

4.1.2 Activity初窥门径

分类 [Android 基础入门教程](#)

本节引言：

上一节中我们对Activity一些基本的概念进行了了解，什么是Activity，Activity的生命周期，如何去启动一个Activity等，本节我们继续来学习Activity，前面也讲了一个App一般都是又多个Activity构成的，这就涉及到了多个Activity间数据传递的问题了，那么本节继续学习Activity的使用！另外关于传递集合，对象，数组，Bitmap的我们会在Intent那里进行讲解，这里只介绍如何传递基本数据！

1.Activity间的数据传递：



代码示例：

效果图：



代码下载：[ActivityTest1.zip](#)

2.多个Activity间的交互(后一个传回给前一个)

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)

①使用startActivityForResult(Intent intent, int requestCode)
来启动一个Activity
②在启动的Activity中重写
onActivityResult(int requestCode, int resultCode, Intent data)
requestCode是用于区分在该Activity中不同的启动方式,比如有两个不同的按钮,启动的是同一个Activity,传递的数据可能不同,这里就可以用这个requestCode来区别
resultCode:子Activity通过setResult()返回的码
③在子Activity重写:
setResult(int resultCode, Intent data)

代码示例：

效果图：



代码下载：[ActivityTest2.zip](#)

3.知晓当前是哪个Activity

让所有Activity继承一个自定义的BaseActivity类,在OnCreate()方法中添加下述语句即可:
`Log.d("BaseActivity", getClass().getSimpleName());`

4.随时关闭所有Activity

有时我们可能会打开了很多个Activity,突然来个这样的需求,在某个页面可以关掉所有的Activity并退出程序!好吧,下面提供一个关闭所有Activity的方法,就是用一个list集合来存储所有Activity!

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 AutoCompleteText Vie...

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

1.创建一个Activity管理器类:ActivityCollector
定义三个共有静态的方法:定义存储Activity的list集合,方法如下

①addActivity:往集合添加Activity对象
②removeActivity:移除Activity中的对象
③finishAll:增强for循环遍历所有Activity调用:
if(!activity.isFinishing())activity.finish();

2.BaseActivity中

①onCreate()方法添加
ActivityCollector.addActivity(this);
②onDestory()方法添加:
ActivityCollector.removeActivity(this);
③可以在任意一个Activity中调用:
ActivityCollector.finishAll();
从而关闭所有Activity,退出app!

具体代码如下：

```
public class ActivityCollector {
    public static LinkedList<Activity> activities = new Linkedi
ist<Activity>();
    public static void addActivity(Activity activity)
    {
        activities.add(activity);
    }
    public static void removeActivity(Activity activity)
    {
        activities.remove(activity);
    }
    public static void finishAll()
    {
        for(Activity activity:activities)
        {
            if(!activity.isFinishing())
            {
                activity.finish();
            }
        }
    }
}
```

5.完全退出App的方法

上面说的是关闭所有Activity的，但是有些时候我们可能想杀死整个App，连后台任务都杀死 杀得一干二净的话，可以使用搭配着下述代码使用：

实现代码：

```
/**
 * 退出应用程序
 */
public void AppExit(Context context) {
    try {
        ActivityCollector.finishAll();
    }
```

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

```

ActivityManager activityMgr = (ActivityManager) context

        .getSystemService(Context.ACTIVITY_SERVICE);
    activityMgr.killBackgroundProcesses(context.getPackageName());
    System.exit(0);
} catch (Exception ignored) {}
}

```

6. 双击退出程序的两种方法：

1) 定义一个变量，来标识是否退出

```

// 定义一个变量，来标识是否退出
private static boolean isExit = false;
Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        super.handleMessage(msg);
        isExit = false;
    }
};

public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK) {
        if (!isExit) {
            isExit = true;
            Toast.makeText(getApplicationContext(),
                "再按一次退出程序",
                    Toast.LENGTH_SHORT).show();

            // 利用handler延迟发送更改状态信息
            mHandler.sendEmptyMessageDelayed(0, 2000);
        } else {
            exit(this);
        }
        return false;
    }
    return super.onKeyDown(keyCode, event);
}

```

2) 保存点击时间：

```

//保存点击的时间
private long exitTime = 0;
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK) {
        if ((System.currentTimeMillis() - exitTime) > 2000) {
            Toast.makeText(getApplicationContext(),
                "再按一次退出程序",

```

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...

```

        Toast.LENGTH_SHORT).show();

        w();

        exitTime = System.currentTimeMillis();
    } else {
        exit();
    }

    return false;
}

return super.onKeyDown(keyCode, event);
}

