



# 《移动终端开发技术》 电子教案

## 第三单元 ListView 的功能和用法

所属专业（教研室）： 计算机软件技术

制定人： 陈媛媛

合作人：

制定时间： 2018年2月

日照职业技术学院



单元标题	ListView 的功能和用法	单元教学学时	4 课时
		在整体设计中的位置	第 7 次
授课班级		上课地点	一体化教室
上课时间	周 月 日第 节		
教学目标	能力目标	知识目标	素质目标
	能够熟练使用 ListView 制作简单大方的应用界面。	1、要求学生掌握 ListView 的使用方法； 2、要求学生掌握 ArrayAdapter 的使用方法； 3、要求学生掌握使用 ArrayAdapter 创建 ListView 的方法；	1、养成积极主动学习意识； 2、养成勤于动手的习惯。
教学重点、难点	<p>教学重点：使用 ArrayAdapter 创建 ListView 的方法</p> <p>教学难点：使用 ArrayAdapter 创建 ListView 的方法</p>		
教学方法	<p>采用反转课堂教学模式，课前学生学习微课了解知识点，课上采用教师引导、演示，学生分组练习、讨论等教学方法。</p> <p>运用多媒体、AndroidStudio 开发环境、实训助手、教学平台等辅助授课。</p>		
课前需掌握的知识点	<p>列表的显示需要三个元素：</p> <p>(1) ListView：用来展示列表的 View。</p> <p>(2) 适配器 (ArrayAdapter)：用来把数据映射到 ListView 上的中介。</p> <p>(3) 数据：具体的将被映射的字符串，图片，或者基本组件。</p>		
教学任务分解	<p>任务一 实现简单列表</p> <p>任务二 定制 ListView 界面</p> <p>任务三 实现 ListView 的点击事件</p>		
教学总结			



## 一、情景导入

我们知道手机的很多软件如 QQ 聊天记录、微信聊天记录及翻阅微博最新消息等都以列表的形式显示信息，并且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，同时屏幕上原有的数据则会滚动出屏幕。我们也希望做一个类似的界面，可以显示水果图片、水果信息，并且点击后将水果价格显示出来。并且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，同时屏幕上原有的数据则会滚动出屏幕。

## 二、知识讲解

列表的显示需要三个元素：

- (1) ListView 用来展示列表的 View。
- (2) 适配器 (ArrayAdapter) 用来把数据映射到 ListView 上的中介。
- (3) 数据 具体的将被映射的字符串，图片，或者基本组件。

学习资料，思考问题：

### 任务一 实现简单列表

#### 1、ListView 的用法

ListView 绝对可以称得上 Android 中最常用的控件之一，几乎所有的应用程序都会用到它。由于手机屏幕空间比较有限，能够一次性在屏幕上显示的内容并不多，当我们的程序中有大量的数据需要展示的时候，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，同时屏幕上原有的数据则会滚动出屏幕。

ListView 需要用到 Adapter 适配器，因为只有通过 Adapter 才可以把列表中的数据映射到 ListView 中。

- (1) 常用 xml 属性

android:divider//在列表条目之间显示的 drawable 或 color

android:dividerHeight//用来指定 divider 的高度

android:entries//构成 ListView 的数组资源的引用。对于某些固定的资源，这个属性提供了比在程序中添加资源更加简便的方式

android:footerDividersEnabled//当设为 false 时，ListView 将不会在各个 footer 之间绘制 divider.默认为 true。

android:headerDividersEnabled//当设为 false 时，ListView 将不会在各个 header 之间绘制 divider.默认为 true。

- (2) 常用方法：

思考 1、为 ListView 注册监听器：setOnItemClickListener()

```
如： listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> adapterView, View view, int i,  
long l) {  
        Fruit fruit=fruitList.get(i);
```



```
Toast.makeText(MainActivity.this,fruit.getName(),Toast.LENGTH_LONG).show();  
    }  
});
```

为 ListView 添加适配器: setAdapter()

如: FruitAdapter adapter=new

```
FruitAdapter(MainActivity.this,R.layout.fruit_item,fruitList);
```

```
    ListView listView=(ListView) findViewById(R.id.list_view);
```

```
    listView.setAdapter(adapter);
```

## 2、ArrayAdatper 的用法

Adapter 是将数据展示在 ListView (GridView) 中的一个接口。通过不同的 Adapter, 可以很方便地展示数据, 也可以实现各种自定义的效果了。

在 Android 日常的开发中, 常用的 Adapter 有如下三种:

