**Sass学习路线**

**课程介绍**

这不是一个CSS教学过程，但会有部分CSS的讲解

将为大家展示Sass编写CSS的便捷性和可维护性

**学习对象**

1、有CSS基础（Sass基于CSS），最好是HTML+CSS+JS基础（Sass具有编程语言的特点）

2、对前端有兴趣，希望编写出高质量的CSS代码

3、喜欢尝试新技术带来的便利性

**培训目标**

1、了解Sass与Compass

2、能使用Sass编写CSS代码

3、能合理利用Compass编写优质的CSS代码

4、熟练使用Sass的命令，如自动压缩CSS

5、轻松的写出最新的跨浏览器的CSS代码

6、能利用Sass进行大型前端项目开发

**开发工具**

Chrome --流行的浏览器，及前端开发调试工具

WebStorm --强大的跨平台前端集成开发环境

Sublime Text --神器级别的代码编辑器，如vim般强大，而上手难度极低。

**课后任务**

安装好所需的开发工具，包含浏览器，IDE或者是适合的文本编辑器

**Sass简介**

**为什么需要CSS预处理器？**

对一般的项目直接用CSS就可以完成，但是对于一些大项目，这个时候如果单纯的写CSS相对复杂而且不太好维护。

**为什么使用Sass？**

使用变量

自动转换RGBA颜色值

忘记浏览器前缀

嵌套规则

media query更简单

自动压缩CSS

**什么是Sass**

[http://sass-lang.com/（官方网站）](http://sass-lang.com/%EF%BC%88%E5%AE%98%E6%96%B9%E7%BD%91%E7%AB%99%EF%BC%89)

**Sass与CSS**

CSS并不能算是一门真正意义上的“编程”语言，无法完成嵌套、继承、设置变量等工作

解决CSS的不足，开发者想到了编写一种对css进行预处理的“中间语言”

**sass与.scss**

最初它是为了配合haml而设计，所以和haml有着一样的缩进式风格

从第三代开始，保留缩进式风格，完全向下兼容普通的css代码

**课后任务**

1、了解sass和scss的区别

2、将一个css文件修改为sass

**Compass简介**

**什么是Compass**

Compass是Sass的工具库

在Sass的基础上封装了一系列有用的模块和模板，补充Sass的功能

Compass与Sass的关系类似于jQucry与javaScript的关系

**Sass与Compass能做什么**

使用变量

自动转换RGBA颜色

忘记浏览器前缀

嵌套规则

Media query更简单

自动压缩CSS

**Sass与compass的安装**

依赖ruby。所以要先安装Ruby

如何安装Ruby呢？在Windows下通过RubyInstaller来安装

安装Compass：gem install compass

**Sass与Compass**

搭建项目骨架

编译Sass

提供常用模块

**Sass,Less与Stylus**

**背景**

**Sass：**

Sass是对CSS（层叠样式表）的语法的一种扩充，诞生于2007年，最早也是最成熟的一款CSS预处理器语言，它可以使用变量、常量、嵌套、混入、函数等功能，可以更有效有弹性的写出CSS。Sass最后还是会编译出合法的CSS让浏览器使用，也就是说它本身的语法并不太容易让浏览器识别，因为它不是标准的CSS格式，在它的语法内部可以使用动态变量等，所以它更像一种极简单的动态语言。

Sass已经有了两套语法规则：一个依旧是用缩进作为分隔符来区分代码块的；另一套规则和CSS一样采用了大括号（｛｝）作为分隔符。后一种语法规则又名SCSS，在Sass3之后的版本都支持这种语法规则。

**Less：**

2009年开源的一个项目，受Sass的影响较大，但又使用CSS的语法，让大部分开发者和设计师更容易上手。LESS提供了多种方式能平滑的将写好的代码转化成标准的CSS代码，在很多流行的框架和工具中已经能经常看到LESS的身影了。

其实LESS是Alexis Sellier受Sass的影响创建的一个开源项目。当时SASS采用了缩进作为分隔符来区分代码块，而不是CSS中广为使用的大括号（｛｝）。为了让CSS现有的用户使用起来更佳方便，Alexis开发了LESS并提供了类似CSS的书写功能。

**Stylus：**

Stylus，2010年产生，来自于Node.js社区，主要用来给Node项目进行CSS预处理支持，在此社区之内有一定支持者，在广泛的意义上人气还完全不如Sass和LESS。

