目录

1：Sass介绍 2

2：环境搭建及编译指令 2

3：Sass变量定义和注释 5

4：Sass导入 10

5：Sass嵌套 11

6：Sass判断语句 13

7：Sass循环语句 14

8：Sass-mixin 17

9：继承 20

10：Sass-函数 22

应用： 23

1：gulp介绍 23

2：工程化的作用 24

3：node环境安装 24

4：gulp安装 27

5：gulp插件安装（压缩、合并、及时刷新） 28

6：gulp使用 28

7：结合Sass的使用 31

应用： 32

综合应用： 32

周六练习： 32

## 1：Sass介绍

CSS 不是编程语言，虽然可以用它来开发网页样式，但是没有办法用它编程。SASS 的出现，让 CSS 实现了通过代码编程的方式来实现。

、‘？。，

，Bootstrap-》less-》sass

SASS 扩展了 CSS3，增加了规则、变量、混入、选择器、继承等特性， 可以生成风格良好的 CSS 样式表文件，易于组织和维护。

SASS 官网：http://sass-lang.com/

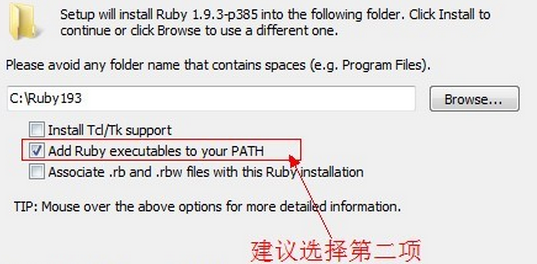
## 2：环境搭建及编译指令

**安装ruby：**

因为sass依赖于ruby环境，所以装sass之前先确认装了ruby。

下载地址： http://rubyinstaller.org/downloads

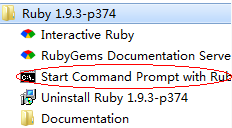
在安装的时候，请勾选Add Ruby executables to your PATH这个选项，添加环境变量，不然以后使用编译软件的时候会提示找不到ruby环境。



**安装SASS：**

安装完ruby之后，在开始菜单中，找到刚才我们安装的ruby，打开Start Command Prompt with Ruby，然后直接在命令行中输入：

gem install sass。按回车键确认，等待一段时间就会提示你sass安装成功。最近因为墙的比较厉害，如果你没有安装成功，那么请参考下面的ruby china的RubyGems镜像安装sass，如果成功则忽略。



**Ruby-china RubyGems镜像安装 sass：**

**1-查看gem软件源 的地址** gem sources –l

2- 删除原来的源 gem sources --remove https://rubygems.org/

3-添加新的软件源 gem sources -a http://gems.ruby-china.org/

4-**查看gem软件源 的地址** gem sources -l 如果更改成功(确保current sources 中只有一个http://gems.ruby-china.org/，然后执行第5步命令

5- 输入 gem install sass

Gem 是什么 ruby软件包管理系统

注：如果你系统不支持https，请将软件源更换成：gem sources -a http://gems.ruby-china.org/

查看sass版本：sass -v

升级sass版本：gem update sass（不需要升级）

**SASS编译：**

单文件转换命令：sass style.scss style.css

单文件监听命令：sass --watch style.scss:style.css

文件夹监听命令：sass --watch sassFileDirectory:cssFileDirectory

sass --style compressed --sourcemap=none --watch sass:css

命令行其他配置选项：

--style表示解析后的css是什么格式，有四种取值分别为：nested，expanded，compact，compressed。

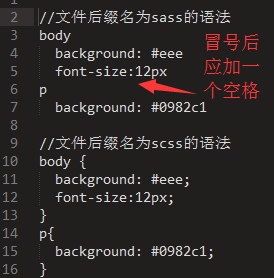
--sourcemap表示开启sourcemap调试。开启sourcemap调试，会生成一个后缀名为.css.map文件。--sourcemap=none禁止输出该文件。

查看帮助文档：sass –h

sass --style compressed --sourcemap=none --watch sass:css

**文件后缀名：**

sass有两种后缀名文件：一种后缀名为sass，不使用大括号和分号；另一种就是我们这里使用的scss文件，这种和我们平时写的css文件格式差不多，使用大括号和分号。建议使用后缀名为scss的文件，以避免sass后缀名的严格格式要求报错。

