# [Android—— ListView 的简单用法及定制ListView界面](https://www.cnblogs.com/edensyd/p/8664239.html)

当程序中有大量的数据需要展示的时候，就可以借助ListView来实现。ListView允许用户通过手指上下滑动的方式将屏幕外的数据滚动到屏幕内。

案例(Example)1： ListView的简单用法

案例(Example)2: 定制ListView界面。

**一、案例(Example)1：ListView的简单用法**

目标：

1) 掌握 ListView 控件的使用

2) 掌握 Adapter 桥梁的作用

实现步骤：

1）首先在已有的项目中新建一个Model， 并让Android Studio自动帮我们创建好Activity。然后修改activity\_main.xml 中的代码，如下所示：

*<?***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"**>  
  
 <**ListView  
 android:id="@+id/listView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />  
  
</**LinearLayout**>

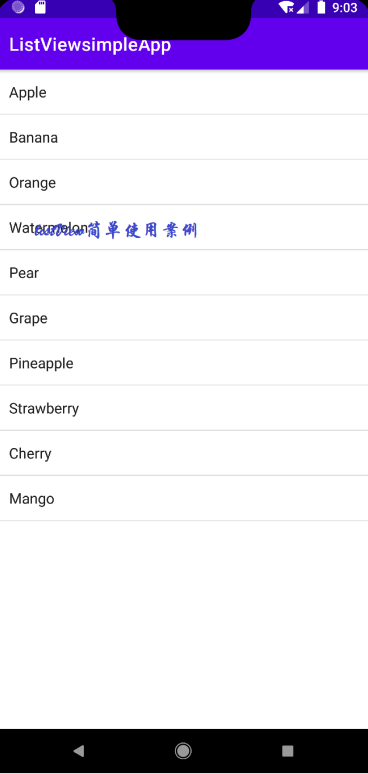
在布局中加入ListView 控件，并为ListView 指定了一个id， 设置成match\_parent 占满整个空间。

1. 在MainActivity 中

**public class** MainActivity **extends** Activity {  
 **private** String[] **data** = { **"Apple"**, **"Banana"**, **"Orange"**, **"Watermelon"**,  
 **"Pear"**, **"Grape"**, **"Pineapple"**, **"Strawberry"**, **"Cherry"**, **"Mango"** };  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(  
 MainActivity.**this**, android.R.layout.***simple\_list\_item\_1***, **data**);  
 ListView listView = (ListView) findViewById(R.id.**listView**);  
 listView.setAdapter(adapter);  
 }  
}

1. 在这里运用了系统包含的一个TextView的布局文件：android.R.layout.simple\_expandable\_list\_item\_1,调用这个比较方便，
2. ArrayAdapter<String> adapter = new ArrayAdapter<String>(  MainActivity.this, android.R.layout.simple\_list\_item\_1, data);  的意思是：创建一个数组适配器的代码，里面有三个参数，第一个参数是上下文，就是当前的Activity, 第二个参数是android sdk中内置的一个布局，它里面只有一个TextView，这个参数是表明我们数组中每一条数据的布局是这个view，也就是将每一条数据都显示在这个 view上面；第三个参数就是我们要显示的数据。listView会根据这三个参数，遍历data里面的每一条数据，读出一条，显示到第二个参数对应的布局中，这样就形成了我们看到的listView。
3. ArrayAdapter是BaseAdapter的子类。最后调用ListView的setAdapter方法，传入构建好的适配器对象。

3）运行效果图：



1. **案例(Example)2: 定制ListView界面**

1.目标：

1) 掌握 ListView 控件的使用

2) 掌握如何自定义 Adapter 的使用

2. 主要参考步骤及代码

1) 定义一个实体类 Fruit，作为 ListView 适配器的适配类型。

2) 为 ListView 的子项指定一个我们自定义的布局 fruit\_item.xml。

3) 创建一个自定义的适配器 FruitAdapter，这个适配器继承自 ArrayAdapter。重写构造方法和 getView 方法。

4) 在MainActivity中编写，初始化水果数据

1、定义一个实体类Fruit

**package** com.example.listviewapp;  
  
**public final class** Fruit {  
 **private final** String **name**;  
 **private final int imageId**;  
  
 **public** Fruit( String name, **int** imageId) {  
 **this**.**name** = name;  
 **this**.**imageId** = imageId;  
 }  
  
 **public final** String getName() {  
 **return this**.**name**;  
 }  
 **public final int** getImageId() {  
 **return this**.**imageId**;  
 }  
}

1. 为 ListView 的子项指定一个我们自定义的布局 fruit\_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"**>  
  
 <**ImageView  
 android:id="@+id/fruit\_image"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="20dp"** />  
  
 <**TextView  
 android:id="@+id/fruit\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="15dp"  
 android:gravity="center\_vertical"** />  
  
