了很多函数可供使用，以@function开始。比如颜色函数，lighten为减淡，darken为加深，其调用方法为：  
lighten(\$color,\$amount),  
darken(\$color,\$amount)，它们的第一个参数都是颜色值，第二个参数都是百分比。

```
$baseFontSize: 10px !default;
$gray: #ccc !default;

// pixels to rems
@function pxToRem($px) {
  @return $px / $baseFontSize * 1rem;
}

body{
  font-size:$baseFontSize;
  color:lighten($gray,10%);
}

.test{
  font-size:pxToRem(16px);
  color:darken($gray,10%);
}
```

<https://sass-lang.com/>