Stylus被称为是一种革命性的新语言，提供一个高效、动态、和使用表达方式来生成CSS，以供浏览器使用。Stylus同时支持缩进和CSS常规样式书写规则。

**官网**

Sass  http://www.sass-lang.com/

Less  http://lesscss.org/

Stylus  http://learnboost.github.io/stylus/

**安装**

**1、Sass需要安装Ruby，然后通过gem安装sass**

**2、Less有两种安装方式：**

客户端安装：

引入less.js，然后就可以直接使用.less文件

<link rel="stylesheet/less" type="text/css" href="文件路径/styles.less">

<script src="文件路径/less.js" type="text/javascript"></script>

服务器安装：

先安装node，然后使用npm安装less。

**3、Stylus的安装类似于Less的服务端安装**

**结论**

1、三者都是开源项目；

2、Sass诞生是最早也是最成熟的CSS预处理器，有Ruby社区和Compass支持；Stylus早期服务器Node JS项目，在该社区得到一定支持者；LESS出现于2009年，支持者远超于Ruby和Node JS社区；

3、Sass和LESS语法较为严谨、严密，而Stylus语法相对散漫，其中LESS学习起来更快一些，因为他更像CSS的标准；

4、Sass和LESS相互影响较大，其中Sass受LESS影响，已经进化到了全面兼容CSS的SCSS；

5、Sass和LESS都有第三方工具提供转译，特别是Sass和Compass是绝配；

6、Sass、LESS和Stylus都具有变量、作用域、混合、嵌套、继承、运算符、颜色函数、导入和注释等基本特性，而且以“变量”、“混合”、“嵌套”、“继承”和“颜色函数”称为五大基本特性，各自特性实现功能基本相似，只是使用规则上有所不同；

7、Sass和Stylus具有类似于语言处理的能力，比如说条件语句、循环语句等，而LESS需要通过When等关键词模拟这些功能，在这一方面略逊一层；

**Sass命令行与注释**

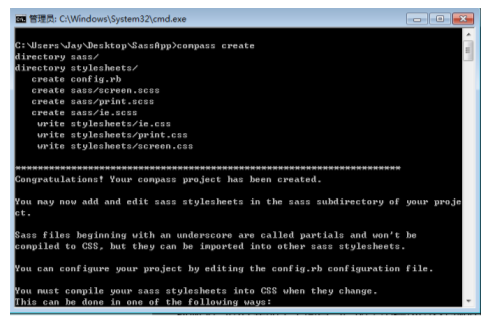
**Sass基础**

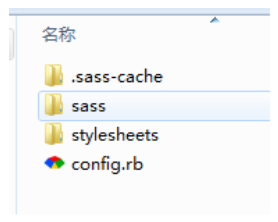
**1.创建工程**

使用Sass创建

利用Compass创建工程：

打开指定目录的控制台窗口，输出指令“copmpass create”即可创建。同时，可以带参数创建：compass create --bare --sass-dir "sass" --css-dir "css" --images-dir "img" --javascripts-dir "js"





**使用命令行**

**编译Sass:**

sass <sass file> <css file>

可以指定输出风格：sass <sass file> <css file> --style [nested|expanded|compact|compressed]

**监视Sass文件:**

sass --watch <sass file>:<css file>

**监视文件夹:**

sass --watch <sass folder>:<css folder>

**编译Sass:**

compass compile此命令为Compass命令。

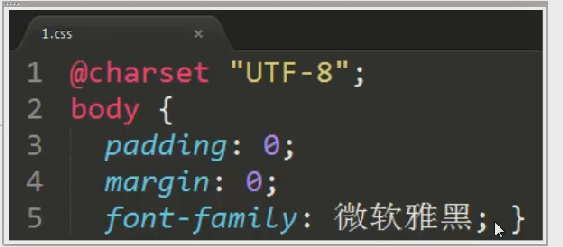
此命令只会编译有变化是Sass文件，如果要强制编译所有的Sass文件，可以使用：compass compile –force

