

搜索.....

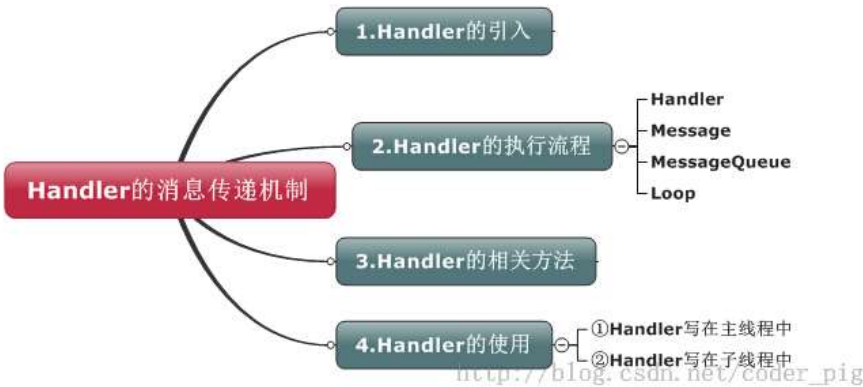
# 3.3 Handler消息传递机制浅析

分类 **Android 基础入门教程**

## 本节引言

前两节中我们对Android中的两种事件处理机制进行了学习，关于响应的事件响应就这两种；本节给大家讲解的是Activity中UI组件中的信息传递Handler，相信很多朋友都知道，Android为了线程安全，并不允许我们在UI线程外操作UI；很多时候我们做界面刷新都需要通过Handler来通知UI组件更新！除了用Handler完成界面更新外，还可以使用runOnUiThread()来更新，甚至更高级的事务总线，当然，这里我们只讲解Handler，什么是Handler，执行流程，相关方法，子线程与主线程中使用Handler的区别等！

## 1.学习路线图：



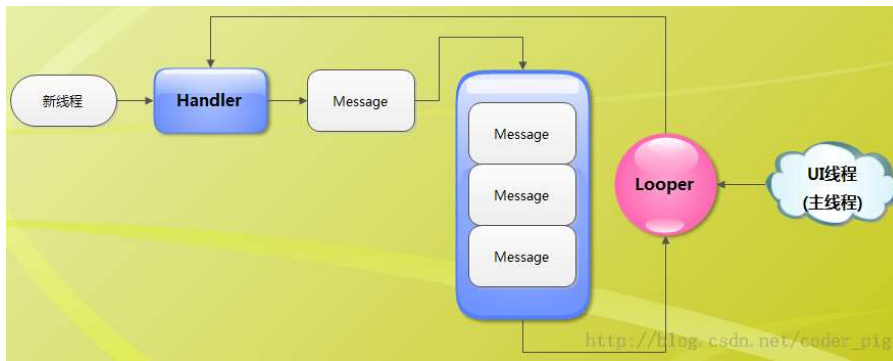
## 2.Handler类的引入：



## 3.Handler的执行流程图：

**Android 基础入门教程(Q群号：153836263)**

- 1.0 Android基础入门教程
  - 1.0.1 2015年最新Android基...
- 1.1 背景相关与系统架构分析
- 1.2 开发环境搭建
  - 1.2.1 使用Eclipse + ADT + S...
  - 1.2.2 使用Android Studio开...
- 1.3 SDK更新不了问题解决
- 1.4 Genymotion模拟器安装
- 1.5.1 Git使用教程之本地仓...
- 1.5.2 Git之使用GitHub搭建...
- 1.6 .9(九妹)图片怎么玩
- 1.7 界面原型设计
- 1.8 工程相关解析(各种文件...
- 1.9 Android程序签名打包
- 1.11 反编译APK获取代码&...
- 2.1 View与ViewGroup的概念
  - 2.2.1 LinearLayout(线性布局)
  - 2.2.2 RelativeLayout(相对布...
  - 2.2.3 TableLayout(表格布局)
  - 2.2.4 FrameLayout(帧布局)
  - 2.2.5 GridLayout(网格布局)
  - 2.2.6 AbsoluteLayout(绝对...
- 2.3.1 TextView(文本框)详解
- 2.3.2 EditText(输入框)详解
- 2.3.3 Button(按钮)与ImageB...
- 2.3.4 ImageView(图像视图)
- 2.3.5.RadioButton(单选按钮...
- 2.3.6 开关按钮ToggleButton...
- 2.3.7 ProgressBar(进度条)
- 2.3.8 SeekBar(拖动条)
- 2.3.9 RatingBar(星级评分条)
- 2.4.1 ScrollView(滚动条)



