# [yeayee](https://www.cnblogs.com/yeayee/)

**------>更多技巧------>更多源码------>www.yeayee.com**

## **[Python 一篇学会多线程](https://www.cnblogs.com/yeayee/p/4952022.html)**

多线程和多进程是什么自行google补脑，廖雪峰官网也有，但是不够简洁，有点晕，所以就整个简单的范例。

　　对于python 多线程的理解，我花了很长时间，搜索的大部份文章都不够通俗易懂。所以，这里力图用简单的例子，让你对多线程有个初步的认识。

**单线程**

　　在好些年前的MS-DOS时代，操作系统处理问题都是单任务的，我想做听音乐和看电影两件事儿，那么一定要先排一下顺序。

（好吧！我们不纠结在DOS时代是否有听音乐和看影的应用。^\_^）

[IMG_256](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_257

from time import ctime,sleep

def music():

for i in range(2):

print "I was listening to music. %s" %ctime()

sleep(1)

def move():

for i in range(2):

print "I was at the movies! %s" %ctime()

sleep(5)

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

music()

move()

print "all over %s" %ctime()

IMG_258

[IMG_259](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

 　　我们先听了一首音乐，通过for循环来控制音乐的播放了两次，每首音乐播放需要1秒钟，sleep()来控制音乐播放的时长。接着我们又看了一场电影，

每一场电影需要5秒钟，因为太好看了，所以我也通过for循环看两遍。在整个休闲娱乐活动结束后，我通过

print "all over %s" %ctime()

看了一下当前时间，差不多该睡觉了。

运行结果：

[IMG_260](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_261

>>=========================== RESTART ================================

>>>

I was listening to music. Thu Apr 17 10:47:08 2014

I was listening to music. Thu Apr 17 10:47:09 2014

I was at the movies! Thu Apr 17 10:47:10 2014

I was at the movies! Thu Apr 17 10:47:15 2014

all over Thu Apr 17 10:47:20 2014

IMG_262

[IMG_263](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

　　其实，music()和move()更应该被看作是音乐和视频播放器，至于要播放什么歌曲和视频应该由我们使用时决定。所以，我们对上面代码做了改造：

[IMG_264](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_265

#coding=utf-8import threadingfrom time import ctime,sleep

def music(func):

for i in range(2):

print "I was listening to %s. %s" %(func,ctime())

sleep(1)

def move(func):

for i in range(2):

print "I was at the %s! %s" %(func,ctime())

sleep(5)

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

music(u'爱情买卖')

move(u'阿凡达')

print "all over %s" %ctime()

IMG_266

[IMG_267](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

　　对music()和move()进行了传参处理。体验中国经典歌曲和欧美大片文化。

运行结果：

[IMG_268](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_269

>>> ======================== RESTART ================================

>>>

I was listening to 爱情买卖. Thu Apr 17 11:48:59 2014

I was listening to 爱情买卖. Thu Apr 17 11:49:00 2014

I was at the 阿凡达! Thu Apr 17 11:49:01 2014

I was at the 阿凡达! Thu Apr 17 11:49:06 2014

all over Thu Apr 17 11:49:11 2014

IMG_270

[IMG_271](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

**多线程**

　　科技在发展，时代在进步，我们的CPU也越来越快，CPU抱怨，P大点事儿占了我一定的时间，其实我同时干多个活都没问题的；于是，操作系统就进入了多任务时代。我们听着音乐吃着火锅的不在是梦想。

　　python提供了两个模块来实现多线程thread 和threading ，thread 有一些缺点，在threading 得到了弥补，为了不浪费你和时间，所以我们直接学习threading 就可以了。

继续对上面的例子进行改造，引入threadring来同时播放音乐和视频：

[IMG_272](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_273

#coding=utf-8import threadingfrom time import ctime,sleep

def music(func):

for i in range(2):

print "I was listening to %s. %s" %(func,ctime())

sleep(1)

def move(func):

for i in range(2):

print "I was at the %s! %s" %(func,ctime())

sleep(5)

threads = []

t1 = threading.Thread(target=music,args=(u'爱情买卖',))

threads.append(t1)

t2 = threading.Thread(target=move,args=(u'阿凡达',))

threads.append(t2)