**监视文件夹:**

compass watch

**四种输出风格：**

nested



expanded



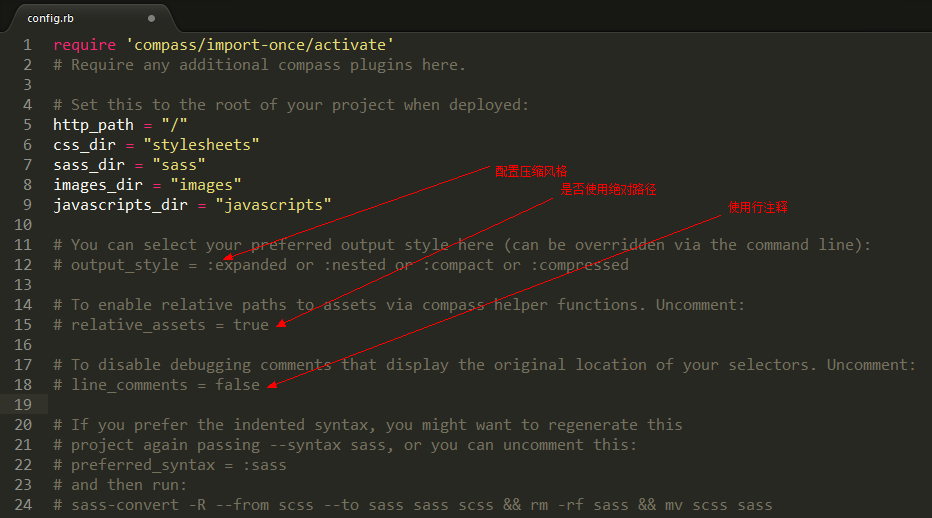
Compact



compressed



**理解config.rb**



**Sass的注释语法**

**注释：**

单行注释，不会生成到css文件中

**/\*\*/注释：**

标准CSS注释，会被编译到CSS文件中

/\*!xxx\*/  重要注释，压缩模式也会编译到css文件。

**中文注释：**

注意事项：

Encoding.default\_external = Encoding.find('utf-8')

engine.rb文件路径：

C:\Ruby22-x64\lib\ruby\gems\2.2.0\gems\sass-3.4.15\lib\sass

**代码部分**

body{

  background: lightgray;

  font-size: 14px;

  // this is the header style.

  header{

    font-weight: bold;

  }

  footer{/\*! This is the footer style.重要注释，压缩模式也会编译到css文件\*/

    text-align: center;

  }

}

**课后任务**

1、创建一个Sass项目

2、使用命令行转换sass文件到css

3、使用命令行相互转换sass和scss

4、使用命令行自动编译sass文件

**Sass变量与导入**

**Sass变量**

**1.局部变量**

**2.全局变量**

$color: red !global;

这样即使变量是在某一个代码块中定义，也可以在此代码块外使用

**3.变量默认值**

$color: red !default;

**4.多值变量**

$colors: red blue green;

color: nth($colors, 1);

$map: (color : red , wdth : 100px); // 其中的key不要加$符号，否则使用时会报错

map-get($map , 1);

**5.变量特殊用法**

1. 变量用在属性或者选择器上，使用#{变量名}

$className : container;

.#{$className}{

Color:red;

}

1. 变量用中横线，还是下划线，取决于你的喜好，都代表同一个变量

$my\_class : box; 等价于 $my-class : box;

**样式导入**

**1.部分文件**

专门为导入而编写的 sass文件，并不需要生成对应的独立文件，这个时候，sass如何判别呢？

依靠约定，sass局部文件的文件名以下划线开头 : \_part.scss

局部文件可以被多个不同的文件引用。当一些样式需要在多个页面甚至多个项目中使用时，这非常有用。在这种情况下，有时需要在你的样式表中对导入的样式稍作修改，sass有一个功能刚好可以解决这个问题，即默认变量值。

**2.嵌套导入**

**3.原生CSS导入**

（1）被导入文件的名字以.css结尾；

（2）被导入文件的名字是一个URL地址（比如http://xxx/css.css）

（3）被导入文件的名字是CSS的url()值。

如：

@import “1.css”;

@import “http://www.xxx”;

@import “url(1.css)”;

正确导入部分文件：

@import “part”;

@import “\_part.scss”;

可以写后缀和前面的下滑先线，也可以不写，完全看你喜欢

**课后任务**

利用所学知识，实现一组提示框的样式：

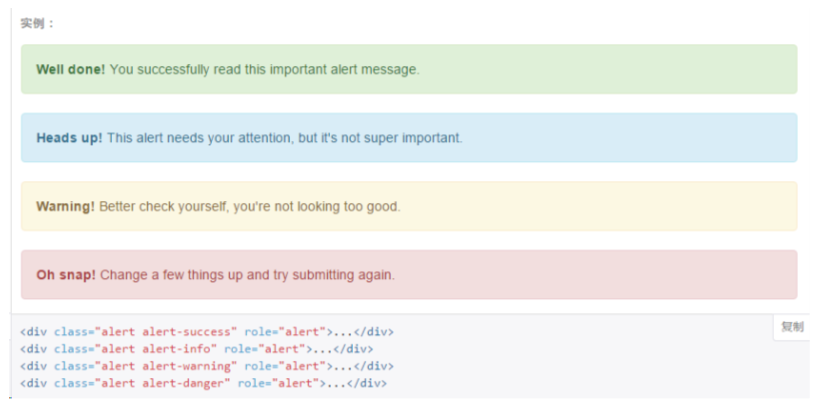
alert-success

alert-info

alert-primary

alert-warning

**实例**



**Sass继承与嵌套**

**嵌套**

**1．选择器嵌套**

body{

background-color:red;

header{

background-color:green;

}

}

**2．属性嵌套**

footer{

background:{

color : red;

size : 100% 50%;

}

}

**3．&--引用父选择器**

body{

background-color:red;

header{

background-color:green;

