sass是基于ruby语言写的;所以使用前需要先安装ruby;

<https://rubyinstaller.org/downloads/>    ruby下载地址

安装完成后在命令行输入 ruby -v检验是否安装成功

devkit配置:

创建config.yml文件,打开命令行,进入devkit的解压目录,

在命令行输入:

ruby  dk.rb init 初始化创建config.yml文件

回到命令行,执行审查和安装:

ruby dk.rb install

ruby dk.rb review

ruby dk.rb init

(全新版本的ruby自带devkit,所以配置以上设置可以跳过)

ruby安装完成后进行sass安装:

命令行输入gem install sass;(一般会成功);

如果安装失败,可以尝试使用淘宝镜像站安装http://ruby.taobao.org/

步骤:$ gem sources --add [https://ruby.taobao.org/](http://ruby.taobao.org/) --remove [https://rubygems.org/](http://rubygems.org/)

    $ gem sources -l

最后输入gem install sass就完成了

sass -v检查是否安装成功

sass的scss文件转css文件的命令行方法:

当前scss文件路径下的cmd:sass in.scss out.css;

还可以命令sass监视文件的改动并更新css:

sass --watch in.scss:out.css

如果目录里有很多sass文件,还可以命令sass监视整个目录:

sass --watch app/sass:public/stylesheets   (app和stylesheets都是目录名)

sass --help 可以看更多sass相关的命令

注意: 如果是在框架中使用,那么以上设置均可忽略,直接在框架中npm下载

1  sass中使用$声明变量,less中使用@声明变量:   $bcolor:blue;

2  属性的规则:

nav{

border-style:solid;

border-color:red;

border-width;1px;

}

用sass写为:(注意border后面要跟:)

nav{

border:{

style:solid;

width:1px;

color:red;

}

}

navv{

border :1px solid red;

border-left:1px;

border-right:2px;

background-color:green;

}

sass写为:

navv{

border:1px solid red{

left:1px;

right:2px;

}

background-color:green;

}

3  sass中的@import导入文件,可以省略sass或者scss文件后缀名:

@import "文件名";

变量在导入文件之后是通用的,因为在导入之后,被导入的scss文件就已经被编译在同一css文件上了,变成了一家人.

4 一次性编译一个文件夹里所有.scss文件到另一个目录中的相关命令如下:

sass --update in:out    (其中的in和out指的是输入  输出文件夹).

5  sass局部文件的文件名以下划线开头,这样,sass就不会在编译时单独编译这个文件输出css,而是只把这个文件用作导入使用.

sass源文件中可以通过@import指令导入其他Sass源文件,被导入的文件就是局部文件,局部文件让Sass模块化编写更加容易。

6默认变量值:!default

eg:

@import "text.scss";

@import "tests.scss";

$first:200px !default;

.class{width:$first;}

7  嵌套导入:

假设有一个局部文件 \_blue-self;那么嵌套导入的形式如下:

.blue{@import "blue-self"};

8  导入原生的css文件,只需要把原生的css文件后缀修改成scss就可以了,因为sass完全兼容css;也可以直接导入css文件,只是这样的方式是以css原生导入css文件实现的,会带来原生的css文件加载问题(浏览器必须先下载完css文件才能执行下面的css代码);

9  静默注释: //  (同less),标准注释:  /\*\*/

10  混合器

sass混合器使用 @mixin标识符定义.看上去像css的@标识符,比如@media;

在样式表中通过@include来使用这个混合器;

example:----

@mixin round{

xxxxxx;

xxxxxx;

xxxxxx;

}

引用:

.box{

xxxx;

xxxx;

@include round;

}

实际上,混合器太好用了,以至于会很容易造成滥用;导致样式表过大,导致加载变慢.

