gulp学习之察言观色watch函数

gulp.watch函数是用来监听文件的变化，当文件发生变化以后，可以通过回调函数或者指定的任务执行相应的内容。

gulp.watch(glob[,opts],tasks);

glob的用法和src中的参数glob的用法相同。可以参考src中的用法

opts这个参数不常用，这里略过不题。

tasks 监听到文件变化以后执行的任务

var gulp = require(‘gulp’);

gulp.task(‘onmpw’,function(){

console.log(‘onmpw’);

})

gulp.task(‘jiyi’,function(){

console.log(‘jiyi’);

})

gulp.watch('lib/onmpw.js', ['onmpw','jiyi']);

当lib目录下的onmpw.js文件发生变化就会执行onmpw和jiyi两个任务。执行完成以后继续监听lib目录下的onmpw.js文件。

上面是使用tasks方式，还可以使用回调函数的方式

gulp.watch(glob[,opts,cb]);

例如

var gulp = require(‘gulp’);

gulp.watch('lib/onmpw.js', function(e){

console.log(e.type);

});

当lib/onmpw.js文件有变化时执行回调函数。e.type的类型有增加，删除和修改。也就是说除了修改onmpw.js文件，当删除或者增加lib/onmpw.js时也会触发回调函数。同时会在回调函数中打印出类型。

Starting 'watch'...

Finished 'watch' after 14 ms

changed

deleted

当我修改lib/onmpw.js，或者删除该文件以后会显示上面的结果。当再次新建onmpw.js文件以后，就会再次出现下面的结果。

Starting 'watch'...

Finished 'watch' after 14 ms

changed

deleted

added

但是当我们删除了这个文件，那watch监听的文件就消失了。所以当前的监听任务也就成了一个僵尸任务。除非我们再次新建这个文件，如果不新建是不是就需要将这个监听任务终止掉。

gulp.task('watch',function(){  
 gulp.watch('lib/onmpw.js',function(e){  
 if(e.type == 'deleted'){  
 process.exit(0);  
 }  
 })  
})

执行上面的程序，当lib/onmpw.js文件被删除了以后，该监听任务就相应的退出了。

以上就是gulp.watch的用法。