Gulp前端自动化构建工具

主要方法:

1. gulp.task（‘任务名’，function（）｛｝）//创建任务
2. gulp.src(‘./\*.js’) //指定想处理的文件
3. gulp.dest() //指定处理后的文件存放位置
4. gupl.watch( ) //自动监视 文件变化
5. gulp.run(‘任务名’) //执行相对应的任务。
6. .pipe(其他方法) //管道（很像链式编程）

首先需要安装gulp

1. 初始化npm：npm init
2. 下载gulp：npm install gulp-cli -save
3. 还要在当前文件夹下载，npm install gulp --save

使用gulp

1. 在项目目录下新建一个gulpfile.js的文件
2. 文件中再引入主模块var gulp = require(‘gulp’);
3. gulp.task(‘mingzi’,function(){放压缩代码，引用、处理、输出})
4. 在cmd中执行gulp mingzi
5. 压缩js，

下载：npm install gulp-uglify --save

引入：var uglify = require(‘gulp-uglify’);

使用：uglify()

1. 合并(可以分别合并js或者css)，

下载：npm install gulp-concat --save

引入：var concat = require(‘gulp-concat’);

使用：gulp.src([url1,url2,url3]) concat(‘index.js’) //参数为合并的js名字

1. 压缩css，

下载：npm install gulp-cssnano --save

引入：var concat = require(‘gulp-cssnano’);

使用：cssnano()

1. 压缩html

下载：npm install gulp-htmlmin --save

引入：var htmlmin = require(‘gulp-htmlmin’)

使用：htmlmin({collapseWhitespace:true,minifyJS:true,minifyCSS:true})

1. 监视文件变化watch

gulp.watch([‘监视的文件1’,’监视的文件2’],[‘执行的命令’])

可以配合gulp.run(‘执行的命令’)

代码案例

var gulp = require('gulp');

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

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

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

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

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

console.log(11111111111)

})

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

gulp.src(['./swj/js/cajax.js','./swj/js/wajax.js'])

.pipe(concat('index.js'))

.pipe(uglify())

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

})

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

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

.pipe(concat('index.css'))

.pipe(cssnano())

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

})

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

gulp.src('./swj/cheya.html')

.pipe(htmlmin({collapseWhitespace:true,minifyJS:true,minifyCSS:true}))

.pipe(gulp.dest('./newswj'))

})

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

gulp.run('script')

gulp.watch(['./swj/js/cajax.js','./swj/js/wajax.js'],['script'])

})