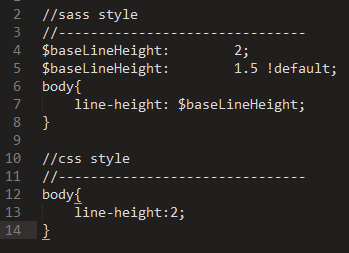
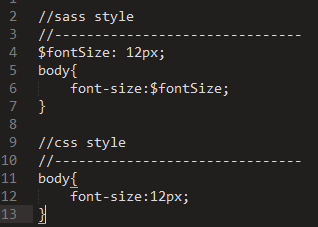


## 3：Sass变量定义和注释

sass的变量必须是$开头，后面紧跟变量名，而变量值和变量名之间就需要使用冒号(:)分隔开（就像CSS属性设置一样），如果值后面加上!default则表示默认值。

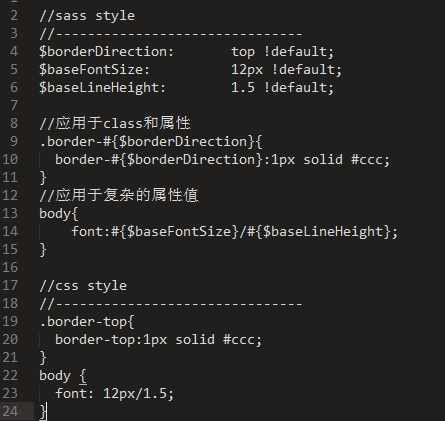
**普通变量**：定义之后可以在全局范围内使用。

**默认变量**：仅需要在值后面加上!default即可。

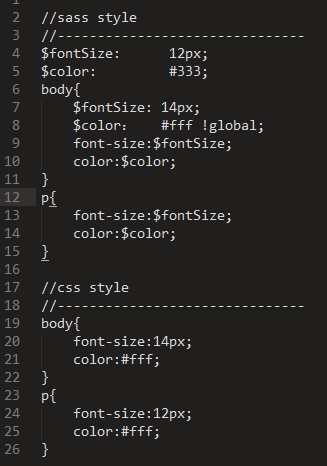


sass的默认变量一般是用来设置默认值，然后根据需求来覆盖的，覆盖的方式也很简单，只需要在默认变量之前重新声明下变量即可。默认变量的价值在进行组件化开发的时候会非常有用。

**特殊变量**：一般我们定义的变量都为属性值，可直接使用，但是如果变量作为属性或在某些特殊情况下等则必须要以#{$variables}形式使用。



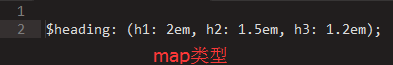
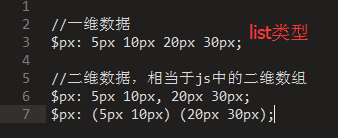
**全局变量**：在变量值后面加上!global即为全局变量。在sass 3.4后的版本中正式应用。



**多值变量**：多值变量分为list类型和map类型，简单来说list类型有点像js中的数组，而map类型有点像js中的对象。

list类型：可通过空格，逗号或小括号分隔多个值。

map类型：以key和value成对出现，其中value又可以是list。格式为：$map: (key1: value1, key2: value2, key3: value3);。



**注释：**

sass提供了三种注释方式：

多行注释：在编译输出的CSS文件中会保留，压缩输出格式中不会保留，注释方式：/\* 注释内容\*/

单行注释：在输出CSS文件时不保留。注释方式：//注释内容

强制注释：在多行注释的开头，添加感叹号!表示强制保留。注释方式：/\*!注释内容\*/

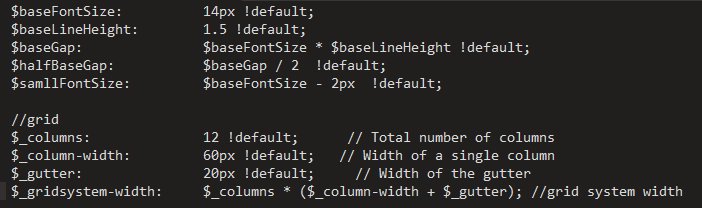
**Sass注释中文报错问题：**

找到ruby的安装目录，里面有sass模块，如这个路径：C:\Ruby23\lib\ruby\gems\2.3.0\gems\sass-3.4.22\lib。

在这个文件里面sass.rb，添加一行代码：Encoding.default\_external = Encoding.find('utf-8')。放在所有的require XXXX 之后即可。（注释在css文件中不保留）

