**SASS学习记录**

**一、什么是SASS**

**SASS是一种CSS的开发工具，提供了许多便利的写法，大大节省了设计者的时间，使得CSS的开发，变得简单和可维护。**

**二、安装和使用SASS**

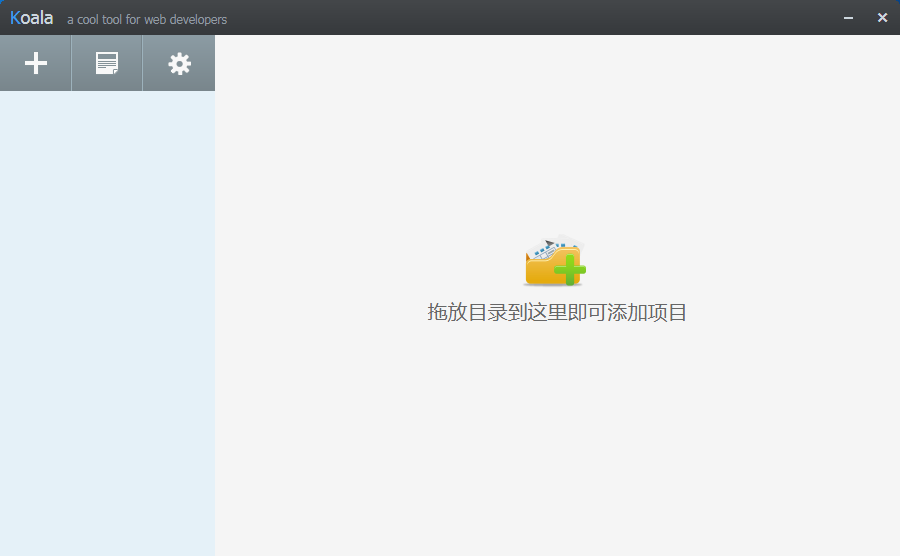
**推荐小白使用方式，免系统环境配置，免命令行操作**

1. **Koala软件编译：**

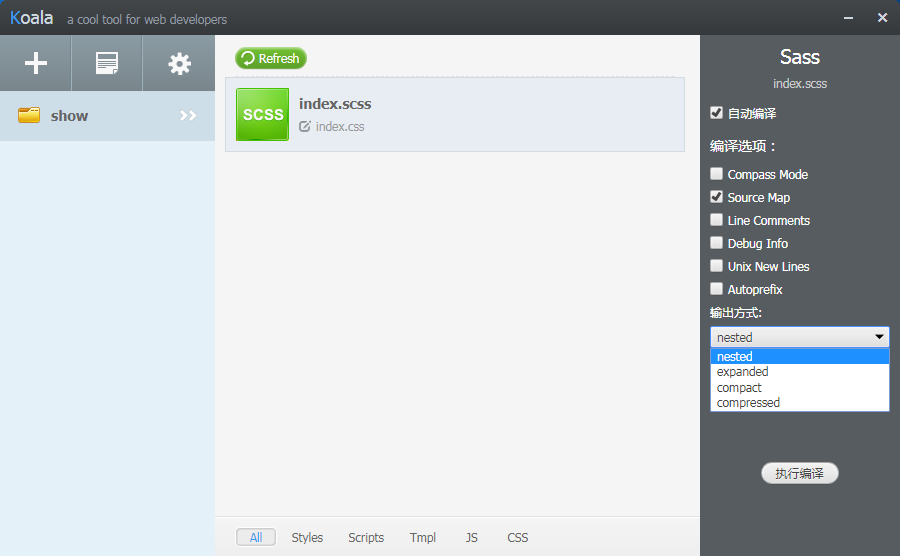
**百度下载考拉：**

****

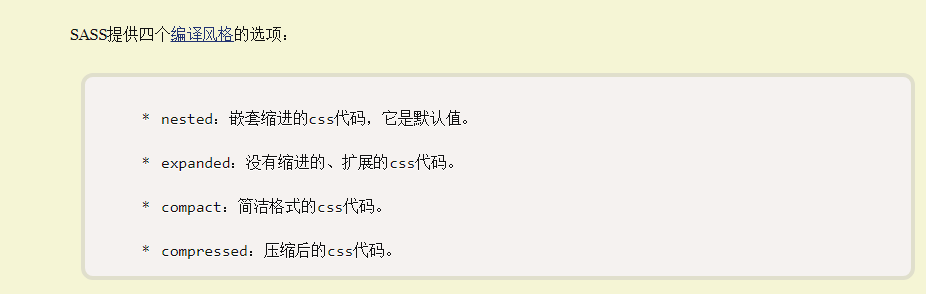
注意系统位数

软件图示

使用sass：

设置你想要输出的方式

四种输出方式介绍：



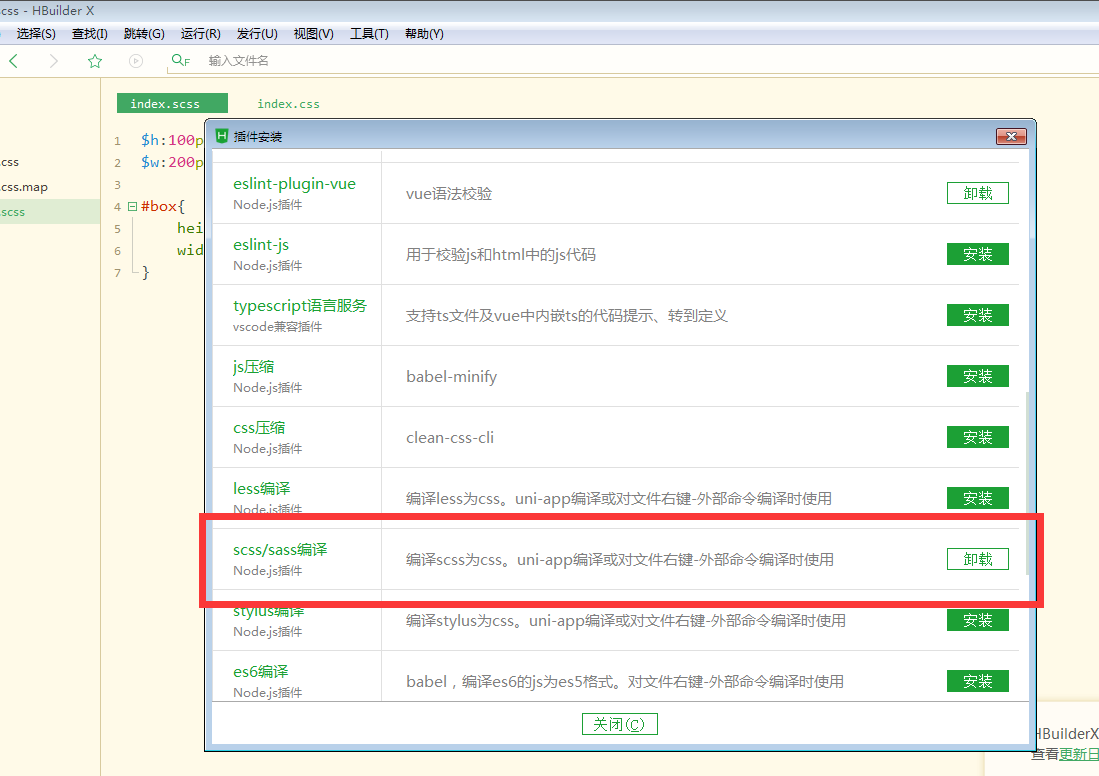
一般开发完成之后都是压缩后在使用，前期可以选择没有压缩的，方便开发

添加需要编译的sass文件 ，选择自动编译即可和确定输出方式及执行编译，，然后就可以酣畅淋漓的写你的sass文件了。

