Gulp使用教程

**一、Gulp简介**

gulp是前端开发过程中一种基于流的代码构建工具，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，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， 它能自动化地完成 javascript、coffee、sass、less、html/image、css 等文件的测试、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官方网址：<http://gulpjs.com>

gulp插件地址：<http://gulpjs.com/plugins>

gulp 官方API：<https://github.com/gulpjs/gulp/blob/master/docs/API.md>

gulp 中文API：<http://www.ydcss.com/archives/424>

gulp常用插件：<http://www.jianshu.com/p/98db023b5b89>

**易于使用**

通过代码优于配置的策略，gulp 让简单的任务简单，复杂的任务可管理。

**构建快速**

利用 Node.js 流的威力，你可以快速构建项目并减少频繁的 IO 操作。

**易于学习**

通过最少的 API，掌握 gulp 毫不费力，构建工作尽在掌握：如同一系列流管道。

**插件高质**

gulp 严格的插件指南确保插件如你期望的那样简洁高质得工作。

**二、安装nodejs**

1、说明：gulp是基于nodejs，理所当然需要安装nodejs；

2、安装：打开[nodejs官网](http://nodejs.org/" \t "http://www.ydcss.com/archives/_blank)，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）。然后像安装QQ一样安装它就可以了（安装路径随意）。

3、使用命令行（如果你熟悉命令行，可以直接跳过）

node -v查看安装的nodejs版本，出现版本号，说明刚刚已正确安装nodejs. ( PS：未能出现版本号，请尝试重启电脑重试.)

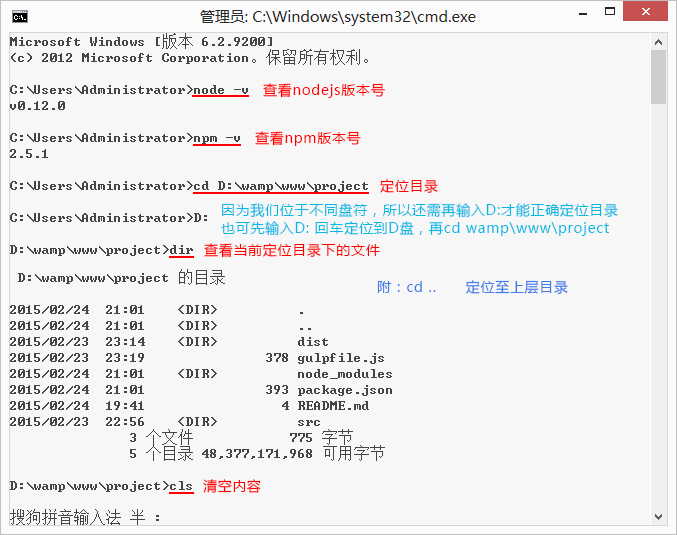
npm -v查看npm版本号， nodejs安装后自动安装了npm命令， 我们需要通过npm命令来安装gulp

cd定位到目录，用法：cd+路径 ；

dir列出文件列表；

D: 进入D盘

cls清空命令提示符窗口内容。

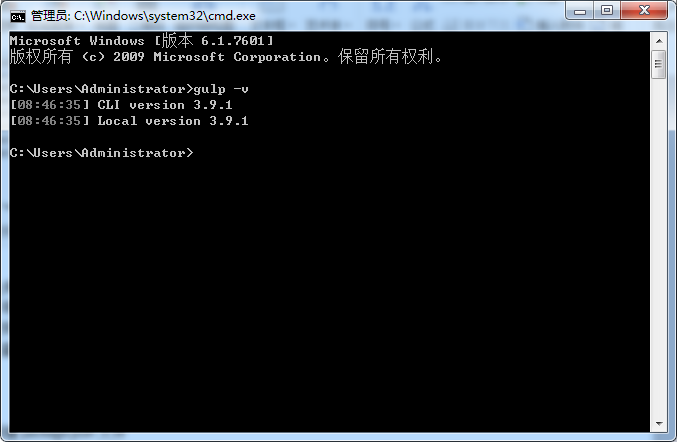


**三、全局安装gulp**

1、说明：全局安装gulp目的是为了通过她执行gulp任务；

2、安装：命令提示符执行npm install gulp -g；安装需要一定的时间.