**运算符：**

sass具有运算的特性，可以对数值型的Value(如：数字、颜色、变量等)进行加减乘除四则运算。请注意运算符前后请留一个空格，不然会出错。



**布尔值：**

SASS 中的布尔值，跟其他语言一样，都是用来表示真/假的逻辑判断的。

取值：true/false，sass 中可以使用比较运算符，返回的就是布尔值。

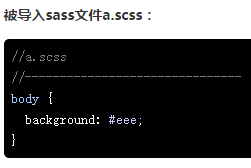
比较运算符：>、 >=、 <、 <=、 !=、 ==

逻辑运算符：and、or、not 1 and 2

## 4：Sass导入

sass的导入(@import)规则和CSS的有所不同，编译时会将@import的scss文件合并进来只生成一个CSS文件。但是如果你在sass文件中导入css文件如@import 'reset.css'，那效果跟普通CSS导入样式文件一样，导入的css文件不会合并到编译后的文件中，而是以@import方式存在。

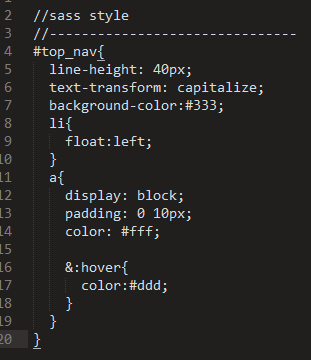
所有的sass导入文件都可以忽略后缀名.scss。文件命名规范是以\_开头，如\_mixin.scss。这种文件在导入的时候可以不写下划线，可写成@import "mixin"。



