

[首页](#) [ANDROID](#) [互联网](#) [杂乱无章](#) [科技资讯](#) [程序员人生](#) [程序员笑话](#) [编程技术](#) [网址导航](#)

## 3.1.1 基于监听的事件处理机制

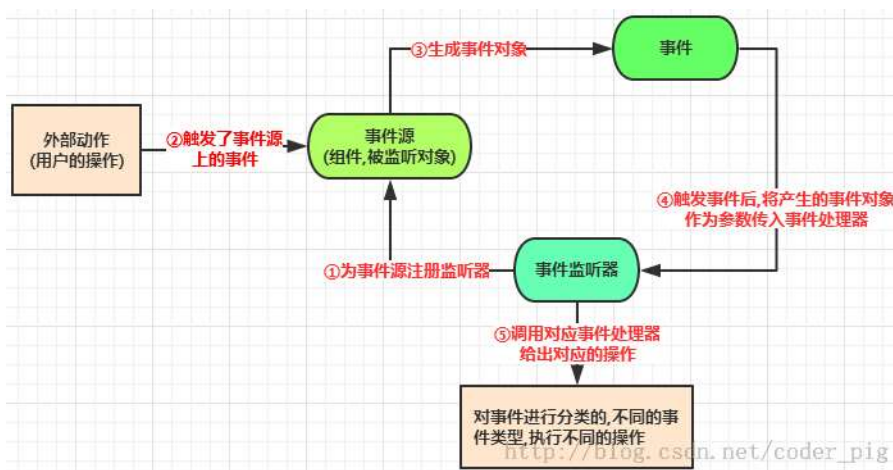
分类 [Android 基础入门教程](#)

### 本节引言：

第二章我们学习的是Android的UI控件，我们可以利用这些控件构成一个精美的界面，但是仅仅是界面而已；下一步就要开始学习逻辑与业务实现了，本章节讲解的是Android的事件处理机制！何为事件处理机制？举个简单的例子，比如点击一个按钮，我们向服务器发送登陆请求！当然，Android中的事件处理机制不止这一种，比如屏幕发生选择，我们点击了屏幕上某个区域...简单点说，事件处理机制就是我和UI发生交互时，我们在背后添加一些小动作而已！本节我们来介绍使用的最频繁的一种：基于监听的事件处理机制！

### 1.基于监听的时间处理机制模型:

流程模型图：



文字表述：

事件监听机制中由**事件源**，**事件**，**事件监听器**三类对象组成 处理流程如下: **Step 1:**为某个事件源(组件)设置一个监听器,用于监听用户操作 **Step 2:**用户的操作,触发了事件源的监听器 **Step 3:**生成了对应的事件对象 **Step 4:**将这个事件源对象作为参数传给事件监听器 **step 5:**事件监听器对事件对象进行判断,执行对应的事件处理器(对应事件的处理方法)

归纳：

**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)

事件监听机制是一种委派式的事件处理机制,事件源(组件)事件处理委托给事件监听器 当事件源发生指定事件时,就通知指定事件监听器,执行相应的操作

## 2.五种不同的使用形式：

我们以下面这个：**简单的按钮点击,提示Toast信息的程序**；使用五种不同的形式来实现！

效果图：



### 1) 直接用匿名内部类

平时最常用的一种:直接setXxxListener后,重写里面的方法即可; 通常是临时使用一次,复用性不高！

实现代码如下：**MainActivity.java**:

```
package com.jay.example.innerlisten;

import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;
import android.app.Activity;

public class MainActivity extends Activity {
    private Button btnshow;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

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

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
btnshow = (Button) findViewById(R.id.btnshow);
btnshow.setOnClickListener(new OnClickListener() {
    //重写点击事件的处理方法onClick()
    @Override
    public void onClick(View v) {
        //显示Toast信息
        Toast.makeText(getApplicationContext(), "你点击了按钮", Toast.LENGTH_SHORT).show();
    }
});
}
}