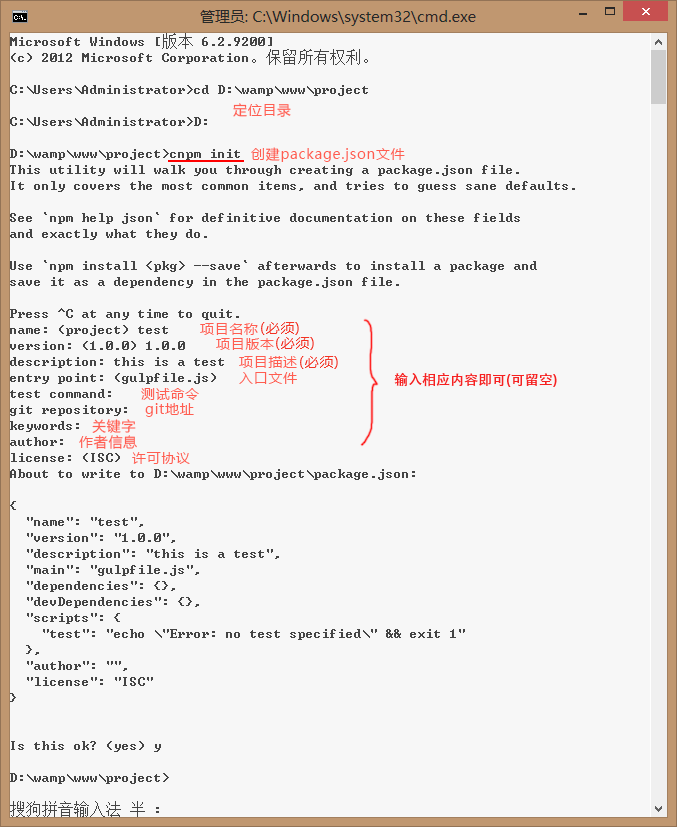
3、查看是否正确安装：命令提示符执行gulp -v，出现版本号即为正确安装。



**四、新建package.json文件**

1. 说明：package.json是基于nodejs项目必不可少的配置文件，它是存放在项目根目录的普通json文件；

2. 进入工程根目录，使用 npm init创建package.json文件 （以下截图中的cnpm 即为npm，实操中请输入npm代替cnpm，文档最后有解释原因）



**五、创建gulpfile.js文件**

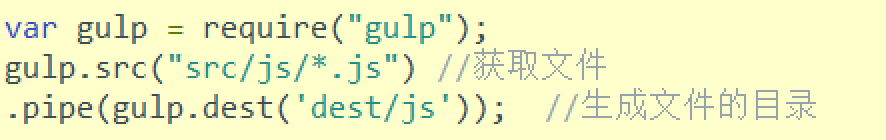
1. gulp也需要一个文件作为它的主文件，在gulp中这个文件叫做**gulpfile.js**。新建一个文件名为gulpfile.js的文件，然后放到你的项目目录中。之后要做的事情就是在gulpfile.js文件中定义我们的任务了。

1. gulp的使用流程一般是：首先通过**gulp.src()**方法获取到想要处理的文件流，然后把文件流通过**pipe()**方法导入到gulp的插件中，最后把经过插件处理后的流再通过pipe方法导入到**gulp.dest()**中，gulp.dest()方法则把流中的内容写入到文件中。

