

[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](#)

## 4.1.1 Activity初学乍练

分类 **Android 基础入门教程**

### 本节引言：

本节开始讲解Android的四大组件之一的Activity(活动)，先看下官方对于Activity的介绍：PS:官网文档：[Activity](#)

介绍如下：

An **Activity** is an application component that provides a screen with which users can interact in order to do something, such as dial the phone, take a photo, send an email, or view a map. Each activity is given a window in which to draw its user interface. The window typically fills the screen, but may be smaller than the screen and float on top of other windows.

大

概意思：

Activity是一个应用程序的组件，他在屏幕上提供了一个区域，允许用户在上面做一些交互性的操作，比如打电话，照相，发送邮件，或者显示一个地图！Activity可以理解成一个绘制用户界面的窗口，而这个窗口可以填满整个屏幕，也可能比屏幕小或者浮动在其他窗口的上方！

从上面这段话，我们可以得到以下信息：

1. Activity用于显示用户界面，用户通过Activity交互完成相关操作
2. 一个App允许有多个Activity

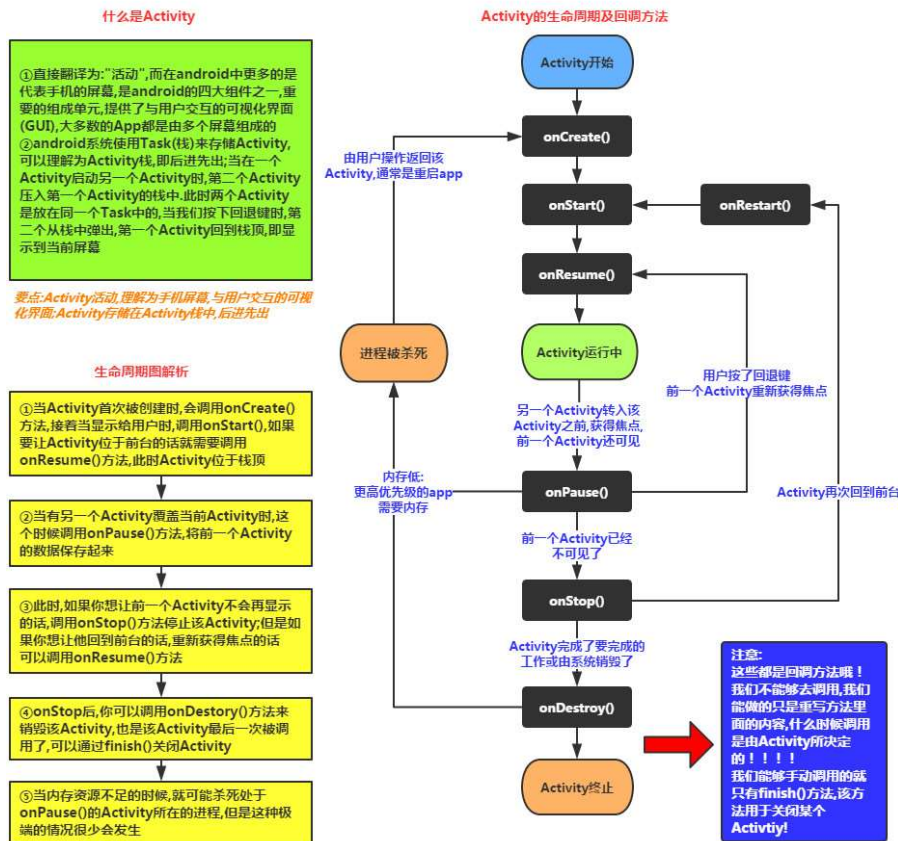
好了，大概的引言就介绍到这里，想深入了解可以继续看API，开始本节内容

~

### 1.Activity的概念与Activity的生命周期图：

**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)



#### 注意事项：

1. onPause()和onStop()被调用的前提是：打开了一个新的Activity！而前者是旧Activity还可见的状态；后者是旧Activity已经不可见！
2. 另外，亲测：AlertDialog和PopWindow是不会触发上述两个回调方法的~

