Handler使用步骤：

1.Looper.prepare(); 先调用，并且只执行一次

Looper.prepare()方法里面执行的操作：

1.在Looper的prepare方法中设置一个 Looper对象：sThreadLocal.set(new Looper());

2.Looper的构造方法里面:创建一个消息队列MessageQueue对象， mQueue = new MessageQueue();

2.创建Handler、使用Handler

在Handler的构造方法中，干了两件事

1.获取到Looper对象

2.获取在Looper中创建好的MessageQueue对象

3.handler发送消息

1. 将Message发送到MessageQueue里面

2. 在MessageQueue里面所有的Message都是按照时间顺序从小到大排列

4.Looper.loop();

1.依次从MessageQueue中取出Message

2. 并执行msg.target.dispatchMessage(msg)方法，msg.target()就是当前的handler对象

①在子线程中无法直接使用Handler对象,要想创建Handler对象必须要先调用Looper.prepare()方法,而在外面(非子线程中)可以直接创建Handler的原因是,系统会在创建Handler对象之前默认调用Looper.prepare()方法，所以如果想要在子线程中创建Handler对象，则需要显示调用Looper.prepare()方法

②在子线程中创建好了Handler对象之后，要想在调用了Handler.sendMessage()方法之后执行Handler的handlerMessage（）方法，则需要在handler.sendMessage()方法执行之后调用Looper.loop()方法对消息队列进行处理，即取出消息队列MessageQueue中的Message对象

如下代码：

//创建一个子线程，在子线程中创建一个Handler对象

**new** Thread(){

**public** **void** run() {

/\*\*

\* Looper.prepare()方法里面执行的操作：

\*

\* 1.在Looper的prepare方法中设置一个 Looper对象：sThreadLocal.set(new Looper());

\*

\* 2.Looper的构造方法里面:创建一个消息队列MessageQueue对象， mQueue = new MessageQueue();

\*

\*/

Looper.*prepare*();

/\*\*

\* 在Handler的构造方法中，干了两件事

\* 1.获取到Looper对象

\* 2.获取在Looper中创建好的MessageQueue对象

\*

\*/

Handler handler = **new** Handler(){

**public** **void** handleMessage(android.os.Message msg) {

Toast.*makeText*(getApplicationContext(), "Message is handleing", 0).show();

};

};

/\*\*

\* 将Message发送到MessageQueue里面

\* 在MessageQueue里面所有的Message都是按照时间顺序从小到大排列

\*

\*/

handler.sendEmptyMessage(0);

/\*\*

\* 依次从MessageQueue中取出Message

\*

\* 并执行msg.target.dispatchMessage(msg)方法，msg.target()就是当前的handler对象

\*

\*/

Looper.*loop*();

};

}.start();

/\*\*

\* 在子线程外面创建Handler对象，系统会默认调用Looper.prepare()方法，无需再去显示调用Looper.prepare()方法

\*/

Handler handler2 = **new** Handler(){

**public** **void** handleMessage(android.os.Message msg) {

};

};