### 流程图解析: 相关名词

**UI线程:**就是我们的主线程,系统在创建UI线程的时候会初始化一个Looper对象,同时也会创建一个与其关联的MessageQueue;

**Handler:**作用就是发送与处理信息,如果希望Handler正常工作,在当前线程中要有一个Looper对象

**Message:**Handler接收与处理的消息对象

**MessageQueue:**消息队列,先进先出管理Message,在初始化Looper对象时会创建一个与之关联的MessageQueue;

**Looper:**每个线程只能有一个Looper,管理MessageQueue,不断地从中取出Message分发给对应的Handler处理!

### 简单点说:

当我们的子线程想修改Activity中的UI组件时,我们可以新建一个Handler对象,通过这个对象向主线程发送信息;而我们发送的信息会先到主线程的MessageQueue进行等待,由Looper按先入先出顺序取出,再根据message对象的what属性分发给对应的Handler进行处理!

## 4.Handler的相关方法:

`void handleMessage(Message msg):`处理消息的方法,通常是用于被重写!

`sendEmptyMessage(int what):`发送空消息

`sendEmptyMessageDelayed(int what,long delayMillis):`指定延时多少毫秒后发送空消息

`sendMessage(Message msg):`立即发送信息

`sendMessageDelayed(Message msg):`指定延时多少毫秒后发送信息

`final boolean hasMessage(int what):`检查消息队列中是否包含what属性为指定值的消息 如果是参数为(int what,Object object):除了判断what属性,还需要判断Object属性是否为指定对象的消息

## 5.Handler的使用示例:

### 1) Handler写在主线程中

在主线程中,因为系统已经初始化了一个Looper对象,所以我们直接创

2.4.2 Date & Time组件(上)

2.4.3 Date & Time组件(下)

2.4.4 Adapter基础讲解

2.4.5 ListView简单实用

2.4.6 BaseAdapter优化

2.4.7 ListView的焦点问题

2.4.8 ListView之checkbox错...

2.4.9 ListView的数据更新问题

2.5.0 构建一个可复用的自定...

2.5.1 ListView Item多布局的...

2.5.2 GridView(网格视图)的...

2.5.3 Spinner(列表选项框)...

2.5.4 AutoCompleteTextVie...

2.5.5 ExpandableListView(...

2.5.6 ViewPager(翻转视图)...

2.5.7 Toast(吐司)的基本使用

2.5.8 Notification(状态栏通...

2.5.9 AlertDialog(对话框)详解

2.6.0 其他几种常用对话框基...

2.6.1 PopupWindow(悬浮框...

2.6.2 菜单(Menu)

2.6.3 ViewPager的简单使用

2.6.4 DrawerLayout(官方侧...

3.1.1 基于监听的事件处理机制

3.2 基于回调的事件处理机制

3.3 Handler消息传递机制浅析

3.4 TouchListener PK OnTo...

3.5 监听EditText的内容变化

3.6 响应系统设置的事件(Co...

3.7 AsyncTask异步任务

3.8 Gestures(手势)

4.1.1 Activity初学乍练

4.1.2 Activity初窥门径

4.1.3 Activity登堂入室

4.2.1 Service初涉

4.2.2 Service进阶

4.2.3 Service精通

4.3.1 BroadcastReceiver牛...

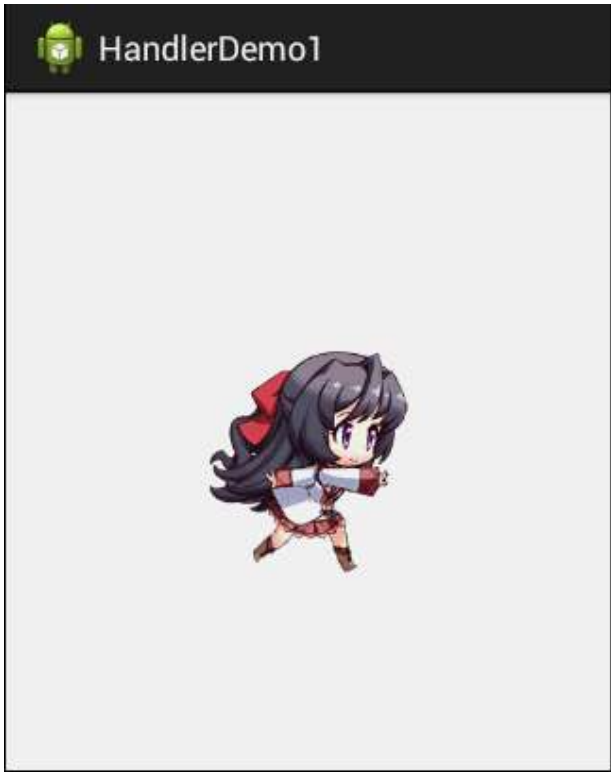
4.3.2 BroadcastReceiver庖...

