

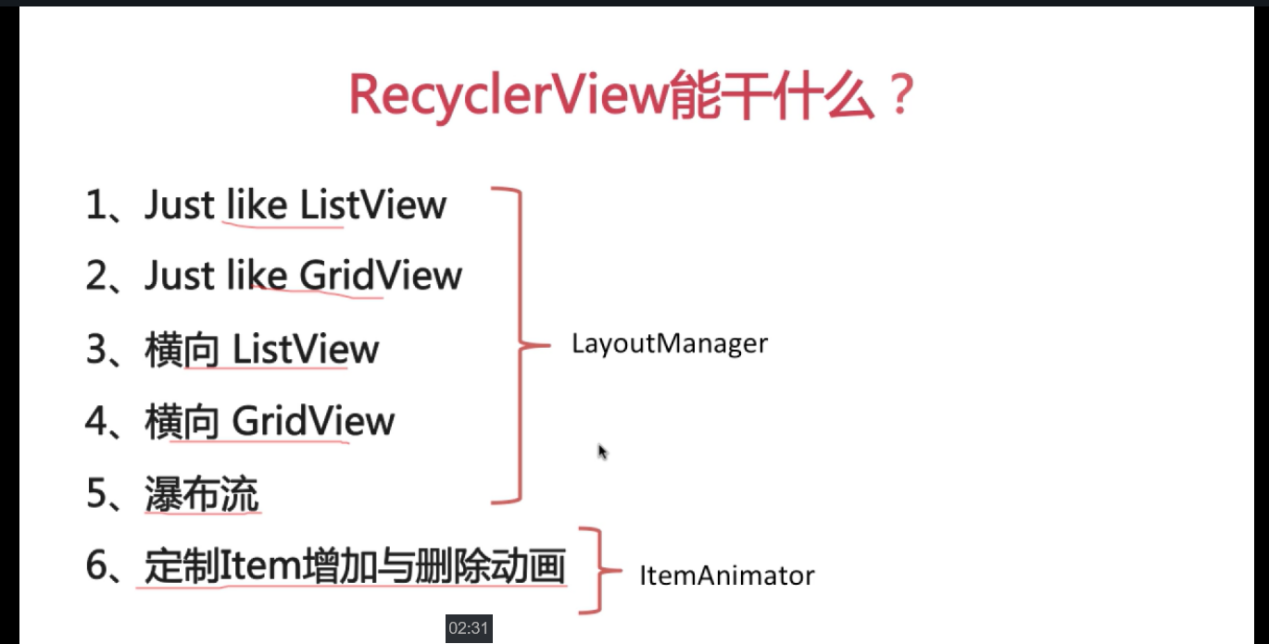
引入LayoutManager来控制RecyclerView显示风格。

通过ItemDecoration来控制分隔

通过ItemAnimator来实现Item增加与删除的动画效果

RecyclerView实际上是一个插件式的控件。





RecyclerView显示效果完全由LayoutManager来决定

瀑布流的实现：

**private** List<Integer> **mHeight**;

**mHeight** = **new** ArrayList<>();  
**for** (**int** i=0;i<mDatas.size();i++){  
 **mHeight**.add((**int**) (100+Math.*random*()\*300));  
}

@Override  
**public void** onBindViewHolder(**final** StaggeredAdapter.MyViewHolder holder, **int** position) {  
 ViewGroup.LayoutParams pl = holder.**itemView**.getLayoutParams();  
 pl.**height** = **mHeight**.get(position);  
 holder.**itemView**.setLayoutParams(pl);

设置高为随机数，并在onBindViewHolder()方法中为item的高赋值

RecyclrView的点击事件：通过回调实现。

具体步骤：

1）编写接口：

**public interface** OnItemClickListener {  
 **public void** OnItemClick(View v,**int** position);  
 **public void** OnItemLongClick(View v , **int** position);  
}

1. 在adapter中实例化接口

OnItemClickListener **onItemClickListener**;

**public void** setOnItemClickListener(OnItemClickListener listener){  
 **onItemClickListener** = listener;  
}

1. 在onBindViewHolder方法中设置点击事件

holder.**itemView**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (**onItemClickListener**!=**null**){  
 **onItemClickListener**.OnItemClick(v,holder.getLayoutPosition());  
 }  
 }  
});  
holder.**itemView**.setOnLongClickListener(**new** View.OnLongClickListener() {  
 @Override  
 **public boolean** onLongClick(View v) {  
 **if** (**onItemClickListener**!=**null**){  
 **onItemClickListener**.OnItemLongClick(v,holder.getLayoutPosition());  
 }  
 **return false**;  
 }  
});

1. **在Activity中实现接口的任务**

删除和添加数据：

**public void** addData(**int** pos){  
 **mDatas**.add(pos,**"insert one"**);  
 *//notifyDataSetChanged();* notifyItemInserted(pos);  
}  
  
**public void** deleteData(**int** pos){  
 **mDatas**.remove(pos);  
 *//notifyDataSetChanged(); //没有动画效果* notifyItemRemoved(pos);  
}

RecyclerView总结：

LayoutManager实现ListView、GridView、横向ListView/GridView、瀑布流

ItemDecoration实现的原理和如何定制

ItemAnimator的使用（系统提供默认的DefaultAnimator）

手动在Adapter中提供OnItemClick和OnItemLongClick