## 2.Activity/ActionBarActivity/AppCompatActivity的区别：

在开始讲解创建Activity之前要说下这三个的一个区别：Activity就不用说啦，后面这两个都是为了低版本兼容而提出的提出来的，他们都在v7包下，ActionBarActivity已被废弃，从名字就知道，ActionBar~，而在5.0后，被Google弃用了，现在用ToolBar...而我们现在在Android Studio创建一个Activity默认继承的会是：AppCompatActivity! 当然你也可以只写Activity，不过AppCompatActivity给我们提供了一些新的东西而已！两个选一个，Just you like~

## 3.Activity的创建流程

- 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

## Activity的使用流程

### ①自定义Activity类名,继承Activity类或者它的子类

```
class MyActivity extends Activity{
```

### ②重写onCreate()方法,在该方法中调用setContentView()设置要显示的视图

```
public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
}
```

### ③在AndroidManifest.xml对Activity进行配置

```
<activity
    android:icon = "图标"
    android:name = "类名"
    android:label = "Activity显示的标题"
    android:theme = "要应用的主题"></activity>
```

### ④启动Activity:调用startActivity(Intent);

```
Intent it = new
Intent(MainActivity.this, MyActivity.class);
startActivity(it);
```

### ⑤关闭Activity:调用finish,直接关闭当前Activity

```
我们可以把他写到启动第二个Activity的方法中,当启动第二个
Activity时,第一个Activity就会被关闭
finish();
```

PS:

好了,上面也说过,可以继承Activity和AppCompatActivity,只不过后者提供了一些新的东西而已! 另外,切记,Android中的四大组件,只要你定义了,无论你用没用,都要在AndroidManifest.xml对这个组件进行声明,不然运行时程序会直接退出,报ClassNotFoundException...

## 4.onCreate()一个参数和两个参数的区别：

相信用as的朋友在重写Act的onCreate()方法时会发现，这玩意有两个参数：

```
@Override
public void onCreate(Bundle savedInstanceState, PersistableBundle persistentState) {
    super.onCreate(savedInstanceState, persistentState);
}
```

可是正常的才只有一个参数啊：

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

恩呢，这就是5.0给我们提供的新的方法，要用它，先要在配置文件中为我们的Activity设置一个属性：

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JavaScip...

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

```
android:persistentMode="persistAcrossReboots"
```

然后我们的Activity就拥有了持久化的能力了，一般我们会搭配另外两个方法来使用：

```
public void onSaveInstanceState(Bundle outState, PersistableBundle outPersistentState)

public void onRestoreInstanceState(Bundle savedInstanceState, PersistableBundle persistentState)
```

相信有些朋友对这两个方法名不陌生吧，前一个方法会在下述情形中被调用：

1. 点击home键回到主页或长按后选择运行其他程序
2. 按下电源键关闭屏幕
3. 启动新的Activity
4. 横竖屏切换时，肯定会执行，因为横竖屏切换的时候会先销毁Act，然后再重新创建 重要原则：当系统"未经你许可"时销毁了你的activity，则 onSaveInstanceState会被系统调用，这是系统的责任，因为它必须要提供一个机会让你保存你的数据（你可以保存也可以不保存）。

而后一个方法，和onCreate同样可以从取出前者保存的数据：一般是在onStart()和onResume()之间执行！之所以有两个可以获取到保存数据的方法，是为了避免Act跳转而没有关闭，然后不走onCreate()方法，而你又想取出保存数据~

**说回来：**说回这个Activity拥有了持久化的能力，增加的这个

PersistableBundle参数令这些方法 拥有了系统**关机后重启**的数据恢复能力！！而且不影响我们其他的序列化操作，卧槽，具体怎么实现的，暂时还不了解，可能是另外弄了个文件保存吧~！后面知道原理的话会告知下大家！另外，API版本需要>=21，就是要5.0以上的版本才有效~