4.4.1 ContentProvider初探

建Handler对象,就可以进行信息的发送与处理了!

**代码示例：** 简单的一个定时切换图片的程序,通过Timer定时器,定时修改ImageView显示的内容,从而形成帧动画

**运行效果图：**



**实现代码：**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/RelativeLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    tools:context="com.jay.example.handlerdemo1.MainActivity" >

    <ImageView
        android:id="@+id/imgchange"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true" />

</RelativeLayout>
```

**MainActivity.java：**

```
public class MainActivity extends Activity {

    //定义切换的图片的数组id
```

4.4.2 ContentProvider再探...

4.5.1 Intent的基本使用

4.5.2 Intent之复杂数据的传递

5.1 Fragment基本概述

5.2.1 Fragment实例精讲—...

5.2.2 Fragment实例精讲—...

5.2.3 Fragment实例精讲—...

5.2.4 Fragment实例精讲—...

5.2.5 Fragment实例精讲—...

6.1 数据存储与访问之——文...

6.2 数据存储与访问之——S...

6.3.1 数据存储与访问之——...

6.3.2 数据存储与访问之——...

7.1.1 Android网络编程要学...

7.1.2 Android Http请求头与...

7.1.3 Android HTTP请求方...

7.1.4 Android HTTP请求方...

7.2.1 Android XML数据解析

7.2.2 Android JSON数据解析

7.3.1 Android 文件上传

7.3.2 Android 文件下载 ( 1 )

7.3.3 Android 文件下载 ( 2 )

7.4 Android 调用 Webservice

7.5.1 WebView(网页视图)基...

7.5.2 WebView和JavaScrip...

7.5.3 Android 4.4后WebVie...

7.5.4 WebView文件下载

7.5.5 WebView缓存问题

7.5.6 WebView处理网页返...

7.6.1 Socket学习网络基础准备

7.6.2 基于TCP协议的Socket...

7.6.3 基于TCP协议的Socket...

7.6.4 基于UDP协议的Socke...

8.1.1 Android中的13种Draw...

8.1.2 Android中的13种Draw...

8.1.3 Android中的13种Draw...

8.2.1 Bitmap(位图)全解析 P...

8.2.2 Bitmap引起的OOM问题

8.3.1 三个绘图工具类详解

8.3.2 绘图类实战示例

```

int imgids[] = new int[]{
    R.drawable.s_1, R.drawable.s_2,R.drawable.s_3,
    R.drawable.s_4,R.drawable.s_5,R.drawable.s_6,
    R.drawable.s_7,R.drawable.s_8
};
int imgstart = 0;

    final Handler myHandler = new Handler()
{
    @Override
    //重写handleMessage方法,根据msg中what的值判断是否执行后续操作

    public void handleMessage(Message msg) {
        if(msg.what == 0x123)
        {
            imgchange.setImageResource(imgids[imgstart++ % 8]);

        }
    }
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    final ImageView imgchange = (ImageView) findViewById(R.
id.imgchange);

    //使用定时器,每隔200毫秒让handler发送一个空信息
    new Timer().schedule(new TimerTask() {
        @Override
        public void run() {
            myHandler.sendMessage(0x123);

