

## 使用 Handler 发送和处理消息

### 1.1 知识解析

可以通过 `Handler` 的 `sendMessage()` 方法，将消息发送给对应的消息队列。对应的 `Looper` 在将这个 `Message` 从 `Message Queue` 取出后，再交还给 `Handler` 处理，此时会使用 `handleMessage()` 去处理这个消息。因为在一般情况下，发送的消息可能需要进行比较耗时的操作（例如：从网上下载），所以，可以在一个单独的线程中进行耗时操作并且在适当的时机将消息发送出去，

而在 `handleMessage()` 中，接收到消息后，即可对 `UI` 进行更新等操作。因为要更新 `UI`，所以这个方法需要在 `UI` 线程中被调用，而不能在其他线程中。

### 1.2 功能演示



## 1.3 实战操作

例如：

```
new Thread() {  
  
    public void run() {  
  
        while (i <= 100) {  
            Message msg = Message.obtain();  
            msg.what = i++;  
            handler.sendMessage(msg);  
            try {  
                Thread.sleep(200);  
            } catch (InterruptedException e) {  
                // TODO Auto-generated catch block  
                e.printStackTrace();  
            }  
        }  
    }  
  
}.start();
```

## 1.4 职业素质

在 Handler 异步实现时,涉及到 Handler, Looper, Message, Thread 四个对象,实现异步的流程是主线程启动 Thread (子线程) 运行并生成 Message-Looper 获取 Message 并传递给 HandlerHandler 逐个获取 Looper 中的 Message, 并进行 UI 变更。

使用的优点:

结构清晰, 功能定义明确

对于多个后台任务时, 简单, 清晰