

---

## Android—DialogActivity

在使用 Dialog 时，如果需要用到很多自己设计的控件，虽然可以让弹出框显示出我们需要的界面，但却无法找到地方完成控制代码的编写，如何解决这个问题呢，我们可以将 Activity 伪装成 Dialog 弹出框，这样既显示了界面，又可以在 Activity 里写控制代码。这就是 DialogActivity。

默认情况下，一个 activity 占据整个屏幕。然而，可以为一个 activity 设置一个“对话框”主题，这样，它就能像一个浮动的对话框显示出来了。举个例子，你可能想要把你的 activity 定义成一个弹出的“对话框”，通知用户进行一些操作。在这种情况下，显示一个“对话框”样式的 activity 是一个引起用户注意的好方法。

首先，问题的关键在 AndroidManifest.xml 中下面这句：

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

这就是 Activity 的 Dialog 风格。

**具体的步骤其实很简单：**

1. 继承 Activity
2. 设置 dialog 提示框的大小
3. AndroidManifest.xml 清单文件中注册该 DialogActivity，并设置其风格为：

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

**代码实现：**

**先创建一个 activity\_main.xml：**

```
*activity_main.xml dialogtext.xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >

    <Button
        android:id="@+id/myButton"
        android:text="点击可打开Dialog"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#99ccff"
        />

    <TextView
        android:id="@+id/showText"
        android:layout_width="fill_parent"
        android:layout_height="100dp"
        android:text="在这里显示dialog里输入的数字:"
        android:background="#ffffff"
        />

</LinearLayout>
```

再创建 dialogtext.xml :

```
*activity_main.xml *dialogtext.xml
<?xml version="1.0" encoding="utf-8"?>
<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/et"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        />

    <Button
        android:id="@+id/returnButton"
        android:text="确定"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        />

</LinearLayout>
```

这是DialogActivity的布局文件，这里可以设置提示框的大小等属性。

创建一个普通的 MainActivity.java :

```

public class MainActivity extends Activity {

    private Button myButton;
    private TextView showText;

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

        myButton = (Button)findViewById(R.id.myButton);
        showText = (TextView)findViewById(R.id.showText);

        myButton.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, TestDialog.class);
                startActivityForResult(intent, 0);
            }
        });
    }
    //利用返回Activity接收输入的数据并显示，证明我们的Dialog式的Activity确实可以完成数据的处理
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        //取出字符串
        Bundle bundle = data.getExtras();
        String str = bundle.getString("str");
        showText.setText(str);
    }
}

```

加载activity\_main布局文件

通过这个可以实现点击按钮时跳转到设置为DialogActivity的activity

再创建一个 TestDialog.java ,你可以看出这个 DialogActivity 和正常的 Activity 没什么区别，但它最后确实可以像 Dialog 一样弹出：

```

public class TestDialog extends Activity {

    private Button returnButton;
    private EditText inputEditor;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.dialogtext);

        returnButton = (Button)findViewById(R.id.returnButton);
        inputEditor = (EditText)findViewById(R.id.et);

        returnButton.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                String input = inputEditor.getText().toString();
                Intent intent = new Intent(TestDialog.this, MainActivity.class);
                Bundle bundle = new Bundle();
                bundle.putString("str", input);
                intent.putExtras(bundle);
                TestDialog.this.setResult(RESULT_OK, intent);
                TestDialog.this.finish();
            }
        });
    }
}

