Gulp学习

1. 什么是gulp以及其特性
2. 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；它不仅能对网站资源进行优化，而且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，使用它，可以大大提高我们的工作效率
3. 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
4. gulp和grunt非常类似，但相比于grunt的频繁io操作，gulp的流操作，能够更快地更便捷地完成构建工作
5. 安装使用
6. 安装nodeJS：gulp是基于nodeJS，所以需要先安装nodeJS
7. 使用命令行：打开命令行（window+r输入cmd回车）

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

npm-v 查看npm的版本号，npm是在安装nodeJS时一同安装的nodeJS包管理器

cd 定位到目录，用法cd + 路径

dir 列出文件列表

cls 清空命令提示符窗口内容

1. npm介绍

3.1 npm nodeJS的包管理器，用于node插件管理（包括安装、卸载、管理依赖等）

3.2 使用npm安装插件：命令提示符执行npm install <name> [-g] [--save-dev]

3.2.1 <name>：node插件名称，例如：npm install gulp-less --save-dev

3.2.2 -g：全局安装，将会安装在c:\users\administrator\appdata\roaming\npm，并且写入环境变量；非全局安装：将会安装在当前定位目录；全局安装可以通过命令行在任何地方调用它，本地安装将安装在定位目录的node\_modules文件下，通过require()调用

3.2.3 --save：将保存配置信息至package.json（package.json是nodeJS项目的配置文件）

3.2.4 --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dependecies的像这些express/ejs/body-parser等等

3.2.4 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，其它开发者对应下载即可（命令提示符执行npm install，则会根据package.json下载所以需要的包，npm install --production 只下载dependencies节点的包）

3.3 使用npm卸载插件：npm uninstall name -g --save-dev（不要直接删除本地插件包）

3.3.1 删除全部插件：借助rimraf：npm install rimraf -g 用法：rimraf node\_modules

3.4 使用npm更新插件：npm update name -g --save-dev

3.4.1 更新全部插件：npm update --save-dev

3.5 查看npm帮助：npm help

3.6 当前目录已安装插件：npm list

1. 选装cnpm

4.1 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，淘宝团队分享了一个完整npmjs.org镜像，可以代替官方版本（只读），同步频率为10分钟一次以保证尽量与官方服务器同步

4.2 官方网址：<http://npm.taobao.org;>

4.3 安装：命令提示符执行npm install cnpm -g --registry=https://registry.npm.taobao.org（npm和cnpm用法完全一致）

1. 全局安装gulp

5.1 全局安装gulp目的是为了通过它执行gulp任务

5.2 安装：命令提示符执行npm install gulp -g

5.3 通过提示符执行gulp-v 查看版本号查看是否正确安装

6.新建package.json文件

6.1 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

{

  "name": "test",   //项目名称（必须）

  "version": "1.0.0",   //项目版本（必须）

  "description": "This is for study gulp project !",   //项目描述（必须）

  "homepage": "",   //项目主页

  "repository": {    //项目资源库

    "type": "git",

    "url": "https://git.oschina.net/xxxx"

  },

  "author": {    //项目作者信息

    "name": "surging",

    "email": "surging2@qq.com"

  },

  "license": "ISC",    //项目许可协议

  "devDependencies": {    //项目依赖的插件

    "gulp": "^3.8.11",

    "gulp-less": "^3.0.0",

"gulp-jshint": "^2.0.0",

  }

}

6.2 也可通过提示符执行npm init 来创建package.json文件

6.3 npm help package.json 可查看package.json帮助文档

1. 本地安装gulp插件

7.1 定位项目目录后命令提示符执行npm install --save-dev

7.2 以colors为例，npm install colors --save-dev，安装成功在node\_modules目录下有colors文件

7.3 为了能正常使用，本地安装gulp（可能同时会安装其它依赖的插件）

1. 新建gulpfile.js文件

8.1 gulpfile.js是gulp项目的配置文件，是位于项目目录的普通js文件

//导入工具包 require('node\_modules里对应模块')

var gulp = require('gulp'), //本地安装gulp所用到的地方

    less = require('gulp-less');

//定义一个testLess任务（自定义任务名称）

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

    gulp.src('src/less/index.less') //该任务针对的文件

        .pipe(less()) //该任务调用的模块

        .pipe(gulp.dest('src/css')); //将会在src/css下生成index.css

});

gulp.task('default',['testLess', 'elseTask']); //定义默认任务 elseTask为其他任务，该示例没有定义elseTask任务

//gulp.task(name[, deps], fn) 定义任务  name：任务名称 deps：依赖任务名称 fn：回调函数

//gulp.src(globs[, options]) 执行任务处理的文件  globs：处理的文件路径(字符串或者字符串数组)

//gulp.dest(path[, options]) 处理完后文件生成路径

1. 运行gulp

9.1 命令提示符执行gulp 任务名称

9.2 编译less：命令提示符执行gulp testLess

9.3 当执行gulp将会调用default任务里的所有任务[‘testLess’，‘elesTask’]

1. 项目使用
2. 全局安装nodeJS，gulp
3. 新建package.json文件（注明需要安装的插件）
4. 本地安装gulp（会下载安装要用的插件及其依赖的插件）
5. 新建gulpfile.js文件，命令提示符运行gulp任务