自动化构建工具gulp

1. **为什么要使用gulp？优势?** **8vA8G2X33z**

**1**. 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。从而使开发更加快速高效。

**2**. 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。

**二、安装步骤简介与安装依赖**

**基本步骤：安装nodejs-->全局安装gulp-->项目安装gulp以及gulp插件-->配置gulpfile文件-->运行任务**

****

**1.为啥要安装nodejs?**

说明：gulp是基于nodejs，理所当然需要安装nodejs；

**2.gulp安装配置、插件安装方式(npm、cnpm)？**

说明：npm（node package manager）nodejs的包管理器，用于node插件管理（包括安装、卸载、管理依赖等）；但是，由于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。如何解决？

npm安装插件命令：

npm install xxx –g 全局安装/npm install xxx –save-dev 安装到当前目录

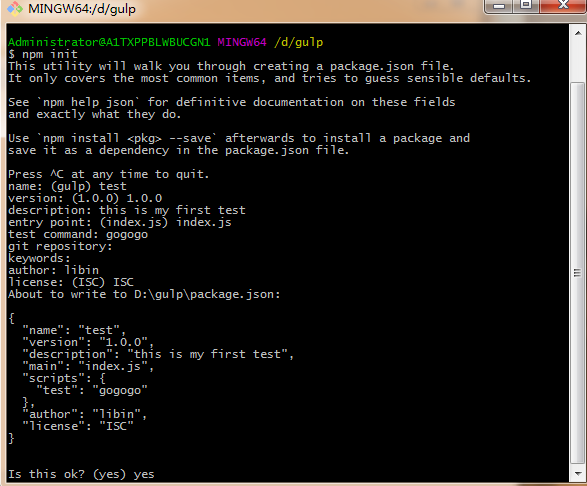
cnpm安装淘宝镜像(选装)：

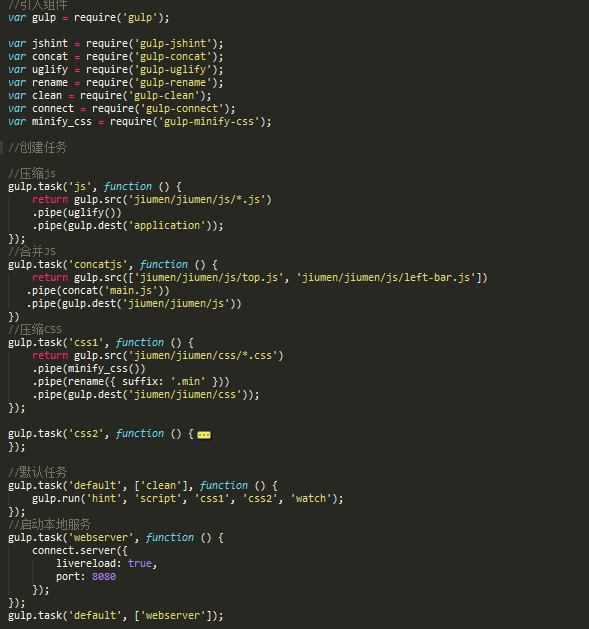
**npm install cnpm -g --registry=https://registry.npm.taobao.org**；

**--save**：将保存配置信息至package.json（package.json是[nodejs项目配置文件](http://www.ydcss.com/archives/18" \l "lesson6)）；

**-dev**：保存至package.json的devDependencies节点，不指定-dev将保存至dependencies节点；一般保存在dependencies的像这些express/ejs/body-parser等等。

为什么要保存至package.json？因为node插件包相对来说非常庞大，所以不加入版本管理，将配置信息写入package.json并将其加入版本管理，其他开发者对应下载即可.如何配置呢？ 1.npm init命令 2.本地创建一个package.json文件 网上去百度一个就行但要注意文件里不能有注释。

**3.配置Gulpfile文件**

****

**4.常用插件介绍**

**Gulp-less/gulp-sass/gulp-livereload/gulp-minify-html/gulp-imagemin(imagemin-pngquant)/gulp-clean**

**注：gulp-clean用法误区，错误示例gulp.task(“default”,[“clean”,”less”,”images”,”js”]);//同步**

**解决：gulp.task(“default”,[‘clean’],function(){**

**gulp.start(‘less’,’images’,’js’);**

**})**

**三、运行任务**

**四、gulp与webpack混搭、区别**

gulp.task("webpack", function(callback) {

       webpack({

          }, function(err, stats) {

       if(err) throw new gutil.PluginError("webpack", err);

       gutil.log("[webpack]", stats.toString({

           // output options

       }));

       callback();

    });

});

**只需要把webpack配置写到webpack（{…}）中去即可，就不需要写webpack.config.js了**

**gulp**是工具链、构建工具，可以配合各种插件做js压缩，css压缩，less编译 替代手工实现自动化工作

1.构建工具

2.自动化

3.提高效率用

webpack是文件打包工具，可以把项目的各种js文、css文件等打包合并成一个或多个文件，主要用于模块化方案，预编译模块的方案

1.打包工具

2.模块化识别

3.编译模块代码方案用

**五、使用外部配置**

好处：让任务更加符合DRY原则，并且config.json可以被其他的任务运行器使用，比如grunt.

Config.json