一、gulp的安装

1、安装gulp命令行工具

npm install –global gulp-cli

2、在项目目录下创建package.json

npm init

3、安装gulp，作为开发时依赖项

npm install –save-dev gulp

4、检查gulp版本

gulp –version

5、创建gulpfile.js文件

在项目的根目录下，创建gulpfile.js。用于配置打包，项目启动等。在运行gulp命令时，会自动加载。

二、gulp的使用

1、gulp创建任务

每个gulp任务（task）都是一个异步的javascript函数，此函数是一个可以接收callback作为参数的函数，也可以是一个返回stream(流)、promise、event emitter(事件发射器)、child process(子进程)或observable(可观测的)类型值的函数。

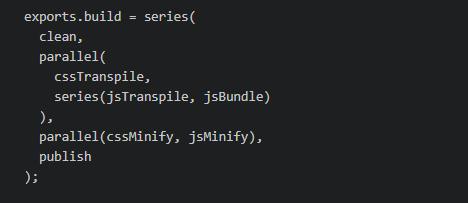
2、导出任务

任务可以是公开的（通过export导出的可以直接使用gulp命令调用的任务），也可以是私有的（只能在内部使用，一般作为series()或parallel()组合的组成部分）。注：在以前的版本中，通常使用task()方法将函数注册为任务，虽然现在这个api还可以使用，但是导出（export）将会是主要的注册机制。

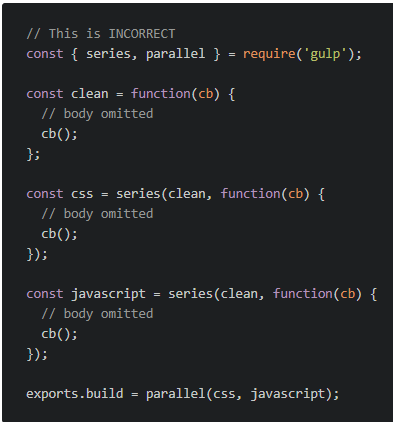
3、组合任务

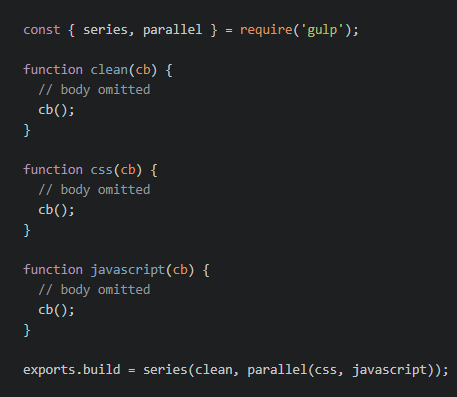
gulp提供了两个将多个独立任务组合成一个更强大的任务的方法：series()和parallel()；这两个方法接受任意数目的任务函数（task）或已经组合过的任务。而且series()和parallel()可以相互无限嵌套。

用法如下：



二者区别在于，series()方法会按照顺序执行任务（task）；而parallel()会以最大并发量来运行任务。

注：当一个组合操作执行时，这个组合中的每一个任务每次被调用时都会被执行，例如在调用组合了两个不同的任务且这两个任务之前都调用了clean任务，这时，操作组合执行时，就会执行两次clean任务，这就会导致产生不可预期的结果。因此最好重构组合中的clean任务。



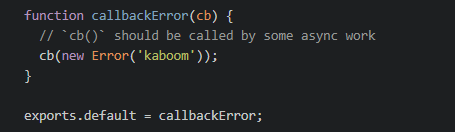
重构前 重构后

4、异步处理

当任务中返回stream(流)、promise、event emitter(事件发射器)、child process(子进程)或observable(可观测的)时，成功或错误值将通知gulp是否继续执行或结束。如果任务出错，gulp会立即结束执行并显示该错误。

当使用series()组合任务时，若一个任务执行报错，将不会执行后续任务；但如果使用parallel()组合任务，一个任务出错将结束整个组合任务，但组合中其他任务是否执行完成不可知。

注：如果一个任务没有任何返回内容，则必须使用callback来指示任务已经完成。如需通过callback把任务（task）中的错误告知gulp，请将Error作为callback的唯一参数。



5、处理文件

src()方法接受glob参数，并从文件系统中读取文件然后生成一个Node流（stream），它将匹配到的所有文件存储到内存中，并通过流（stream）进行处理。由src()产生的流应当从任务中返回并发出异步完成的信号。

