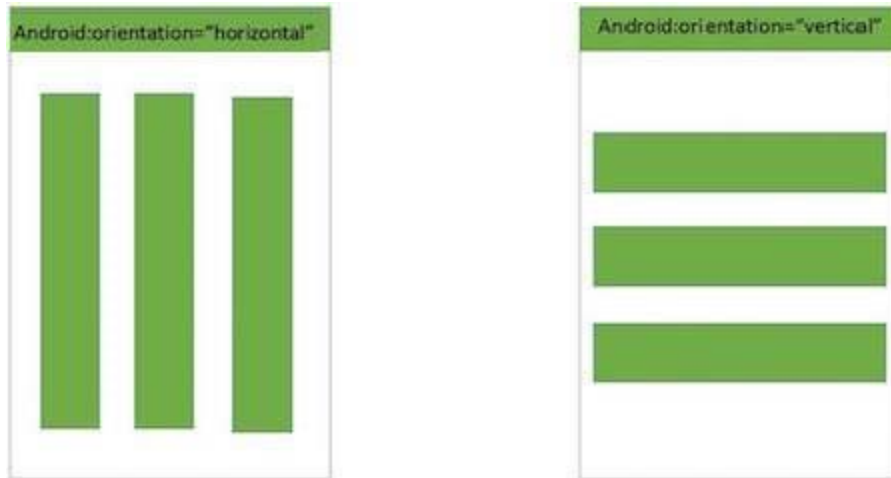


# Layouts

## Android Linear Layout

Android LinearLayout is a view group that aligns all children in either vertically or horizontally.



Linear Layout

### LinearLayout Attributes

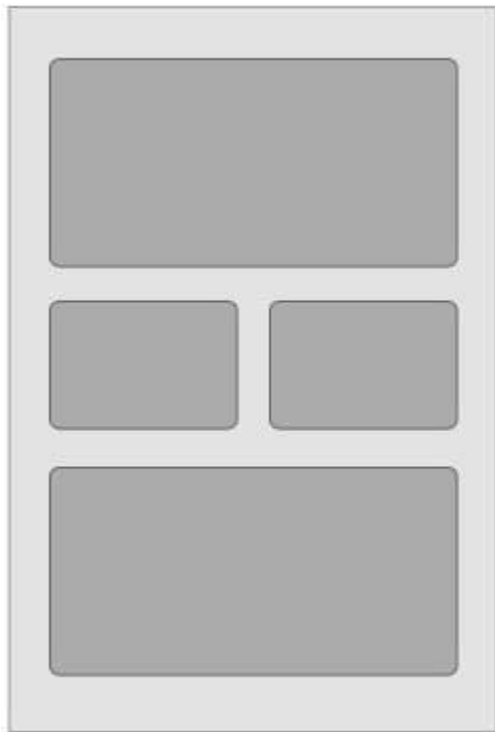
Following are the important attributes specific to LinearLayout –

Sr.No	Attribute & Description
1	<b>android:id</b> This is the ID which uniquely identifies the layout.
2	<b>android:baselineAligned</b> This must be a boolean value, either "true" or "false" and prevents the layout from aligning its children's baselines.
3	<b>android:baselineAlignedChildIndex</b> When a linear layout is part of another layout that is baseline aligned, it can specify which of its children to baseline align.
4	<b>android:divider</b>

	This is drawable to use as a vertical divider between buttons. You use a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".
5	<p>android:gravity</p> <p>This specifies how an object should position its content, on both the X and Y axes. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc.</p>
6	<p>android:orientation</p> <p>This specifies the direction of arrangement and you will use "horizontal" for a row, "vertical" for a column. The default is horizontal.</p>
7	<p>android:weightSum</p> <p>Sum up of child weight</p>

### Android Relative Layout

Android RelativeLayout enables you to specify how child views are positioned relative to each other. The position of each view can be specified as relative to sibling elements or relative to the parent.



## Relative Layout

### RelativeLayout Attributes

Following are the important attributes specific to RelativeLayout –

Sr.No.	Attribute & Description
1	<code>android:id</code> This is the ID which uniquely identifies the layout.
2	<code>android:gravity</code> This specifies how an object should position its content, on both the X and Y axes. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc.
3	<code>android:ignoreGravity</code> This indicates what view should not be affected by gravity.