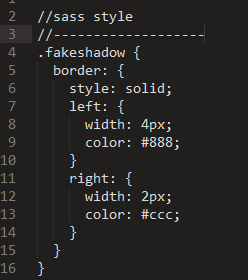
## 5：Sass嵌套

**选择器嵌套**：选择器嵌套指的是在一个选择器中嵌套另一个选择器来实现继承，从而增强了sass文件的结构性和可读性。

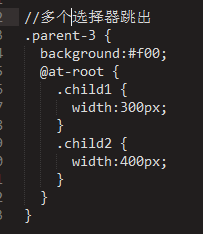
&：在选择器嵌套中，可以使用&表示父元素选择器。



**属性嵌套**：属性嵌套指的是有些属性拥有同一个开始单词，如border-width，border-color都是以border开头。



**@at-root：**用来跳出选择器嵌套的。默认所有的嵌套，继承所有上级选择器，但有了这个就可以跳出所有上级选择器。



## 6：Sass判断语句

**@if判断**：@if可一个条件单独使用，也可以和@else结合多条件使用。

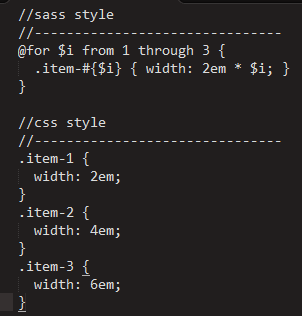


**三目判断**：语法为：if($condition, $if\_true, $if\_false) 。三个参数分别表示：条件，条件为真的值，条件为假的值。

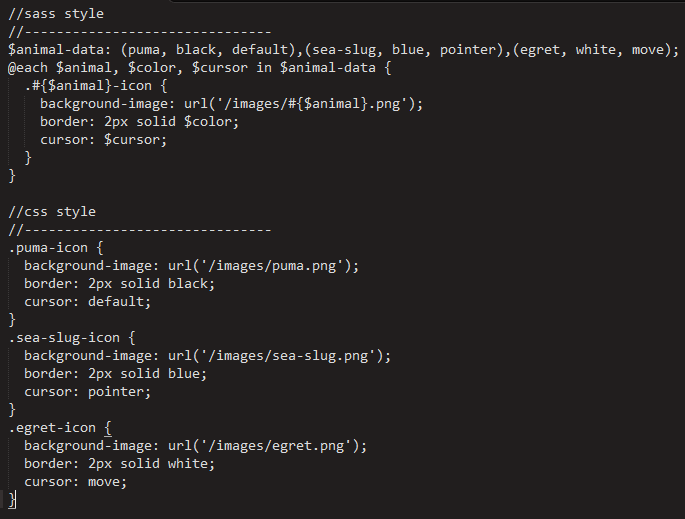


## 7：Sass循环语句

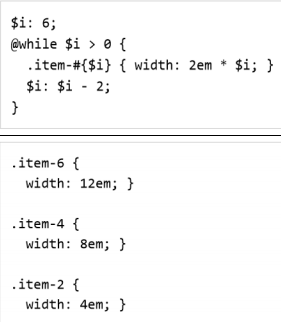
**@for循环**：for循环有两种形式，分别为：@for $var from <start> through <end>和@for $var from <start> to <end>。$i表示变量，start表示起始值，end表示结束值，这两个的区别是关键字through表示包括end这个数，而to则不包括end这个数。



**@each循环**：语法为：@each $var in <list or map>。其中$var表示变量，而list和map表示list类型数据和map类型数据。sass 3.3.0新加入了多字段循环和map数据循环。



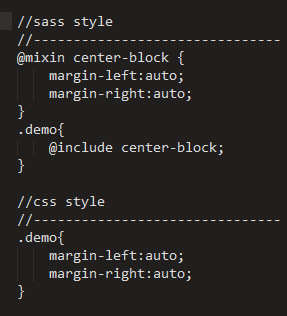
**@while循环**：@while 指令在 SASS 中用于循环重复处理样式，直到@while表达式返回false。



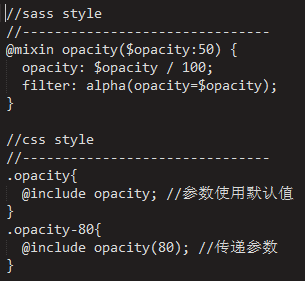
## 8：Sass-mixin

使用@mixin声明混合(宏)，可以传递参数，参数名以$符号开始，多个参数以逗号分开，也可以给参数设置默认值。声明的@mixin通过@include来调用。

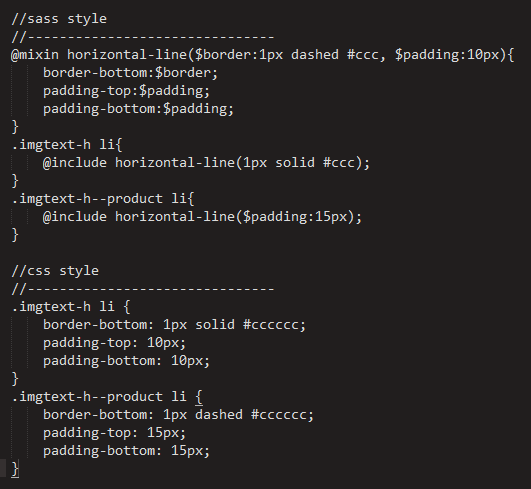
**无参数混合**：不建议使用，如果是这样的代码块，直接使用后面提到的@extend来处理。



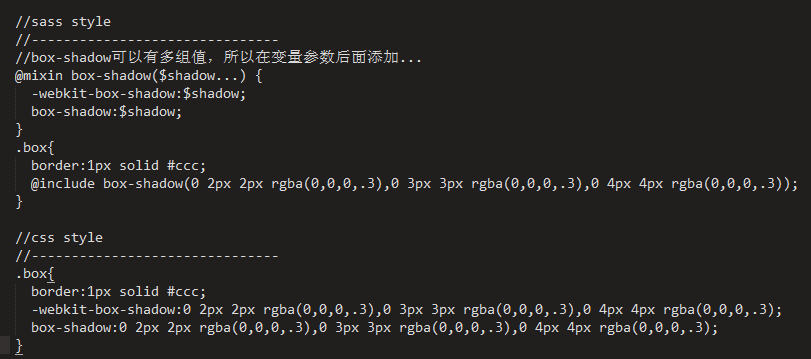
**有参数混合**：



**多个参数混合**：调用时可直接传入值，如@include传入参数的个数小于@mixin定义参数的个数，则按照顺序表示，后面不足的使用默认值，如不足的没有默认值则报错。除此之外还可以选择性的传入参数，使用参数名与值同时传入。



**多组值参数混合**：如果一个参数可以有多组值，如box-shadow、transition等，那么参数则需要在变量后加三个点表示，如$variables...。



下面是一个mixin的实例，用来生成浏览器前缀。

　@mixin rounded($vert, $horz, $radius: 10px) {  
　　　　border-#{$vert}-#{$horz}-radius: $radius;  
　　　　-moz-border-radius-#{$vert}#{$horz}: $radius;  
　　　　-webkit-border-#{$vert}-#{$horz}-radius: $radius;  
　　}

