1)

1.ViewHolder Pattern

In a ListView, it was recommended to use the ViewHolder pattern but it was never a compulsion. In case of RecyclerView, this is mandatory using the RecyclerView.ViewHolder class.

2.LayoutManager

In a ListView, the only type of view available is the vertical ListView. In RecyclerView, we can have

i) LinearLayoutManager

ii) StaggeredLayoutManager

iii) GridLayoutManager

3.Item Animator

RecyclerView uses the ItemAnimator class to make animation easier

4.Item Decorator

The RecyclerView.ItemDecorator class is used.

5.OnItemTouchListener

RecyclerView.OnItemTouchListener is more complicated than AdapterView.OnItemClickListener interface for ListView, but give more control and functionality

2)

Lazy loading is a design pattern is used to make the initialization of an object happens at the point at which it is needed.

3)

An ItemDecoration allows the application to add a special drawing and layout offset to specific item views from the adapter's data set. This can be useful for drawing dividers between items, highlights, visual grouping boundaries and more.

4)

The ViewHolder design pattern enables you to access each list item view without the need for the look up, saving valuable processor cycles. Specifically, it avoids frequent call of findViewById() during ListView scrolling, and that will make it smooth.

5)

You will sometimes need to override onMove(RecyclerView, ViewHolder, ViewHolder) and / or onSwiped(ViewHolder, int), depend on what you want.

Also, this class is designed to work with any LayoutManager but for certain situations, it can be optimized for your custom LayoutManager by extending methods in the ItemTouchHelper.Callback class or implementing ItemTouchHelper.ViewDropHandler interface in your LayoutManager.

By default, ItemTouchHelper moves the items' translateX/Y properties to reposition them. You can customize these behaviors by overriding onChildDraw(Canvas, RecyclerView, ViewHolder, float, float, int, boolean) or onChildDrawOver(Canvas, RecyclerView, ViewHolder, float, float, int, boolean). You will need to override these most of the time.