}

a{

&:hover{

Background-color: orange;

}

}

}

**4．@at-root,跳出嵌套**

默认@at-root只会跳出选择器嵌套，而不能跳出@media或@support，如果要跳出这两种，则需使用@at-root (without: media)，@at-root (without: support)。这个语法的关键词有四个：all（表示所有），rule（表示常规css），media（表示media），support（表示support，因为@support目前还无法广泛使用，所以在此不表）。我们默认的@at-root其实就是@at-root (without:rule)。

Body{

@media screen and (max-width:600px){

@at-root(without : media){

.container{

Background: blue;

}

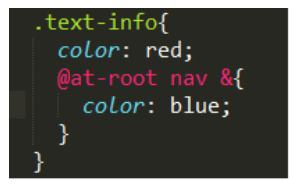
}

}

}

则在css中就不会

**5．@at-root与&**



**继承**

**1.简单继承**

**2.多继承**

（1）多个@extend

（2）@extend selector1 selector2

**3.链型继承**

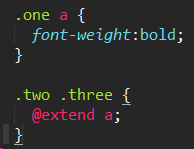
**4.继承的局限性**

虽然能继承的选择器数量很多，但也有很多是不被支持的。如包含选择器（.one .two）或者是相邻兄弟选择器(.one+.two)

如果继承的元素是a，恰巧这个元素a又有hover状态的形式，那么hover状态也会被继承

**5.继承交叉合并**

没有在同一个父级下，会产生交叉合并的编译结果



**6.继承的作用域**

继承在指令中是有作用域问题的，继承是无法使在指令如(@media)之外的选择器继承的，要是继承就只能写在指令中。



**占位选择器%**

优势在于：如果不调用则不会有任何多余的css文件，避免了以前在一些基础的文件中预定义了很多基础的样式，然后实际应用中不管是否使用了@extend去继承相应的样式，都会解析出来所有的样式。占位选择器以%标识定义，通过@extend调用。

**课后任务**

设计一个小页面，使用学到的Sass基础知识来构建样式

**Sass数据类型与mixin**

**数据类型**

**1.Number**

http://sass-lang.com/documentation/Sass/Script/Functions.html#number\_functions

**2.String**

http://sass-lang.com/documentation/Sass/Script/Functions.html#string\_functions

**3.List**

http://sass-lang.com/documentation/Sass/Script/Functions.html#list-functions

**4.Map**

http://sass-lang.com/documentation/Sass/Script/Functions.html#map-functions

**5.Color**

**6.Boolean**

**7.Null**

**变量操作**

**1. == ,!=**

所有数据类型都支持

**2. <,>,<=,>=**

仅仅支持数字类型

**3. +,-,\*,/,%**

**Mixin**

1.简单示例

2.传递多值参数

3.传递内容

**代码部分**

1.

/\*! 数字类型 \*/

$n1: 1.2;

$n2: 12;

$n3: 14px;

p{

  font-size: $n3;

}

/\*! 字符串类型\*/

$s1: container;

$s2: 'container';

$s3: "container";

.#{$s3}{

  font-size: $n3;

}

/\*! Bool类型\*/

$bt: ture;

$bf: false;

/\*! Null类型\*/

$null: null;

body{

  @if $null == null{

    color: red;

  }

}

/\*! 颜色类型\*/

$c1: blue;

$c2: #fff;

$c3: rgba(255,255,0,0.5);

body{

  color: $c3;

}

$width1: 10px;

$width2: 15px;

span{

  width: $width1 - $width2;

}

a:hover{

  cursor: e + -resize;

}

a:before{

  content: "A" + bc;

}

a:before{

  content: A + 'bc';

}

p{

  padding: 3px + 4px auto;

}

$version: 3;

p:before{

  content: 'Hello,Sass #{$version}';

}

p{

  font: 20px / 10px;

  width: $width2 / 2;

  width: round($width2) / 2;

  height: (100px / 2);

}

2.

@mixin cont($color: red, $fontSize: 14px){

  color: $color;

  font-size: $fontSize;

}

body{

  @include cont($fontSize: 18px);

}

@mixin box-shadow($shadow...){

  -moz-box-shadow: $shadow;

  -webkit-box-shadow: $shadow;

  box-shadow: $shadow;

}

.shadows{

  @include box-shadow(0px 4px 4px #555, 2px 6px 10px #6dd3ee);

}

@mixin style-for-iphone{

  @media only screen and (min-device-width: 320px) and (max-device-width:568px){

    @content;

  }

}

@include style-for-iphone{

  font-size: 12px;

}

**课后任务**

1、定义一组变量，用到Sass中所有的数据类型；

2、使用这组变量进行四则运算；

3、实现一个简单的media query mixin。

# Sass函数与调试

**内置函数**

1. http://sass-lang.com/documentation/Sass/Script/Functions.html

2. rgb(),rgba()