- (1) ArrayAdapter
- (2) SimpleAdapter
- (3) BaseAdapter

思考 2、使用 ArrayAdapter 填充 ListView 步骤:

- (1) 定义一个数组来存放 ListView 中 item 的内容(数据源);
- (2) 通过实现 ArrayAdapter 的构造方法创建一个 ArrayAdapter 对象;
- (3) 通过 ListView 的 setAdapter(...)方法绑定 ArrayAdapter。

练习题:

1、以下哪个属性是构成 ListView 的数组资源的引用 ( )。

- A. android:divider //在列表条目之间显示的 drawable 或 color
- B. android:dividerHeight //用来指定 divider 的高度
- C. android:entries //构成 ListView 的数组资源的引用
- D. android:id

2、为 ListView 添加适配器的方法为 ( )。

- A. setOnItemClickListener() (为 ListView 注册监听器)
- B. setAdapter()
- C. findViewById()
- D. setEmptyView() (给 ListView 设置在空数据下显示的默认提示)

3、在 Android 日常的开发中, 常用的 Adapter 有 (ABC)。

- A. ArrayAdapter
- B. SimpleAdapter
- C. BaseAdapter
- D. HeaderViewListAdapter

4、ListView 的直接父类是 AbsListView (正确)。



5、ArrayAdapter 继承了 SimpleAdapter（错误）。继承 BaseAdapter

### 三、案例实现

#### 任务二 定制 ListView 界面、任务三 ListView 点击事件

- 1、主页面 Activity\_main.xml，一个 ListView 组件。
- 2、定义一个实体类 Fruit，作为 ListView 适配器的适配类型。
- 3、（每项有一个图片和文字说明）自定义布局 Fruit\_item.xml，需要 ImageView 和 TextView。
- 4、自定义适配器 FruitAdapter，继承 ArrayAdapter，泛型指定为 Fruit。
- 5、修改 MainActivity 中的代码，

#### Activity\_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="match_parent"
    android:layout_width="match_parent">
    <ListView
        android:id="@+id/list_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </ListView>
</LinearLayout>
```

#### Fruit\_item.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="match_parent"
    android:layout_width="match_parent">
    <ImageView
        android:id="@+id/fruit_image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:id="@+id/fruit_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:layout_marginLeft="10dp"/>
</LinearLayout>
```

#### Fruit.java

```
public class Fruit {
    private String name;
    private int imageId;
    public Fruit(String name, int imageId) {
```



```
        this.name=name;
        this.imageId=imageId;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getImageId() {
        return imageId;
    }
    public void setImageId(int imageId) {
        this.imageId = imageId;
    }
}
```

#### FruitAdapter.java

```
public class FruitAdapter extends ArrayAdapter {
    private int resourceId;

    public FruitAdapter(Context context, int resource, List<Fruit> objects) {
        super(context, resource, objects);
        resourceId=resource;
    }

    public View getView(int position, View convertView, ViewGroup parent){
        Fruit fruit= (Fruit) getItem(position);//获取当前项的Fruit实例
        View view =
        LayoutInflater.from(getContext()).inflate(resourceId, parent, false);
        ImageView fruitImage=(ImageView) view.findViewById(R.id.fruit_image);
        TextView fruitName=(TextView) view.findViewById(R.id.fruit_name);
        fruitImage.setImageResource(fruit.getImageId());
        fruitName.setText(fruit.getName());
        return view;
    }
}
```

#### MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private List<Fruit> fruitList=new ArrayList<>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

