学习方法

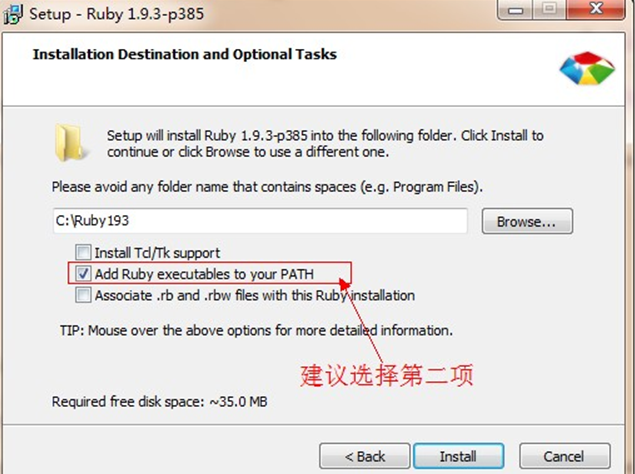
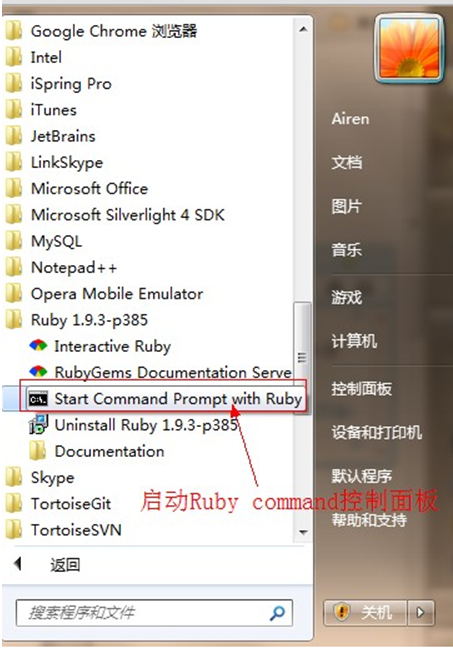
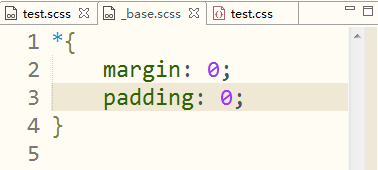
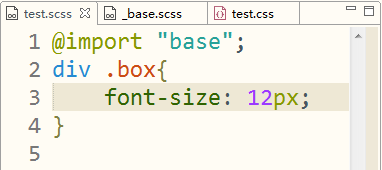
1. 建立逻辑思维
   1. 多分析
   2. 理解逻辑算法
   3. 多练、多敲、多读
2. 编程习惯
3. 大胆动手
4. 保持好奇好
5. 乐观面对错误

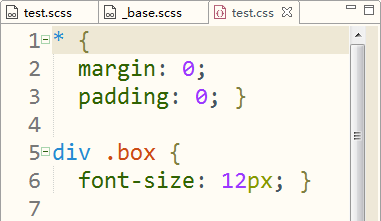
就业能力

1. 分析能力
2. 独立思考能力
3. 程序阅读能力
4. 程序排错能力
5. 独立编写程序能力
6. 团队协作能力
7. 学习能力
8. 善于总结的能力

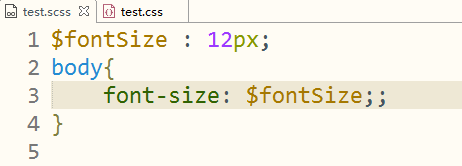
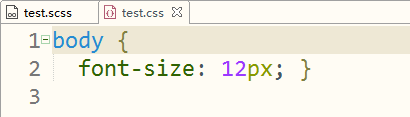
专业技能

