### Threading

### 创建Thread对象：

**import threading**

**threading.Thread(target=None,name=None,args=())**

或

**th1=threading.Thread(ta rget=Nona,args=(args1,args2))**

import threading

import time

def one():

for i in range(3):

print('第一个线程%d'%i)

time.sleep(1)

def two():

for i in range(4):

print('第二个线程%d'%i)

time.sleep(1)

if \_\_name\_\_=='\_\_main\_\_':

th1=threading.Thread(target=one)

th2=threading.Thread(target=two)

th1.start()

#th2.start()

print(th1.isAlive())

th1.join()

print(th1.isAlive())

th1.run()

print('线程1终止')

参数说明：

target 是函数名字，需要调用的函数。

name 设置线程名字。

args 函数需要的参数，以元祖( tuple)的形式传入

### threading模块的方法：

Threading.enumerate() 返回当前的线程都有哪些

Threading.current\_thread() 返回当前的线程对象名，放在线程执行的函数里面

### Thread的对象主要方法说明：

run(): 用以表示线程活动的方法。

start(): 启动线程活动。

join(): 等待至线程中止。Join后的代码在线程终止后执行

isAlive(): 返回线程是否活动的。

getName(): 返回线程名。

setName(): 设置线程名。

### 读写锁

lock=Threading.lock()

def run():

lock.acquire()

balance+=1

balance-=1

lock.release()

1. 用新建类的方法创建一个线程

class mythread(threading.Thread):

def \_\_init\_\_(self,n):

super().\_\_init\_\_()#需要使用super父类的方法

self.n=n

def run(self):

for i in range(3):

print('第%d个线程'%self.n)

time.sleep(1)

mt1=mythread(1)

mt2=mythread(2)

mt1.start()

mt2.start()