```

## 2) 使用内部类

和上面的匿名内部类不同哦！使用优点:可以在该类中进行复用,可直接访问外部类的所有界面组件！

实现代码如下：**MainActivity.java**:

```

package com.jay.example.innerlisten;

import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;
import android.app.Activity;

public class MainActivity extends Activity {
    private Button btnshow;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnshow = (Button) findViewById(R.id.btnshow);
        //直接new一个内部类对象作为参数
        btnshow.setOnClickListener(new BtnClickListener());
    }
    //定义一个内部类,实现View.OnClickListener接口,并重写onClick()方法
    class BtnClickListener implements View.OnClickListener {
        @Override
        public void onClick(View v) {
            Toast.makeText(getApplicationContext(), "按钮被点击了", Toast.LENGTH_SHORT).show();
        }
    }
}

```

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

```
}
```

### 3) 使用外部类 :

就是另外创建一个处理事件的Java文件,这种形式用的比较少!因为外部类不能直接访问用户界面 类中的组件,要通过构造方法将组件传入使用;这样导致的结果就是代码不够简洁!

ps:为了演示传参,这里用TextView代替Toast提示!



实现代码如下 : **MyClick.java**:

```
package com.jay.example.innerlisten;

import android.view.View;
import android.view.View.OnClickListener;
import android.widget.TextView;

public class MyClick implements OnClickListener {
    private TextView textshow;
    //把文本框作为参数传入
    public MyClick(TextView txt)
    {
        textshow = txt;
    }
    @Override
    public void onClick(View v) {
        //点击后设置文本框显示的文字
        textshow.setText("点击了按钮!");
    }
}
```

- 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...

## MainActivity.java

```
package com.jay.example.innerlisten;

import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;
import android.app.Activity;

public class MainActivity extends Activity {
    private Button btnshow;
    private TextView txtshow;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnshow = (Button) findViewById(R.id.btnshow);
        txtshow = (TextView) findViewById(R.id.textshow);
        //直接new一个外部类，并把TextView作为参数传入
        btnshow.setOnClickListener(new MyClick(txtshow));
    }
}
```

## 4) 直接使用Activity作为事件监听器

只需要让Activity类实现XxxListener事件监听接口,在Activity中定义重写对应的事件处理器方法 eg:Activity实现了OnClickListener接口,重写了onClick(view)方法在为某些组建添加该事件监听对象 时,直接setXxx.Listener(this)即可

实现代码如下：**MainActivity.java**:

```
package com.jay.example.innerlisten;

import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;
import android.app.Activity;

//让Activity方法实现OnClickListener接口
public class MainActivity extends Activity implements OnClickListener{
    private Button btnshow;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnshow = (Button) findViewById(R.id.btnshow);
        //直接写个this
    }
}
```

```

        btnshow.setOnClickListener(this);
    }
    //重写接口中的抽象方法
    @Override
    public void onClick(View v) {
        Toast.makeText(getApplicationContext(), "点击了按钮", Toast.LENGTH_SHORT).show();
    }
}

```

## 5) 直接绑定到标签:

就是直接在xml布局文件中对应得Activity中定义一个事件处理方法  
 eg:public void myClick(View source) source对应事件源(组件) 接着布局文件中对应要触发事件的组建,设置一个属性:onclick = "myclick"即可

实现代码如下：**MainActivity.java**:

```

package com.jay.example.caller;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    //自定义一个方法,传入一个view组件作为参数
    public void myclick(View source)
    {
        Toast.makeText(getApplicationContext(), "按钮被点击了",
        Toast.LENGTH_SHORT).show();
    }
}

```

**main.xml**布局文件:

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res
/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <Button
        android:layout_width="wrap_content"

```

```
        android:layout_height="wrap_content"
        android:text="按钮"
        android:onClick="myclick"/>
</LinearLayout>
```

## 本节小结

本节给大家介绍了Android中的事件处理机制，例子中的是onClickListener点击事件，当然除了这个以外还有其他的事件，比如onItemClickListener，凡是需要通过setXxxListener这些，基本上都是基于事件监听的！另外这五种方式用的比较多的是：1，2，3，5几种，看具体情况而定~

← 2.6.4 DrawerLayout(官方侧滑菜单)的简单使用 3.2 基于回调的事件处理机制 →



### 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



反馈

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