Sass&Gulp

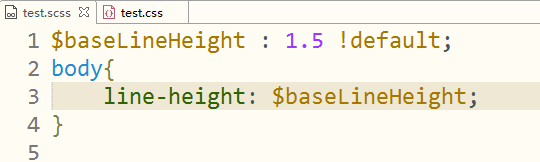
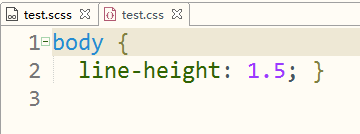
1. sass介绍
   1. SASS是一种CSS的开发工具，提供了许多便利的写法，大大节省了设计者的时间，使得CSS的开发，变得简单且可维护。
   2. SASS的本质是一种帮助你简化CSS工作流程的方式，帮助你更容易的维护和开发CSS内容。
2. 环境搭建及编译指令
   1. 在 Windows 平台下安装 Ruby 需要先有 Ruby 安装包，大家可以到 Ruby 的官网（http://rubyinstaller.org/downloads）下载对应需要的 Ruby 版本。
   2. Ruby 安装文件下载好后，可以按应用软件安装步骤进行安装 Ruby。在安装过程中，个人建议将其安装在 C 盘下，在安装过程中选择第二个选项（不选中，就会出现编译时找不到Ruby环境的情况），如下图所示：  
      
   3. Ruby 安装完成后，在开始菜单中找到新安装的 Ruby，并启动 Ruby 的 Command 控制面板，如下图所示：
   4. 当你的电脑中安装好 Ruby 之后，接下来就可以安装 Sass 了。同样的在windows下安装 Sass 有多种方法。但这几种方法都是非常的简单，只需要在你的命令终端输入一行命令即可。
   5. 通过命令安装 Sass
      1. 打开电脑的命令终端，输入下面的命令：
      2. gem install sass
      3. 提醒一下，在使用 Mac 的同学，可能需要在上面的命令前加上"sudo"，才能正常安装：
      4. sudo gem install sass
   6. 如果不能正常安装sass?
      1. gem sources --remove https://rubygems.org/
      2. 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]
      3. gem sources -l
      4. get install sass
   7. 检测Sass
      1. sass -v
   8. 升级sass版本的命令行为
      1. gem update sass
   9. 卸载（删除）sass
      1. 在常期使用的时候难免会碰到无法解决的问题，有时候可能需要卸载 Sass，然后再重新安装 Sass。那么怎么卸载 Sass 呢？
      2. 其实他也就是一句命令的事情：
         1. gem uninstall sass
3. 基本用法
   1. sass提供四个编译风格的选项：
      1. nested : 嵌套缩进的css代码，它是默认值
      2. expanded:没有缩进的、扩展的css代码
      3. compact:简洁格式的css代码
      4. compressed:压缩后的css代码
   2. sass导入
      1. Css本身包含一个指令@import,但是CSS中的@import每次执行都会发送一次新的请求都会消耗一定的资源  
         SASS中扩展了这个指令，会将包含的编译成一个CSS文件，切割成小的部分（partials）包含进来进行处理。  
         Partials这样的文件，命名规范是以下划线开头的，这样的SCSS文件不会被编译成CSS文件。  
         Partials是用来定义公共样式或者组件的样式的，专门用于被其他的SCSS文件import进行使用的。  
         在SCSS文件中引入指令@import在引入Partials文件时，不需要添加下划线。如下： 