        }
    }, 0,200);
}
}

```

## 2 ) Handler写在子线程中

如果是Handler写在了子线程中的话,我们就需要自己创建一个Looper对象了! 创建的流程如下:

**1 )**直接调用Looper.prepare()方法即可为当前线程创建Looper对象,而它的构造器会创建配套的MessageQueue; **2 )**创建Handler对象,重写handleMessage( )方法就可以处理来自于其他线程的信息了! **3 )**调用Looper.loop()方法启动Looper

使用示例：输入一个数，计算后通过Toast输出在这个范围内的所有质数

8.3.3 Paint API之—— Mask...  
8.3.4 Paint API之—— Xferm...  
8.3.5 Paint API之—— Xferm...  
8.3.6 Paint API之—— Xferm...  
8.3.7 Paint API之—— Xferm...  
8.3.8 Paint API之—— Xferm...  
8.3.9 Paint API之—— Color...  
8.3.10 Paint API之—— Colo...  
8.3.11 Paint API之—— Colo...  
8.3.12 Paint API之—— Path...  
8.3.13 Paint API之—— Sha...  
8.3.14 Paint几个枚举/常量值...  
8.3.15 Paint API之——Type...  
8.3.16 Canvas API详解(Part 1)  
8.3.17 Canvas API详解(Part...  
8.3.18 Canvas API详解(Part...  
8.4.1 Android动画合集之帧...  
8.4.2 Android动画合集之补...  
8.4.3 Android动画合集之属...  
8.4.4 Android动画合集之属...  
9.1 使用SoundPool播放音...  
9.2 MediaPlayer播放音频与...  
9.3 使用Camera拍照  
9.4 使用MediaRecord录音  
10.1 TelephonyManager(电...  
10.2 SmsManager(短信管理...  
10.3 AudioManager(音频管...  
10.4 Vibrator(振动器)  
10.5 AlarmManager(闹钟服务)  
10.6 PowerManager(电源服...  
10.7 WindowManager(窗口...  
10.8 LayoutInflater(布局服务)  
10.9 WallpaperManager(壁...  
10.10 传感器专题(1)——相...  
10.11 传感器专题(2)——方...  
10.12 传感器专题(3)——加...  
10.12 传感器专题(4)——其...  
10.14 Android GPS初涉  
11.0 《2015最新Android基...

实现代码：**main.xml**：

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <EditText
        android:id="@+id/etNum"
        android:inputType="number"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="请输入上限"/>

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="cal"
        android:text="计算"/>

</LinearLayout>
```

**MainActivity.java:**

```
public class CalPrime extends Activity
{
    static final String UPPER_NUM = "upper";
    EditText etNum;
    CalThread calThread;
    // 定义一个线程类
    class CalThread extends Thread
    {
        public Handler mHandler;

        public void run()
        {
            Looper.prepare();
            mHandler = new Handler()
            {
                // 定义处理消息的方法
                @Override
                public void handleMessage(Message msg)
                {
                    if(msg.what == 0x123)
                    {
                        int upper = msg.getData().getInt(UPPER_
NUM);

                        List<Integer> nums = new ArrayList<Inte
ger>();

                        // 计算从2开始、到upper的所有质数
                        outer:
                        for (int i = 2 ; i <= upper ; i++)
                        {
```

```

        // 用 i 处于从 2 开始、到 i 的平方根的所有数

        for (int j = 2 ; j <= Math.sqrt(i)

; j++)

        {

            // 如果可以整除，表明这个数不是质数

            if(i != 2 && i % j == 0)

            {

                continue outer;

            }

        }

        nums.add(i);

    }

    // 使用 Toast 显示统计出来的所有质数

    Toast.makeText(CalPrime.this , nums.toS

tring()

        , Toast.LENGTH_LONG).show();

    }

}

};

Looper.loop();

}

}

@Override

public void onCreate(Bundle savedInstanceState)

{

    super.onCreate(savedInstanceState);

    setContentView(R.layout.main);

    etNum = (EditText) findViewById(R.id.etNum);

    calThread = new CalThread();

    // 启动新线程

    calThread.start();

}

// 为按钮的点击事件提供事件处理函数

public void cal(View source)

{

    // 创建消息

    Message msg = new Message();

    msg.what = 0x123;

    Bundle bundle = new Bundle();

    bundle.putInt(UPPER_NUM ,

        Integer.parseInt(etNum.getText().toString()));

    msg.setData(bundle);

    // 向新线程中的 Handler 发送消息

    calThread.mHandler.sendMessage(msg);

}

}

```

PS:本例子来自于《Android疯狂讲义》~

## 本节小结

本节对Android中的Handler事件传递进行了简单的分析，要分清楚Handler，Message，MessageQueue，Loop的概念，以及Handler写在主线程中以及子线程中的区别！

← 3.2 基于回调的事件处理机制  
3.4 TouchListener PK OnTouchEvent + 多点触碰 →



在线实例

- HTML 实例
- CSS 实例
- JavaScript 实例
- Ajax 实例
- jQuery 实例
- XML 实例
- Java 实例

字符集&工具

- HTML 字符集设置
- HTML ASCII 字符集
- HTML ISO-8859-1
- HTML 实体符号
- HTML 拾色器
- JSON 格式化工具

最新更新

- Swift 正式开源
- PHP 7 正式发布
- Shell 编程快速入门
- Shell 文件包含
- Shell 输入/输出...
- Shell printf 命令
- Shell 基本运算符

站点信息

- 意见反馈
- 免责声明
- 关于我们
- 文章归档

关注微信



Copyright © 2013-2015 菜鸟教程 **runoob.com** All Rights Reserved. 备案号：闽ICP备15012807号-1