

Specifying the Code to Run on a Thread

This lesson shows you how to implement a [Runnable](/reference/java/lang/Runnable.html) (</reference/java/lang/Runnable.html>) class, which runs the code in its `Runnable.run()` ([/reference/java/lang/Runnable.html#run\(\)](/reference/java/lang/Runnable.html#run())) method on a separate thread. You can also pass a [Runnable](/reference/java/lang/Runnable.html) (</reference/java/lang/Runnable.html>) to another object that can then attach it to a thread and run it. One or more [Runnable](/reference/java/lang/Runnable.html) (</reference/java/lang/Runnable.html>) objects that perform a particular operation are sometimes called a *task*.

[Thread](/reference/java/lang/Thread.html) (</reference/java/lang/Thread.html>) and [Runnable](/reference/java/lang/Runnable.html) (</reference/java/lang/Runnable.html>) are basic classes that, on their own, have only limited power. Instead, they're the basis of powerful Android classes such as [HandlerThread](/reference/android/os/HandlerThread.html) (</reference/android/os/HandlerThread.html>), [AsyncTask](/reference/android/os/AsyncTask.html) (</reference/android/os/AsyncTask.html>), and [IntentService](/reference/android/app/IntentService.html) (</reference/android/app/IntentService.html>). [Thread](/reference/java/lang/Thread.html) (</reference/java/lang/Thread.html>) and [Runnable](/reference/java/lang/Runnable.html) (</reference/java/lang/Runnable.html>) are also the basis of the class [ThreadPoolExecutor](/reference/java/util/concurrent/ThreadPoolExecutor.html) (</reference/java/util/concurrent/ThreadPoolExecutor.html>). This class automatically manages threads and task queues, and can even run multiple threads in parallel.

THIS LESSON TEACHES YOU TO

1. [Define a Class that Implements Runnable](#)
2. [Implement the run\(\) Method](#)

YOU SHOULD ALSO READ

- [Processes and Threads](#)

TRY IT OUT

[Download the sample](#)

ThreadSample.zip

Define a Class that Implements Runnable

Implementing a class that implements [Runnable](/reference/java/lang/Runnable.html) (</reference/java/lang/Runnable.html>) is straightforward. For example:

```
public class PhotoDecodeRunnable implements Runnable {  
    ...  
    @Override  
    public void run() {  
        /*  
         * Code you want to run on the thread goes here  
         */  
        ...  
    }  
    ...  
}
```

Implement the run() Method

In the class, the [Runnable.run\(\)](/reference/java/lang/Runnable.html#run()) ([/reference/java/lang/Runnable.html#run\(\)](/reference/java/lang/Runnable.html#run())) method contains the code that's executed. Usually, anything is allowable in a [Runnable](/reference/java/lang/Runnable.html) (</reference/java/lang/Runnable.html>). Remember, though, that the [Runnable](/reference/java/lang/Runnable.html) (</reference/java/lang/Runnable.html>) won't be running on the UI thread, so it can't directly modify UI objects such as [View](/reference/android/view/View.html) (</reference/android/view/View.html>) objects. To

communicate with the UI thread, you have to use the techniques described in the lesson [Communicate with the UI Thread](#) ([communicate-ui.html](#)).

At the beginning of the `run()` ([/reference/java/lang/Runnable.html#run\(\)](#)) method, set the thread to use background priority by calling `Process.setThreadPriority()` ([/reference/android/os/Process.html#setThreadPriority\(int\)](#)) with `THREAD_PRIORITY_BACKGROUND` ([/reference/android/os/Process.html#THREAD_PRIORITY_BACKGROUND](#)). This approach reduces resource competition between the `Runnable` ([/reference/java/lang/Runnable.html](#)) object's thread and the UI thread.

You should also store a reference to the `Runnable` ([/reference/java/lang/Runnable.html](#)) object's `Thread` ([/reference/java/lang/Thread.html](#)) in the `Runnable` ([/reference/java/lang/Runnable.html](#)) itself, by calling `Thread.currentThread()` ([/reference/java/lang/Thread.html#currentThread\(\)](#)).

The following snippet shows how to set up the `run()` ([/reference/java/lang/Runnable.html#run\(\)](#)) method:

```
class PhotoDecodeRunnable implements Runnable {
    ...
    /*
     * Defines the code to run for this task.
     */
    @Override
    public void run() {
        // Moves the current Thread into the background
        android.os.Process.setThreadPriority(android.os.Process.THREAD_PRIORI
        ...
        /*
         * Stores the current Thread in the PhotoTask instance,
         * so that the instance
         * can interrupt the Thread.
         */
        mPhotoTask.setImageDecodeThread(Thread.currentThread());
        ...
    }
    ...
}
```