

**项目构建**

目录

[目录 2](#_Toc460567866)

[第1章 项目构建 3](#_Toc460567867)

[1.1 Gulp 3](#_Toc460567868)

[1.1.1 Gulp基础 3](#_Toc460567869)

[1.1.2 Gulp工作原理 5](#_Toc460567870)

[1.1.3 Gulp API 5](#_Toc460567871)

[1.1.4 常用Gulp插件 6](#_Toc460567872)

# 项目构建

多个开发者共同开发一个项目，每位开发者负责不同的模块，这就会造成一个完整的项目实际上是由许多的“代码版段”组成的；

使用less、sass等一些预处理程序，降低CSS的维护成本，最终需要将这些预处理程序进行解析；

合并css、javascript，压缩html、css、javascript、images可以加速网页打开速度，提升性能；

这一系列的任务完全靠手动完成几乎是不可能的，借助构建工具可以轻松实现。

所谓构建工具是指通过简单配置就可以帮我们实现合并、压缩、校验、预处理等一系列任务的软件工具。

常见的构建工具包括：Grunt、Gulp、F.I.S（百度出品）、webpack

## Gulp

Gulp是基于Nodejs开发的一个构建工具，借助[gulp插件](http://gulpjs.com/plugins/)可以实现不同的构建任务，以其简洁的配置和卓越的性能成为目前主流的构建工具。

全局安装 npm install -g gulp

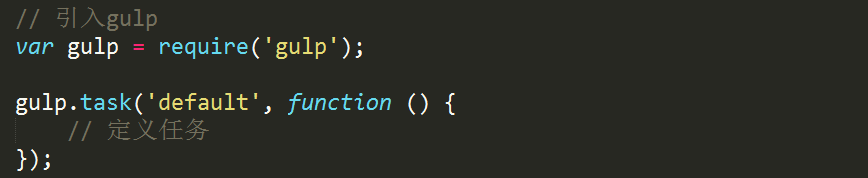
### Gulp基础

1、本地安装gulp

进入项目根目录执行npm install gulp --save-dev（添加--save-dev会在package.json记录依赖关系）。

2、任务清单

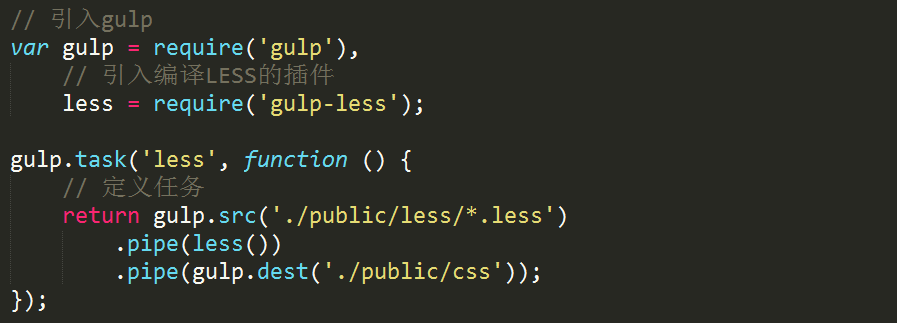
在项目根目录中创建gulpfile.js（这是一个配置文件）



3、定义任务

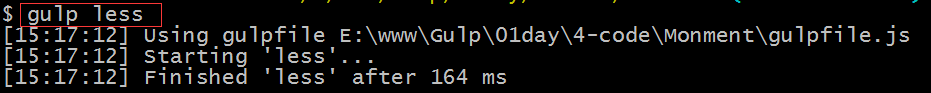
在gulpfile.js定义构建任务，如压缩、合并，Gulp自身并不执行任何任务，是通过调用具体插件来完成的。

