

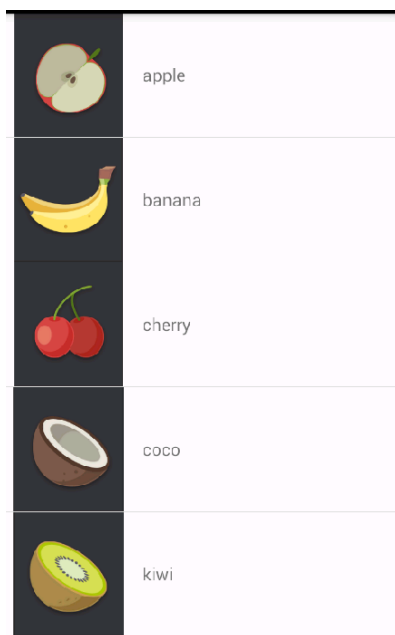
## Intent, Bundle 的使用及 ListView 的应用

### 【实验目的】

- (1) 复习 Button 点击事件处理
- (2) 了解 Intent, Bundle 机制，并且使用 Intent, Bundle 实现页面跳转
- (3) 了解 ListView 使用及其 Item 点击事件处理

### 【实验内容】

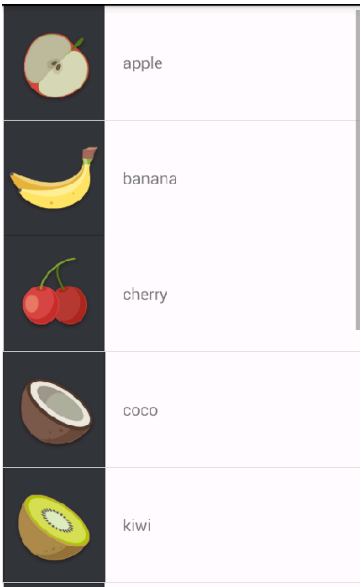
- (1) 进入主页面后，使用 ListView 实现如下页面：



- (2) 点击其中任何一项水果，跳转到另外一个活动，使用 Intent 转换活动，并且使用 Bundle 传递数据（必须使用 Bundle），跳转页面如下：



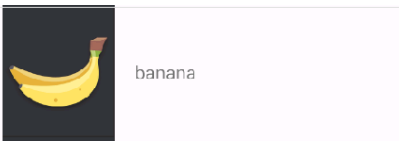
其中水果名为 Bundle 传递进来数据，点击 Back 按钮，返回原来的 Activity



【参考内容】

(1) **ListView** 是 **Android** 中最常用的控件之一，同时也是比较难用的控件。与前面学习的控件相比，**ListView** 用法相对复杂，因此能够使得手机程序界面丰富度和灵活性大大提高。**ListView** 以列表形式动态展示数据，其中列表中的项 (**item**) 不仅仅是文字，同时也可以图片和文字。因此，**ListView** 需要给列表中的每一项添加布局文件 (**xml**)，从而控制每一项展示方式。

(2) 本实验中需要用到布局文件 (**xml**) 对 **ListView** 每一项进行安排 (自己设计 **xml** 文件):



可以看出，每一项都有一张图片和该水果的名称，因此，考虑使用一个类，对 **Fruit** 进行封装 (传说中的面向对象，原来可以这样玩):

```

public class Fruit {
    private String name;
    private int imageId;

    public Fruit(String name, int imageId) {
        this.name = name;
        this.imageId = imageId;
    }

    public int getImageId() {
        return imageId;
    }

    public String getName() {
        return name;
    }
}

```

然后用线性表，将所有的 Fruit 对象储存起来：

```

private List<Fruit> fruitList = new ArrayList<Fruit>();

```

那么 imageId 是什么呢？在 ListView 中，每一项 (Item) 都需要图片和水果名称，原来 imageId 存储的是每一张图片的 id (android 为每一个文件都使用了一个唯一的 id，定义在 R 文件中)：

```

fruitList.add(new Fruit("apple", R.mipmap.apple));

```

然后怎么办？接下来需要创建一个自定义的适配器，继承自 ArrayAdapter，并且泛型指定为 Fruit 类，重写该类中的方法：

将控制 item 的 xml 的 id 传进去，方便我们设置每一项 item 的图片和名字：

```

FruitAdapter adapter = new FruitAdapter(MainActivity.this, R.layout.fruit_item, fruitList);

```

```

public class FruitAdapter extends ArrayAdapter<Fruit> {
    private int resourceId;

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

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        [...]
    }
}

```

其中，最关键的是重写 getView 函数，getView 函数代码如下：

```

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    Fruit fruit = getItem(position);

    View view = LayoutInflater.from(this.getContext()).inflate(resourceId, null);
    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;
}

```

接下来就是如何使得 `ListView` 的每一项具有相应能力：

```

fruits.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

```

可以看出是通过设置监听器实现的，并且重写里面的 `onItemClick` 函数

（3）在活动之间跳转，几乎是每一个 `android` 程序必备的，而活动之间的跳转是通过 `Intent` 实现的，`android` 组件之间传递数据，都是通过 `Intent` 来实现：

```

Intent intent = new Intent(MainActivity.this, SecondActivity.class);

startActivity(intent);

```

因此，需要新建一个 `Activity`，并且在 `AndroidManifest.xml` 进行注册，否则程序会崩溃（`android` 所有的组件只有注册后才能够使用）：

```

<activity
    android:name=".SecondActivity"
    android:label="@string/app_name" />

```

### 【课后作业】

（1）参考下面网址资料，了解 `Android` 活动不同状态生命周期：

<http://www.cnblogs.com/feisky/archive/2010/01/01/1637427.html>

请分别说明活动生存期在什么时候调用下列函数：

a) `onCreate()`

- b) onStart()
- c) onResume()
- d) onPause()
- e) onStop()
- f) onDestroy()
- g) onRestart()

(2) 活动的启动模式有四种，standard, singleTop, singleTask 以及 singleInstance，请参考网上资料，列表说明不同的启动方式有什么作用

(3) 为了让程序拥有更好的屏幕适配能力，仅仅使用 match\_parent 和 wrap\_content 无法满足我们的要求，这个时候就必须给控件宽和高指定一个固定值。简要说明以下单位区别：

- a) px 和 pt
- b) dp 和 sp

(4) 在 Activity 注册文件中，默认生成的 MainActivity 在注册文件中有如下代码：








```
<activity
    android:name=".MainActivity"
    android:label="@string/ThirdExperience">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
```

请说明<action>标签以及<category>标签中的 android:name 后面的字符串作用，为什么需要加这样的字符串？








```
android:name="android.intent.action.MAIN"
```

```
android:name="android.intent.category.LAUNCHER"
```

(5) 在原来实验基础上，添加功能，使得长按 item 后自动删除该 item，比如：

	apple
	banana
	cherry
	coco
	kiwi
	orange
	pear

长按 apple 后，该 item 被删除，如下图：

	banana
	cherry
	coco
	kiwi
	orange
	pear
	strawberry

提示：setOnItemClickListener

提交格式：

(1) 课堂实验代码: 学号\_姓名\_code0.zip

(2) 课后作业代码: 学号\_姓名\_code.zip

(3) 实验报告: 学号\_姓名\_reportX.pdf

(4) 将上述压缩成一个文件: 学号\_姓名\_expX\_vX.zip

(注意用英文小写，其中 x 为实验序号或者是提交序号，比如第一次实验就是 exp1，第一次提交就是 v1)

DDL: 周二晚上 12 点