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

for t in threads:

t.setDaemon(True)

t.start()

print "all over %s" %ctime()

IMG_274

[IMG_275](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

import threading

首先导入threading 模块，这是使用多线程的前提。

threads = []

t1 = threading.Thread(target=music,args=(u'爱情买卖',))

threads.append(t1)

　　创建了threads数组，创建线程t1,使用threading.Thread()方法，在这个方法中调用music方法target=music，args方法对music进行传参。 把创建好的线程t1装到threads数组中。

　　接着以同样的方式创建线程t2，并把t2也装到threads数组。

for t in threads:

　　t.setDaemon(True)

　　t.start()

最后通过for循环遍历数组。（数组被装载了t1和t2两个线程）

setDaemon()

　　setDaemon(True)将线程声明为守护线程，必须在start() 方法调用之前设置，如果不设置为守护线程程序会被无限挂起。子线程启动后，父线程也继续执行下去，当父线程执行完最后一条语句print "all over %s" %ctime()后，没有等待子线程，直接就退出了，同时子线程也一同结束。

start()

开始线程活动。

运行结果：

[IMG_276](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

>>> ========================= RESTART ================================

>>>

I was listening to 爱情买卖. Thu Apr 17 12:51:45 2014 I was at the 阿凡达! Thu Apr 17 12:51:45 2014 all over Thu Apr 17 12:51:45 2014

[IMG_277](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

　　从执行结果来看，子线程（muisc 、move ）和主线程（print "all over %s" %ctime()）都是同一时间启动，但由于主线程执行完结束，所以导致子线程也终止。

继续调整程序：

[IMG_278](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_279

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

for t in threads:

t.setDaemon(True)

t.start()

t.join()

print "all over %s" %ctime()

IMG_280

[IMG_281](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

　　我们只对上面的程序加了个join()方法，用于等待线程终止。join（）的作用是，在子线程完成运行之前，这个子线程的父线程将一直被阻塞。

　　注意:  join()方法的位置是在for循环外的，也就是说必须等待for循环里的两个进程都结束后，才去执行主进程。

运行结果：

[IMG_282](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_283

>>> ========================= RESTART ================================

>>>

I was listening to 爱情买卖. Thu Apr 17 13:04:11 2014 I was at the 阿凡达! Thu Apr 17 13:04:11 2014

I was listening to 爱情买卖. Thu Apr 17 13:04:12 2014

I was at the 阿凡达! Thu Apr 17 13:04:16 2014

all over Thu Apr 17 13:04:21 2014

IMG_284

[IMG_285](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

　　从执行结果可看到，music 和move 是同时启动的。

　　开始时间4分11秒，直到调用主进程为4分22秒，总耗时为10秒。从单线程时减少了2秒，我们可以把music的sleep()的时间调整为4秒。

[IMG_286](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_287

...def music(func):

for i in range(2):

print "I was listening to %s. %s" %(func,ctime())

sleep(4)

...

IMG_288

[IMG_289](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

执行结果：

[IMG_290](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_291

>>> ====================== RESTART ================================

>>>

I was listening to 爱情买卖. Thu Apr 17 13:11:27 2014I was at the 阿凡达! Thu Apr 17 13:11:27 2014

I was listening to 爱情买卖. Thu Apr 17 13:11:31 2014

I was at the 阿凡达! Thu Apr 17 13:11:32 2014

all over Thu Apr 17 13:11:37 2014

IMG_292

[IMG_293](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

　　子线程启动11分27秒，主线程运行11分37秒。

　　虽然music每首歌曲从1秒延长到了4 ，但通多程线的方式运行脚本，总的时间没变化。

本文从感性上让你快速理解python多线程的使用，更详细的使用请参考其它文档或资料。

 ==========================================================

class threading.Thread()说明：

class threading.Thread(group=None, target=None, name=None, args=(), kwargs={})