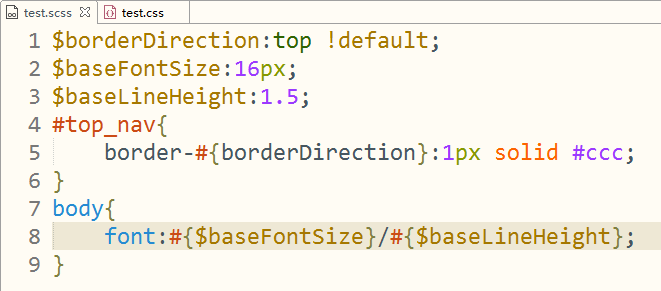
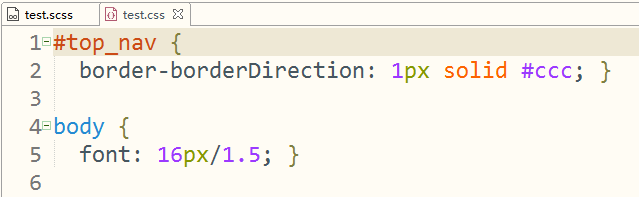
1. sass嵌套
   1. Sass 中还提供了选择器嵌套功能，但这也并不意味着你在 Sass 中的嵌套是无节制的，因为你嵌套的层级越深，编译出来的 CSS 代码的选择器层级将越深，这往往是大家不愿意看到的一点。这个特性现在正被众多开发者滥用。
   2. 选择器嵌套为样式表的作者提供了一个通过局部选择器相互嵌套实现全局选择的方法，Sass 的嵌套分为三种：
      1. 选择器嵌套
      2. 属性嵌套
      3. 伪类嵌套(了解)
   3. 选择器嵌套
      1. 假设我们有一段这样的结构：  
         <header>  
         <nav>  
          <a href = “#”>Home</a>  
          <a href = “#”>About</a>  
          <a href = “#”>Blog</a>  
         </nav>  
         </header>
      2. 想选中header中的a标签，在写CSS，会这样写：  
         nav a{  
          color: red;  
         }  
         header nav a{  
          color : green;  
         }
      3. 那么在SASS中，就可以使用选择器的嵌套来实现：  
         //在嵌套的过程中，如果需要用到父元素，在SASS中通过&符号引用父属性  
         nav {  
          a{  
          color: red;  
          header & {  
          color:green;  
          }  
          }  
         }
   4. 属性嵌套——不常用
      1. 所谓属性嵌套，是指某些属性拥有同样的单词开头，如：border-left,border-color都是以border开头的，所以就出现了属性嵌套语法，如：  
         demo.scss  
         #top\_nav\_li{  
          border:{  
          style: solid;  
          left:{  
          width:4px;  
          color : #888;  
          }  
          right:{  
          width:2px;  
          color:#ccc;  
          }  
          }  
          }  
         demo.css  
         #top\_nav\_li{  
          border-style:solid;  
          border-left-width:4px;  
          border-left-color:#888;  
          border-right-width: 2px;  
          border-right-color: #ccc;  
         }
2. sass变量定义
   1. SASS中的变量，必须是$符号开头，后面紧跟变量名，变量名称和变量值之间要使用冒号：进行分隔（参考CSS属性和值的设定语法），如果值后面加上[!default]就表示默认值。  
      引用变量的值，直接使用变量名称，即可引用定义的变量的值。
   2. 普通变量：定义之后可以在全局范围内使用

* 1. 默认变量：sass的默认变量需要在值的后面加上!default进行标识

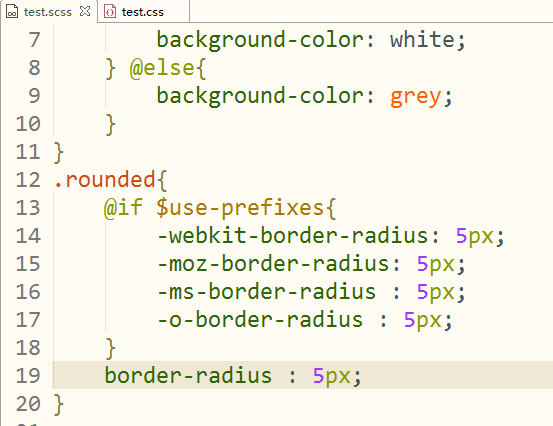
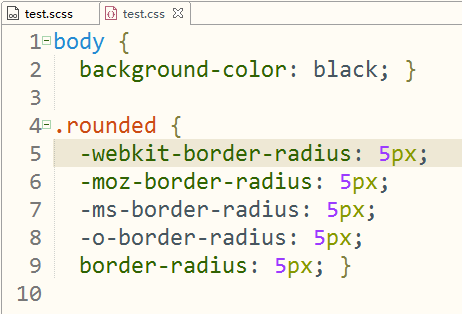
 

