

gulp 简介

gulp 是前端开发过程中对代码进行构建的工具,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;大大提高我们的工作效率。

gulp 是基于 Nodejs 的自动任务运行器,她能自动化地完成 javascript/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。

第一步:安装 Node

首先,最基本也是最重要的,我们需要搭建 node 环境。

第二步:使用命令行测试

为了确保 Node 已经正确安装,我们执行几个简单的命令。

```
node -v
```

```
npm -v
```

得到相应的版本号则代表 node 和 npm 安装成功。

第三步:安装 gulp

NPM 是基于命令行的 node 包管理工具,它可以将 node 的程序模块安装到项目中,在它的官网中可以查看和搜索所有可用的程序模块。

在命令行中输入

```
npm install -g gulp
```

1. npm 是安装 node 模块的工具,执行 install 命令。

2. -g 表示在全局环境安装,以便任何项目都能使用它。

3. 最后,gulp 是将要安装的 node 模块的名字。

运行时注意查看命令行有没有错误信息,安装完成后,你可以使用下面的命令查看 gulp 的版本号以确保 gulp 已经被正确安装。

```
gulp -v
```

接下来,我们需要将 gulp 安装到项目本地

```
npm install --save-dev gulp
```

第四步:新建 Gulpfile 文件,运行 gulp

1、安装好 gulp 后需要告诉它要为我们执行哪些任务,首先,我们自己需要弄清楚项目需要哪些任务。

- 检查 Javascript
- 编译 Sass （或 Less 之类的）文件
- 合并 Javascript
- 压缩并重命名合并后的 Javascript

2、安装依赖

```
npm install gulp-jshint gulp-concat gulp-uglify gulp-rename gulp-jshint-html-reporter
```

```
gulp-imagemin --save-dev
```

插件介绍

gulp-jshint ： 检查语法错误

gulp-concat ： 合并

gulp-uglify ： 压缩

gulp-rename ： 重命名

gulp-jshint-html-reporter ： 语法错误报告

gulp-imagemin ： 图片压缩

3、新建 gulpfile 文件

现在，组件都安装完毕，我们要新建 gulpfile 文件以指定 gulp 需要为我们完成什么任务。

gulp 只有五个方法：task，run，watch，src，和 dest，在项目根目录新建一个 js 文件并命名为 gulpfile.js。

压缩、合并 js 代码

1、引入 gulp

```
var gulp = require('gulp');
```

2、引入组件

```
var jshint = require('gulp-jshint');//检查语法错误
```

```
var concat = require('gulp-concat');//合并
```

```
var uglify = require('gulp-uglify');//压缩
```

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

```
var imagemin = require('gulp-imagemin');//图片压缩
```

3、检查脚本

gulp.task('jshint', function() { //jshint 任务会检查 js/ 目录下得 js 文件有没有报错或警告。

```
    gulp.src('js/*.js')
      .pipe(jshint())
      .pipe(jshint.reporter('gulp-jshint-html-reporter', { filename: 'error.html' })); // 输出结果到
自定义的 html 文件
});
```

4、合并，压缩js 文件

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

//scripts 任务会合并 js/ 目录下得所有得 js 文件并输出到 dist/ 目录

//然后 gulp 会重命名、压缩合并的文件，也输出到 dist/ 目录。

```
    gulp.src('js/*.js')
      .pipe(concat('all.js'))//合并
      .pipe(gulp.dest('./dist'))//输出
      .pipe(rename('all.min.js'))
      .pipe(uglify())//压缩
      .pipe(gulp.dest('./dist'))//输出
  });
```

5、使用 .run() 方法关联和运行我们上面定义的任务，使用 .watch() 方法去监听指定目录的文件变化，当有文件变化时，会运行回调定义的其他任务。

```
gulp.task('default', function() {
  gulp.run('jshint', 'scripts');
  // 监听文件变化
  gulp.watch('js/*.js', function() {
    gulp.run('lint', 'scripts');
  });
});
```

```
//合并,压缩 css 文件

var gulp = require('gulp');

var cssmin = require('gulp-minify-css');

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

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

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

        .pipe(concat('all.css'))//合并

        .pipe(gulp.dest('distcss/'))//输出*/

        .pipe(cssmin())//压缩

        .pipe(gulp.dest('distcss1/'))//输出

});

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

    gulp.run('testCssmin');

    // 监听文件变化

    gulp.watch('css/*.css', function() {

        gulp.run('testCssmin');

    });

});
```

```
//合并 , 压缩 html 文件

var gulp = require('gulp');

var minifyHtml = require("gulp-minify-html");

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

    gulp.src('*.html') // 要压缩的 html 文件

        .pipe(minifyHtml()) // 压缩

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

});
```