1、概述脚手架实现的过程，并使用 NodeJS 完成一个自定义的小型脚手架工具

项目地址：https://github.com/sqw2016/wsq-cli.git

脚手架实现的过程：脚手架的功能主要是提供项目初始结构。其实现步骤主要分为以下几步：

1. 发起命令行询问
2. 根据命令行交互的答案读取文件
3. 根据情况对文件进行处理，如通过模板语言处理文件，如ejs处理html文件
4. 输出文件到目标位置
5. 尝试使用 Gulp 完成项目的自动化构建

项目地址：

项目的自动化构建的实现：自动化构建主要实现开发到生产代码的转换。通过gulp的实现时，首先对文件进行基础的编译处理，需要新建以下基础任务：

1. page: 编译html，使用gulp-swig模板对HTML文件进行处理，并输出到临时文件目录temp。
2. style: 编译sass文件，通过gulp-sass将.scss文件编译为css文件，并输出到临时文件目录temp。
3. js: 编译js文件，通过gulp-babel将.js文件编译为es5，并输出到临时文件目录temp。
4. image: 压缩图片文件，通过gulp-imagemin对图片进行压缩处理，并将压缩后的文件存入到最终的dist文件目录中。
5. font: 处理字体文件，对字体文件中的图片进行压缩处理，处理方法和图片的处理方法一致。
6. other: 对其他文件进行处理，主要是public中的文件进行处理，直接复制到最终的dist文件目录中。
7. useref: 资源合并，通过temp目录html文件中的编译注释将引用的文件合并到一个文件，并最终输出到dist目录中。由于模式不同时，useref中的处理也不同，因此定义为一个接收是否为production的参数，返回一个任务。production为true时表示默认为production模式，需要对文件进行压缩。production为false时表示默认为develop模式，不需要对文件进行压缩。



由于可以通过命令行参数来指定是否为production模式，需要在useref任务时接收并解析命令行参数，并根据参数来指定模式。

上面基础任务中的临时目录temp的创建是为了防止useref合并资源文件时的读写冲突。

然后根据package中需要提供的命令新建任务：

1. clean: 清理dist和temp目录，通过 del 组件删除dist和temp目录
2. lint: 对js和sass代码进行检查，使用gulp-eslint和 gulp-sass-lint 分别处理js和sass文件。
   1. 安装gulp-eslint、gulp-sass-lint
   2. 定义eslint和sassLint任务，分别检查.js文件和.scss文件
   3. 定义lint任务，使用gulp.parallel组合eslint和sassLint，执行lint任务时同时开始执行eslint和sassLint任务

3、compile: 编译css、js、html。通过gulp.parallel并行执行page、style和js任务

4、serve: 以开发模式启动项目，需要启动一个本地的web服务器，并且在源文件改变时自动重新编译并刷新浏览器。在启动时可以通过 --open 指定是否自动打开浏览器，通过--part num 指定启动服务的端口

(1) 要启动项目，首先要编译文件，即执行compile任务

(2) 定义一个server任务启动本地web服务，通过browser-sync 来实现启动web服务。由于开发模式和生产模式下启动web服务会从不同的目录下读取文件，因此定一个接收开始目录参数的函数，并将并返回一个server任务。



在server任务中，初始化web服务时，需要通过配置选项中的server属性的baseDir属性来正确访问图片等源文件。server属性的routes来将指向/node\_modules的路径指到本地node\_modules



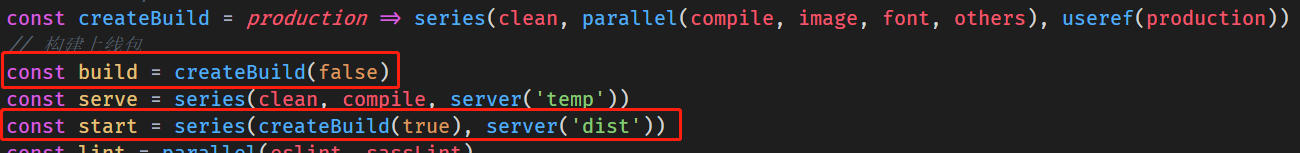
由于命令行参数--open和--port可以改变web服务的初始配置，因此在这里通过process.argv来获取到命令行执行时的参数，并将对应的值填充到服务的配置选项中。

(3) 执行compile任务时会产生temp临时目录，因此每次在执行serve任务时首先应该执行clean任务，清除产生的temp目录。故此serve任务就是依次执行clean、compile和server命令

5、build: 将文件编译输出到dist目录中，并且在production模式下需要对文件进行压缩。

build任务需要依次执行clean、编译（同时执行compile、image、font和others）和默认为develop模式的useref任务。

6、start: 和serve任务基本一致，只是以生产模式构建项目，同时web服务启动时从dist目录读取文件。



7、deploy: 将文件上传到指定的位置。通过gulp-gh-pages插件来实现。解析命令行参数--branch，通过命令行中传入的branch参数来指定插件配置项中的branch。配置项中的remoteUrl是git提交的地址，默认使用当前项目的git提交地址，如果使用其它可以通过remoteUrl进行设置。