</**LinearLayout**>

1. 创建一个自定义的适配器 FruitAdapter，这个适配器继承自 ArrayAdapter。重写构造方法和 getView 方法。

**package** com.example.listviewapp;  
  
**import** android.content.Context;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
**import** androidx.annotation.NonNull;  
**import** androidx.annotation.Nullable;  
**import** java.util.List;  
**public class** FruitAdapter **extends** ArrayAdapter<Fruit> {  
 **private final int resourceId**;  
 **public** FruitAdapter(Context context, **int** textViewResourceId, List<Fruit> objects){  
 **super**(context,textViewResourceId,objects);  
 **resourceId**=textViewResourceId;  
 }  
 @NonNull  
 @Override  
 **public** View getView(**int** position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 *//每次滑动屏幕时，就会调用这个方法，创建新的实例并布置好布局返回  
 //convertView会缓存之前的布局* Fruit fruit=getItem(position);*//获取当前项的fruit实例* View view;  
 ViewHolder viewHolder;*//创建存储对象  
 //如果之前没有缓存过布局，就进行加载* **if**(convertView==**null**){  
 view=LayoutInflater.*from*(getContext()).inflate(**resourceId**,parent,**false**);*//给当前实例加载传入的布局* viewHolder=**new** ViewHolder();*//之前没有创建过，现在就创建一个存储对象* viewHolder.**fruitImage**=view.findViewById(R.id.***fruit\_image***);*//获取到ImageView实例* viewHolder.**fruitName**=view.findViewById(R.id.***fruit\_name***);*//获取到TextView实例* view.setTag(viewHolder);*//将viewHolder存储到view中，为了下次使用* }**else**{  
 view=convertView;  
 viewHolder= (ViewHolder) view.getTag();  
 }  
  
 viewHolder.**fruitImage**.setImageResource(fruit.getImageId());*//将当前实例的内容装入到实例当中* viewHolder.**fruitName**.setText(fruit.getName());*//同上* **return** view;*//将设定好的view布局传回去* }  
 **class** ViewHolder{  
 ImageView **fruitImage**;  
 TextView **fruitName**;  
 }  
}

上面代码中View view = LayoutInflater.from(getContext()).inflate(resourceId, null);使用Inflater对象来将布局文件解析成一个View

1. 在MainActivity中编写，初始化水果数据

**package** com.example.listviewapp;  
**import** androidx.appcompat.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;  
**import** java.util.ArrayList;  
**import** java.util.List;  
**public class** MainActivity **extends** AppCompatActivity {  
 **private** List<Fruit> **fruitList**=**new** ArrayList<Fruit>();  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 init();*//初始化数据* FruitAdapter fruitAdapter=**new** FruitAdapter(MainActivity.**this**,R.layout.***fruit\_item***,**fruitList**);*//创建适配器* ListView listView=findViewById(R.id.***list\_view***);*//获取ListView* listView.setAdapter(fruitAdapter);*//加载适配器* listView.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {  
 Fruit fruit=**fruitList**.get(position);*//从列表中取得点击的实例，通过position* Toast.*makeText*(MainActivity.**this**,fruit.getName(),Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 }  
 **private void** init(){  
 *//建立Fruit实例并装入fruitList中* **for**(**int** i=0;i<2;i++){  
 Fruit a=**new** Fruit(**"Apple"**, R.drawable.***apple\_pic***);  
 **fruitList**.add(a);  
 Fruit b=**new** Fruit(**"Banana"**, R.drawable.***banana\_pic***);  
 **fruitList**.add(b);  
 Fruit c=**new** Fruit(**"Orange"**, R.drawable.***orange\_pic***);  
 **fruitList**.add(c);  
 Fruit d=**new** Fruit(**"Watermelon"**, R.drawable.***watermelon\_pic***);  
 **fruitList**.add(d);  
 Fruit e=**new** Fruit(**"Pear"**, R.drawable.***pear\_pic***);  
 **fruitList**.add(e);  
 Fruit f=**new** Fruit(**"Grape"**, R.drawable.***grape\_pic***);  
 **fruitList**.add(f);  
 Fruit g=**new** Fruit(**"Pineapple"**, R.drawable.***pineapple\_pic***);  
 **fruitList**.add(g);  
 Fruit h=**new** Fruit(**"Strawberry"**, R.drawable.***strawberry\_pic***);  
 **fruitList**.add(h);  
 Fruit l=**new** Fruit(**"Cherry"**, R.drawable.***cherry\_pic***);  
 **fruitList**.add(l);  
 Fruit j=**new** Fruit(**"Mango"**, R.drawable.***mango\_pic***);  
 **fruitList**.add(j);  
 }  
 }  
}

1. 运行项目，效果如下：

