什么时候用线程，什么时候用线程池？

线程池：创建并保存了一定数量的线程的容器，可以随机从该容器中获取可用的线程进行操作，和每次去创建一个新的线程相比，节约了系统资源

------------------------------------------------------------------------------------

使用差别：如果每次执行的操作类似，如都是通过url从网络去加载图片，则可以使用线程池，如果每次执行的操作差异很大，则每次创建一个新的线程，在新的线程里面实现相应的操作

------------------------------------------------------------------------------------

线程和线程池使用方法对比：

一、线程：

1.继承Thread类，并重写Thread类的run方法，通过start()方法开启线程

线程： new Thread(){

run(){

//toDo 实现目的操作

}

}.start();

2.实现Runnable接口

Class MyThread implements Runnable{

run(){

//toDo重写run方法

}

}

new MyThread().start();

二、线程池

1.创建线程池：

/\*\*

\* 创建的线程池对象，设置参数为指定该线程池容纳的线程数 \*/

**private** ExecutorService newFixedThreadPool;

newFixedThreadPool = Executors.*newFixedThreadPool*(5);

2.使用线程池：

通过线程池的execute()方法去获取一个该线程池中已经创建好的线程，并去完成相应的操作：

GetInternetPictureTask是一个继承了Thread线程类的类，并重写了run方法

newFixedThreadPool.execute(**new** GetInternetPictureTask(iv, url));