

## python\_多线程join和setDaemon

1、join ()方法：主线程A中，创建了子线程B，并且在主线程A中调用了B.join()，那么，主线程A会在调用的地方等待，直到子线程B完成操作后，才可以接着往下执行，那么在调用这个线程时可以使用被调用线程的join方法。

```
import time,threading
```

```
def lajifenlei():
```

```
    time.sleep(2)
```

```
    print("haha jjj")
```

```
for i in range(10):
```

```
    th=threading.Thread(target=lajifenlei) #声明子线程
```

```
    th.start() # start在join前
```

```
    th.join() #主线程等待子线程
```

```
print("main theading") #主线程
```

```
C:\Users\zhaow\AppData\Local\Programs\Python\Python37\python.exe D:/study/python/atp/lib/t.py
```

```
haha jjj
```

```
haha jjj
```

```
haha jjj
```

```
haha jjj
```

```
haha jjj
```

```
haha jjj
```

```
haha jjj
```

```
haha jjj
```

```
haha jjj
```

```
haha jjj
```

```
main theading #等待所有的子线程运行完才会运行主进程
```

2、setDaemon()方法。主线程A中，创建了子线程B，并且在主线程A中调用了B.setDaemon(),这个的意思是，把主线程A设置为守护线程，这时候，要是主线程A执行结束了，就不管子线程B是否完成,一并和主线程A退出.这就是setDaemon方法的含义，这基本和join是相反的。此外，还有个要特别注意的：必须在start() 方法调用之前设置

```
import time,threading
```

```
def lajifenlei():
```

```
    time.sleep(2)
```

```
    print("haha jjj")
```

```
for i in range(10):
```

```
    th=threading.Thread(target=lajifenlei) #声明子线程
```

```
    th.setDaemon(True) #子线程设置为守护线程
```

```
    th.start() #start在守护线程后面
```

```
print("main theading") #主线程
```

C:\Users\zhaow\AppData\Local\Programs\Python\Python37\python.exe D:/study/python/atp/lib/t.py

```
main theading #主线程未等子线程运行，直接结束了
```