

- ViewHolder是用来保存视图引用的类，无论是ListView亦或是RecyclerView。只不过在ListView中，ViewHolder需要自己来定义，且这只是一种推荐的使用方式，不使用当然也可以，这不是必须的。只不过不使用ViewHolder的话，ListView每次getView的时候都会调用findViewById(int)，这将导致ListView性能展示迟缓。而在RecyclerView中使用RecyclerView.ViewHolder则变成了必须，尽管实现起来稍显复杂，但它却解决了ListView面临的上述不使用自定义ViewHolder时所面临的问题。
- 我们知道ListView只能在垂直方向上滚动，Android API没有提供ListView在水平方向上面滚动的支持。或许有多种方式实现水平滑动，但是请相信我，ListView并不是设计来做这件事情的。但是RecyclerView相较于ListView，在滚动上面的功能扩展了许多。它可以支持多种类型列表的展示要求，主要如下：
  1. LinearLayoutManager，可以支持水平和竖直方向上滚动的列表。
  2. StaggeredGridLayoutManager，可以支持交叉网格风格的列表，类似于瀑布流或者Pinterest。
  3. GridLayoutManager，支持网格展示，可以水平或者竖直滚动，如展示图片的画廊。
- 列表动画是一个全新的、拥有无限可能的维度。起初的Android API中，删除或添加item时，item是无法产生动画效果的。后面随着Android的进化，Google的Chat Hasse推荐使用ViewPropertyAnimator属性动画来实现上述需求。  
相比较于ListView，RecyclerView.ItemAnimator则被提供用于在RecyclerView添加、删除或移动item时处理动画效果。同时，如果你比较懒，不想自定义ItemAnimator，你还可以使用DefaultItemAnimator。
- ListView的Adapter中，getView是最重要的方法，它将视图跟position绑定起来，是所有神奇的事情发生的地方。同时我们也能够通过registerDataObserver在Adapter中注册一个观察者。RecyclerView也有这个特性，RecyclerView.AdapterDataObserver就是这个观察者。ListView有三个Adapter的默认实现，分别是ArrayAdapter、CursorAdapter和SimpleCursorAdapter。然而，RecyclerView的Adapter则拥有除了内置的内DB游标和ArrayList的支持之外的所有功能。RecyclerView.Adapter的实现的，我们必须采取措施将数据提供给Adapter，正如BaseAdapter对ListView所做的那样。
- 在ListView中如果我们想要在item之间添加间隔符，我们只需要在布局文件中对ListView添加如下属性即可：

```
android:divider="@android:color/transparent"
android:dividerHeight="5dp"
```

- ListView通过AdapterView.OnItemClickListener接口来探测点击事件。而RecyclerView则通过RecyclerView.OnItemTouchListener接口来探测触摸事件。它虽然增加了实现的难度，但是却给予开发人员拦截触摸事件更多的控制权限。
- ListView可以设置选择模式，并添加MultiChoiceModeListener，如下所示：

```
listView.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE_MODAL);
listView.setMultiChoiceModeListener(new MultiChoiceModeListener() {
    public boolean onCreateActionMode(ActionMode mode, Menu menu) { ... }
    public void onItemCheckedStateChanged(ActionMode mode, int position,
        long id, boolean checked) { ... }
        public boolean onActionItemClicked(ActionMode mode, MenuItem item) {
            switch (item.getItemId()) {
                case R.id.menu_item_delete_crime:
                    CrimeAdapter adapter = (CrimeAdapter)getListAdapter();
                    CrimeLab crimeLab = CrimeLab.get(getActivity());
```

```
        for (int i = adapter.getCount() - 1; i >= 0; i--) {
            if (getListView().isItemChecked(i)) {
                crimeLab.deleteCrime(adapter.getItem(i));
            }
        }
        mode.finish();
        adapter.notifyDataSetChanged();
        return true;
    default:
        return false;
}

public boolean onPrepareActionMode(ActionMode mode, Menu menu) { ... }
public void onDestroyActionMode(ActionMode mode) { ... }
});
```

而RecyclerView则没有此功能。