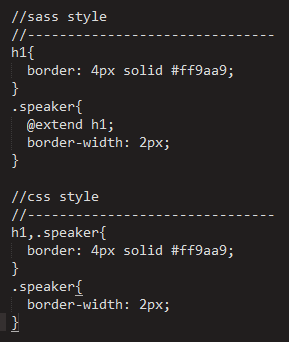
使用的时候，可以像下面这样调用：

#navbar li { @include rounded(top, left); }

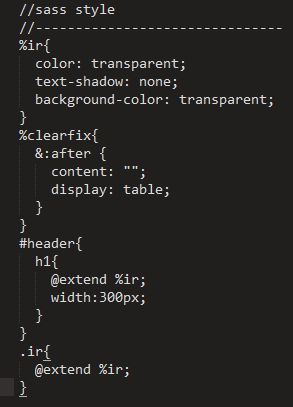
　　#footer { @include rounded(top, left, 5px); }

## 9：继承

选择器继承可以让选择器继承另一个选择器的所有样式，并联合声明。使用选择器的继承，要使用关键词@extend，后面紧跟需要继承的选择器。



**占位选择器%**：这种选择器的优势在于：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。



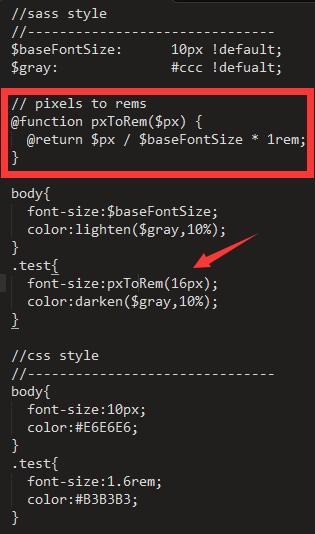
**@mixin和继承的区别：**

@mixin通过@include调用后解析出来的样式是以拷贝形式存在的，而继承则是以联合声明的方式存在的，所以建议传递参数的用@mixin，而非传递参数类的使用继承%。

## 10：Sass-函数

sass定义了很多函数可供使用，当然你也可以自己定义函数，以@fuction开始。实际项目中我们使用最多的应该是颜色函数，而颜色函数中又以lighten减淡和darken加深为最，其调用方法为lighten($color,$amount)和darken($color,$amount)，它们的第一个参数都是颜色值，第二个参数都是百分比。

**自定义函数：**



## 应用：

1：使用Sass改造飞机大战中的CSS文件。

## 1：gulp介绍

基于node流的自动化构建工具。

**gulp能干什么？**

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sass。

5：自动压缩图片。

6：自动刷新浏览器。

等等。

gulp网址：http://www.gulpjs.com.cn

## 2：工程化的作用

把一整套前端工作流程中能用工具搞定的部分，用工具搞定。也就是将前端开发流程、技术、工具、经验等规范化，标准化。

前端工程自动化工具：grunt、gulp、fis3等。

## 3：node环境安装

**nodeJS**：Node.js是一个Javascript运行环境(runtime)。实际上它是对Google V8引擎进行了封装。V8引 擎执行Javascript的速度非常快，性能非常好。

**NPM**：全称是Node Package Manager，是随同NodeJS一起安装的包管理和分发工具，它很方便让JavaScript开发者下载、安装、上传以及管理已经安装的包。 ，它已经成为了非官方的发布Node模块（包）的标准。

nodejs自身提供了基本的模块，但是开发实际应用过程中仅仅依靠这些基本模块则还需要较多的工作。幸运的是，Nodejs库和框架为我们提供了帮助，让我们减少工作量。但是成百上千的库或者框架管理起来又很麻烦，有了NPM，可以很快的找到特定服务要使用的包，进行下载、安装以及管理已经安装的包。