3. lighten(),darken()

4. str-length(),str-index()

**自定义函数**

**@debug、@warn、@error**

**课后任务**

1、学习5个内置函数，并列出它们的常规用法。

2、自定义一个函数，通过给定的层次，获取特定的z-index。

@function getzIndex($layer: default){

}

$layer可选：default,modal,dropdown,grid

**Sass条件控制**

**@if**

**1.三目运算**

@if($condition, $condition\_true, $condition\_false)

**2. @if...@else if...@else**

**@for**

@for $i from 1 through 5 {

  .span#{$i} { width: 20% \* $i; }

}

**@while**

$i: 5;

@while $i > 0 {

  .span#{$i} { width: 2em \* $i; }

  $i: $i - 2;

**@each**

**1.常规遍历**

@each $state in default, primary, info, success, warning, danger {

  .text-#{$state} {

    width: 100%;

  }

}

**2.遍历List**

@each $animal, $color, $cursor in (puma, black, default),

                                  (sea-slug, blue, pointer),

                                  (egret, white, move) {

  .#{$animal}-icon {

    background-image: url('/images/#{$animal}.png');

    border: 2px solid $color;

    cursor: $cursor;

  }

}

**3.遍历Map**

@each $key, $value in (h1: 2em, h2: 1.5em, h3: 1.2em) {

  #{$key} {

    font-size: $value;

  }

}

**代码部分**

1.

@function getzIndex($layer: default){

  $zindexMap: (default: 1, modal: 1000, dropdown: 500, grid: 300);

  $zindex: 1;

  @if map-has-key($zindexMap, $layer){

    $zindex: map-get($zindexMap, $layer);

  }

  @return $zindex;

}

@debug getzIndex('afd');

//$layer可选：default,modal,dropdown,grid

2.

$screenWidth: 500;

body{

  color: if($screenWidth > 768, blue, red);

}

@if $screenWidth > 768{

  body{

    color: red;

  }

}

@else if $screenWidth >400{

  span{

    color: green;

  }

}

@else{

  p{

    color: blue;

  }

}

@for $i from 1 to 5{

  .span#{$i}{

    width: 20% \* $i;

  }

}

$j: 6;

@while $j > 0{

  .p#{$j}{

    width: 20% \* $j;

  }

  $j: $j - 3;

}

$k: 1;

@each $c in blue, red, green {

  .div#{$k}{

    color: $c;

  }

  $k: $k+1;

}

@each $key,$color in (default, blue), (info, green), (danger, red){

  .text-#{$key}{

    color: $color;

  }

}

@each $key, $value in (default: blue, info: green, danger: red){

  .label-#{$key}{

    color: $value;

  }

}

**课后任务**

1、使用自定义function和@each实现栅格布局。

@function buildLayout($num: 5){

}

**结果：**

col1{width: 20%}

col2{width: 40%}

...

