# Gulp使用教程

## 简介

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；她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；使用她，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。

gulp是基于Nodejs的**自动任务运行器**，她能自动化地完成 javascript/coffee/sass/less/html/image/css 等文件的的测试、检查、合并、压缩、格式化、浏览器自动刷新、部署文件生成，并监听文件在改动后重复指定的这些步骤。在实现上，她借鉴了Unix操作系统的管道（pipe）思想，前一级的输出，直接变成后一级的输入，使得在操作上非常简单。

gulp 和 grunt 非常类似，但相比于 grunt 的频繁 IO 操作，gulp 的流操作，能更快地更便捷地完成构建工作。

## 安装nodejs

2.1、说明：gulp是基于nodejs，理所当然需要安装nodejs；

2.2、安装：打开nodejs官网（https://nodejs.org/en/），点击硕大的绿色Download按钮，它会根据系统信息选择对应版本（.msi文件）。

注：安装完成之后注意将其配置到环境变量（系统变量path中），这样你就可以在全局范围内去使用它了

我的路径是在c:\Program Files\nodejs\node\_modules\npm\bin，记得把它也配置进去

2.3测试：按window + r 输入cmd回车，然后输入node -v

然后输入npm -v

如果都能打印出相应版本信息，那么说明你配置成功了

2.4 npm的解释：

npm是Node.js的包管理工具（package manager）。

在Node.js上开发时，会用到很多别人写的JavaScript代码。如果需要使用别人写的某个包，每次都根据名称搜索一下官方网站，下载代码，解压，再使用，非常繁琐。于是一个集中管理的工具应运而生：大家都把自己开发的模块打包后放到npm官网上，如果要使用，直接通过npm安装就可以直接用，不用管代码存在哪，应该从哪下载。

更重要的是，如果我们要使用模块A，而模块A又依赖于模块B，模块B又依赖于模块X和模块Y，npm可以根据依赖关系，把所有依赖的包都下载下来并管理起来。否则，靠我们自己手动管理，肯定既麻烦又容易出错。

npm已经在Node.js安装的时候顺带装好了。我们在命令提示符或者终端输入npm -v

## 必装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](http://npm.taobao.org/" \t "http://www.ydcss.com/archives/_blank)

安装：命令提示符执行

npm install cnpm -g --registry=https://registry.npm.taobao.org

注意：安装完后最好查看其版本号cnpm -v或关闭命令提示符重新打开，安装完直接使用有可能会出现错误；

将路径C:\Users\asus\AppData\Roaming\npm\node\_modules\cnpm配置到环境变量中，这样就可以在全局范围内使用cnpm命令了

注：cnpm跟npm用法完全一致，只是在执行命令时将npm改为cnpm（以下操作将以cnpm代替npm）。

## 全局安装gulp

在cmd中执行 cnpm install gulp -g

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

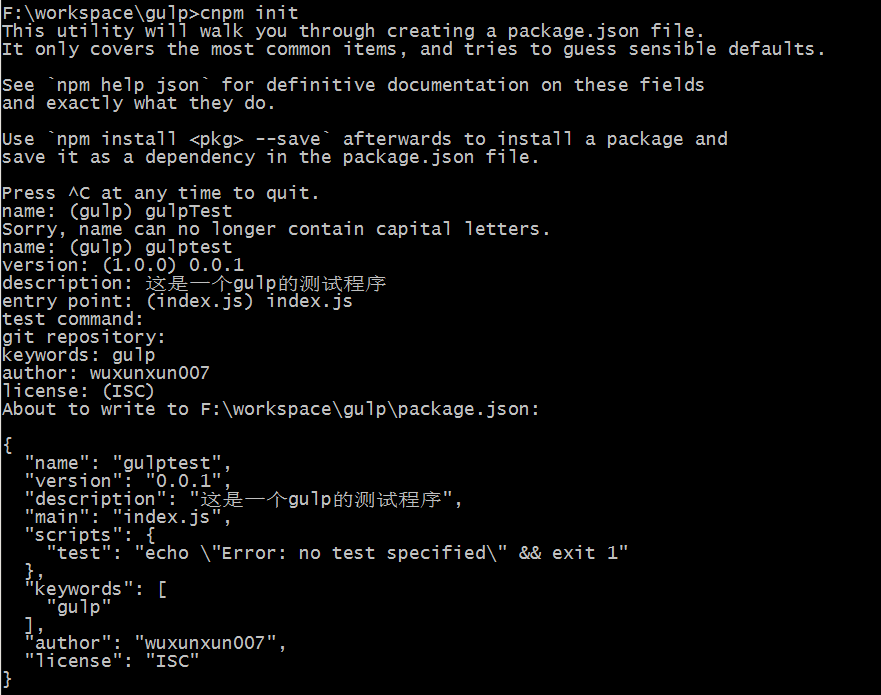
注意配置环境变量（C:\Users\asus\AppData\Roaming\npm\gulp）

