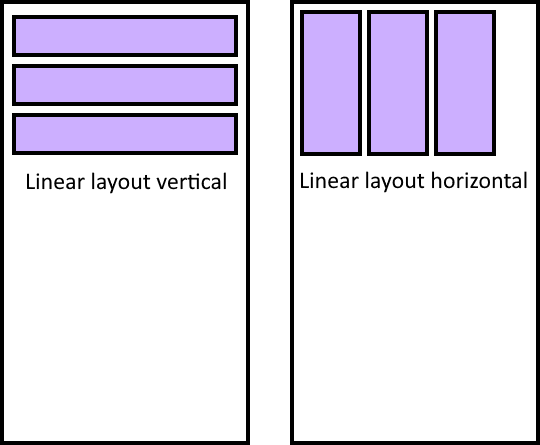
Linear Layout

LinearLayout is a view group that aligns all children in a single direction, vertically or horizontally.

All children of a LinearLayout are stacked one after the other, so a vertical list will only have one child per row, no matter how wide they are, and a horizontal list will only be one row high (the height of the tallest child, plus padding).



Downsides:

* No more than one element per row/column
* Limited to one direction

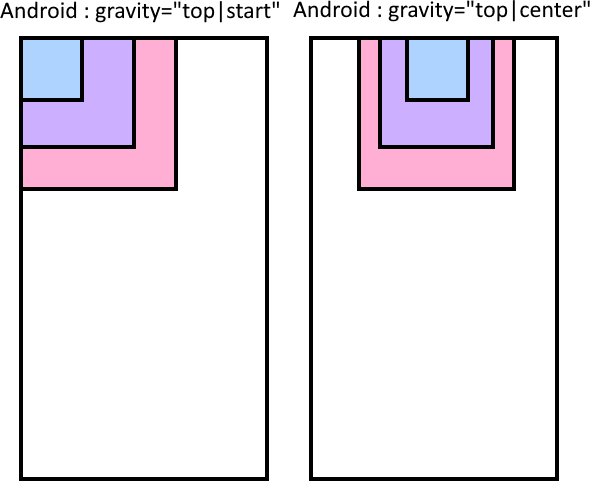
Upsides:

* Easy to manage order of children

Frame Layout

FrameLayout is a view group used stack child views on top of each other, with the most recent child on top of the stack. You can add multiple children to a FrameLayout and control their position within the FrameLayout by assigning gravity to each

The size of the FrameLayout is the size of its largest child and its padding.



Downsides:

* Little control on position of elements
* Hard to control order of elements

Upsides:

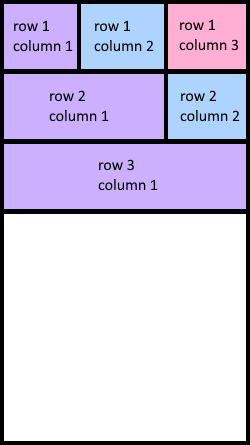
* The ability to overlay elements over each other

Table Layout

TableLayout positions its children into rows and columns, the border lines of which are not displayed. The table will have as many columns as the row with the most cells.

A table can leave cells empty and cells can span multiple columns, as they can in HTML,using using the span field in the TableRow.LayoutParams class.

TableRow objects are the child views of a TableLayout, that define the rows, they can have 0 or more cels created by adding children to them.



Downsides:

* Location of elements is stuck to a grid made up by rows and columns

Upsides:

* Easy to manage location and order of elements