**Sass基础项目结构**

以下实战以Sass官网为例

**基础项目结构**

**1.base**

**2.components**

**3.helpers**

（1）variables

（2）functions

（3）mixins

**4.layout**

（1）header

（2）footer

**5.pages**

（1）home

（2）about

（3）……

**6.themes**

**课后任务**

搭建Sass官网的项目环境

**Sass自动化构建**

**自动化构建**

**1. gulp构建工具**

**2. gulp-compass**

https://www.npmjs.com/package/gulp-compass

**3. browser-sync**

**4. Css3：linear-gradient()**

https://developer.mozilla.org/en-US/docs/Web/CSS/linear-gradient

**代码部分**

<!DOCTYPE html>

<html>

<head>

  <meta charset="UTF-8">

  <title>Sass:Syntactiically Awesome Style Sheets</title>

  <link rel="stylesheet" href="css/main.css">

</head>

<body>

  <div>

      <div></div>

      <div></div>

      <div></div>

      <div></div>

      <div></div>

      <div></div>

      <div></div>

      <div></div>

    </div>

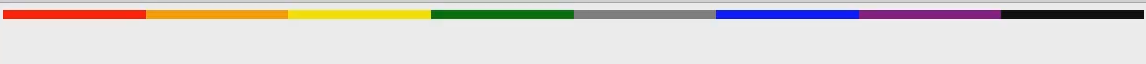
    <div style="clear:both;"></div>

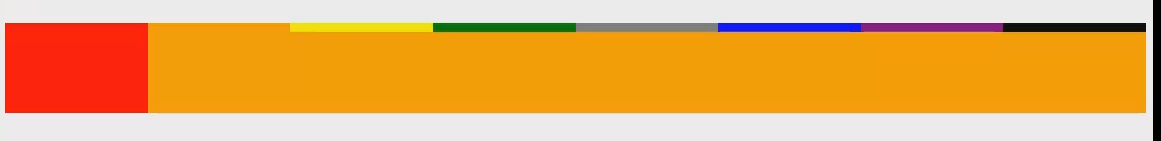
    <div></div>

</body>

</html>

**运行结果**





**课后任务**

利用Sass特性，来编写动态渐变代码

**Sass网站首页布局**

**网站首页布局**

**1.reset**

**2.normalized**

（1）html5

（2）base

（3）links

（4）typography

（5）embeds

（6）groups

（7）forms

（8）tables

**3.susy**

**4.breakpoint**

**5.页面Layout**

**代码部分**

<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <meta content="IE=edge" http-equiv="X-UA-Compatible">

    <title>Sass: Syntactically Awesome Style Sheets</title>

    <meta content="Syntatically Awesome Style Sheets" name="description">

    <meta content="width=device-width, initial-scale=1" name="viewport">

    <link rel="stylesheet" href="css/main.css">

  </head>

  <body>

    <div>

      <div class="alert stickers">

        <div>

          <p> <strong>Sass stands with trans detainees.</strong>

            <a href="https://www.tfaforms.com/369883">Donate to the Trans Justice Funding Project

            </a>.

          </p>

        </div>

      </div>

      <div></div>

      <header role="banner">

        <div>

          <h1>

          <a href="/">

          <img height="48" alt="Sass" src="assets/img/logos/logo-b6e1ef6e.svg"></a>

          </h1>

          <nav class="navigation collapse" role="navigation">

            <ul>

              <li><a href="/install">Install</a></li>

              <li><a href="/guide">Learn Sass</a></li>

              <li><a href="http://blog.sass-lang.com/">Blog</a></li>

              <li><a href="../documentation/file.SASS\_REFERENCE.html">Documentation</a></li>

              <li><a href="/community">Get Involved</a></li>

              <li><a href="/libsass">libSass</a></li>

            </ul>

          </nav>

          <div>

            <button data-target=".navigation" data-toggle="collapse" type="button"> <span></span> <span></span> <span></span> </button>

          </div>

        </div>

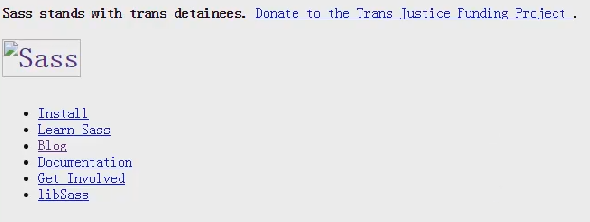
      </header>

    </div>

  </body>

</html>

**运行结果：**



**Sass网站首页实现**

**网站首页的实现**

**一.reset**

**二.normalized**

（1）html5

（2）base

（3）links

（4）typography

（5）embeds

（6）groups

（7）forms

（8）tables

**代码部分**

<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <meta content="IE=edge" http-equiv="X-UA-Compatible">

    <title>Sass: Syntactically Awesome Style Sheets</title>

    <meta content="Syntatically Awesome Style Sheets" name="description">

    <meta content="width=device-width, initial-scale=1" name="viewport">

    <link rel="stylesheet" href="css/main.css">

  </head>

  <body>

    <div>

      <div class="alert stickers">

        <div>

          <p> <strong>Sass stands with trans detainees.</strong>

            <a href="https://www.tfaforms.com/369883">Donate to the Trans Justice Funding Project

            </a>.

          </p>

        </div>

      </div>

      <div></div>

      <header role="banner">

        <div>

          <h1>

          <a href="/">

          <img height="48" alt="Sass" src="images/logo.svg"></a>

          </h1>

          <nav class="navigation collapse" role="navigation">

            <ul>

              <li><a href="/install">Install</a></li>

              <li><a href="/guide">Learn Sass</a></li>

              <li><a href="http://blog.sass-lang.com/">Blog</a></li>

              <li><a href="../documentation/file.SASS\_REFERENCE.html">Documentation</a></li>

              <li><a href="/community">Get Involved</a></li>

              <li><a href="/libsass">libSass</a></li>

            </ul>

          </nav>

          <div>

            <button data-target=".navigation" data-toggle="collapse" type="button"> <span></span> <span></span> <span></span> </button>

          </div>

        </div>

      </header>

    </div>

  </body>

</html>

**运行结果**



**课后任务**

根据自己的理解，完成首页公共样式的编写

**Sass网站公共组件实现**

**公共组件**

1.alert

2.banner

3.page-header

4.section-bottom

5.contentinfo

**代码部分**

.banner{

margin;0;

@include padding-leader(0.5);

.size-brand{

@include span(3);

text-align;left;

margin;0;

img{

height;96px;