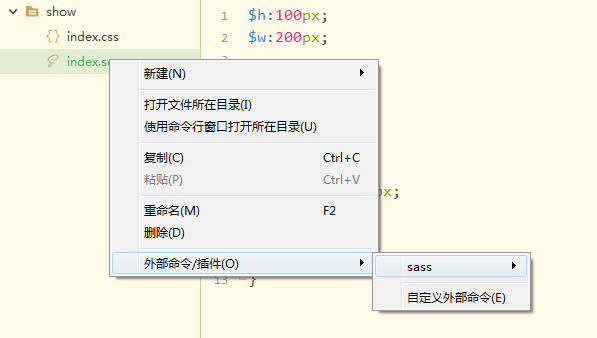
1. **编辑器插件安装：**

**这里以HubliderX为例：**

**工具————》插件安装————》选择sass插件下载**

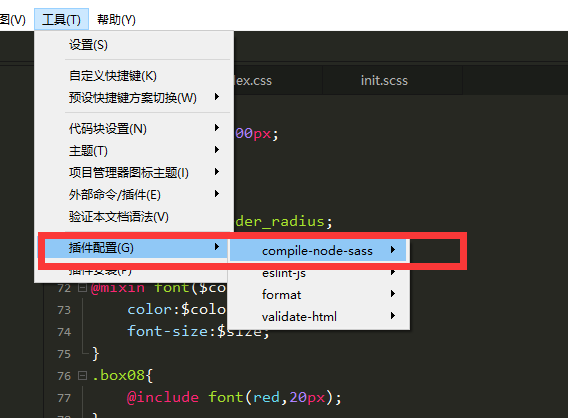


**安装完后，**



**右击文件选择插件命令，去编译sass 文件即可**

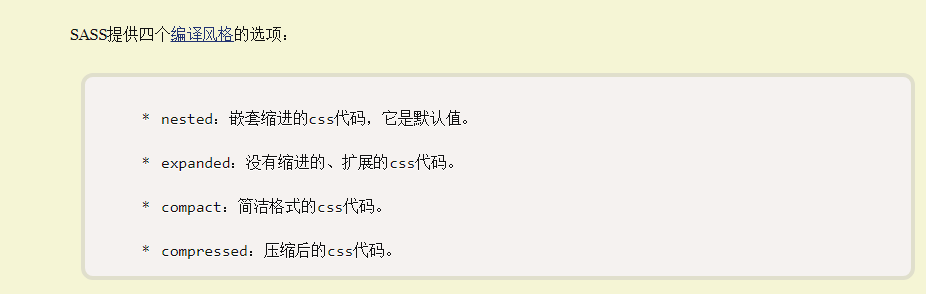
**如果需要保存时自动编译，修改sass插件配置即可**



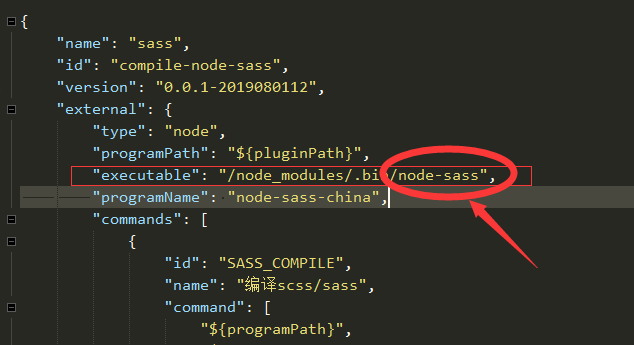
**修改如下：修改后重启生效【注意自动保存时编译，需要先去编译一遍，即右击选择插件命令编译一次】**

****

四种编译风格修改【默认是nosted的格式】



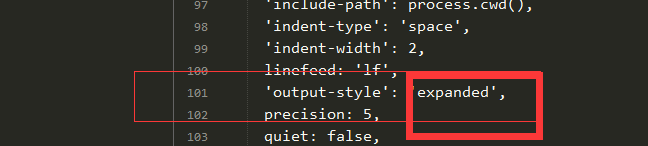
去修改node-sass



Node-sass文件具体位置