This constructor should always be called with keyword arguments. Arguments are:

　　group should be None; reserved for future extension when a ThreadGroup class is implemented.

　　target is the callable object to be invoked by the run() method. Defaults to None, meaning nothing is called.

　　name is the thread name. By default, a unique name is constructed of the form “Thread-N” where N is a small decimal number.

　　args is the argument tuple for the target invocation. Defaults to ().

　　kwargs is a dictionary of keyword arguments for the target invocation. Defaults to {}.

If the subclass overrides the constructor, it must make sure to invoke the base class constructor (Thread.\_\_init\_\_()) before doing

anything else to the thread.

        是不是感觉感觉讲的意犹未尽，其实，多线程非常有意思。因为我们在使用电脑的过程中无时无刻都在多进程和多线程。我们可以接着之前的例子继续讲：

　　从上面例子中发现线程的创建是颇为麻烦的，每创建一个线程都需要创建一个tx（t1、t2、...），如果创建的线程多时候这样极其不方便。下面对通过例子进行继续改进：

player.py

[IMG_294](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_295

#coding=utf-8from time import sleep, ctime import threading

def muisc(func):

for i in range(2):

print 'Start playing： %s! %s' %(func,ctime())

sleep(2)

def move(func):

for i in range(2):

print 'Start playing： %s! %s' %(func,ctime())

sleep(5)

def player(name):

r = name.split('.')[1]

if r == 'mp3':

muisc(name)

else:

if r == 'mp4':

move(name)

else:

print 'error: The format is not recognized!'

list = ['爱情买卖.mp3','阿凡达.mp4']

threads = []

files = range(len(list))

#创建线程for i in files:

t = threading.Thread(target=player,args=(list[i],))

threads.append(t)

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

#启动线程

for i in files:

threads[i].start()

　for i in files:

　　 threads[i].join()

#主线程

print 'end:%s' %ctime()

IMG_296

[IMG_297](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

　　有趣的是我们又创建了一个player()函数，这个函数用于判断播放文件的类型。如果是mp3格式的，我们将调用music()函数，如果是mp4格式的我们调用move()函数。哪果两种格式都不是那么只能告诉用户你所提供有文件我播放不了。

　　然后，我们创建了一个list的文件列表，注意为文件加上后缀名。然后我们用len(list) 来计算list列表有多少个文件，这是为了帮助我们确定循环次数。

　　接着我们通过一个for循环，把list中的文件添加到线程中数组threads[]中。接着启动threads[]线程组，最后打印结束时间。

     split()可以将一个字符串拆分成两部分，然后取其中的一部分。

[IMG_298](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_299

>>> x = 'testing.py'

>>> s = x.split('.')[1]

>>> if s=='py':

print s

py

IMG_300

[IMG_301](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

运行结果：

Start playing： 爱情买卖.mp3! Mon Apr 21 12:48:40 2014

Start playing： 阿凡达.mp4! Mon Apr 21 12:48:40 2014

Start playing： 爱情买卖.mp3! Mon Apr 21 12:48:42 2014

Start playing： 阿凡达.mp4! Mon Apr 21 12:48:45 2014

end:Mon Apr 21 12:48:50 2014

现在向list数组中添加一个文件，程序运行时会自动为其创建一个线程。

**继续改进例子：**

　　通过上面的程序，我们发现player()用于判断文件扩展名，然后调用music()和move() ，其实，music()和move()完整工作是相同的，我们为什么不做一台超级播放器呢，不管什么文件都可以播放。经过改造，我的超级播放器诞生了。

super\_player.py

[IMG_302](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_303

#coding=utf-8from time import sleep, ctime import threading

def super\_player(file,time):

for i in range(2):

print 'Start playing： %s! %s' %(file,ctime())

sleep(time)