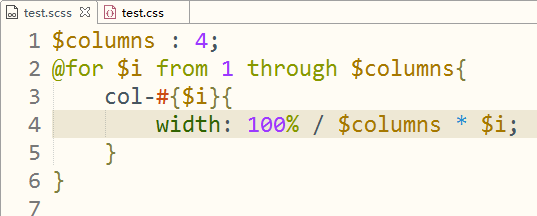
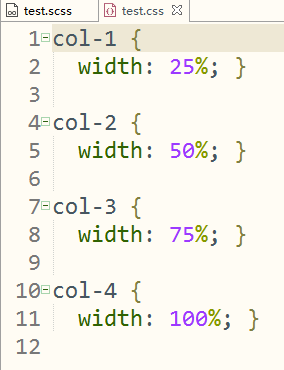
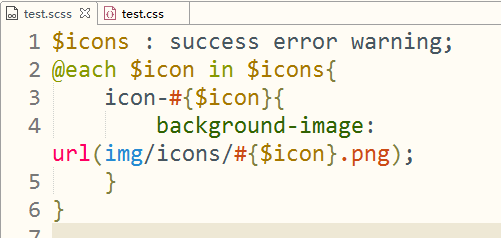
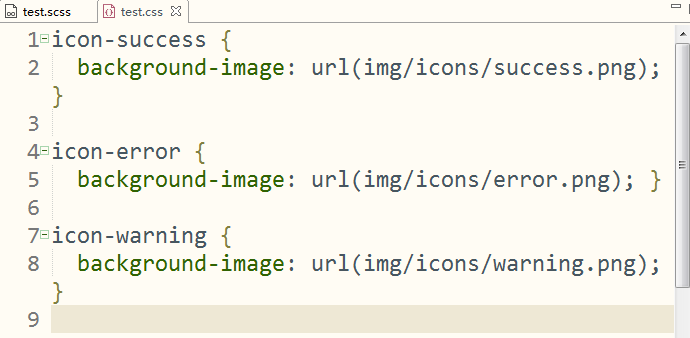
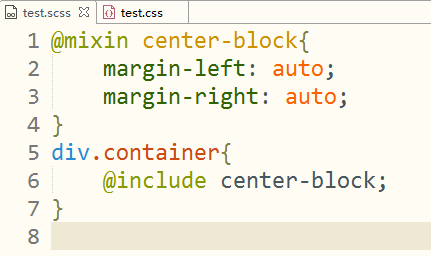
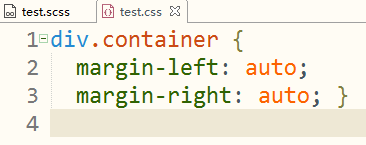
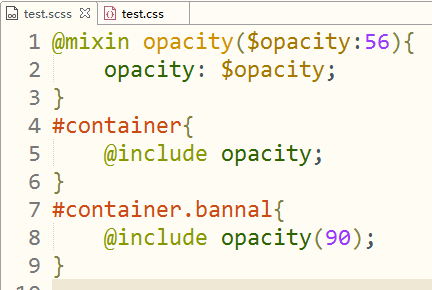
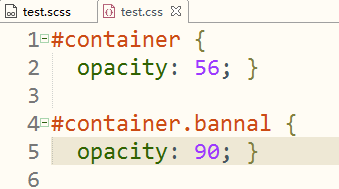
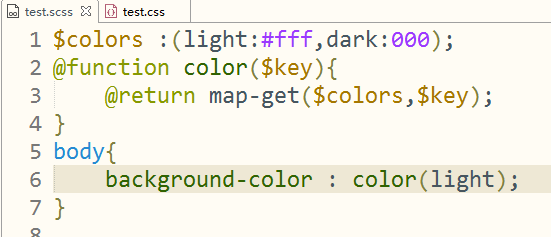
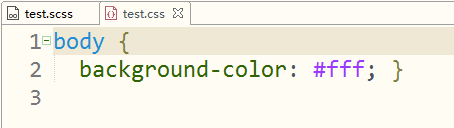
* + 1. 默认变量的覆盖：在默认变量前后，重新声明变量并赋值即可
    2. 默认变量是基于组件化开发的过程中，进行优化处理的
  1. 特殊变量：一般情况下，我们定义的变量都是属性值，可以直接使用，但是如果变量作为属性或者其他的特殊情况下，必须使用#{$variable}的形式进行调用。
     1. #{$varialbe}就是取值的一种特殊形式，符合特殊用法。

1. sass判断语句
   1. @if指令是SASS中的一个控制指令，用于在表达式满足条件(true)的时候输出指定的样式，在不满足条件（false）或者表达式为null的情况下输出其他的样式