}

}

.navigation{

@include span(9);

Text-align;right;

Ul{

margin;0;

li{

list-style;none;

display;inline-block;

line-height;96px;

}

}

a{

padding;0.75em 1em;

color;#c69;

font-weight;600;

text-lrunsform;uppercase;

text-decoration;none;

&;visited{

Color;#b37399;

}

&;hover,&;focus{

Color;#036;

}

}

}

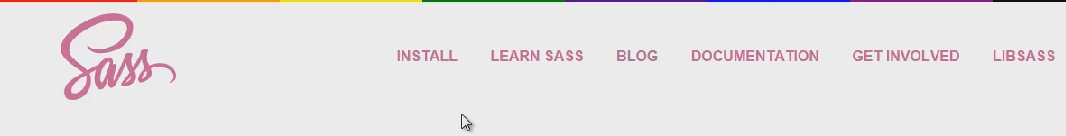
.banner-loggle{

Display;none;

}

}

**运行结果：**



**课后任务**

将公共组件中的样式进行整合，提取出公共部分放到\_base.scss中

**Sass网站内容页实现**

**通用布局**

**避免多层嵌套**

**代码部分**

<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <meta content="IE=edge" http-equiv="X-UA-Compatible">

    <title>Sass: Syntactically Awesome Style Sheets</title>

    <meta content="Syntatically Awesome Style Sheets" name="description">

    <meta content="width=device-width, initial-scale=1" name="viewport">

    <link rel="stylesheet" href="css/main.css">

  </head>

  <body>

    <div>

      <div class="alert stickers">

        <div>

          <p> <strong>Sass stands with trans detainees.</strong>

            <a href="https://www.tfaforms.com/369883">Donate to the Trans Justice Funding Project

            </a>.

          </p>

        </div>

      </div>

      <div></div>

      <header role="banner">

        <div>

          <h1>

          <a href="/">

          <img height="48" alt="Sass" src="images/logo.svg"></a>

          </h1>

          <nav class="navigation collapse" role="navigation">

            <ul>

              <li><a href="/install">Install</a></li>

              <li><a href="/guide">Learn Sass</a></li>

              <li><a href="http://blog.sass-lang.com/">Blog</a></li>

              <li><a href="../documentation/file.SASS\_REFERENCE.html">Documentation</a></li>

              <li><a href="/community">Get Involved</a></li>

              <li><a href="/libsass">libSass</a></li>

            </ul>

          </nav>

          <div>

            <button data-target=".navigation" data-toggle="collapse" type="button"> <span></span> <span></span> <span></span> </button>

          </div>

        </div>

      </header>

      <div>

        <div>

          <h1> CSS with superpowers </h1>

        </div>

      </div>

      <div class="alert release">

        <div>

          <ul>

            <li>Current Release: <span>Selective Steve (3.4.17)</span></li>

            <li><a href="documentation/file.SASS\_CHANGELOG.html">Release Notes</a></li>

            <li><a href="https://github.com/sass/sass">Fork on Github</a></li>

          </ul>

        </div>

      </div>

      <section>

        <div>

          <ul>

            <li> <h3>CSS Compatible</h3> <p> Sass is completely compatible with all versions of CSS. We take this compatibility seriously, so that you can seamlessly use any available CSS libraries. </p> </li>

            <li> <h3>Feature Rich</h3> <p> Sass boasts more features and abilities than any other CSS extension language out there. The Sass Core Team has worked endlessly to not only keep up, but stay ahead. </p> </li>

            <li> <h3>Mature</h3> <p> Sass has been actively supported for over 8 years by its loving Core Team. </p> </li>

            <li> <h3>Industry Approved</h3> <p> Over and over again, the industry is choosing Sass as the premier CSS extension language. </p> </li>

            <li> <h3>Large Community</h3> <p> Sass is actively supported and developed by a consortium of several tech companies and hundreds of developers. </p> </li>

            <li> <h3>Frameworks</h3> <p> There are endless number of frameworks built with Sass. <a href="http://compass-style.org/">Compass</a>, <a href="http://bourbon.io/">Bourbon</a>, and <a href="http://susy.oddbird.net/">Susy</a> just to name a few. </p> </li>

          </ul>

        </div>

      </section>

    </div>

    <footer role="contentinfo">

      <nav>

        <ul> <li> <p>Sass © 2006–2015 <a href="http://www.hamptoncatlin.com/">Hampton Catlin</a>, <a href="http://nex-3.com/">Natalie Weizenbaum</a>, <a href="http://chriseppstein.github.io/">Chris Eppstein</a>, and numerous contributors.</p> <p>It is available for use and modification under the <a href="http://sass-lang.com/docs/yardoc/file.MIT-LICENSE.html">MIT License</a>.</p> </li> <li> <ul> <li><a href="https://github.com/sass/sass">Sass Repo</a></li> <li><a href="https://github.com/sass/sass-site">Website Repo</a></li> <li><a href="/styleguide">Style Guide</a></li> <li><a href="/community-guidelines">Community Guidelines</a></li> </ul> </li> <li> <a data-show-count="false" data-size="large" href="https://twitter.com/SassCSS">Follow @SassCSS</a></li> </ul>

