一、DrawerLayout 布局，可以直接放入两个子控件，

二、一个主屏、

1. 头部：toolbar
2. 中间：RecyclerView占满；

你想要控制其显示的方式，请通过布局管理器LayoutManager

你想要控制Item间的间隔（可绘制），请通过ItemDecoration

你想要控制Item增删的动画，请通过ItemAnimator

你想要控制点击、长按事件，请自己写

在开发RecyclerView时充分考虑了扩展性，因此用它可以创建想到的任何种类的的布局。但在使用上也稍微有些不便，比如使用步骤更加复杂，特别是一些控制点击、长压事件需要自己完成。使用RecyclerView开发的项目结构大致如下图所示：要使用RecyclerView，需要先了解清楚LayoutManager和Adapter元素，分别如下：

LayoutManager：用来确定每一个item如何进行排列摆放，何时展示和隐藏。回收或重用一个View的时候，LayoutManager会向适配器请求新的数据来替换旧的数据，这种机制避免了创建过多的View和频繁的调用findViewById方法。目前RecyclerView库提供了如下三种子Manager：

LinearLayoutManager：展示了水平或者垂直的滚动列表，相当于之前学习的ListView，但是没有页眉和页尾。

GridLayoutManager：在网格中展示条目，相当于之前学习的GridView。

StaggeredGridLayoutManager： 在错落的网格中展示条目，比如常见的瀑布流。

Adapter：这是一种新型适配器，不同于之前使用的BaseAdapter了。在使用RecyclerView之前，需要自定义一个继承自RecyclerView.Adapter的适配器，将数据与每一个item的界面进行绑定。需要注意的是编写Adapter面向的是ViewHolder而不在是View了，复用的逻辑被封装了起来，实现更加简单。使用时需要重写以下两个主要方法：

除了上面两个主要元素，通常还会使用到如下三个类：

ViewHolder：维持了所有被数据填充的实体的视图的引用。

ItemDecoration：一个实体的周围的装饰。

ItemAnimator：条目增加删除时重新排序所产生动画。

* viewholder内部类，布局中控件；
* 数据：list<>;
* onCreateViewHolder：用来展现视图和它的持有者。
* onBindViewHolder：主要用来把数据绑定到视图上。
* 数据个数，getItemCount

**自己的感悟：adapter是这样安排的，1、定义一个对象，new viewhoder;2，导入要显示的数据：list<>;**

* **3、viewholder安排显示在xml 表达的布局中；**onCreateViewHolder：用来展现视图和它的持有者。监听器在这里准备；

Android加载View一般有两种方式，第一种方式就是在Activity中直接使用setContentView()方法来加载，第二种方式则是使用LayoutInflater的inflate方法来映射View。

View view = LayoutInflater.*from*(**mContext**).inflate(R.layout.***fruit\_item***, parent, **false**)

**4、传入2的数据，一条：list.get(position);** onBindViewHolder：主要用来把数据绑定到视图上

**5、定义传入数据的多少：getItemCount**

RecyclerView是一个大框，里面排的是ViewHolder,Viewholder 怎么排，由调用的LayoutManager决定。

1. 左下（bottom|end）FloatingActionButton;
2. 下面：Snackbar,3的点击事件引出

三、一个滑动菜单（）

**mDrawerLayout**.openDrawer(GravityCompat.START);打开

**mDrawerLayout**.closeDrawers();关闭