logo.png

Css预处理器

H5 web前端

[www.bufanui.com](http://www.bufanui.com)

css痛点：

1. Div1 .d1 p span

Div1.d1 p span i

---代码重复，尤其是后代选择器

1. 比如bgc :red ; color:green;

Bgc2:red;

能不能像js一样，预定义一个变量，只修改变量就可以全局修改呢？

1. 能不能使用运算符？

# css预处理器

## 1.1 什么是 CSS 预处理器

CSS 预处理器是一种语言，用来为 CSS 增加一些编程的的特性，无需考虑浏览器的兼容性问题，例如你可以在 CSS 中使用变量、简单的程序逻辑、函数等等在编程语言中的一些基本技巧，可以让你的 CSS 更见简洁，适应性更强，代码更直观等诸多好处。

## 1.2 有哪些css预处理器

当下比较流行的预处理器有Sass、Less CSS、Stylus。

目前Sass用的最多，然后是less。比如bootstrap2就是用less实现的，bootstrap3改成了sass实现。

# 二、Less

中文官网：http://www.bootcss.com/p/lesscss/

## 2.1安装less

$ npm install -g less

## 2.2 命令行使用less

* 直接编译结果

$ lessc styles.less

* 将结果输出到固定文件

$ lessc styles.less > styles.css

* 若要输出压缩过的 CSS，只需添加 -x 选项。

## 2.3 客户端js编译less

## 2.4 语法

### 2.4.1 变量

@bgColor:red;

### 2.4.2 混合

### 2.4.3 带参数混合

2.4.5 带默认值的混合

…

# Sass

中文网址：https://www.sass.hk/

## 3.1 安装

### 3.1.1 下载ruby

window下安装SASS首先需要安装Ruby，先从官网下载Ruby并安装。

Ruby官网：<https://rubyinstaller.org/downloads/>

### 3.1.2 查看安装成功

//更新sass

~~gem update sass~~

//查看sass版本

sass -v

//查看sass帮助

sass -h

=========================添加gem源===========

//1.删除原gem源

gem sources --remove https://rubygems.org/

//2.添加国内淘宝源

gem sources -a [https://gems.ruby-china.com](https://gems.ruby-china.com/)

//3.打印是否替换成功

gem sources -l

//4.更换成功后打印如下

\*\*\* CURRENT SOURCES \*\*\*

<https://ruby.taobao.org/>

=====================安装sass============

//安装如下(如mac安装遇到权限问题需加 sudo gem install sass)

gem install sass

gem install compass

## 3.2 sass文件后缀

**~~.sass~~  不推荐**

|  |
| --- |
| //文件后缀名为sass的语法  body  background: #eee  font-size:12px  p  background: #0982c1 |

**.scss scss完全兼容css**

|  |
| --- |
| //文件后缀名为scss的语法  body {  background: #eee;  font-size:12px;  }  p{  background: #0982c1;  } |

## 3.3 编译sass

### 3.3.1 命令行编译

|  |
| --- |
| //单文件转换命令  sass input.scss output.css  //单文件监听命令  sass --watch input.scss:output.css  //如果你有很多的sass文件的目录，你也可以告诉sass监听整个目录：  sass --watch app/sass:public/stylesheets |

### 3.3.2 命令行编译配置选项

命令行编译sass有配置选项，如编译过后css排版、生成调试map、开启debug信息等，可通过使用命令sass -v查看详细。

//编译格式 简单的压缩格式

* sass --watch input.scss:output.css --style compact

//编译添加调试map

* sass --watch input.scss:output.css --sourcemap

//选择编译格式并添加调试map

* sass --watch input.scss:output.css --style expanded --sourcema

//开启debug信息

* sass --watch input.scss:output.css --debug-info

# sass语法

## 4.1 变量

### 4.1.1 变量声明

$ sass使用$符号来标识变量

|  |
| --- |
| $highlight-color: #F90; |

4.1.2 引用

|  |
| --- |
| $highlight-color: #F90;  $highlight-border: 1px solid $highlight-color;  .selected {  border: $highlight-border;  }  //编译后  .selected {  border: 1px solid #F90;  } |

## 4.2 css嵌套

### 4.2.1 嵌套语法

|  |
| --- |
| #content {  article {  h1 { color: #333 }  p { margin-bottom: 1.4em }  }  #content aside { background-color: #EEE }  } |

### 4.2.2 父选择器的标识符&

|  |
| --- |
| article a {  color: blue;  &:hover { color: red }  } |

### 4.2.3 >、+和~

|  |
| --- |
| article {  ~ article { border-top: 1px dashed #ccc }  > section { background: #eee }  dl > {  dt { color: #333 }  dd { color: #555 }  }  nav + & { margin-top: 0 }  } |

### 4.2.4 嵌套属性

|  |
| --- |
| nav {  border: {  style: solid;  width: 1px;  color: #ccc;  }  } |

## 4.3 @import

### 4.3.1 导入公共样式

可以省略.scss后缀

### 4.3.2 变量冲突

反复声明一个变量，只有最后一处声明有效且它会覆盖前边的值。

### ~~4.3.3 嵌套导入~~

|  |
| --- |
| .blue-theme {@import "blue-theme"}  //生成的结果跟你直接在.blue-theme选择器内写\_blue-theme.scss文件的内容完全一样。  .blue-theme {  aside {  background: blue;  color: #fff;  }  } |

### 4.3.4 原生的css导入

因为sass的语法完全兼容css，所以把原始的css文件改名为.scss后缀，即可直接导入了。

## 4.4 注释

|  |
| --- |
| body {  color: #333; // 这种注释内容不会出现在生成的css文件中  padding: 0; /\* 这种注释内容会出现在生成的css文件中 \*/  } |

## 4.5 混合器

### 4.5.1 混合器语法

混合器使用@mixin标识符定义。

使用@include 引用混合器

|  |
| --- |
| @mixin rounded-corners {  -moz-border-radius: 5px;  -webkit-border-radius: 5px;  border-radius: 5px;  }  // 使用  notice {  background-color: green;  border: 2px solid #00aa00;  @include rounded-corners;  } |

### 4.5.2 带参数混合器

### 4.5.3 带默认参数的混合器

## 4.6 继承

|  |
| --- |
| //通过选择器继承继承样式  .error {  border: 1px solid red;  background-color: #fdd;  }  .seriousError {  @extend .error;  border-width: 3px;  } |

注意：不要继承后代选择器。