# Adapter

Adapter：复杂数据展示和转换的载体。

不同Adapter的不同直接类和间接类。

常用的Adapter：

1，ArrayAdapter

支持泛型操作，最为简单，只能展示一行字。

2，SimpleAdapter

较好的扩充性，可自定义多种效果。

3，BaseAdapter

抽象类，继承需要实现多种方法，较高的灵活性。

4，SimpleCursorAdapter

可以适用于简单的纯文字型ListView，它需要Cursor的字段和UI的id对应起来。

如需要实现更复杂的UI也可以重写其他方法。可以认为是SimpleAdapter对数据库的简单结合，可以方便地把数据库的内容以列表的形式展示出来。

Adapter：

数据源

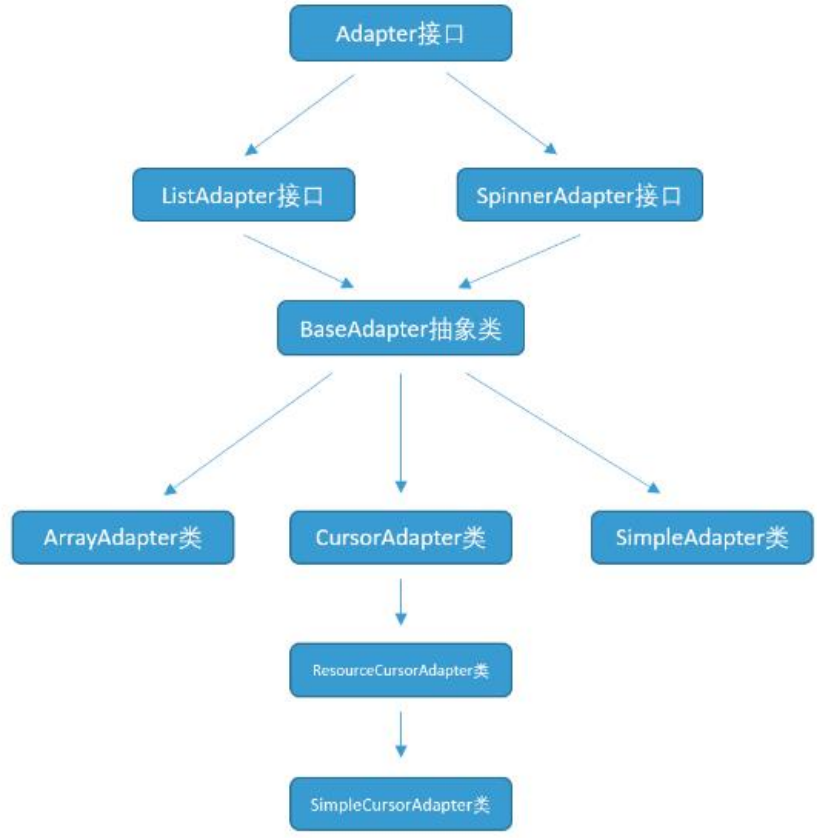
常见AdapterView子类：

1，ListView

2，GridView

3，Spinner

4，ExpandableListView等




Adapter接口定义的方法：

1，数据发生变化时，通知的AdapterView做出改变。--->观察者模式

2，返回数据数量。

3，根据索引取出数据。

4，获取指定数据项ID.。

5，数据源发生改变，判断原有数据ID是否发生变化。

6，getView方法。根据数据项索引，创建对应的UI。

ListAdapter接口：

ListAdapter可以作为AbsListView的数据源， AbsListView的子类有ListView、GridView和ExpandableListView。

ListAdapter相比Adapter新增了areAllItemsEnabled和isEnabled两个方法。

SpinnerAdapter接口：

SpinnerAdapter可以作为AbsSpinner的数据源， AbsSpinner的子类有Gallery, Spinner和AppCompatSpinner。

simpleCursorAdapter：

一定要以数据库作为数据源的时候,才能使用SimpleCursorAdapter，这里特别需要注意的一点是：不要忘了在

AndroidManifest.xml文件中加入权限

<uses-permission android:name="android.permission.READ_CONTACTS"></uses-permission>

BaseAdapter：

有时候，列表不光会用来做显示用，我们同样可以在在上面添加按钮。添加按钮首先要写一个有按钮的xml文件，然后自然会想到用上面的方法定义一个适配器，然后将数据映射到布局文件上。但是事实并非这样，因为按钮是无法映射的，即使你成功的用布局文件显示出了按钮也无法添加按钮的响应，这时就要研究一下ListView是如何现实的了，而且必须要重写一个类继承BaseAdapter。

---

---

理解：

1，ArrayAdapter显示一行数据：

a，在布局中添加listview（或者其他）。

b，创建adapter。

通过数组

通过list<>方法添加数据。

c，通过setAdapter添加即可。

2，simpleAdapter

可自定义视图显示方式，变化较为灵活。

a，创建视图布局样式。

b，在布局中添加listview（或者其他）。

c，创建simpleAdapter。

d，添加数据。list<map<,>>。HashMap

注意：

simpleAdapter的数据一般都是HashMap构成的list。

```
public class SimpleAdapterActivity extends ListActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        SimpleAdapter adapter = new SimpleAdapter(this, getData(), R.layout.simple, new String[] { "title",  "img"
        setListAdapter(adapter);
    }

    private List<Map <String , Object > > getData() {
        //map.put(参数名字,参数值)
        List<Map <String , Object > > list = new ArrayList<Map <String , Object > >();
        Map <String , Object >  map = new HashMap <String , Object > ();
        map.put("title", "摩托罗拉");
        map.put("img", R.drawable.icon);
        list.add(map);

        map = new HashMap <String , Object > ();
        map.put("title", "诺基亚");
        map.put("img", R.drawable.icon);
        list.add(map);

        map = new HashMap <String , Object > ();
        map.put("title", "三星");
        map.put("img", R.drawable.icon);
        list.add(map);
        return list;
        }

}
```