```

7.为Activity设置过场动画

所谓的过场动画就是切换到另外的Activity时加上一些切换动画，比如淡入淡出，放大缩小，左右互推等！当然，我们并不在这里详细讲解动画，后面有专门的章节来讲解这个，这里只教大家如何去加载动画，另外给大家提供了一些比较常用的过渡动画，只要将相关动画文件添加到res/anim目录下，然后下述方法二选一 就可以实现Activity的切换动画了！

1) 方法一：

**A跳转到B,在startActivity (intent) 后面加上
overridePendingTransition(R.anim.anim_in,
R.anim.anim_out);**

**B返回A,要在finish()后面加上
overridePendingTransition(R.anim.anim_in,
R.anim.anim_out);**

**anim_in是进入的Activity的动画
anim_out是退出的Activity的动画**

2) 方法二：

通过style进行配置，这个是全局的哦，就是所有的Activity都会加载这个动画！

实现代码如下：

①在style.xml中自定义style：

```

<!-- 默认Activity跳转动画 -->
<style name="default_animation" mce_bogus="1" parent="@android:style/Animation.Activity">
    <item name="android:activityOpenEnterAnimation">@anim/default_anim_in</item>
    <item name="android:activityOpenExitAnimation">@anim/anim_s
tay</item>
    <item name="android:activityCloseEnterAnimation">@anim/anim
_stay</item>
    <item name="android:activityCloseExitAnimation">@anim/default_anim_out</item>

```

```
</style>
```

解释：

4个item分别代表:

Activity A跳转到Activity B时Activity B进入动画;

Activity A跳转到Activity B时Activity A退出动画;

Activity B返回Activity A时Activity A的进入动画

Activity B返回Activity A时ActivityB的退出动画

②然后修改下AppTheme:

```
<style name="AppTheme" mce_bogus="1" parent="@android:style/The
me.Light">
    <item name="android:windowAnimationStyle">@style/default
t_animation</item>
    <item name="android:windowNoTitle">true</item>
</style>
```

③最后在application设置下：

```
<application
    android:icon="@drawable/logo"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
```

好的，动画特效就这样duang一声设置好了~

3) 其他

好的，除了上面两种方法以外，还可以使用**TransitionManager**来实现，但是需求版本是API 19以上的，另外还有一种**addOnPreDrawListener**的转换动画，这个用起来还是有点麻烦的，可能不是适合初学者 这里也不讲，最后提供下一些常用的动画效果打包，选择需要的特效加入工程即可！[Activity常用过渡动画.zip](#)

8.Bundle传递数据的限制

在使用Bundle传递数据时，要注意，Bundle的大小是有限制的 < 0.5MB，如果大于这个值 是会报TransactionTooLargeException异常的！！！！

9.使用命令行查看当前所有Activity的命令：

使用下述命令即可，前提是你为SDK配置了环境变量:adb shell
dumpsys activity



10.设置Activity全屏的方法：

1) 代码隐藏ActionBar

在Activity的onCreate方法中调用getActionBar.hide();即可

2) 通过requestWindowFeature设置

requestWindowFeature(Window.FEATURE_NO_TITLE); 该代码需要在 setContentView ()之前调用，不然会报错！！

3) 通过AndroidManifest.xml的theme

在需要全屏的Activity的标签内设置 theme =
@android:style/Theme.NoTitleBar.FullScreen

11.onWindowFocusChanged方法妙用：

我们先看下官方对这个方法的介绍：

```
abstract void onWindowFocusChanged (boolean hasFocus)  
Callback method to be invoked when the window focus changes in the view tree.
```

就是，当Activity得到或者失去焦点的时候，就会回调该方法！如果我们想监控Activity是否加载完毕，就可以用到这个方法了~ 想深入了解的可移步到这篇文章：[onWindowFocusChanged触发简介](#)

12.定义对话框风格的Activity

在某些情况下，我们可能需要将Activity设置成对话框风格的，Activity一般是占满全屏的，而Dialog则是占据部分屏幕的！实现起来也很简单！

直接设置下Activity的theme:

```
android:theme="@android:style/Theme.Dialog"
```

这样就可以了，当然你可以再设置下标题，小图标！

```
//设置左上角小图标  
requestWindowFeature (Window.FEATURE_LEFT_ICON);  
setContentView (R.layout.main);  
getWindow().setFeatureDrawableResource (Window.FEATURE_LEFT_ICON  
, android.R.drawable.ic_lion_icon);  
//设置文字：  
setTitle (R.string.actdialog_title); //XML代码中设置:android:lab  
el="@string/activity_dialog"
```

本节小结：

好的，本节我们又学习了一下Activity在实际开发中的一些常见问题，相信在实际开发中会帮到大家的！下节我们来学习Activity的栈的概念，以及四种加

载模式！敬请期待~谢谢~

← 4.1.1 Activity初学乍练

4.1.3 Activity登堂入室 →

坚持一个月，听懂CNN



海量英语视频，互动单词学习卡片 和900万用户一起天天向上！



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