**gulp.src()**用于找到要处理的文件。支持“\*”语法，如："main/\*" , "js/\*.js" ，"\*\*"代表所有文件夹。

**pipe()**返回处理结果以便后面操作的衔接。node函数。

**gulp.dest()**被处理文件的输出路径



**gulp.task()**方法用来定义任务，内部使用的是Orchestrator(用于排序、执行任务和最大并发依赖关系的模块)，其语法为：

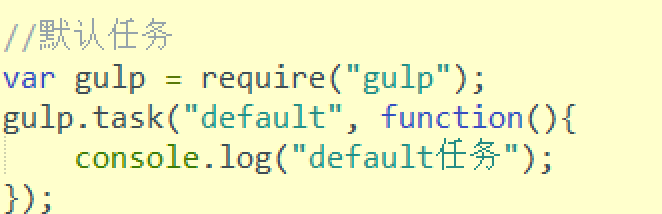
|  |
| --- |
| gulp.task(name[, deps], fn) |

name 为任务名；

deps 是当前定义的任务需要依赖的其他任务，为一个数组。当前定义的任务会在所有依赖的任务执行完毕后才开始执行。如果没有依赖，则可省略这个参数；

fn 为任务函数，我们把任务要执行的代码都写在里面。该参数也是可选的。当你定义一个简单的任务时，需要传入任务名字和执行函数两个.

**默认任务**



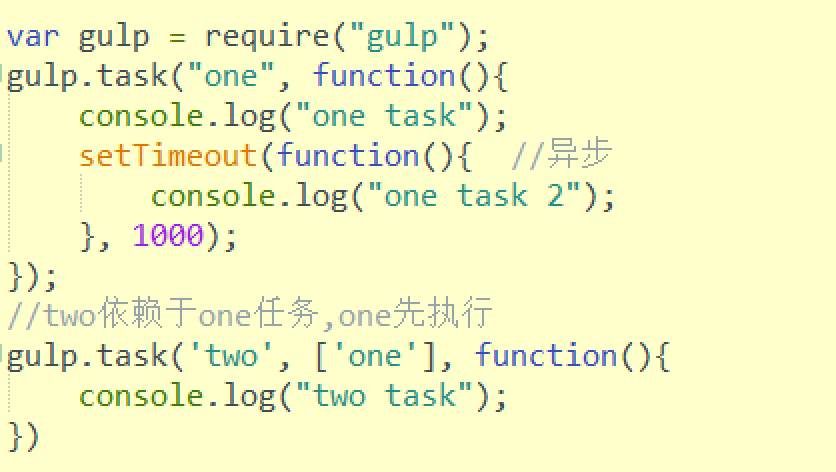
**任务依赖**

可以通过任务依赖来实现。例如我想要执行one,two,three这三个任务，那我们就可以定义一个空的任务，然后把那三个任务当做这个空的任务的依赖就行了：

//只要执行default任务，就相当于把one,two,three这三个任务执行了

gulp.task('default',['one','two','three']);

(Ps:如果任务相互之间没有依赖，任务就会按你书写的顺序来执行，如果有依赖的话则会先执行依赖的任务)但是如果某个任务所依赖的任务是异步的，就要注意了，gulp并不会等待那个所依赖的异步任务完成, 而是会接着执行后续的任务。例如：



**六、安装gulp插件**

1. 安装gulp插件使用命令 npm install --save-dev 插件名称

2. 本示例以html插件**gulp-htmlmin** 为例，执行npm install --save-dev gulp-htmlmin

安装插件(注意要先cd到项目的根目录)

**3. 使用html插件gulp-htmlmin**

在gulpfule.js中输入代码:

var gulp = require("gulp");

var htmlmin = require("gulp-htmlmin");

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

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

.pipe(htmlmin(obj))

.pipe(gulp.dest('dest/html'));

});

gulp.task('default', ['htmlTask']);

**参数(可选)：**

var obj = {

removeComments: true, //清除HTML注释

collapseWhitespace: true, //压缩HTML

collapseBooleanAttributes: true,//省略布尔属性的值<input checked="true"/> ==> <input checked/>

removeEmptyAttributes: true, //删除所有空格作属性值 <input id="" /> ==> <input />

removeScriptTypeAttributes: true, //删除<script>的type="text/javascript"

removeStyleLinkTypeAttributes: true, //删除<style>和<link>的type="text/css"

