# [Executors创建的4种线程池的使用](https://www.cnblogs.com/ljp-sun/p/6580147.html)

## newCachedThreadPool

创建一个可缓存线程池，如果线程池长度超过处理需要，可灵活回收空闲线程，若无回收，则新建线程

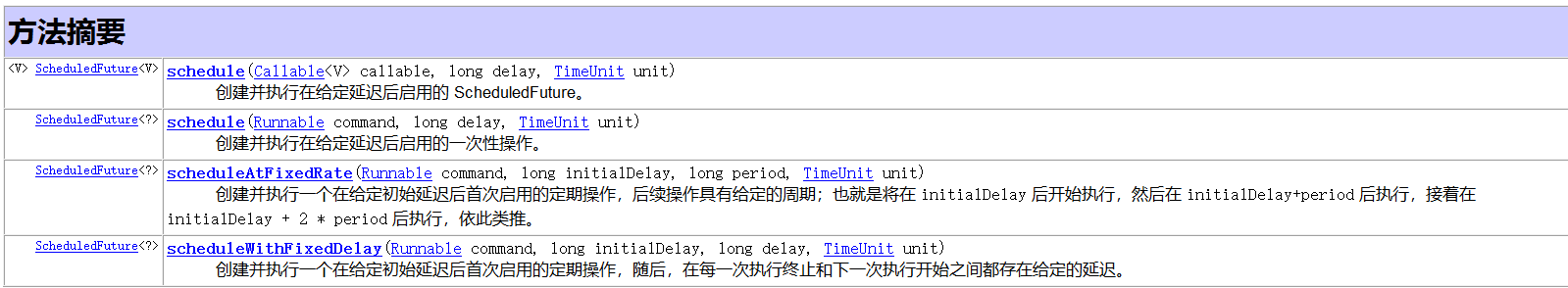
## newFixedThreadPool

创建一个定长的线程池，可控制线程最大并发数，超出线程会在队列中等候

## newscheduleThreadPool

创建一个定长线程池，支持定时及周期性执行任务

### ScheduleExecutorService



#### 四个方法:

* Schedule(实现了Callable或者实现了Runable的类,延时多久开始执行，时间单位),注意要手动shutdow(),否则线程会阻塞
* scheduleAtFixedRate(实现了Runable的类，延时多久开始执行，周期时间，时间单位)，注意周期时间要减去实现了Runable的类的休眠时间

executorService.scheduleAtFixedRate(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **try** {  
 System.***out***.println(**"run 线程名称"**+Thread.*currentThread*().getName()+ System.*currentTimeMillis*());  
 TimeUnit.***SECONDS***.sleep(2);  
 System.***out***.println(**"runB 线程名称2"**+Thread.*currentThread*().getName()+System.*currentTimeMillis*());  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
}, 3, 5, TimeUnit.***SECONDS***);

比如说休眠了2秒，但是线程周期是5秒，所以5-2=3秒，间隔周期会变成3秒，如果周期小于执行代码所需的时间，则参考API

* scheduleWithFixedDelay(实现了Runable接口的类，延时多久执行，周期，时间单位)，这个方法不会因为Runable接口的类休眠多久，而影响周期