

# WebApp实战系列课程 01-CSS编程框架(sass)

SASS 是成熟、稳定、强大的 CSS 扩展语言。

## SASS 安装与配置

### 第一步：安装Ruby

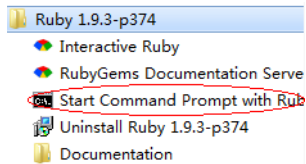
因为sass依赖于ruby环境，所以装SASS之前先确认装了ruby。

ruby官方网站下载地址: <http://rubyinstaller.org/downloads>

在安装的时候，请勾选Add Ruby executables to your PATH这个选项，添加环境变量，不然以后使用编译软件的时候会提示找不到ruby环境。

### 第二步: 通过ruby命令 gem安装sass

安装完ruby之后，在cmd命令行中执行 `gem install sass`



```
gem install sass
```

### 淘宝RubyGems镜像安装 sass

```
$ gem sources --remove https://rubygems.org/
$ gem sources -a https://ruby.taobao.org/ 【如果你系统不支持https，请将淘宝源更换成: gem sources -a http://gems.ruby-china.org】
$ gem sources -l
*** CURRENT SOURCES ***

https://ruby.taobao.org
# 请确保只有 ruby.taobao.org
$ gem install sass
```

### 升级sass版本的命令行为

```
gem update sass
```

## SASS编译

SASS编译的方式有多种：

1. 通过命令行的方式进行编译
2. 通过webstorm开发工具进行编译
3. 通过gulp进行集成编译 /kaola

### 通过命令行的方式进行编译

`sass scss`目录地址：输出目录地址

如 `sass scss/main.scss:style/main.css`

SASS命令行工具可以对目录进行自动监视，这样我们的目录下的文件发生了变化，SASS帮助我们自动编译。

`sass --watch` 监视的目录地址：自动编译的目录地址

通过 `--watch`命令让sass工具自动监视目录，并自动进行编译

`sass --watch watchdir:outputdir --style nested|compact|compressed|expanded`

通过 `--style nested`(嵌套-默认)|`compact`(紧凑型)|`compressed`(压缩)|`expanded`(扩展) 命令，可以为sass生成css样式指定生成的格式

### 通过webstorm开发工具进行编译

`--no-cache --update --style expanded $FileName$:$FileNameWithoutExtension$.css`

### 通过gulp进行集成编译

# SASS语法结构

<http://sass.bootcss.com/docs/sass-reference/>

## 1.变量的定义

```
`$+变量名 : 值` 如 $color : #ffff | $border : 1px solid red(多值)
```

## 2.嵌套语法使用

使用嵌套语法的好处，我们可以去掉样式里面的公共重复的部份

在使用嵌套语法时，如果使用: hover这种伪类选择器时，嵌套语法会保留空格，我们为元素定义伪类是不能在元素和伪类之间有空格。(ie: a: hover)。

**& : 用于引用父类名称** (如: `& &-title` => `.nav .nav-title`)

这个时候我们就必须使用一个符号 "`&: hover`" 的方式，直接使用父类名称，不保留空格。

可以在属性中使用嵌套:

```
.circle {
  background-color: red;
  background-origin: border-box;
  background-repeat: no-repeat;
  background-size: cover;
}

.circle2 {
  background :{ //注意需要添加 ":"
    color: #000;
    origin: border-box;
    repeat: no-repeat;
    size: cover;
  }
}
```

## 3.@mixin语法

mixin 是一种预先定义好的样式块，我们可以在其它的任何地方重复使用这个样式块。

语法结构:

```
@mixin 名称 ($参数..., $参数) {
  .....
}
//使用@include进行mixin的调用
@include 名称 ($参数, ... ) {
  .....
}
```

## 4.@extend 继承语法

我们通过继承来减少重复的定义

```
.mm {
  background-color: red;
  a {
    text-align: center;
  }
}

.mm2 {
  @extend .mm;
  background-origin: border-box;
}
```

当子类从父类继承后，子类可以继承父类相关的引用 ie: a的嵌套

## 5.占位符 %placeholder

```
%icon {
  transition: background-color ease .2s;
  margin: 0 .5em;
}
```

```
.error-icon {
  @extend %icon;
  /*错误图标指定的样式... */
}

.info-icon {
  @extend %icon;
  /* 信息图标指定的样式... */
}
```

#### 4.@import 引入partial样式语法

SASS扩展了@import的功能, SASS使我们可以可以在一个CSS文件里面可以引入多个文件,并且编译成一个文件; 因些我们就可以把一个大的css文件, 分为多个 partials (partials 需要以下划线开头 ie: \_header.scss), 引入时, 我们只需要在文件的头部添加 @import "header"; 这个时候, header.scss的下划线和扩展名都不需要添加了!

```
@import "header";
```

#### 5.Interpolation 把一个值插入到别一个值中

#{\$变量名}: 用于引用定义的字符串

```
$version:"0.0.1";
/* hello test comments #{ $version} */

$nns : "txt";
.nav-#{ $nns} {
  background-size: cover;
}
```

#### 6.SASS 控制语名

```
$theme : "ddd";
.testif {
  @if($theme == light) {
    background-color: lightblue;
  } @else if($theme == red){
    background-color: red;
  } @else {
    background-color: black;
  }
}

@for $i from 1 through 5 {
.col-#{ $i} {
  width:10% *$i;
}
}

@each $var in <list/map>

$list: zhangsan lisi wangwu;
$map:(red:#ddd) -> map-get($map,key)

@each $name in $list {
  .photo-#{ $name}{
    background: url("#{ $name}.jpg");
  }
}
```

#### 7.SASS 自定义函数

```
$ccs:(light:#ffffff,dark:#000000);
//定义function
@function getColor($key){
  @return map_get($ccs,$key);
}

@mixin aa(){
  //调用@function
  color: getColor(light);
}

//使用mixin
.testff {
  @include aa()
```

```
}
```

8.颜色函数-lighten 与 darken

## SASS的定义的数据类型和函数

sass -i 进入运算模式

type-of(输入值) : 查看输入的数据类型

HSLA 表示一种颜色(色相(0-360), 饱和度(0-100%), 明度(0-100%), 透明度(0-1))