适配器 Adapter

适配器可以为多个不相关的API提供一个公共的接口

Android的适配器负责为选择部件提供数据源，也负责将单独的数据元素转换为显示在选择部件中的特定视图。

一个简单的适配器——ArrayAdapter

ArrayAdapter接受3个参数

要使用的上下文（通常就是当前的activity的实例）

要使用的视图的资源ID

要实际显示的选项数组或列表

列表 ListView

显示数据的经典控件——ListView

ListView调用setAdapter()以提供数据和子视图

然后通过添加一个侦听器，以便在选项变化时作出反应

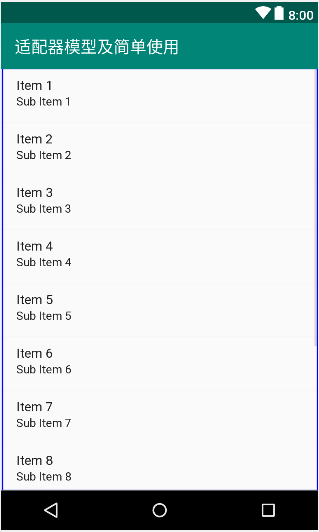
setOnItemClickListener()

**模型结构：**



activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"**>  
  
  
 <**ListView  
 android:id="@+id/listView1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="2dp"  
 android:layout\_marginTop="0dp"** />  
</**RelativeLayout**>



list\_item.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"**>  
  
 <**TextView  
 android:id="@+id/text1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="显示内容"** />  
</**LinearLayout**>



MainActivity.java

**package** com.amaker.hjs.adapterdemo;  
  
**import** android.app.Activity;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.AdapterView;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.ListView;  
**import** android.widget.Toast;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 *//数据源* **private** String[] **data**={**"上海"**,**"北京"**,**"广州"**,**"深圳"**,**"杭州"**,**"南京"**,**"成都"**,**"武汉"**,**"西安"**};  
 *//控件* **private** ListView **listView**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 *//关联控件* **listView** = (ListView)findViewById(R.id.***listView1***);  
 *//定义适配器（使用自己定义的子项目布局）  
// ArrayAdapter<String> adapter = new ArrayAdapter<String>(  
// this, //上下文 显示在自身的Activity里  
// R.layout.list\_item, //项布局 定义了ListView中每一项是以什么样的方式显示的布局文件  
// R.id.text1, //数据要显示的布局文件 的格式控件的id  
// data //添加数据源  
// );  
  
 //改成使用系统定义的子布局* ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(  
 **this**,  
 android.R.layout.***simple\_list\_item\_1***, *//前面的R是我工程文件的，这里的R是安卓系统定义的，所以必须用android.R.* android.R.id.***text1***,  
 **data** );  
 *//设置适配器* **listView**.setAdapter(adapter);  
  
 *//添加监听* **listView**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 */\*\*  
 \** ***@param parent*** *是子项item的父容器，也就是我们的ListView  
 \** ***@param view*** *是子项item  
 \** ***@param position*** *当前点击的子项item的索引，从0开始  
 \** ***@param id*** *当前点击的子项item的id  
 \*/* @Override  
 **public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {  
 *//这里是context是内部类方法的参数，所以必须加上MainActivity.this来指明是在上下文位于Activity里* Toast.*makeText*(MainActivity.**this**,**"你所选择的城市是："**+**data**[position],Toast.***LENGTH\_LONG***).show();  
  
 }  
 });  
 }  
}