## 新建package.json文件(初始化一个项目)

进入项目的目录,如：cd F:\workspace\gulp （这是项目目录）

创建package.json文件

**cnpm init**



在路径下会形成这样的一个.json文件（部分内容为后添加的）

{

"name": "gulptest",//项目名称（必须）不能有大写

"version": "0.0.1",//项目版本（必须）

"description": "这是一个gulp的测试程序",//项目描述（必须）

"homepage": "",   //项目主页

"main": "index.js",//入口文件

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"keywords": [//关键词

"gulp"

],

"author": "wuxunxun007",//作者

"license": "ISC"//项目许可协议

"devDependencies": {    //项目依赖的插件

    "gulp": "^3.8.11",

    "gulp-less": "^3.0.0"

  }

}

查看package.json帮助文档，命令提示符执行**cnpm help package.json**

## 本地安装gulp(在项目目录下安装gulp)：

1）、把当前目录转到项目目录：cd 项目目录

**安装：定位目录命令后提示符执行**cnpm install gulp --save-dev**；**

**注意：这个命令的执行，必须放在项目的目录下。即上例中的**F:\workspace\gulp。

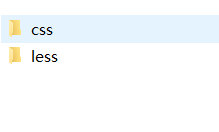
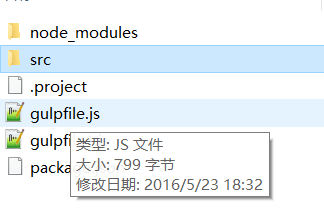
## 本地安装gulp插件

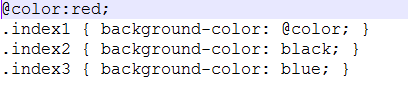
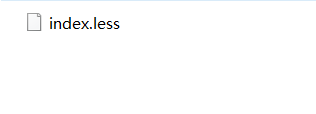
**本示例以gulp-less为例（编译less文件），命令提示符执行cnpm install gulp-less --save-dev；**



将会安装在node\_modules的gulp-less目录下，该目录下有一个gulp-less的使用帮助文档README.md；

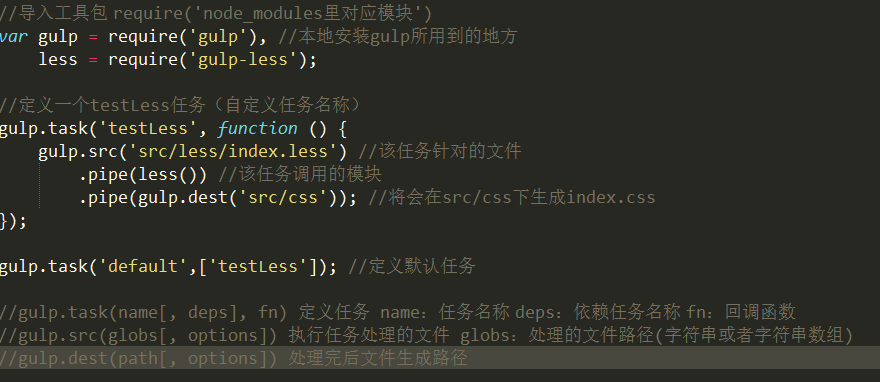
## 创建文件

在目录中创建src文件夹，在该文件夹中创建less文件夹和css文件夹

在less文件中创建一个index.less文件，写一些代码

然后在你的项目的根目录下创建一个gulpfile.js文件

在该文件中写入



然后打开cmd窗口，或者你安装了git环境的话，直接在该文件夹下右键点解Git Base Here，然后输入gulp命令(gulp testLess)，那么你就在你的src/css文件夹下就可以看到一个index.css文件，这个文件就是我们编译less文件之后生成的一个css文件