流（stream）所提供的主要API是.pipe()方法，用于连接转换流或可写流

dest()接受一个输出目录作为参数，并且他还会产生一个Node流，通常作为终止流。当他接收到通过管道传输过来的文件时，它会将文件内容和文件属性写入到指定的目录中。

gulp还提供了symlink()方法，功能与dest()类似，但是创建的链接而不是文件。

6、glob详解

glob是由普通字符和/或通配符组成的字符串，用于匹配文件路径。

src()方法接收一个glob字符串或由多个glob字符串组成的数组作为参数，用于确定哪些文件需要被操作。glob或glob数组必须至少匹配到一个匹配项，否则src()将会报错。当使用glob数组时，将按照每个glob在数组中的位置依次匹配，这在取反操作时有用。

字符串片段与分隔符：/(不区分操作系统)

转译符：\\

\*（一个星）：在一个字符片段中匹配任意数量的字符，包括零个匹配。对于匹配单级（当前目录或指定目录的下，不匹配更深一级目录）目录下的文件很有用。

\*\*（两个星）：在多个字符片段中匹配任意数量的字符，包括零个匹配。对于匹配嵌套目录下的文件很有用。请确保适当的限制两个星号的glob的使用，以避免匹配到大量不必要的目录。

！：取反，由于 glob 匹配时是按照每个 glob 在数组中的位置依次进行匹配操作的，所以 glob 数组中的取反（negative）glob 必须跟在一个非取反（non-negative）的 glob 后面。第一个 glob 匹配到一组匹配项，然后后面的取反 glob 删除这些匹配项中的一部分。

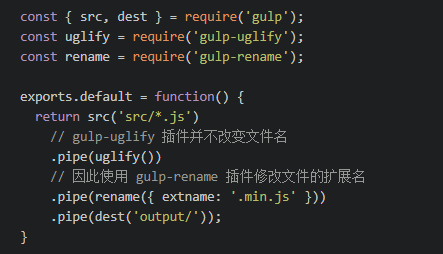


匹配重叠：两个或多个glob匹配到了相同的文件。若在同一个src()中，glob会尽力去重，若在不同的src()中不会去重。

7、插件

gulp的插件实际上是Node转换流，他封装了通过管道转换文件的常见功能，通常是使用.pipe()放在src()和dest()之间。他们可以更改通过流的每个文件的文件名、元数据或文件内容。

注：托管到npm的gulp插件标记有“gulpplugin”和“gulpfriendly”关键词，可在<https://gulpjs.com/plugins/>上搜索。

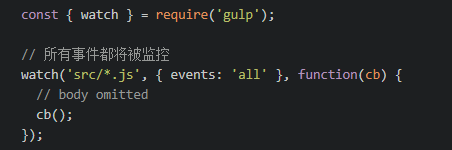


8、文件监控

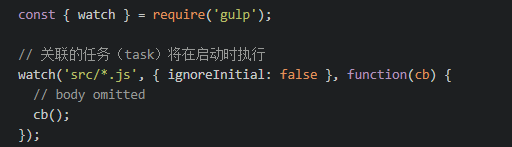
gulp api 中的watch()方法利用文件系统的监控程序将globs和文件相关联。他对匹配glob的文件进行监控，如果有文件被修改了就执行相关联的任务。如果被执行的任务没有触发异步完成信号，他将永远不会再次执行。（执行报错场景下吗？）



默认情况下，只要创建、更改或删除文件，文件监控程序就会执行关联的任务（task）。 如果你需要使用不同的事件，你可以在调用 watch() 方法时通过 events 参数进行指定。可用的事件有 'add'、'addDir'、'change'、'unlink'、'unlinkDir'、'ready'、'error'。此外，还有一个 'all' 事件，它表示除 'ready' 和 'error' 之外的所有事件。



调用 watch() 之后，关联的任务（task）是不会被立即执行的，而是要等到第一次文件修之后才执行。如需在第一次文件修改之前执行，也就是调用 watch() 之后立即执行，请将 ignoreInitial 参数设置为 false。



watch()方法默认执行完一个任务再执行下一个，按照队列执行所有任务，将 queue 参数设置为 false后，可以禁止队列，这样每次文件修改后都会马上执行关联任务，就会导致有可能会产生并发。

文件更改后，会默认有200毫秒的延迟才会触发文件监控程序关联的任务。可通过delay参数更改延迟时间（正整数）。