#播放的文件与播放时长

list = {'爱情买卖.mp3':3,'阿凡达.mp4':5,'我和你.mp3':4}

threads = []

files = range(len(list))

#创建线程for file,time in list.items():

t = threading.Thread(target=super\_player,args=(file,time))

threads.append(t)

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

#启动线程

for i in files:

threads[i].start()

　　for i in files:

　　 threads[i].join()

#主线程

print 'end:%s' %ctime()

IMG_304

[IMG_305](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

　　首先创建字典list ，用于定义要播放的文件及时长（秒），通过字典的items()方法来循环的取file和time，取到的这两个值用于创建线程。

　　接着创建super\_player()函数，用于接收file和time，用于确定要播放的文件及时长。

　　最后是线程启动运行。运行结果：

[IMG_306](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_307

Start playing： 爱情买卖.mp3! Fri Apr 25 09:45:09 2014

Start playing： 我和你.mp3! Fri Apr 25 09:45:09 2014

Start playing： 阿凡达.mp4! Fri Apr 25 09:45:09 2014

Start playing： 爱情买卖.mp3! Fri Apr 25 09:45:12 2014

Start playing： 我和你.mp3! Fri Apr 25 09:45:13 2014

Start playing： 阿凡达.mp4! Fri Apr 25 09:45:14 2014

end:Fri Apr 25 09:45:19 2014

IMG_308

[IMG_309](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

**创建自己的多线程类**

[IMG_310](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_311

#coding=utf-8import threading from time import sleep, ctime

class MyThread(threading.Thread):

def \_\_init\_\_(self,func,args,name=''):

threading.Thread.\_\_init\_\_(self)

self.name=name

self.func=func

self.args=args

def run(self):

apply(self.func,self.args)

def super\_play(file,time):

for i in range(2):

print 'Start playing： %s! %s' %(file,ctime())

sleep(time)

list = {'爱情买卖.mp3':3,'阿凡达.mp4':5}

#创建线程

threads = []

files = range(len(list))

for k,v in list.items():

t = MyThread(super\_play,(k,v),super\_play.\_\_name\_\_)

threads.append(t)

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

#启动线程

for i in files:

threads[i].start()

　　for i in files:

　　 threads[i].join()

#主线程

print 'end:%s' %ctime()

IMG_312

[IMG_313](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

MyThread(threading.Thread)

创建MyThread类，用于继承threading.Thread类。

\_\_init\_\_()

使用类的初始化方法对func、args、name等参数进行初始化。

apply()

　　apply(func [, args [, kwargs ]]) 函数用于当函数参数已经存在于一个元组或字典中时，间接地调用函数。args是一个包含将要提供给函数的按位置传递的参数的元组。如果省略了args，任何参数都不会被传递，kwargs是一个包含关键字参数的字典。

apply() 用法：

[IMG_314](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

IMG_315

#不带参数的方法

>>> def say():

print 'say in'

>>> apply(say)

say in

#函数只带元组的参数

>>> def say(a,b):

print a,b

>>> apply(say,('hello','虫师'))

hello 虫师

#函数带关键字参数

>>> def say(a=1,b=2):

print a,b

>>> def haha(\*\*kw):

apply(say,(),kw)

>>> haha(a='a',b='b')

a b

IMG_316

[IMG_317](https://www.cnblogs.com/yeayee/p/javascript:void(0);)

MyThread(super\_play,(k,v),super\_play.\_\_name\_\_)

由于MyThread类继承threading.Thread类，所以，我们可以使用MyThread类来创建线程。

运行结果：

Start playing： 爱情买卖.mp3! Fri Apr 25 10:36:19 2014

Start playing： 阿凡达.mp4! Fri Apr 25 10:36:19 2014

Start playing： 爱情买卖.mp3! Fri Apr 25 10:36:22 2014

Start playing： 阿凡达.mp4! Fri Apr 25 10:36:24 2014

all end: Fri Apr 25 10:36:29 2014

Python Life