**使用NPM，必须安装nodeJS。**

**package.json文件的作用**：npm的配置文件。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！

**package.json文件字段列表：（简单了解）**

name字段：必须字段。不要在name中包含js, node字样；这个名字最终会是URL的一部分、命令行的参数、目录名，所以不能以点号或下划线开头；这个名字可能在require()方法中被调用，所以应该尽可能短。

version字段：必须字段。

description字段：可选字段，必须是字符串。 npm search 的时候会用到。

keywords字段：可选字段，字符串数组。 npm search 的时候会用到。

homepage字段：可选字段，没有http://等带协议前缀的URL。

bugs字段：填写一个bug提交地址或者一个邮箱。

license字段：可选字段。包的开源协议名称。

author字段：包的作者。

contributors, maintainers字段：包的贡献者，是一个数组。

main字段：包的入口文件。

repository字段：可选字段。用于指示代码存放的位置。

scripts字段：可选字段，object。Key是生命周期事件名，value是在事件点要跑的命令。参考npm-scripts。

dependencies字段：指定依赖的其它包，这些依赖是指包发布后正常执行时所需要的，也就是线上需要的包。使用命令安装：npm install --save packageName

devDependencies字段：这些依赖只有在开发时候才需要。使用命令安装：npm install --save-dev packageName

**npm常用命令：**

npm -v：查看npm安装的版本。

npm root：查看当前包的安装路径。

npm root -g：查看全局的包的安装路径。

npm init：会引导你创建一个package.json文件，包括名称、版本、作者等信息。

npm help：查看帮助命令。

npm list：查看当前目录下已安装的node包。

npm ls：查看当前目录下已安装的node包。

npm install moduleNames：安装Node模块到本地目录node\_modules下。

npm install <name> -g：将包安装到全局环境中。

npm install <name> --save：安装的同时，将信息写入package.json中项目路径中如果有package.json文件时，直接使用npm install方法就可以根据dependencies配置安装所有的依赖包，这样代码提交到github时，就不用提交node\_modules这个文件夹了。

npm install <name> --save-dev：安装的同时，将信息写入package.json中项目路径中如果有package.json文件时，直接使用npm install方法就可以根据devDependencies配置安装所有的依赖包，这样代码提交到github时，就不用提交node\_modules这个文件夹了。

npm update moduleName：更新node模块。

npm uninstall moudleName：卸载node模块。

npm remove moudleName：卸载node模块。

npm search packageName：发布一个npm包的时候，需要检验某个包名是否已存在。

## 4：gulp安装

1. 在项目中根目录输入 npm init 命令，一直按回车，最后输入 yes，会产生一个package.json文件
2. 先全局安装 gulp npm install --global gulp(如果安装失败可以切换源试试)
3. 在项目中根目录本地安装gulp npm install --save-dev gulp

因为它基于nodeJS，因此需要先安装node环境。

安装完成后，打开你的命令行窗口：输入：node -v，检测一下node是否安装好。

1-在项目里创建一个package.json

2. 指定这个字段

devDependencies：{ "gulp": "^3.9.1"

}

3. 执行命令 npm install

切换源的方法

1.

注：由于网络经常不好，考虑把npm换成cnpm，可以使用淘宝提供的镜像服务器。（npm install -g cnpm --registry=https://registry.npm.taobao.org

）

2.

或者用下面方法切换

nrm 是一个管理npm软件源的工具，用nrm可以切换国外的源和国内的源。

全局安装nrm npm i nrm -g

查看nrm里面的源 nrm ls

切换到国内的源 nrm use taobao

**全局安装gulp**：npm install --global gulp

Npm install gulp -g

作为项目的开发依赖（devDependencies）安装：npm install --save-dev gulp

## 5：gulp插件安装（压缩、合并、及时刷新）

文件重命名插件：npm i -D gulp-rename

js文件压缩插件：npm i -D gulp-uglify

css文件压缩插件：npm i -D gulp-clean-css

html文件压缩插件：npm i -D gulp-minify-html

js/css文件合并插件：npm i -D gulp-concat

自动刷新插件：npm i -D gulp-connect

可以使用上面的方式一个一个安装gulp插件，也可以在package.json中指定devDependencies 字段当中 依赖的插件

"devDependencies": {

"gulp": "^3.9.1",

"gulp-autoprefixer": "^4.0.0",

"gulp-concat": "2.6.1",

"gulp-connect": "",

"gulp-htmlmin": "",

"gulp-imagemin": "",

"gulp-jshint": "^2.0.4",

"gulp-rename": "",

"gulp-ruby-sass": "",

"gulp-uglify": "^2.0.0"

