ListView

1. 创建ListView 布局

```
<ListView
    android:id="@+id/list_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

2.给listview 每个 item 创建自定义对象

```
fruit.java
    private String text;
    private int imageId;
```

3. 创建自定义适配器（关键）

ViewHolder.java(存储每个item中的所有控件)

```
public class ViewHolder {
    public ImageView imageView;
    public TextView textView;

}
```

MyArrayAdapter.java

```
public class MyArrayAdapter extends ArrayAdapter<Fruit> {
    private int resourceId;   // 因为 要调用 inflate  方法，需要保存布局 item 的 id

    public MyArrayAdapter(@NonNull Context context, @LayoutRes int resource, @NonNull List<Fruit> objects) {
        super(context, resource, objects);
        resourceId = resource;
    }

//  position:
// The position of the item within the adapter's data set of the item whose view we want.

// convertView - The old view to reuse, if possible.
//  Note: You should check that this view is non-null and of an appropriate type before
//  using. If it is not possible to convert this view to display the correct data, this
//  method can create a new view.
// parent - The parent that this view will eventually be attached to
//加载每项时调用getView 函数

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        // 获取当前项对象，实际上调用 传入的集合的 get方法，或者 数组
        Fruit fruit = getItem(position);
        // 动态加载布局（每一项的布局id）,返回 view
        ViewHolder viewHolder;
        View view=null; // view实际上获得的是每个item对象，即 fruit.xml
        if(convertView==null){
            view= LayoutInflater.from(getContext()).inflate(resourceId,parent,false);
            viewHolder = new ViewHolder();
            viewHolder.imageView = (ImageView)view.findViewById(R.id.image);
            viewHolder.textView = (TextView)view.findViewById(R.id.text_view);
            view.setTag(viewHolder);  //将当前item 中的空间与 view 对象绑定，方便下次直接使用

        }else{
            view = convertView;
            viewHolder = (ViewHolder)view.getTag();
        }
        viewHolder.imageView.setImageResource(fruit.getImageId());
        viewHolder.textView.setText(fruit.getText());

        return view;
    }
}
```

4. 主活动

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //构建适配器对象
    initFruits();

    MyArrayAdapter adapter = new MyArrayAdapter(MainActivity.this,R.layout.fruit_item,list);
    ListView listView = (ListView)findViewById(R.id.list_view);
    listView.setAdapter(adapter);
    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
            Fruit fruit = list.get(position);
            Toast.makeText(MainActivity.this,fruit.getText()+position,Toast.LENGTH_SHORT).show();
        }
    });

}
```