@if条件{  
 //当条件为真时执行的样式  
 }  
同样，也可以通过@else if 和@else 指令结合，进行多条件的判断  

1. sass数组/map（了解）
   1. list表示列表类型的值  
      在CSS中就是表示属性的一串值  
      列表中的值可以使用空格或者逗号分隔，如
      1. border:#ccc solid 1px;值就是列表
      2. font – family:Courier , “Lucide Console”,monospace;值也是列表
   2. 列表中可以包含其他的列表，如：
      1. padding:10px 5px,5px 5px;值的列表中有两个列表，用逗号分隔
      2. padding(10px 5px) (5px 5px);可以用括号分开，编译成css时会去掉这些括号
   3. map就是列表项目中带名称的列表
      1. $map : (key1 : value1,key2:value2,key3:value3)
         1. $var (key1:value1,key2:value2..):声明一个Map
         2. length($map):获取map中的元素对个数
         3. map-get($map,key):获取$map中名称为key的值
         4. map-keys($map):获取指定$map中所有的key
         5. map-values($map):获取指定$map中所有的value
         6. map-has-key($map,key):判断在$map中是否包含指定的key
         7. map-merge($map1,$map2):将$map1和$map2合并在一起
         8. map-remove($map,key):将指定名称的key从$map中移除
2. sass循环语句
   1. @for指令在SASS中用于重复处理一组指令
   2. 有两种表现形式
      1. @for $var from <开始值> through < 结束值>
      2. @for $var from <start> to <end>
      3. to和throuch都是表示一个区间，唯一的区别就是停止循环的地方不一样。$var可以是任意一个变量名称如$i，<start>和<end>是SASS表达式并且必须是整数  
          
   3. @each在SASS中主要被用来进行列表或者映射数据的循环
      1. 主要表示形式：@each $var in <list>
      2. $var可以是任意变量名称，<list>是SASS表达式并且必须是list  
          
3. sass-mixin函数
   1. sass中可以通过@mixin声明混合，可以传递参数，参数名称以$开始，多个参数之间使用逗号分隔，@mixin的混合代码块由@include来调用
   2. 无参数混合——不建议使用，如果是这样的代码块，直接使用后面提到的@extend来处理  
       
   3. 有参数混合  
       
4. sass-function介绍
   1. 函数的功能主要是数据的运算，SASS中可以将一些值交给函数进行处理，具体的处理方式由定义的函数具体的设计确定。
   2. @function 函数名称（参数列表）{  
       //数据处理  
      }  
       

