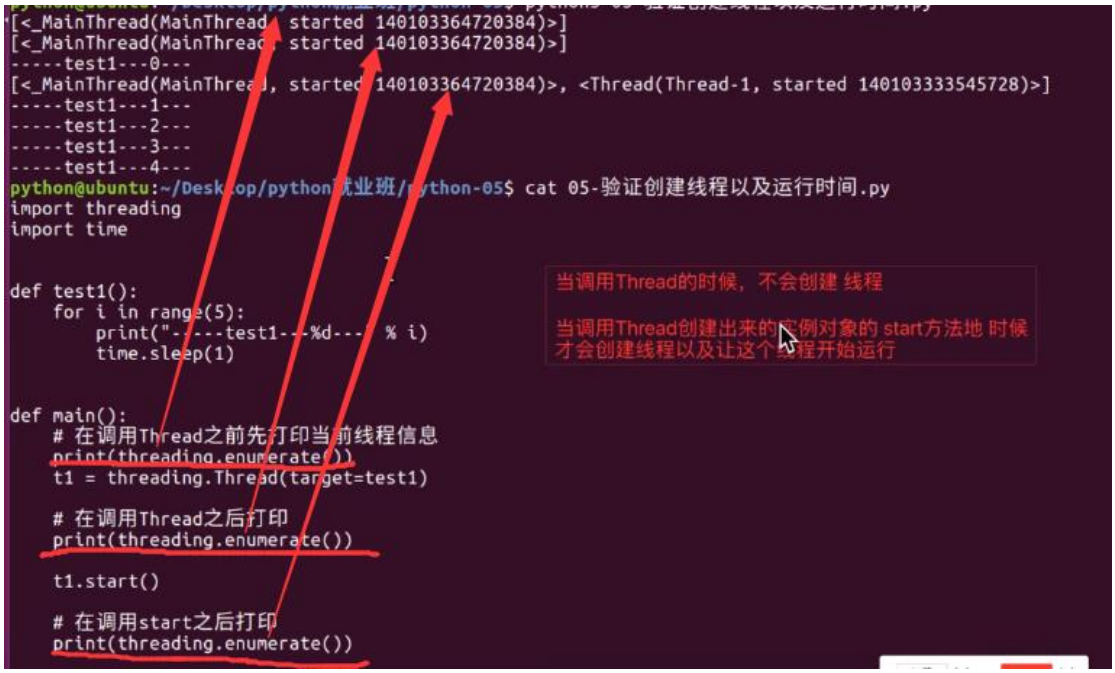
Threading

查看有多少线程

len(threading.enumerte(1))

threading 什么时候创建



Threading 什么时候结束

线程调用的函数执行完成之后线程结束

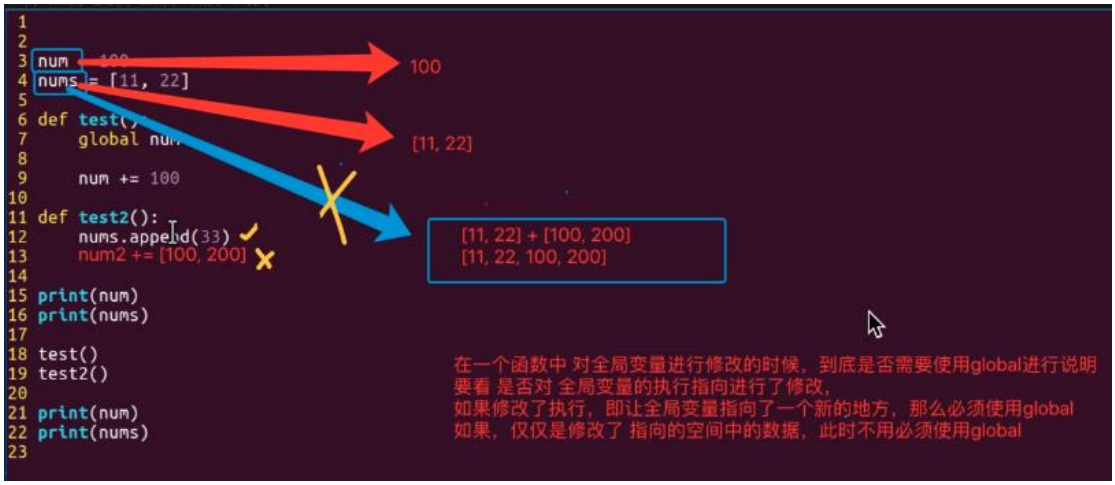
使用类作为多线程的执行函数

```python
#coding=utf-8
import threading
import time

class MyThread(threading.Thread):
    def run(self):
        for i in range(3):
            time.sleep(1)
            msg = "I'm "+self.name+' @ '+str(i) #name属性中保存的是当前线程的名字
            print(msg)


if __name__ == '__main__':
    t = MyThread()
    t.start()
```

Threading 中传参数

注意 args 中必须是元组

Global



锁

```
# 创建锁
mutex = threading.Lock()


# 锁定
mutex.acquire()


# 释放
mutex.release()
```