解释：gulp.src() 是读取将要处理的文件，即源文件

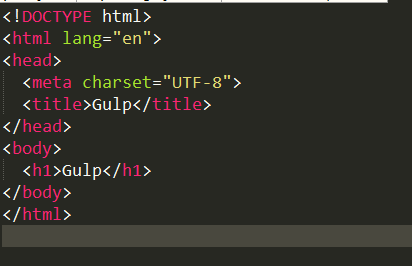
pipe() 管道，是如何处理该文件（如：sass编译，复制文件，压缩文件等）。

gulp.dest() 是处理后的文件的路径，即目的文件。

## 复制文件

8.1复制页面

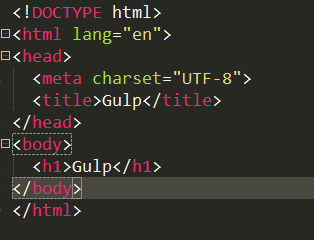
在根目录下创建一个index.html写一点内容，



然后将gulpfile.js修改为



然后在命令行末执行gulp copy-index 那么我们就会在根目录下发现一个index.html文件



注：gulp.dest（）里可以写绝对路径。

如：

gulp.src("\*.html").pipe(gulp.dest("e:/dist"));

表示拷贝到e盘下dist下。

要复制项目的所有文件和文件夹（包括子文件夹，不管有多少级）：“./\*\*/\*”

当前路径用 “.”表示

gulp.task('copyallfile',function(){

gulp.src('./\*\*/\*')

.pipe(gulp.dest('D:\\phpStudy\\WWW\\day32test'));

});