## 4.启动一个Activity的几种方式

在Android中我们可以通过下面两种方式来启动一个新的Activity,注意这里是怎么启动，而非 启动模式！！分为显示启动和隐式启动！

### 1. 显式启动：通过包名来启动，写法如下：

#### ①最常见的：

```
startActivity(new Intent(当前Act.this,要启动的Act.class));
```

#### ②通过Intent的ComponentName：

```
ComponentName cn = new ComponentName("当前Act的全限定类名",
```

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...



```
"启动Act的全限定类名") ;
Intent intent = new Intent() ;
intent.setComponent(cn) ;
startActivity(intent) ;
```

### ③初始化Intent时指定包名：

```
Intent intent = new Intent("android.intent.action.MAIN");
intent.setClassName("当前Act的全限定类名","启动Act的全限定类名");
startActivity(intent);
```

**2.隐式启动：通过Intent-filter的Action,Category或data来实现** 这个是通过Intent的 intent-filter\*\*来实现的，这个Intent那章会详细讲解！这里知道个大概就可以了！

```
<activity android:name=".SecondActivity"
    android:label="第二个Activity">
    <intent-filter>
        <action android:name="my_action"/>
        <category android:name="my_category"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
</activity>
```

！这个一定要写哦~

#### Java文件中启动:

```
Intent it = new Intent();
it.setAction("my_action");
it.addCategory("my_category");
startActivity(it);
```

### 3. 另外还有一个直接通过包名启动apk的：

```
Intent intent = getPackageManager().getLaunchIntentForPackage
("apk第一个启动的Activity的全限定类名") ;
if(intent != null) startActivity(intent) ;
```

## 5.横竖屏切换与状态保存的问题

前面也也说到App横竖屏切换的时候会销毁当前的Activity然后重新创建一个，你可以自行在生命周期 的每个方法里都添加打印Log的语句，来进行判断，又或者设一个按钮一个TextView点击按钮后，修改TextView 文本，然后横竖屏切换，会神奇的发现TextView文本变回之前的内容了！横竖屏切换时Act走下述生命周期：

**onPause-> onStop-> onDestroy-> onCreate-> onStart-> onResume**

关于横竖屏切换可能遇到下述问题：

1.先说下如何**禁止屏幕横竖屏自动切换**吧，很简单，在AndroidManifest.xml 中为Act添加一个属性：**android:screenOrientation**，有下述可选值：

**unspecified**:默认值 由系统来判断显示方向.判定的策略是和设备相关的，所以不同的设备会有不同的显示方向。

**landscape**:横屏显示（宽比高要长）

**portrait**:竖屏显示(高比宽要长)

**user**:用户当前首选的方向

**behind**:和该Activity下面的那个Activity的方向一致(在Activity堆栈中的)

**sensor**:有物理的感应器来决定。如果用户旋转设备这屏幕会横竖屏切换。

**nosensor**:忽略物理感应器，这样就不会随着用户旋转设备而更改了（"unspecified"设置除外）。

## 2.横竖屏时想加载不同的布局：

1) 准备两套不同的布局，Android会自己根据横竖屏加载不同布局：创建两个布局文件夹：**layout-land**横屏,**layout-port**竖屏 然后把这两套布局文件丢这两文件夹里，文件名一样，Android就会自行判断，然后加载相应布局了！

2) 自己在代码中进行判断，自己想加载什么就加载什么：

我们一般是在onCreate()方法中加载布局文件的，我们可以在这里对横竖屏的状态做下判断，关键代码如下：

```
if (this.getResources().getConfiguration().orientation == Configuration.ORIENTATION_LANDSCAPE) {
    setContentView(R.layout.横屏);
}

else if (this.getResources().getConfiguration().orientation == Configuration.ORIENTATION_PORTRAIT) {
    setContentView(R.layout.竖屏);
}
```