第二节：应用

1. 使用sass改造飞机大战的CSS文件

第三节

1. gulp介绍
   1. gulp是前端开发过程中对代码进行构建的工具，是自动化项目的构建利器；它不仅能对网站资源进行优化，而且在开发过程中很多重复的任务能够使用正确的工具自动完成；使用它，我们不仅可以很愉快的编写代码，而且大大提高我们的工作效率。
2. 工程化的作用
   1. 自动压缩JS文件
   2. 自动压缩CSS文件
   3. 自动合并文件
   4. 自动编译cass
   5. 自动压缩图片
   6. 自动刷新浏览器
3. Node环境安装
   1. window下，下载node安装即可。 node下载地址：http://nodejs.cn/download/ 安装比较简单，基本一直下一步即可，安装路径随意。
   2. 安装完成后，终端输入如下命令 显示版本号即安装成功。
   3. node -v
4. gulp安装
   1. 推荐安装cnpm：因为npm安装插件是从国外服务器下载，受网络影响大，可能出现异常，如果npm的服务器在中国就好了，所以我们乐于分享的淘宝团队干了这事。网址：http://npm.taobao.org
   2. 安装：命令提示符执行  
      npm install cnpm -g --registry=https://registry.npm.taobao.org
   3. 全局安装gulp
      1. 在com中执行cnpm install gulp –g
      2. 查看是否正确安装：命令提示符执行gulp –v，出现版本号即为正确安装
5. gulp插件安装（编译/压缩/合并/代码检查/即时刷新）
   1. 项目文件根目录新建package.json  
      注：package.json是基于nodejs项目必不可少的配置文件，它是存在项目根目录的普通json文件重点内容
      1. 进入你的项目文件
      2. 执行命令cnpm init来新建package.json
      3. 检测package.json是否成功新建查看项目文件根目录，是否新建package.json，且内容是否和你终端中输入的一致。 package.json内容如下：{   "name": "test",  "version": "1.0.0",  "description": "我是描述",  "main": "index.js",  "scripts": {    "test": "echo \"Error: no test specified\" && exit 1"  },  "author": "",  "license": "ISC"}注：可不使用cnpm init方式，可选择手动创建package.json配置文件
      4. 在项目中新建配置文件packkage.json  
         配置文件是为了将来项目迁移使用，测试demo时，可以不需要  
         {  
          “name”:”test”, //项目名称（必须）  
          “version”: “1.0.0”, //项目版本（必须）  
          “description”:”This is for study gulp project!”, //项目描述（必须）  
          “author”:{ //项目作者信息  
          “name”:”zzl”,  
          “email”:[zzl@qq.com](mailto:zzl@qq.com)  
          },  
          “devDependencies”:{ //项目依赖的插件，这个不用写，待会装完插件自动生成  
          “gulp”:”^3.8.11”,  
          “gulp-less”:”^3.0.0”  
          }  
          }  
         注：由于json文件里是不能写注释的，所以实际开发的时候，记得把注释去掉
      5. Package.json的作用  
         当有一个新的项目需要建立，或者有人跟你合作开发时，我们不需要把项目中所有用到的模块（文件大小可能上百兆）都上传到git,只需要给项目带上一个配置文件  
         而对方拿到项目后，只需要一个命令npm install  
         Node就可以自动读取项目的配置文件，并将所有依赖模块自动下载，这样不是更快捷，更方便吗？
   2. 本地安装gulp插件  
      注：全局安装gulp是为了执行gulp任务，本地安装gulp则是为了调用gulp插件的功能
   3. 进入你的项目文件路径中后，执行cnpm install gulp --save-dev
   4. 安装gulp-sass插件（拿该插件进行示例）进入你的项目文件路径中后，执行cnpm install gulp-sass --save-dev
   5. 常用插件
      1. sass的编译（gulp-sass）
      2. less编译 （gulp-less）
      3. 重命名（gulp-rename）
      4. 自动添加css前缀（gulp-autoprefixer）
      5. 压缩css（gulp-clean-css）
      6. js代码校验（gulp-jshint）
      7. 合并js文件（gulp-concat）
      8. 压缩js代码（gulp-uglify）
      9. 压缩图片（gulp-imagemin）
      10. 自动刷新页面（gulp-livereload，谷歌浏览器亲测，谷歌浏览器需安装livereload插件）
      11. 图片缓存，只有图片替换了才压缩（gulp-cache）
      12. 更改提醒（gulp-notify）
   6. 安装以上插件
      1. $ npm install gulp-ruby-sass gulp-autoprefixer gulp-minify-css gulp-jshint gulp-concat gulp-uglify gulp-imagemin gulp-clean gulp-notify gulp-rename gulp-livereload gulp-cache --save-dev
6. gulp使用
   1. 载入插件
      1. 接下来，我们需要建立一个gulpfile.js档案，并且载入这些插件：  
         var gulp = require('gulp'),   
          sass = require('gulp-ruby-sass'),  
          autoprefixer = require('gulp-autoprefixer'),  
          minifycss = require('gulp-minify-css'),  
          jshint = require('gulp-jshint'),  
          uglify = require('gulp-uglify'),  
          imagemin = require('gulp-imagemin'),  
          rename = require('gulp-rename'),  
          clean = require('gulp-clean'),  
          concat = require('gulp-concat'),  
          notify = require('gulp-notify'),  
          cache = require('gulp-cache'),  
          livereload = require('gulp-livereload');
7. 结合sass的使用
   1. 建立任务
      1. 编译sass,Autoprefix及缩小化
      2. 首先，我们设置编译Sass。我们将编译Sass、接着通过Autoprefixer，最后储存结果到我们的目的地。接着产生一个缩小化的.min版本、自动重新整理页面及通知任务已经完成：  
         gulp.task('styles', function() {   
          return gulp.src('src/styles/main.scss')  
          .pipe(sass({ style: 'expanded' }))  
          .pipe(autoprefixer('last 2 version', 'safari 5', 'ie 8', 'ie 9', 'opera 12.1', 'ios 6', 'android 4'))  
          .pipe(gulp.dest('dist/assets/css'))  
          .pipe(rename({suffix: '.min'}))  
          .pipe(minifycss())  
          .pipe(gulp.dest('dist/assets/css'))  
          .pipe(notify({ message: 'Styles task complete' }));  
         });
      3. 说明：gulp.task('styles', function() { ... )}; 这个gulp.taskAPI用来建立任务。可以透过终端机输入$ gulp styles指令来执行上述任务。
      4. return gulp.src('src/styles/main.scss') 用来定义一个或多个来源档案。
      5. .pipe(sass({ style: 'expanded' })) 使用pipe()来找到哪个插件
      6. .pipe(gulp.dest('dist/assets/css')); 这个是用来设定目的路径。

第四节：应用

1. 掌握项目打包

第五节：综合应用

1. 掌握gulp工具的使用
2. 掌握sass的编写

周六练习

1. Sass、Gulp、RequireJS、JQuery、JQ插件的整合使用