8.2 拷贝图片文件images



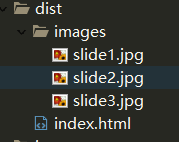
修改gulpfile.js

复制imges文件加下的所有jpg文件

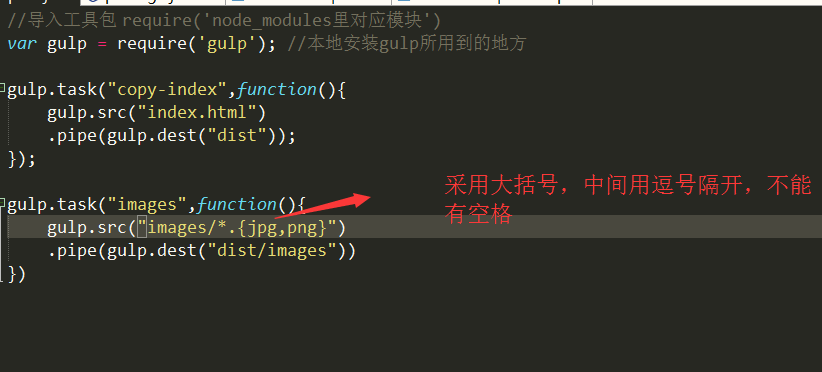


执行命令 gulp images

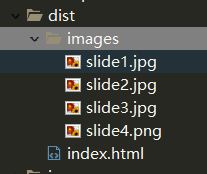
在dist/images下有这样三个文件



复制iamges文件加下的所有jpg和png文件



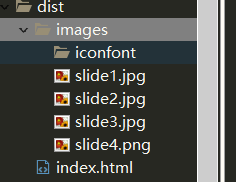
执行命令



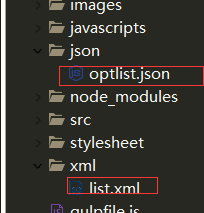
复制iamges文件加下的所有文件



执行命令



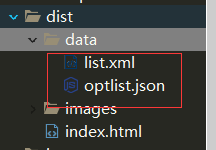
8.3 同时拷贝两个文件夹,拷贝到dist文件夹下的data中

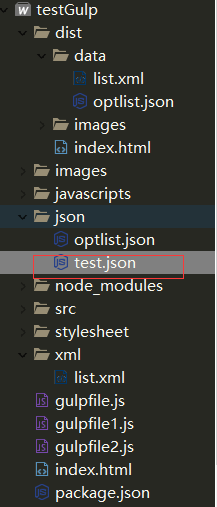


修改gulpfile.js



执行命令gulp data

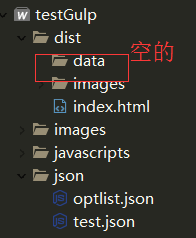


8.4 拷贝文件是过滤文件,在json文件中新建一个json为test.json，

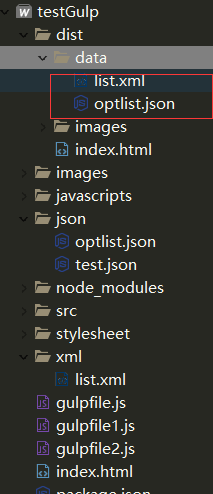
然后修改gulpfile.js



请手工删除dist/data文件下的文件



执行命令

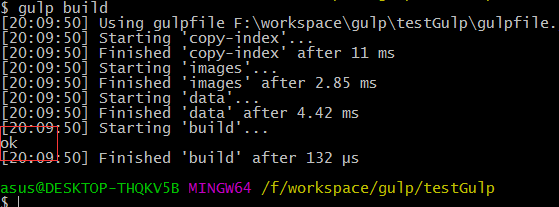


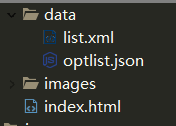
我们发现，并没有把test.json拷贝进来

8.5如何将以上命令一次性执行，采用bulid命令



执行命令





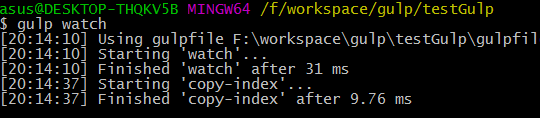