## 3. 如何让模拟器横竖屏切换

如果你的模拟器是GM的话。直接按模拟器上的切换按钮即可，原生模拟器可按ctrl + f11/f12切换！

## 4. 状态保存问题：

这个上面也说过了，通过一个Bundle savedInstanceState参数即可完成！三个核心方法：

```
onCreate(Bundle savedInstanceState);
onSaveInstanceState(Bundle outState);
onRestoreInstanceState(Bundle savedInstanceState);
```

你只重写onSaveInstanceState()方法，往这个bundle中写入数据，比如：

```
outState.putInt("num",1);
```

这样，然后你在onCreate或者onRestoreInstanceState中就可以拿出里面存储的数据，不过拿之前要判断下是否为null哦！

```
savedInstanceState.getInt("num");
```

然后想干嘛就干嘛~

## 6.系统给我们提供的常见的Activity

好的，最后给大家附上一些系统给我们提供的一些常见的Activitiy吧！

```
//1.拨打电话
// 给移动客服10086拨打电话
Uri uri = Uri.parse("tel:10086");
Intent intent = new Intent(Intent.ACTION_DIAL, uri);
startActivity(intent);

//2.发送短信
// 给10086发送内容为“Hello”的短信
Uri uri = Uri.parse("smsto:10086");
Intent intent = new Intent(Intent.ACTION_SENDTO, uri);
intent.putExtra("sms_body", "Hello");
startActivity(intent);

//3.发送彩信（相当于发送带附件的短信）
Intent intent = new Intent(Intent.ACTION_SEND);
intent.putExtra("sms_body", "Hello");
Uri uri = Uri.parse("content://media/external/images/media/23");
;
intent.putExtra(Intent.EXTRA_STREAM, uri);
intent.setType("image/png");
startActivity(intent);

//4.打开浏览器：
// 打开Google主页
Uri uri = Uri.parse("http://www.baidu.com");
Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);

//5.发送电子邮件：(阉割了Google服务的没戏!!!!)
// 给someone@domain.com发邮件
Uri uri = Uri.parse("mailto:someone@domain.com");
Intent intent = new Intent(Intent.ACTION_SENDTO, uri);
startActivity(intent);
// 给someone@domain.com发邮件发送内容为“Hello”的邮件
Intent intent = new Intent(Intent.ACTION_SEND);
intent.putExtra(Intent.EXTRA_EMAIL, "someone@domain.com");
intent.putExtra(Intent.EXTRA_SUBJECT, "Subject");
intent.putExtra(Intent.EXTRA_TEXT, "Hello");
intent.setType("text/plain");
startActivity(intent);
// 给多人发邮件
```

```
Intent intent=new Intent(Intent.ACTION_SEND);
String[] tos = {"1@abc.com", "2@abc.com"}; // 收件人
String[] ccs = {"3@abc.com", "4@abc.com"}; // 抄送
String[] bccs = {"5@abc.com", "6@abc.com"}; // 密送
intent.putExtra(Intent.EXTRA_EMAIL, tos);
intent.putExtra(Intent.EXTRA_CC, ccs);
intent.putExtra(Intent.EXTRA_BCC, bccs);
intent.putExtra(Intent.EXTRA_SUBJECT, "Subject");
intent.putExtra(Intent.EXTRA_TEXT, "Hello");
intent.setType("message/rfc822");
startActivity(intent);

//6.显示地图:
// 打开Google地图中国北京位置(北纬39.9, 东经116.3)
Uri uri = Uri.parse("geo:39.9,116.3");
Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);

//7.路径规划
// 路径规划: 从北京某地(北纬39.9, 东经116.3) 到上海某地(北纬31.2, 东经121.4)
Uri uri = Uri.parse("http://maps.google.com/maps?f=d&saddr=39.9
116.3&daddr=31.2 121.4");
Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);

//8.多媒体播放:
Intent intent = new Intent(Intent.ACTION_VIEW);
Uri uri = Uri.parse("file:///sdcard/foo.mp3");
intent.setDataAndType(uri, "audio/mp3");
startActivity(intent);

//获取SD卡下所有音频文件,然后播放第一首==
Uri uri = Uri.withAppendedPath(MediaStore.Audio.Media.INTERNAL_
CONTENT_URI, "1");
Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);

//9.打开摄像头拍照:
// 打开拍照程序
Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
startActivityForResult(intent, 0);
// 取出照片数据
Bundle extras = intent.getExtras();
Bitmap bitmap = (Bitmap) extras.get("data");

//另一种:
//调用系统相机应用程序,并存储拍下来的照片
Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
time = Calendar.getInstance().getTimeInMillis();
intent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(new File(
Environment
.getExternalStorageDirectory().getAbsolutePath()+"/tucue", time
```



```
+ ".jpg"))));
startActivityForResult(intent, ACTIVITY_GET_CAMERA_IMAGE);

//10.获取并剪切图片
// 获取并剪切图片
Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
intent.setType("image/*");
intent.putExtra("crop", "true"); // 开启剪切
intent.putExtra("aspectX", 1); // 剪切的宽高比为1: 2
intent.putExtra("aspectY", 2);
intent.putExtra("outputX", 20); // 保存图片的宽和高
intent.putExtra("outputY", 40);
intent.putExtra("output", Uri.fromFile(new File("/mnt/sdcard/temp"))); // 保存路径
intent.putExtra("outputFormat", "JPEG");// 返回格式
startActivityForResult(intent, 0);
// 剪切特定图片
Intent intent = new Intent("com.android.camera.action.CROP");
intent.setClassName("com.android.camera", "com.android.camera.CropImage");
intent.setData(Uri.fromFile(new File("/mnt/sdcard/temp")));
intent.putExtra("outputX", 1); // 剪切的宽高比为1: 2
intent.putExtra("outputY", 2);
intent.putExtra("aspectX", 20); // 保存图片的宽和高
intent.putExtra("aspectY", 40);
intent.putExtra("scale", true);
intent.putExtra("noFaceDetection", true);
intent.putExtra("output", Uri.parse("file:///mnt/sdcard/temp"));
;
startActivityForResult(intent, 0);

//11.打开Google Market
// 打开Google Market直接进入该程序的详细页面
Uri uri = Uri.parse("market://details?id=" + "com.demo.app");
Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);

//12.进入手机设置界面:
// 进入无线网络设置界面（其它可以举一反三）
Intent intent = new Intent(android.provider.Settings.ACTION_WIRELESS_SETTINGS);
startActivityForResult(intent, 0);

//13.安装apk:
Uri installUri = Uri.fromParts("package", "xxx", null);
returnIt = new Intent(Intent.ACTION_PACKAGE_ADDED, installUri);

//14.卸载apk:
Uri uri = Uri.fromParts("package", strPackageName, null);
Intent it = new Intent(Intent.ACTION_DELETE, uri);
startActivity(it);

//15.发送附件:
Intent it = new Intent(Intent.ACTION_SEND);
```

```
it.putExtra(Intent.EXTRA_SUBJECT, "The email subject text");

it.putExtra(Intent.EXTRA_STREAM, "file:///sdcard/eoe.mp3");

sendIntent.setType("audio/mp3");
startActivity(Intent.createChooser(it, "Choose Email Client"));

//16.进入联系人页面：
Intent intent = new Intent();
intent.setAction(Intent.ACTION_VIEW);
intent.setData(People.CONTENT_URI);
startActivity(intent);

//17.查看指定联系人：
Uri personUri = ContentUris.withAppendedId(People.CONTENT_URI,
info.id); //info.id联系人ID
Intent intent = new Intent();
intent.setAction(Intent.ACTION_VIEW);
intent.setData(personUri);
startActivity(intent);
```

本节小结：

好吧，写着写着就不像入门教程了，哈哈，不过学多点没事的，本节初窥门径就到这里吧~下节我们会继续来研究这个Activity，比如数据传递，启动模式等~敬请期待~



^

QR Code

反馈

消息

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

- 意见反馈
- 免责声明
- 关于我们
- 文章归档

关注微信

算符



15012807号-1