```

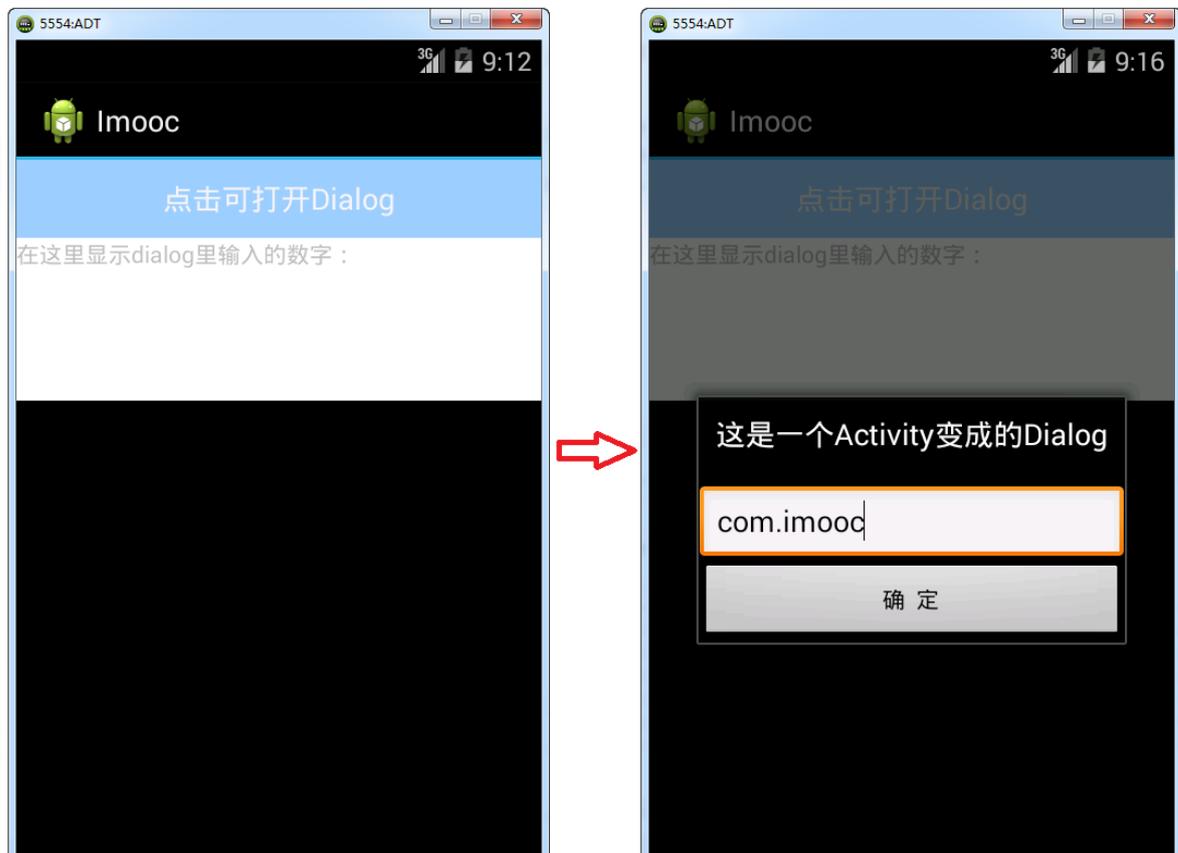
最后，最关键的一步，设置 Activity 的 Dialog 风格。在 AndroidManifest.xml 清单文

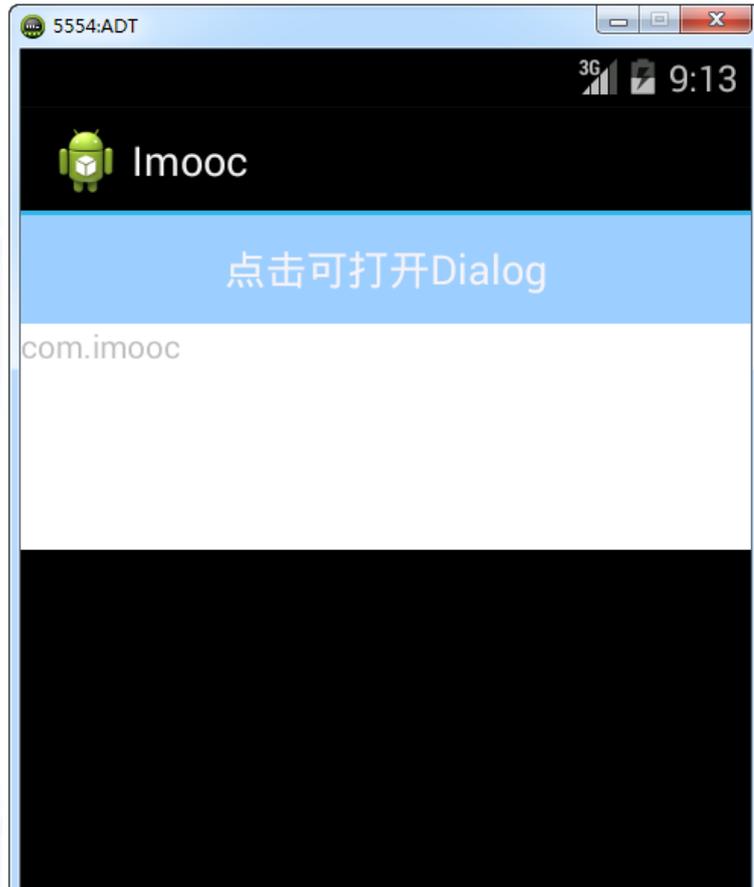
件中注册该 Activity

```
<activity
    android:name="com.example.dialog.MainActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
    android:name=".TestDialog"
    android:label="这是一个Activity变成的Dialog"
    android:theme="@android:style/Theme.Dialog" >
</activity>
```

效果演示：





慕课网  
imooc.com

慕课网  
imooc.com