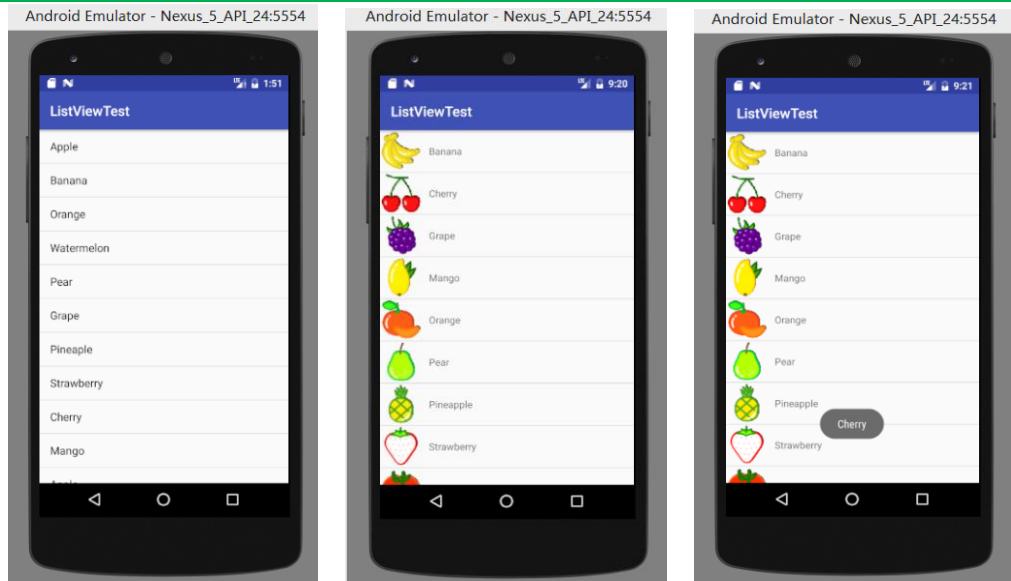


```
        initFruits();
        FruitAdapter adapter=new
FruitAdapter(MainActivity.this,R.layout.fruit_item,fruitList);
        ListView listView=(ListView) findViewById(R.id.list_view);
        listView.setAdapter(adapter);
        //为ListView注册监听器，效果见案例三，点击每个项会 Toast 将水果名字
显示出来。
        listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view,
int i, long l) {
                Fruit fruit=fruitList.get(i);

Toast.makeText(MainActivity.this,fruit.getName(),Toast.LENGTH_LONG).sho
w();

            }
        });
    }
    private void initFruits(){
        for(int i=0;i<2;i++){
            Fruit banana=new Fruit("Banana",R.drawable.banana1);
            fruitList.add(banana);
            Fruit cherry=new Fruit("Cherry",R.drawable.cherry1);
            fruitList.add(cherry);
            Fruit grape=new Fruit("Grape",R.drawable.grape1);
            fruitList.add(grape);
            Fruit mango=new Fruit("Mango",R.drawable.mango1);
            fruitList.add(mango);
            Fruit orange=new Fruit("Orange",R.drawable.orange1);
            fruitList.add(orange);
            Fruit pear=new Fruit("Pear",R.drawable.pear1);
            fruitList.add(pear);
            Fruit pineapple=new Fruit("Pineapple",R.drawable.pineapple1);
            fruitList.add(pineapple);
            Fruit strawberry=new Fruit("Strawberry",R.drawable.strawberry1);
            fruitList.add(strawberry);
            Fruit tomato=new Fruit("Tomato",R.drawable.tomato1);
            fruitList.add(tomato);

        }
    }
}
```



案例 1 效果图

案例 2 效果图

案例 3 效果图

```
public class FruitAdapter extends ArrayAdapter {
    private int resourceId;

    public FruitAdapter(Context context, int resource, List<Fruit> objects) {
        super(context, resource, objects);
        resourceId=resource;
    }

    public View getView(int position, View convertView, ViewGroup parent) {
        Fruit fruit= (Fruit) getItem(position); //获取当前项的 Fruit 实例
        //提升 ListView 的运行效率
        View view;
        ViewHolder viewHolder;
        if (convertView==null) {
            view=LayoutInflater.from(getContext()).inflate(resourceId, parent, false);
            viewHolder=new ViewHolder();
            viewHolder.fruitImage=(ImageView) view.findViewById(R.id.fruit_image);
            viewHolder.fruitName=(TextView) view.findViewById(R.id.fruit_name);
            view.setTag(viewHolder); //将 ViewHolder 存储在 View 中。
        }
        else {
            view=convertView;
            viewHolder=(ViewHolder) view.getTag(); //重新获取 ViewHolder
        }
        viewHolder.fruitImage.setImageResource(fruit.getImageId());
        viewHolder.fruitName.setText(fruit.getName());
        /*
        View                view                =
        LayoutInflater.from(getContext()).inflate(resourceId, parent, false);
        ImageView fruitImage=(ImageView) view.findViewById(R.id.fruit_image);
```





```
        TextView fruitName=(TextView) view.findViewById(R.id.fruit_name);
        fruitImage.setImageResource(fruit.getImageId());
        fruitName.setText(fruit.getName());*/
        return view;
    }
    class ViewHolder{
        ImageView fruitImage;
        TextView fruitName;
    }
}
```

#### 四、知识巩固

- 1、总结知识点，使用教学平台中的随堂练习题巩固本所学知识。
- 2、使用教学平台中的测试题给学生布置作业。

拓展 作业	制作 Android 手机应用市场界面
教学 后记	