D:\HBuilderX\plugins\compile-node-sass\node\_modules\node-sass-china\bin

HbuilderX 每个人的安装位置按照自己的来，后边路径不变。



101行修改，四种不同编译风格按照自己的来

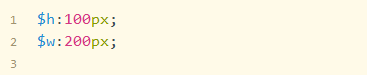
重启即可

然后你就可以愉快的编写sass文件了，再也不想什么命令行，什么环境了

**三、sass语法及使用**

（一）.变量使用

变量 SASS允许使用变量，所有变量以$开头。

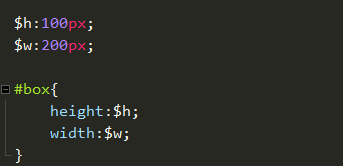
如：

此时$h 和$w就是一个变量

变量引用：

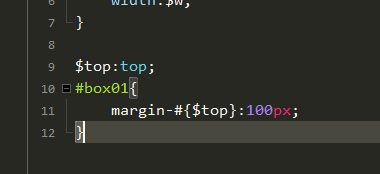
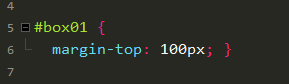
Css生成时，变量会被它们的值所替代。之后，如果你需要一个不同的值，只需要改变这个变量的值，则所有引用此变量的地方生成的值都会随之改变。