## 实时监测文件的更新

9.1 新建一个任务watch

Gulpfile.js

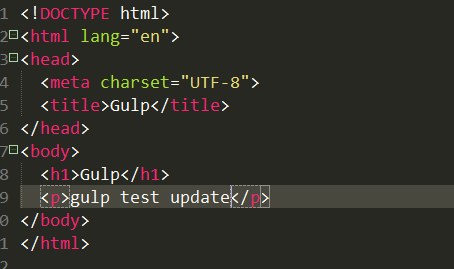


执行命令 gulp watch

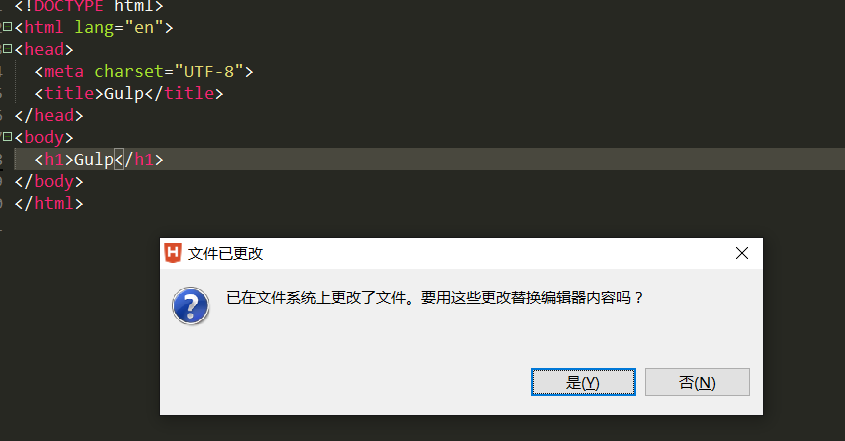


除非手动关闭，否则一直更新

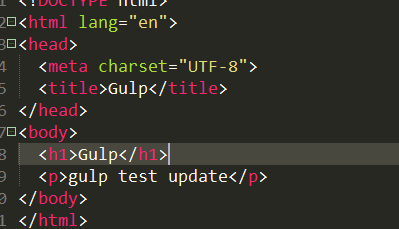
修改index.html



打开dist文件下的index.html



点击是



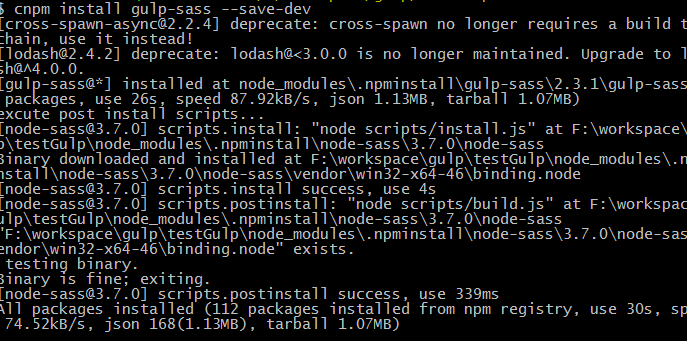
## gulp插件

gulp本身只做一些文件的拷贝，监视等等，但是它提供了很多的接口，由插件完成更多对应的任务，如：sass，less文件的编译，合并文件，压缩文件，优化图片的尺寸，创建本地的开发服务器。下面先介绍常用的，更多的可以找gulp的插件。

10.1 编译sass文件

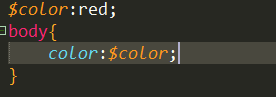
输入命令：cnpm install gulp-sass --save-dev

安装sass插件， --save-dev表示安装到开发依赖里。

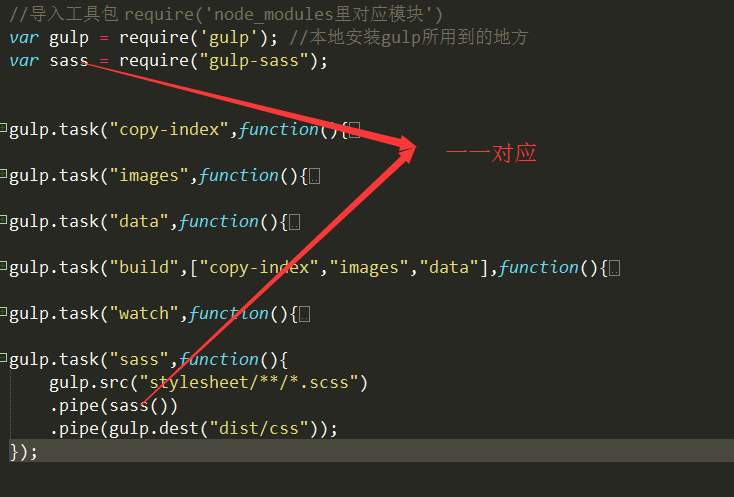


安装完成之后使用即可

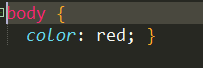
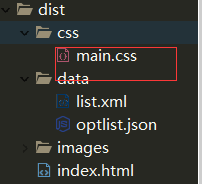
创建一个main.scss



修改gulpfile.js

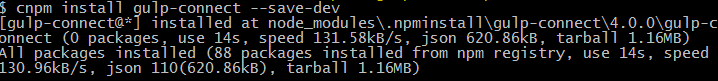


执行命令gulp sass



10.2 启动服务（web服务器）

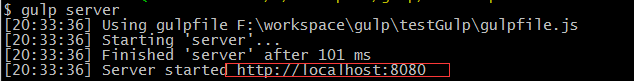
安装插件 cnpm install gulp-connect --save-dev