以编译LESS为例，安装npm install gulp-less，如下图定义任务



4、执行任务

输入命令 gulp less



这样我们的LESS文件便会编译成CSS了。

### Gulp工作原理



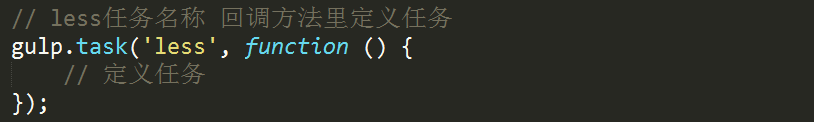
通过不同的插件实现构建任务，Gulp只是按着配置文件调用执行了这些插件。

### Gulp API

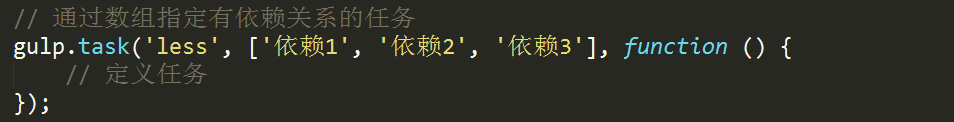
Gulp是基于NodeJS的，通过require可以引入一个NodeJS的包（模块），其作用类似于浏览器中的script标签引入资源，被引入的包存放在node\_modules目录下。

引入gulp包（模块）后返回一个对象，习惯赋值给变量gulp，通过该对象提供的方法（API）完成任务的配置。

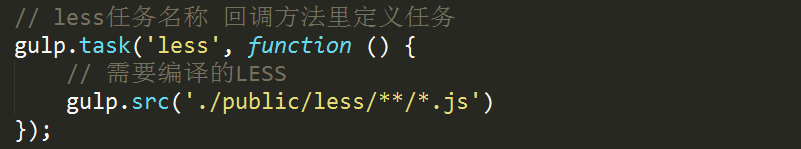
1、gulp.task() 定义各种不同的任务，如下图有两个参数



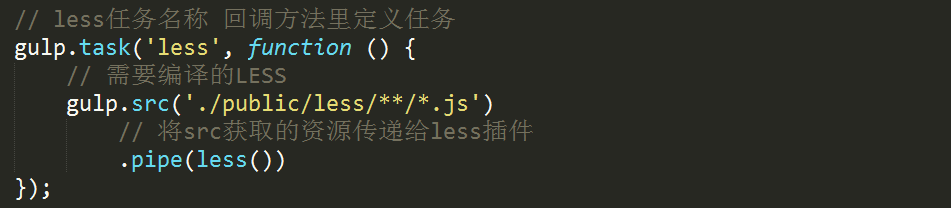
不同任务间存在依赖关系时，可以指定依赖，如下图



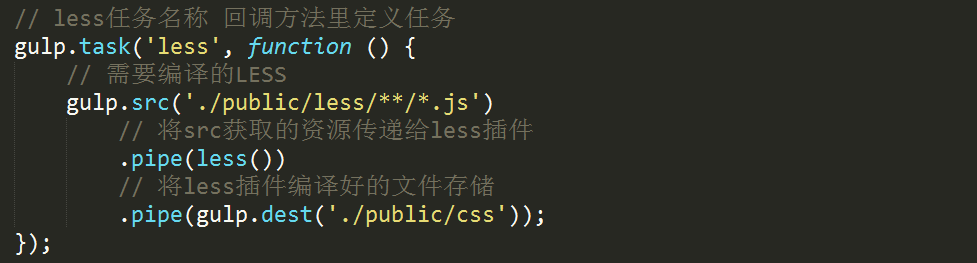
2、gulp.src() 需要构建资源的路径，字符串或数组（可以正则方式书写）



2、gulp.pipe() 管道，将需要构建的资源“输送”给插件。



3、gulp.dest() 构建任务完成后资源存放的路径（会自动创建）



4、gulp.watch()

### 常用Gulp插件

gulp-less 编译LESS文件

gulp-autoprefixer 添加CSS私有前缀

gulp-cssmin 压缩CSS

gulp-rname重命名

gulp-imagemin 图片压缩

gulp-uglify 压缩Javascript

gulp-concat 合并

gulp-useref 合并js文件并且替换HTML中的引用

gulp-if

gulp-htmlmin 压缩HTML

gulp-rev 添加版本号 (对所有内容MD5加密 浏览器再次请求资源的时候如果内容发生变化 MD5的值就不一致了 浏览器会再次请求)

gulp-rev-collector 内容替换

gulp-clean清除内容