Using RelativeLayout, you can align two elements by right border, or make one below another, centered in the screen, centered left, and so on. By default, all child views are drawn at the top-left of the layout, so you must define the position of each view using the various layout properties available from RelativeLayout.LayoutParams and few of the important attributes are given below –

Sr.No.	Attribute & Description
1	<code>android:layout_above</code> Positions the bottom edge of this view above the given anchor view ID and must be a reference to another resource, in the form " <code>@+[package:]type:name</code> "
2	<code>android:layout_alignBottom</code> Makes the bottom edge of this view match the bottom edge of the given anchor view ID and must be a reference to another resource, in the form " <code>@+[package:]type:name</code> ".
3	<code>android:layout_alignLeft</code>

	<p>Makes the left edge of this view match the left edge of the given anchor view ID and must be a reference to another resource, in the form "@[+][package:]type:name".</p>
4	<p><code>android:layout_alignParentBottom</code></p> <p>If true, makes the bottom edge of this view match the bottom edge of the parent. Must be a boolean value, either "true" or "false".</p>
5	<p><code>android:layout_alignParentEnd</code></p> <p>If true, makes the end edge of this view match the end edge of the parent. Must be a boolean value, either "true" or "false".</p>
6	<p><code>android:layout_alignParentLeft</code></p> <p>If true, makes the left edge of this view match the left edge of the parent. Must be a boolean value, either "true" or "false".</p>
7	<p><code>android:layout_alignParentRight</code></p> <p>If true, makes the right edge of this view match the right edge of the parent. Must be a boolean value, either "true" or "false".</p>
8	<p><code>android:layout_alignParentStart</code></p> <p>If true, makes the start edge of this view match the start edge of the parent. Must be a boolean value, either "true" or "false".</p>
9	<p><code>android:layout_alignParentTop</code></p> <p>If true, makes the top edge of this view match the top edge of the parent. Must be a boolean value, either "true" or "false".</p>
10	<p><code>android:layout_alignRight</code></p> <p>Makes the right edge of this view match the right edge of the given anchor view ID and must be a reference to another resource, in the form "@[+][package:]type:name".</p>
11	<p><code>android:layout_alignStart</code></p> <p>Makes the start edge of this view match the start edge of the given anchor view ID and must be a reference to another resource, in the form "@[+][package:]type:name".</p>

12	<p><code>android:layout_alignTop</code></p> <p>Makes the top edge of this view match the top edge of the given anchor view ID and must be a reference to another resource, in the form "<code>@+[package:]type:name</code>".</p>
13	<p><code>android:layout_below</code></p> <p>Positions the top edge of this view below the given anchor view ID and must be a reference to another resource, in the form "<code>@+[package:]type:name</code>".</p>
14	<p><code>android:layout_centerHorizontal</code></p> <p>If true, centers this child horizontally within its parent. Must be a boolean value, either "true" or "false".</p>
15	<p><code>android:layout_centerInParent</code></p> <p>If true, centers this child horizontally and vertically within its parent. Must be a boolean value, either "true" or "false".</p>
16	<p><code>android:layout_centerVertical</code></p> <p>If true, centers this child vertically within its parent. Must be a boolean value, either "true" or "false".</p>
17	<p><code>android:layout_toEndOf</code></p> <p>Positions the start edge of this view to the end of the given anchor view ID and must be a reference to another resource, in the form "<code>@+[package:]type:name</code>".</p>
18	<p><code>android:layout_toLeftOf</code></p> <p>Positions the right edge of this view to the left of the given anchor view ID and must be a reference to another resource, in the form "<code>@+[package:]type:name</code>".</p>
19	<p><code>android:layout_toRightOf</code></p> <p>Positions the left edge of this view to the right of the given anchor view ID and must be a reference to another resource, in the form "<code>@+[package:]type:name</code>".</p>
20	<p><code>android:layout_toStartOf</code></p>

	Positions the end edge of this view to the start of the given anchor view ID and must be a reference to another resource, in the form "@[+][package:]type:name".
--	--