变量使用：

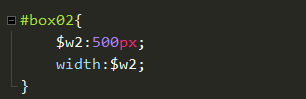
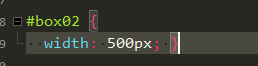


如果变量是使用在字符串中。则需要使用一个#{}将其包裹

例如

 Css: 

当变量定义在css规则块内，那么该变量只能在此规则块内使用。

此时$w2就只能在#box02中使用

变量名称

sass的变量名可以与css中的属性名和选择器名称相同，包括中划线和下划线。这完全取决于个人的喜好，sass并不想强迫任何人一定使用中划线或下划线，所以这两种用法相互兼容。用中划线声明的变量可以使用下划线的方式引用，反之亦然。

（二）嵌套CSS 规则

css中重复写选择器是非常恼人的。如果要写一大串指向页面中同一块的样式时，往往需要一遍又一遍地写同一个ID:

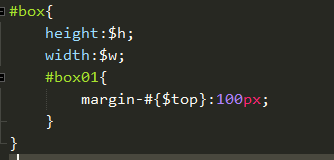
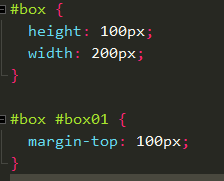
#content article h1 { color: #333 }

#content article p { margin-bottom: 1.4em }

#content aside { background-color: #EEE }

sass可以让你只写一遍，且使样式可读性更高。在Sass中，你可以像俄罗斯套娃那样在规则块中嵌套规则块。sass在输出css时会帮你把这些嵌套规则处理好，避免你的重复书写

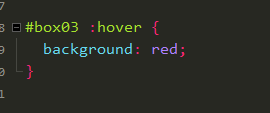
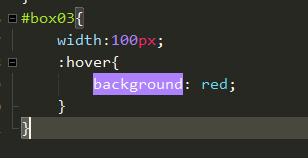
例如：

css

大多数情况下这种简单的嵌套都没问题，但是有些场景下不行，比如你想要在嵌套的选择器里边立刻应用一个类似于:hover的伪类。为了解决这种以及其他情况，sass提供了一个特殊结构&。

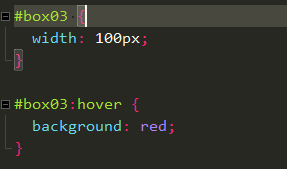
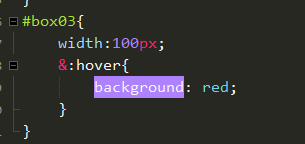
# 父选择器的标识符&

sass在解开一个嵌套规则时就会把父选择器(#content)通过一个空格连接到子选择器的前边, 这种在CSS里边被称为后代选择器写:hover这种伪类时，你并不希望以后代选择器的方式连接。比如说，下面这种情况sass就无法正常工作:



解决可以使用一个特殊的sass选择器，即父选择器。在使用嵌套规则时，父选择器能对于嵌套规则如何解开提供更好的控制。它就是一个简单的&符号，且可以放在任何一个选择器可出现的地方当包含父选择器标识符的嵌套规则被打开时，它不会像后代选择器那样进行拼接，而是&被父选择器直接替换

