# 2018.04.26

## 构建工具

构建工具是自动从源代码创建可执行应用程序的程序。对于需要反复重复的任务，例如压缩（minification）、编译、单元测试、linting等，自动化工具可以简化开发工作，提高效率。

## grunt

grunt安装

1. 全局安装grunt-cli cnpm i grunt-cli -g
2. 新建项目，需要准备package.json和Gruntfile.js
   1. package.json需要添加开发依赖

“devDependencies”: {

“grunt”: “~1.0.1”,

“grunt-contrib-uglify”:”~0.5.0” //压缩丑化插件

}

b) Gruntfile.js

下载插件，安装并保存到开发依赖。

cnpm i 插件名 --save-dev

导出函数，把grunt作为参数传入。

module.exports=function(grunt){

grunt.initConfig({

在配置文件中加载插件

grunt.loadNpmTask(“插件名”);

注册任务

grunt.registerTask(“default”,[任务名]);

})

}

## gulp

易于使用，构建快速（基于node.js流，减少IO操作），插件高质，易于学习。

gulp安装

1. 全局安装gulp cnpm i gulp -g
2. 新建项目，需要准备package.json(npm init -y)和gulpfile.js
3. 在命令行运行gulp（自动找默认任务）

gulp的基本使用步骤

1. 在当前项目安装本地gulp模块，并且保存到开发依赖。

cnpm i gulp --save-dev

1. 在gulpfile.js里面，引入gulp模块

var gulp=required(‘gulp’)

1. 使用gulp模块，开启任务（需要什么任务就使用什么插件）

//引入所有的插件

var gulp=require('gulp');

var cleanCss=require('gulp-clean-css');

var imagemin=require('gulp-imagemin');

var less=require('gulp-less');

var rename=require('gulp-rename');

var uglify=require('gulp-uglify');

var livereload = require('gulp-livereload');

//编译less

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

gulp.src('./style/less/index.less')

.pipe(less())

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

.pipe(livereload()) //在任务中开启热更新

});

//压缩css+重命名

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

gulp.src('./style/css/index.css')

.pipe(cleanCss())

.pipe(rename({

suffix:'.min'

}))

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

.pipe(livereload())

});

//压缩js+重命名

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

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

.pipe(uglify())

.pipe(rename({

suffix:'.min'

}))

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

.pipe(livereload())

});

//压缩图片

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

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

.pipe(imagemin())

.pipe(gulp.dest('./dist/images'))

});

//开始观察（gulp自带，不需要下载）

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

livereload.listen();

gulp.watch('./style/less/index.less',['less']);

gulp.watch('./style/css/index.css',['cleanCss']);

gulp.watch('./js/\*.js',['uglify']);

})