

# 线程常用方法

---

## 一、sleep ()——线程睡眠

- 使线程停止运行一段时间，将处于阻塞状态
- 如果调用了sleep方法之后，没有其他等待执行的线程，这个时候当前线程不会马上恢复执行！

## 二、线程让步——yield()

1、作用：

- 让当前正在执行线程暂停，不是阻塞线程，而是将线程转入就绪状态
- 如果调用了yield方法之后，没有其他等待执行的线程，这个时候当前线程就会马上恢复执行！

## 三、线程合并——join ()

### 1、作用：

阻塞指定线程等到另一个线程完成以后再继续执行

join是Thread类的一个方法，启动线程后直接调用，即join()的作用是：“等待该线程终止”，这里需要理解的就是该线程是指的主线程等待子线程的终止。也就是在子线程调用了join()方法后面的代码，只有等到子线程结束了才能执行。

### 2、使用场景

在很多情况下，主线程生成并起动了子线程，如果子线程里要进行大量的耗时的运算，主线程往往将子线程之前结束，但是如果主线程处理完其他的事务后，需要用到子线程的处理结果，也就是主线程需要等待子线程执行完成之后再结束，这个时候就要用到join()方法了。

## 四、设置线程的优先级

## 1、函数

- `setPriority(int newPriority)`
- `getPriority()`

## 2、线程的优先级用数字表示，范围从1到10

- `Thread.MIN_PRIORITY = 1`
- `Thread.MAX_PRIORITY = 10`
- `Thread.NORM_PRIORITY = 5`

## 3、注意：

优先级低只是意味着获得调度的概率低。并不是绝对先调用优先级高后调用优先级低的线程

# 五、后台线程 `setDaemon()`

## 1、基本概念

- 可以将指定的线程设置成后台线程
- 创建后台线程的线程结束时，后台线程也随之消亡
- 只能在线程启动之前把它设为后台线程

## 2、基本语法

`setDaemon(true)` 将其设置为守护线程

## 3、举例

JVM的垃圾回收、内存管理等线程都是守护线程。还有就是在做数据库应用时候，使用的数据库连接池，连接池本身也包含着很多后台线程，监控连接个数、超时时间、状态等 等。