

sass介绍

Sass变量

在Sass中，声明变量使用的是"\$"（美元符号）开头。语法：

在Sass中，对于变量的取值有2种方式：（1）一般值；（2）默认值。

定义变量的默认值很简单，只需要在“变量值”后面加上"default"就可以。

变量的作用域

在Sass中，变量根据作用域可以分为2种：（1）全局变量；（2）局部变量。

Sass数据类型

在Sass中，数字（Number）是最基本的数据类型，可以是正数、0或负数。数字在Sass中使用非常广泛，大多数都是结合CSS单位来实现的，例如10px、10em或者10%。虽然它们带有单位，但是技术上依然算是数字。

字符串

在JavaScript中，使用单引号（'）或双引号（"）包含的都是字符串，就算它们包含的是一个空格，那也是字符串。但是Sass中的字符串跟JavaScript中的字符串有点不一样。

在Sass中，共有2种字符串：

- （1）有引号的字符串；
- （2）无引号的字符串；

无引号字符串，我们在CSS中是经常遇到的，例如"font-weight:bold"中的bold。"font-family:sans-serif"中的sans-serif等。Sass引入无引号字符串的目的也是为了与CSS语法一致。

布尔值

Sass中的布尔值只有2种取值：true和false。

在Sass中，布尔值一般用于"@if...@else..语句"条件判断，只有条件表达式结果是false或null才会返回false，其他一切将返回true。

在Sass中，有一种特殊的数据类型，那就是“颜色值”。Sass中的颜色值共有4种：

- （1）关键字颜色值，如red；
- （2）十六进制颜色值，如#FFF00；
- （3）RGB颜色值，如rgb(255,255,0)；
- （4）HSL颜色值，如：hsl(360,50%,50%)；

这几种颜色值都是可以互相转换的，在Sass的颜色运算中，都是统一转换为十六进制颜色值然后再计算。

列表值

“列表值”的数据类型跟JavaScript中的数组是相似的，我们可以把它比作“Sass中的数组”。

Sass列表值有2种语法格式，一种是由英文逗号隔开的分隔符，另外一种是由空格隔开的分隔符。

在Sass中，还为我们提供了另外一种特殊数据类型：Map值。Map值跟JSON值是非常相似的，数据都是以“键/值”的方式成对出现。

空值null

Sass流程控制

跟JavaScript一样，Sass作为一门编程式的语言，也有属于自己的一套流程控制语法：

- @if语句；
- @for循环；
- @while循环；
- @each循环；
- @if语句

在Sass中，可以使用"@if语句"来进行条件选择判断。Sass的条件选择语句共有3种：

- @if...（单向选择）；
- @if...@else...（双向选择）；
- @for循环

在Bootstrap中，有.col1-.col12；在Sass中，可以使用"@for"来实现循环操作。其中，Sass中的@for循环有2种方式。

这2种方式是相似的，唯一的区别是：方式1包括结束值，方式2不包括结束值。其中“开始值”和“结束值”都是正整数。

- @while循环

在Sass中，也可以使用@while语句来实现循环操作。跟JavaScript一样，@while语句既是循环语句，也是条件判断语句。

说明：在Sass中，没有类似于JavaScript的“do...while...”这种语句。

- @each循环

除了@for以及@while这2种循环方式之外，Sass还为我们提供了另外一种循环方式：each循环。Sass中的each循环跟jQuery中的each循环很相似。

\$var是一个变量，也可以根据后面列表值表示的内容需要写成\$item、\$color等。列表值，是Sass中的一种数据类型，可以称之为“Sass中的数组”。

Sass代码重用-混合宏

在Sass中，使用“混合宏（mixin）”来处理经常被多个地方使用的相同的CSS代码块。混合宏，跟JavaScript中的函数很相似，我们可以称之为“Sass中的函数”。

一、混合宏的定义和调用

在Sass中，使用"@mixin"来定义一个混合宏，然后使用"@include"来调用一个混合宏。

语法：

Sass中的混合宏跟C语言的宏是非常相似的。所谓的“宏”，指的是可重用的代码块。

@mixin用来定义一个混合宏，@include用来调用一个混合宏。此外，@mixin跟CSS3中的@font-face和@media语法是一样的。

Sass代码重用-占位符%placeholder

在Sass中，引入了占位符"%placeholder"来进一步优化"@extend"。

从这个例子可以看出，继承"@extend"是非常好用的。通过@extend，可以直接在".btn-primary"和".btn-second"中插入定义好的".btn"，这样每次只要你修改了".btn"中的样式，".btn-primary"和".btn-second"中的样式都会同时修改。

但是如果我们的HTML中没有任何一个元素使用".btn"的话，".btn"存在的唯一目的就是仅仅用来给".btn-primary"和".btn-second"继承。也就是说，".btn"这个类在编译出来的CSS中是多余的。那有没有更好的办法来实现预期效果呢？

在新版本的Sass中，引入了“占位符%placeholder"来优化"继承@extend"的输出。

“占位符%placeholder"并非用来替换"继承@extend"的，而是用来配合"继承@extend"来使用的。从上面我们可以知道，继承@extend有2种输出方式：