例如：



（三） 群组选择器的嵌套

# 类似

# 

# 这种选择器称为群组选择器。群组选择器的规则会对命中群组中任何一个选择器的元素生效。

# 非常幸运，sass的嵌套特性在这种场景下也非常有用。当sass解开一个群组选择器规则内嵌的规则时，它会把每一个内嵌选择器的规则都正确地解出来:

# 

# 

# 处理这种群组选择器规则嵌套上的强大能力，正是sass在减少重复敲写方面的贡献之一。尤其在当嵌套级别达到两层甚至三层以上时，与普通的css编写方式相比，只写一遍群组选择器大大减少了工作量。

# 有利必有弊，你需要特别注意群组选择器的规则嵌套生成的css。虽然sass让你的样式表看上去很小，但实际生成的css却可能非常大，这会降低网站的速度。

# 子组合选择器和同层组合选择器：>、+和~

# 上边这三个组合选择器必须和其他选择器配合使用，以指定浏览器仅选择某种特定上下文中的元素

# 

# 同层相邻组合选择器+ 选择.box05元素后紧跟的.box05\_child:

# 同层全体组合选择器~， 选择所有跟在.box05\_child后的同层span元素，不管它们之间隔了多少其他元素:

# 子组合选择器>选择一个元素的直接子元素。，第一个选择器会选择ul下的所有li元素

# 嵌套属性

# 在sass中，除了CSS选择器，属性也可以进行嵌套。尽管编写属性涉及的重复不像编写选择器那么糟糕，但是要反复写border-style、border-width、border-color以及border-\*等也是非常烦人的。在sass中，嵌套属性的规则是这样的:把属性名从中划线-的地方断开，在根属性后边添加一个冒号:，紧跟一个{ }块，把子属性部分写在这个{ }块中

# 

# 

（四）导入SASS文件

# 当通过@import把sass样式分散到多个文件时，你通常只想生成少数几个css文件。那些专门为@import命令而编写的sass文件，并不需要生成对应的独立css文件，这样的sass文件称为局部文件，@import ‘url’ 的方式导入

# 先定义一个init.scss 的清除默认样式的

# 

# 文件中导入

# 

# 编译结果

# 

# 嵌套导入

# sass允许@import命令写在css规则内。这种导入方式下，生成对应的css文件时，局部文件会被直接插入到css规则内导入它的地方。

# 

# 效果

# 

（五）静默注释

# sass另外提供了一种不同于css标准注释格式/\* ... \*/的注释语法，即静默注释，其内容不会出现在生成的css文件中。它们以//开头，注释内容直到行末。

# 

# /\*\*/ -------该注释可以在css文件中显示处理啊

# //---------该注释在css文件中不显示

（六） 混合器

# 但是当你的样式变得越来越复杂，你需要大段大段的重用样式的代码，独立的变量就没办法应付这种情况了。你可以通过sass的混合器实现大段样式的重用。混合器使用@mixin标识符定义这个标识符给一大段样式赋予一个名字，这样你就可以轻易地通过引用这个名字重用这段样式。然后就可以在你的样式表中通过@include来使用这个混合器，放在你希望的任何地方。@include调用会把混合器中的所有样式提取出来放在@include被调用的地方

# 例如：

# 

# 效果

# 

# 混合器中不仅可以包含属性，也可以包含css规则，包含选择器和选择器中的属性，如下代码

# 

# 给混合器传参

# 混合器并不一定总得生成相同的样式。可以通过在@include混合器时给混合器传参，来定制混合器生成的精确样式。当@include混合器时，参数其实就是可以赋值给css属性值的变量。如果你写过JavaScript，这种方式跟JavaScript的function很像:

# 

# 

# 当你@include混合器时，有时候可能会很难区分每个参数是什么意思，参数之间是一个什么样的顺序。为了解决这个问题，sass允许通过语法$name: value的形式指定每个参数的值。这种形式的传参，参数顺序就不必再在乎了，只需要保证没有漏掉参数即可:

# 

# 尽管给混合器加参数来实现定制很好，但是有时有些参数我们没有定制的需要，这时候也需要赋值一个变量就变成很痛苦的事情了。所以sass允许混合器声明时给参数赋默认值。

# 默认参数值

# 

# 默认参数就是在参数后加上 ： 和默认的值即可

# 使用选择器继承来精简CSS

# 使用sass的时候，最后一个减少重复的主要特性就是选择器继承。选择器继承是说一个选择器可以继承为另一个选择器定义的所有样式。这个通过@extend语法实现，如下代码:

# 

# 我都学习记录到此结束，如有不当之处还请批评指正

# 学习人：赵钊坤