minifyJS: true, //压缩页面JS

minifyCSS: true //压缩页面CSS

}

1. **使用其他插件**

**压缩js插件gulp-uglify**

**解决中文乱码插件gulp-bom**

**压缩css插件gulp-minify-css**

**压缩html插件gulp-minify-html**

**压缩图片插件gulp-imagemin**

**压缩png图片插件imagemin-pngquant**

**重命名插件gulp-rename**

执行npm install --save-dev 插件名称 安装上面的插件

**在gulpfule.js中输入代码 ：**

var gulp = require('gulp');

var uglify = require('gulp-uglify'); //js压缩插件

var bom = require('gulp-bom'); //解决中文乱码插件

var minifyCss = require('gulp-minify-css'); //css压缩插件

var minifyHtml = require('gulp-minify-html'); //html压缩插件

var imagemin = require('gulp-imagemin'); //图片压缩相关插件

var pngquant = require('imagemin-pngquant'); //png图片压缩插件

var rename = require('gulp-rename'); //重命名插件

//压缩js

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

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

// .pipe(bom())

.pipe(uglify())

.pipe(gulp.dest('dest/js'));

});

//压缩css

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

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

.pipe(minifyCss())

.pipe(gulp.dest('dest/css'));

});

//压缩html

gulp.task('htmlTask2', function(){

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

.pipe(minifyHtml())

.pipe(gulp.dest('dest/html'));

});

//压缩图片

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

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

.pipe(imagemin({

progressive: true, //类型：Boolean 默认：false 无损压缩jpg图片

use: [pngquant()] //使用pngquant插件来深度压缩png图片

// optimizationLevel: 5, //类型：Number 默认：3 取值范围：0-7（优化等级）

// interlaced: true, //类型：Boolean 默认：false 隔行扫描gif进行渲染

// multipass: true //类型：Boolean 默认：false 多次优化svg直到完全优化

}))

.pipe(gulp.dest('dest/img'));

});

//重命名

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

gulp.src('src/js/bullet.js')

.pipe(uglify())

.pipe(rename('bullet.min.js'))

.pipe(gulp.dest('dest/js'));

});

gulp.task('default', ['jsTask', 'cssTask', 'htmlTask2', 'imgTask', 'renameTask']);

gulp-babel： es6转es5插件

使用方法：

1,安装插件:

安装 Gulp 上 Babel 的插件

npm install --save-dev gulp-babel

安装 Babel 上将 ES6 转换成 ES5 的插件

npm install --save-dev babel-preset-es2015

安装 Babel 上将 ES6 转换成 ES5 的插件

npm install --save-dev babel-core

2,在gulpfile.js中添加以下代码

//先es6转es5，再压缩js

var gulp = require('gulp');

var babel = require('gulp-babel'); //es6转es5

var uglify = require('gulp-uglify'); //js压缩插件

//压缩js

gulp.task('jsTask2', function(){

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

.pipe(babel({"presets": ["es2015"]})) //es6转es5

.pipe(uglify()) //js压缩

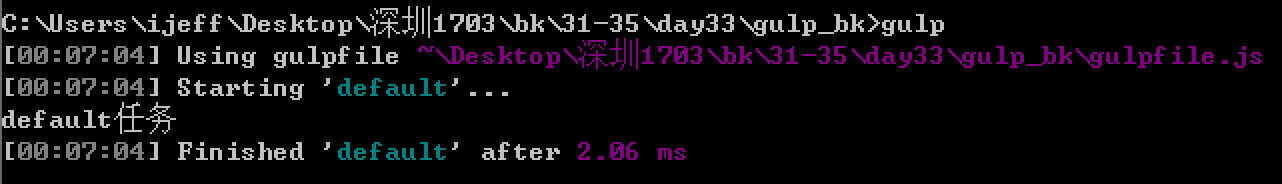
.pipe(gulp.dest('dest/js'));

});

gulp.task('default', ['jsTask2']);

**七、执行gulp**

1. 输入命令：gulp 来执行gulp



**淘宝镜像**

ps：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。这是一个完整 npmjs.org 镜像，你可以用此代替官方版本(只读)，同步频率目前为 10分钟 一次以保证尽量与官方服务同步。

官方网址：[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；

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）。