      </nav> </footer>

    </body>

  </html>

**课后任务**

完成一个Sass官网页面的完全模仿

**Sass网站响应式布局**

**屏幕尺寸**

1.<768

2.>=768

3.>=992

4.>=1200

**代码部分**

<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <meta content="IE=edge" http-equiv="X-UA-Compatible">

    <title>Sass: Syntactically Awesome Style Sheets</title>

    <meta content="Syntatically Awesome Style Sheets" name="description">

    <meta content="width=device-width, initial-scale=1" name="viewport">

    <link rel="stylesheet" href="css/main.css">

  </head>

  <body>

    <div>

      <div class="alert stickers">

        <div>

          <p> <strong>Sass stands with trans detainees.</strong>

            <a href="https://www.tfaforms.com/369883">Donate to the Trans Justice Funding Project

            </a>.

          </p>

        </div>

      </div>

      <div></div>

      <header role="banner">

        <div>

          <h1>

          <a href="/">

          <img height="48" alt="Sass" src="images/logo.svg"></a>

          </h1>

          <nav class="navigation collapse" role="navigation">

            <ul>

              <li><a href="/install.html">Install</a></li>

              <li><a href="/guide.html">Learn Sass</a></li>

              <li><a href="http://blog.sass-lang.com/">Blog</a></li>

              <li><a href="../documentation/file.SASS\_REFERENCE.html">Documentation</a></li>

              <li><a href="/community.html">Get Involved</a></li>

              <li><a href="/libsass.html">libSass</a></li>

            </ul>

          </nav>

          <div>

            <button data-target=".navigation" data-toggle="collapse" type="button"> <span></span> <span></span> <span></span> </button>

          </div>

        </div>

      </header>

      <div>

        <div>

          <h1> Install Sass </h1>

        </div>

      </div>

      <div class="body install">

        <div>

          <div>

            <main role="main">

            <h2>There are a couple of ways to start using Sass:</h2>

            <ol>

              <li> <h3>Applications</h3>

                <p><img height="160" alt="Mouse" src="images/install\_1.svg"></p>

              </li>

              <li> <h3>Command Line</h3>

                <p><img height="160" alt="Keyboard" src="images/install\_2.svg"></p>

              </li>

            </ol>

            </main>

          </div>

        </div>

      </div>

      <div class="alert release">

        <div>

          <ul>

            <li>Current Release: <span>Selective Steve (3.4.17)</span></li>

            <li><a href="documentation/file.SASS\_CHANGELOG.html">Release Notes</a></li>

            <li><a href="https://github.com/sass/sass">Fork on Github</a></li>

          </ul>

        </div>

      </div>

    </div>

    <footer role="contentinfo">

      <nav>

        <ul> <li> <p>Sass © 2006–2015 <a href="http://www.hamptoncatlin.com/">Hampton Catlin</a>, <a href="http://nex-3.com/">Natalie Weizenbaum</a>, <a href="http://chriseppstein.github.io/">Chris Eppstein</a>, and numerous contributors.</p> <p>It is available for use and modification under the <a href="http://sass-lang.com/docs/yardoc/file.MIT-LICENSE.html">MIT License</a>.</p> </li> <li> <ul> <li><a href="https://github.com/sass/sass">Sass Repo</a></li> <li><a href="https://github.com/sass/sass-site">Website Repo</a></li> <li><a href="/styleguide">Style Guide</a></li> <li><a href="/community-guidelines">Community Guidelines</a></li> </ul> </li> <li> <a data-show-count="false" data-size="large" href="https://twitter.com/SassCSS">Follow @SassCSS</a></li> </ul>

      </nav> </footer>

    </body>

  </html>

**课后任务**

实现项目的多端显示

**Sass总结**

**项目总结**

1. 利用编程语言书写CSS

2. Sass项目结构

3. 栅格化布局

4. 响应式布局

**不足之处**

1. 项目选择不佳，模仿网站不易利用compass的种种方便

2. 使用Compass的东西比较少，更多的是Sass的基础用法

**课程总结**

1. 基础+简单项目

2. Sass常用的工具库

**期望**

1. 能起到抛砖引玉的作用，促使大家喜欢Sass

2. Sass基础学习，让大家更容易的在项目中使用

3. 项目实战，让大家在使用Sass开发时，对目录结构不疑惑