- 需要保留基类的：只使用@extend来实现；
- 不需要保留基类的：使用@extend配合%placeholder来实现；

Sass代码重用-继承@extend

- （1）继承"@extend"；
- （2）占位符"%placeholder"；
- （3）混合宏"@mixin"；

在实际开发中，经常会碰到不同地方都用到相同的CSS样式的情况。如果相同的CSS样式只有一个CSS属性，此时只需要使用Sass变量实现就可以满足了。如果相同的CSS样式有多个CSS属性，这个时候我们就希望把“相同的CSS样式”当做整块来处理。

Sass基本运算-颜色运算

颜色值是Sass中的一种特殊的数据类型。在Sass中，我们也是可以对颜色值进行运算的。颜色运算支持加、减、乘、除，并且是分段进行计算的。也就是说，红、绿、蓝这3个部分的颜色是单独进行计算的。

此外，如果颜色值不是十六进制颜色值（如#010203），而是RGB或者HSL的话，都是将RGB或HSL先转换为十六进制颜色值，然后再进行加、减、乘、除运算的。

Sass基本运算-字符串运算

在Sass中，我们可以使用"+"（加号）来实现字符串的拼接。对于字符串的连接，我们分为3种情况来考虑：

- 如果左右两个字符串都是有引号的，结果是一个有引号的字符串；
- 如果左边字符串是有引号的，右边字符串是没有引号的，结果是一个有引号的字符串
- 如果左边字符串是没有引号的，右边字符串是有引号的，结果是一个没有引号的字符串；

概括：结果字符串是否有引号，取决于左边字符串是否有引号。

Sass嵌套

选择器嵌套，是Sass中最常见的嵌套方式，这个类似于HTML元素的嵌套。

选择器嵌套这种方式虽然操作起来很方便，但是却有一个很大的弊端：嵌套的层级越深，编译出来的CSS代码的选择器层级也越深。这种嵌套方式会导致CSS样式冗余，并且难以维护。

CSS有些属性的前缀是相同的，例如：

- （1）border-top、border-right、border-bottom、border-left这4个属性拥有相同的前缀"border"；
- （2）margin-top、margin-right、margin-bottom、margin-left这4个属性拥有相同的前缀"margin"；
- （3）font-family、font-size、font-weight、font-variant等属性拥有相同的前缀"font"；

对于这些拥有相同前缀的属性，可以使用属性嵌套的方式来简化操作。

对于属性嵌套，要特别注意一点：在需要嵌套的属性前缀后面一定要加英文冒号“:”

伪类嵌套（或伪元素嵌套）跟属性嵌套很像，只不过它是需要借助"&"符号一起配合使用。

Sass插值

在Sass中，可以通过使用插值的方式来实现“选择器名”、“属性名”以及“属性值”中插入一个“变量的值”，从而来“构造”一个新的选择器名、新的属性名以及新的属性值。

插值这种方式，在Sass代码的几乎任何一个地方都可以插入一个值，包括选择器名、属性名、属性值等。

变量，我们一般都是用于属性值的，是不能直接用于选择器名中的。

如果属性值是一个简单数字或者字符串，我们可以使用变量的方式来实现。但如果属性值是一个复杂的数据类型，我们就必须使用插值的方式来实现。

总而言之，如果需要在Sass中构造一个新的选择器名、新的属性名以及新的属性值，首先考虑的应该是使用“插值#{0}"来实现。

Sass注释

注释内容

在Sass中，这种注释方式在编译后不会保留下来。

注释内容

在Sass中，这种注释方式在编译之后会保留下来。因为这种注释方式跟CSS注释方式是相同的，所以编译后会保留下来。

注释内容

压缩工具会删除所有的注释，有些时候为了保留一些版权声明的注释说明，可以采用以下方式：

在注释内容前面加上一个"!", 这种压缩工具就不会删除这条注释信息

Sass基本运算-数字运算

- （1）数字运算；
- 子主题 1
 - （2）字符串运算；
 - （3）颜色运算；

Sass加法

在Sass中，做加法运算时，数值可以带单位，但是需要运算单位相同。如果不相同就会报错，编译不通过。

Sass减法

在Sass中，做减法运算时，数值可以带单位，但是需要运算单位相同。如果不相同就会报错，编译不通过。这一点跟Sass加法是一样的。

Sass乘法

在Sass中，做乘法运算时，只能有一个数值带单位，另外一个数值只能是不带单位的数字。如果两个都是带单位的数字，则Sass会报错而编译不通过。

Sass除法

"/"在CSS中已经作为一种符号来使用了，例如常见的字体属性缩写"font:Arial 12px/1.5em"。因此在Sass中做除法运算的时候，如果直接使用"/"符号作为除号，将不会生效。

在Sass中，如果想要做除法运算，需要在外面添加一个“小括号()”。

- 数值被小括号()包含；
- 数值是另外一个数学表达式的一部分；
- 数值或它的任意部分存储在一个变量中或者函数的返回值；