**Sass 写法示例**

CSS 本身是非常强大的，但随着样式表变大，变复杂，维护 CSS 变得越来越难。这时候预处理就有用了。Sass 是一种预处理，它能让你使用一些 CSS 中没有的特性，比如：变量，嵌套（nesting），混入（mixins），继承等。这些特性能让 CSS 变的容易维护。更多描述见[官网](https://link.jianshu.com/?t=http://sass-lang.com/)。

想尝试下 Sass ，可以尝试下在线编译，点[这里](https://link.jianshu.com/?t=http://www.sassmeister.com/)。

**指定编码格式**

如果 Sass 文件中有中文，代码第一行必须加如下代码来指定编码格式，否则会报错。

@charset 'utf-8';// 不加这行，文件中有中文时会报错

**注释**

// 表示单行注释，不会输出在 CSS 中。  
/\* \*/ 表示多行注释，会输出在 CSS 中。

// 我是单行注释

/\*

@author: Joel

@describe Sass syntax overview

\*/

**变量**

变量必须以 $ 做为开头。

body {

$primary-color: #333; // 定义变量。作用域为括号。

color: $primary-color; // 使用变量

$base-line: 2px;

$base-line: 1px !default;

$base-line;// 2px。 !default 定义的变量不会覆盖前面定义的同名变量。

$gloable-red-color: #f00 !global; // 全局变量

}

// $primary-color;//报错：Undefined variable: "$primary-color"

$gloable-red-color; // #f00

**数据类型**

Sass 支持字符串，数字，颜色，数组，Map 这几种数据类型。

// 字符串

$str: 'str1' + 'str2';

@debug $str;

// 数字

4 + 5;// 9

5px + 2px;// 7px

5px - 2px;// 3px

10px \* 2; // 20px

10px \* 2px; // 报错

10px / 2;// 10px / 2。注意。

10px / 2px;// 10px / 2px。注意。

'5' == 5; // false

'5' != 5; // true

5px == 5; // true

$width: 10px;

$width / 2;// 5px

$width / 2px;// 5

// 数组。

// 数组的元素可以由空格或逗号分割。

$list: 10px 20px 10px;

nth($list, 2); // 20px。取列表中的第 n 个元素，下标从 1 开始。

$list2: 10,30,50;

list-separator($list2);//comma（逗号）。 数组的分隔符。

// 颜色

$orange: #f60;

red($orange); // 255。$orange 红色部分的值

// Map

$map: ('name': 'joel', 'gender': 'male');

map-get($map, 'name'); // 'joel'

map-get($map, 'gender'); // 'male'

**引入 Sass 或 Scss 文件**

@import 'reset';// 等效于 @import 'reset.scss';如果找不到 reset.scss 也会找\_reset.scss

**逻辑操作符**

$true: true;

$false: false;

not $true;// false

$true and $false; // false

$true or $false; // true

**条件语句**

$isCool: true;

@if $isCool {

.isCool{

color: #fff;

}

}

@if not $isCool {

} @else {

.notCool{

color: #000;

}

}

**循环**

$map: ('a': #f00, 'b': #0f0, 'c': #00f);

@each $key, $value in $map {

.section-#{$key} {

background-color: $value;

}

}

**函数**

@function add($a, $b){

@return $a + $b;

}

add(1, 2);// 3

// 参数的默认值

@function greet($greet: 'Hello', $name: 'Nobody') {

@return $greet + ',I am ' + $name;

}

greet();// Hello,I am Nobody

greet(Hi, Joel);// Hi,I am Joel

**嵌套**

**嵌套规则**

.box {

.title {// 相当于 .box .title

font-size: 18px;

padding-left: 10px;

}

.content {

padding: 10px;

.pager {

text-align: center;

}

}

// &表示外层的元素

&:hover { // 相当于 .box:hover

outline: 1px solid #ddd;

}

}

**嵌套属性**

.font {

font: {

weight: bold;// 相当于 font-bold

size: 14px;

}

}

输出

.font {

font-weight: bold;

font-size: 14px;

}

**混入(Mixins)**

// 定义

@mixin clearfix {

&::after {

content: '';

display: table;

clear: both;

}

}

.wrap {

@include clearfix;// 使用

}

// 带参数

@mixin size($width, $height: $width) {

width: $width;

height: $height;

}

.block1{

@include size(100px);

}

.block2{

@include size(100px, 200px);

}

**扩展**

扩展和混入比较类似，不同点是，扩展不支持参数。

.message {

border: 1px solid #ccc;

padding: 10px;

}

%border-box {

box-sizing: border-box;

}

.success {

@extend %border-box;

@extend .message;

border-color: #0f0;

}

.error {

@extend %border-box;

@extend .message;

border-color: #00f;

}

**插值**

如果想要在选择器或属性上用变量,必须使用插值，否则会报错。

$sel: '.a';

$attr: width;

#{$sel}{

color: #f00;

#{$attr}: 100px;

}

**在命令行输出信息**

@debug xxx