}

在根目录下 执行 npm install，所有需要的插件都会被自动安装

## 6：gulp使用

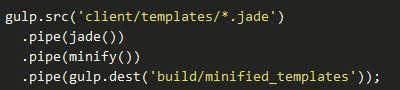
**gulpfile.js文件**：这是gulp的配置文件，放在项目根目录下。

gulp的API：

**gulp.src(globs[, options])**：输出（Emits）符合所提供的匹配模式（glob）或者匹配模式的数组（array of globs）的文件。 将返回一个 Vinyl files 的 stream 它可以被 piped 到别的插件中。

globs参数：类型为String 或 Array，所要读取的 glob 或者包含 globs 的数组。

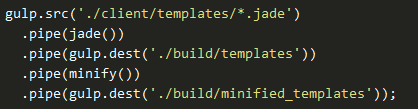
options参数：类型为Object。具体使用方法：请参照官方文档。



**gulp.dest(path[, options])**：能被 pipe 进来，并且将会写文件。并且重新输出（emits）所有数据，因此你可以将它 pipe 到多个文件夹。如果某文件夹不存在，将会自动创建它。

path参数：类型为String 或 Function，文件将被写入的路径（输出目录）。也可以传入一个函数，在函数中返回相应路径。

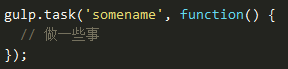
options参数：类型为Object。具体使用方法：请参照官方文档。



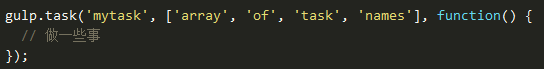
**gulp.task(name[, deps], fn)**：定义一个任务。

name参数：任务的名字，如果你需要在命令行中运行你的某些任务，那么，请不要在名字中使用空格。

fn参数：该函数定义任务所要执行的一些操作。通常来说，它会是这种形式：return gulp.src().pipe(someplugin())。



deps参数：类型为Array。一个包含任务列表的数组，这些任务会在你当前任务运行之前完成。

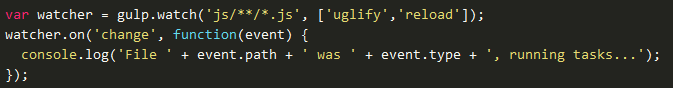


**注意：** 你的任务是否在这些前置依赖的任务完成之前运行了？请一定要确保你所依赖的任务列表中的任务都使用了正确的异步执行方式：使用一个 callback，或者返回一个 promise 或 stream。

**gulp.watch(glob, tasks)**：监视文件，并且可以在文件发生改动时候做一些事情。它总会返回一个 EventEmitter 来发射（emit） change 事件。

glob参数：类型为String or Array。一个 glob 字符串，或者一个包含多个 glob 字符串的数组，用来指定具体监控哪些文件的变动。

tasks参数：类型为Array。需要在文件变动后执行的一个或者多个通过 gulp.task() 创建的 task 的名字。



event.type参数：类型为String。发生的变动的类型：added, changed 或者 deleted。

event.path参数：类型为String。触发了该事件的文件的路径。

## 7：结合Sass的使用

本地安装Sass插件：npm i gulp-ruby-sass -D

我们之前安装过ruby环境，所以选择了gulp-ruby-sass，没有ruby环境的可以安装gulp-sass。两个插件的API略有不同，需要的时候百度一下就行了。





执行任务 gulp watch 执行watch这个任务，可以一直监听，也可以单独执行某个任务

-1-安装nodejs

0-**全局安装gulp**：npm install --global gulp

1-在项目根目录里面创建一个package.json 文件,指定devDependencies 这个参数（项目依赖的工具），”

"devDependencies":{  
 "gulp":"",  
 "gulp-uglify":"^2.0.0",  
 "gulp-concat":"2.6.1",  
 "gulp-connect":"",  
 "gulp-rename":"",  
 "gulp-ruby-sass":""  
}

2- 在项目的根目录下面，执行 npm install 命令，npm就会帮咱们下载 gulp和gulp插件

3- 在根目录下面 gulpfile.js

4-gulp watch

https://www.npmjs.com

## 应用：

1：将gulp应用到微信飞机大战项目中。

## 综合应用：

1：掌握gulp工具使用。

2：掌握Sass的编写。