修改gulpfile.js



执行命令 gulp server

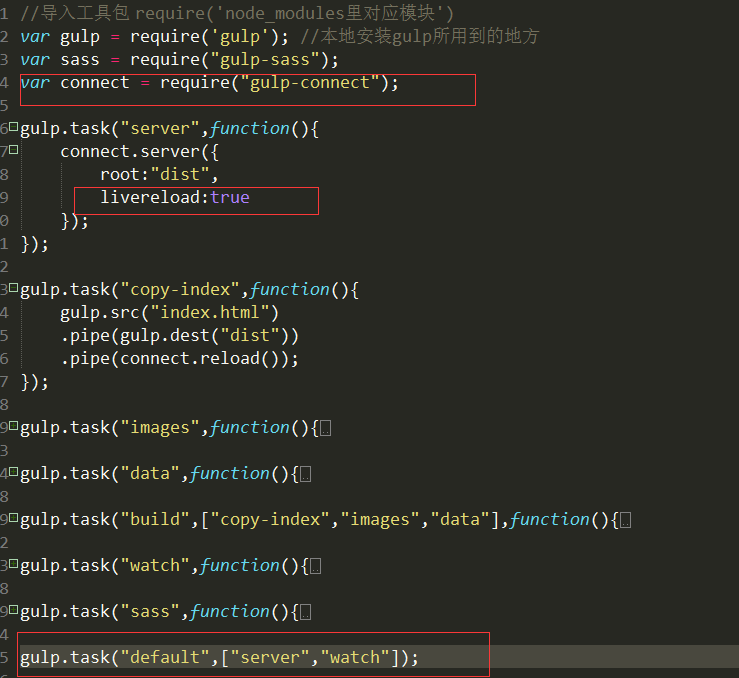


打开浏览器输入

<http://localhost:8080/>，可以查看到



如果需要实时更新数据，那么需要添加一个参数livereload:true



执行gulp命令

然后修改index.html那么会自动更新

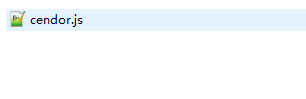
10.3 合并文件

安装命令 cnpm install gulp-concat --save-dev

修改gulpfile.js



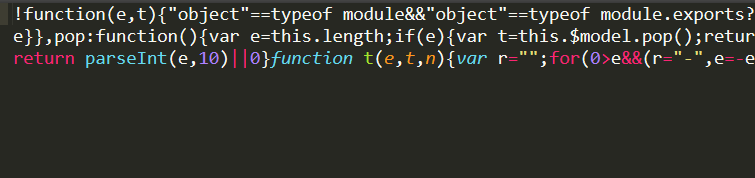
执行命令 gulp scripts



10.4 合并并且压缩

先安装uglify插件

cnpm install gulp-uglify --save-dev

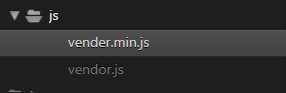


10.5 合并并且压缩,重新起名字

先安装插件rename

cnpm install gulp-rename --save-dev





10.6 压缩css

先安装插件：

cnpm install gulp-minify-css --save-dev





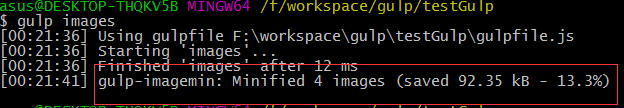
重新起名字：



10.6 压缩图片

先安装插件 imagemin

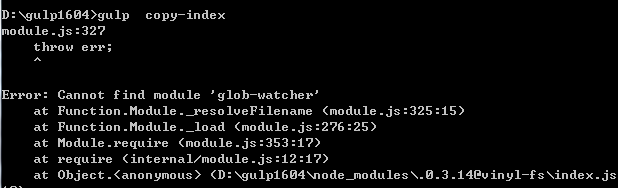
cnpm install gulp-imagemin --save-dev



Pipe：是管道的意思，pipe的参数如果写的是一个（命令）相当于生活中的在管道里增加一个过滤器过滤纯净水一样的道理

**常见问题：**

1. **如果出现了cannot find module ’\*\*\*\*’,表示该模块没有安装成功，解决：执行cnpm install gulp ‘\*\*\*’ –save-dev，就可以单独安装该模块。**

**如：**

**表示，模块glob-watcher没有安装成功。执行以下命令：**

**cnpm install gulp glob-watcher –save-dev。如下图：**