11   给混合器传参:

@mixin link-colors($normal,$hover,$visited){

color:$normal;

&:hover{color:$hover;}

&:visited{color:$visited;}

}

调用:

.box{

@include link-colors(red,blue,green);

}

最终生成的css文件为:

.box{color:red;}

.box:hover{color:blue;}

.box:visited{color:green;}

sass强大的一点是允许通过语法$name:value的形式指定每个参数的值:

.box{

@include link-colors($hover:red,$normal:green,$visited:yellow);

}/\*这时可以看到参数的顺序已经不重要了\*/

sass允许混合器声明时,给参数默认值;

@mixin link-colors($normal,$hover:red,$visited:$normal){

color:$normal;

&:hover:{color:$hover;}

&:visited:{color:$visited;}

}

12   sass的继承:

通过@extend语法实现

.box{xxxxxx}

.wap{@extend .box;height:100px;}

(.wrap不仅会继承.box 自身的所用样式,任何跟.box有关的组合选择器样式也会被.wrap以组合的形式继承)

sass中也可以组合继承:

.one{@extend .two, .three;}

继承的局限性:目前包含选择器(.wrap.box)或者相邻选择器(.wrap+.box)还不支持继承.

继承带%号的东西:(%代表占位符):

#message a%long{

color:red;

}

.notice{

@extend %long;

margin:10px;

}

编译后就是:

#message a.notice{

color:red;

}

.notice{

margin:10px;

}

sass继承也有作用域的问题,比如在@media{}作用域内不能继承其作用域外的东西;

(实际上在sass中不是很建议多使用继承,因为继承会造成子类对父类的依赖,一旦父类改变,事情将会变得更麻烦,值得一提的是,我们完全可以放心的继承有后代选择器修饰规则的选择器,不管后代选择器有多长,但有一个前提条件就是,不要用后代选择器去继承).

12   sass函数;

unquote()函数,用来删除字符串最外侧的引号(单双引号);

.box{

color:unquote("'red'");

}

编译后就是:

.box{

color:'red';

}

quote()函数,给字符串加上双引号的函数,如果原来的单引号,则会把单引号变成双引号;使用quote()函数只能给字符串增加双引号,而且字符串中间单引号或者空格时,需要用单引号或者双引号括起,否则编译时会报错;

sass中的数字函数:

percentage($value):将一个不带单位的数字转换成百分比数值;

round($value):将数值四舍五入;会保留单位

ceil($value):向上取整;

floor($value):向下取整;

abs($value):去绝对值;

min($value):找出几个数值中的最小值;

max($value):找出几个数值中的最大值;

sass中的列表函数:

length($list):返回一个列表的长度;

nth($list,$n):返回一个列表中指定的某个标签值;

join($list1,$list2,[$separator]);将两个列表连接成一个列表;(注意只能连接两个列表,多于两个会报错)   $separator参数有两种情况:comma 以逗号进行分隔,space 以空格进行分隔,不写的话会默认auto,这会带来问题(分隔的形式不好预期)

append($list,$val,[$separator]):将某个值放在列表最后;   $val是要插入的值

zip($list...):将几个列表组合成一个多维的列表;  在使用zip()函数时,每个单一的列表内的个数值必须是相同的; zip(1px 2px 3px,red green blue,solid dashed dotted);编译后就是:((1px 2px 3px),(red green blue),(solid dashed dotted));

index($list,$value):返回一个值在列表中的位置值;

sass中的三元函数:

introspection函数包含了几个判断形函数:

type-of($value):返回一个值的类型;

unit($number):返回一个值的单位;

unitless($number):判断一个值是否带有单位;  返回true和false,没有单位返回true

comparable($number1,$number2):判断两个值是否可做加减和合并,这几个函数主要用来对值做一个判断的作用,依次来看每个函数的功能.  返回true和false;

Miscellaneous函数,三元条件函数:if($true,$if-true,$if-false);比如  if(true,red,green);>>>>>red;

13  sass自定义函数:

@function method(参数){@return  xxxxxx;}