

python\_100个数多线程打印

问题：1-100个数多线程打印

```
import threadpool
```

```
def pool_num(num,p_methond,num_list):
```

```
    pool=threadpool.ThreadPool(num) #声明线程池个数
```

```
    reqs=threadpool.makeRequests(p_methond,num_list) #生成线程池启动参数
```

```
    [pool.putRequest(req) for req in reqs] #循环执行启动线程
```

```
    pool.wait() #等待子线程
```

```
def p_methond(num):
```

```
    print(num)
```

```
num_list=[i for i in range(1,101)]
```

```
pool_num(3,p_methond,num_list)
```

问题：1-100个数，线程数可以自定义，然后多线程打印这1-100个数，要求每个线程打印的数分段连续打印，比如说用2个线程，那么线程1是打印1-50，线程2是打印51-100

线程方式实现

```
import threading
```

```
def func(arg): #打印数字函数
```

```
    for i in arg:
```

```
        print("当前线程: ", threading.currentThread().name, "----", i+1)
```

```
def thread_num(total,num): #传参是打印数字的总数及线程数
```

```
    data = [x for x in range(0, total)] #所有的数字循环放入list
```

```
    split_data = [data[i: i + int(total/num)] for i in range(0, len(data), int(total/num))] #带步长的循环list, 且每段放入一个list, 生成2维数组
```

```
    for d in split_data: #循环二维数组, 每次取一个数组, 作为打印函数的传参
```

```
        t = threading.Thread(target=func, args=(d,)) #生成线程且调用方法及给予传参
```

```
        t = t.start() #启动线程
```

```
    while threading.active_count() != 1: #等待子线程
```

```
        pass
```

```
thread_num(13,2) #调用线程方法
```

numpy模块分段:

```
import threading

import numpy

def func(arg):

    for i in arg:

        print(threading.currentThread().name, "----", i+1)

def thread_num(total,num):

    result=numpy.array_split(range(total),num) #分段

    for i in result:

        t=threading.Thread(target=func,args=(i,)) #声明线程

        t.start() #启动线程

    while threading.active_count()!=1: #等待子线程

        pass
```

随记：问题--为什么等子线程

开了几个线程执行任务，如果不等待线程，如果你的进程挂了的时候，线程也会挂掉

还有一个场景，如果你要等待所有线程处理完后的结果，那就必须等待了