Layouts :

* A layout defines the visual structure for a user interface, such as the UI for an [activity](https://developer.android.com/guide/components/activities.html) or [app widget](https://developer.android.com/guide/topics/appwidgets/index.html).

or

* An Android layout is a class that handles arranging the way its children appear on the screen.  Anything that is a View can be a child of a layout.
* All of the layouts will inherit from ViewGroup

Different types of layout :

1. Absolute Layout

2. Frame Layout

3. Linear Layout

4.Relative Layout

5.Table Layout

6. ListView

7. GridView

1. Absolute Layout :

* AbsoluteLayout is based on the simple idea of placing each control at an absolute position.
* An Absolute Layout lets you specify exact locations based on (x/y coordinates) of its children.
* Absolute layout is deprecated in android studio.

Attribute :

1. android:layout\_x :- This attribute specifies the x-coordinate of the view.

2. android:layout\_y :- This attributr specifies the y-coordinate of the view.

2. Frame Layout :

* Frame Layout is designed to block out an area on the screen to display a single item.

3. Linear Layout :

* [LinearLayout](http://developer.android.com/reference/android/widget/LinearLayout.html)  arranges the “component” in vertical or horizontal order.
* The attribute used to set the component is android : orientation
* The orientation can be in vertical or horizontal View.

4. Relative Layout :

* [RelativeLayout](https://developer.android.com/reference/android/widget/RelativeLayout.html) is a view group that displays child views in relative positions.
* A [RelativeLayout](https://developer.android.com/reference/android/widget/RelativeLayout.html) is a very powerful utility for designing a user interface because it can eliminate nested view groups and keep your layout hierarchy flat, which improves performance.
* The RelativeLayout is very flexible and user friendly.
* We can use “**above**, **below**, **left** and **right**” to arrange the component position in layout.

5. Table Layout :

* TableLayout will arrange groups of views into rows and columns.
* Each row has zero or more number of cells
* A table can leave a cells empty.
* We will use the <TableRow> element to build a row in the table.

6. List View :

* [ListView](https://developer.android.com/reference/android/widget/ListView.html) is a view group that displays a list of scrollable items.
* The list items are automatically inserted to the list using an [Adapter](https://developer.android.com/reference/android/widget/Adapter.html) that pulls content from a source such as an array or database query and converts each item result into a view that's placed into the list.
* An adapter actually bridges between UI components and the data source that fill data into UI Component.

7. Grid View :

* [GridView](https://developer.android.com/reference/android/widget/GridView.html) is a [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup.html) that displays items in a two-dimensional, scrollable grid.
* The grid items are automatically inserted to the layout using a [ListAdapter](https://developer.android.com/reference/android/widget/ListAdapter.html).
* This layout can be used to build applications like image viewer, audio or video players in order to show elements in grid manner.

Note :

The ListView and GridView are subclasses of AdapterView and they can be populated by binding them to an Adapter, which retrieves data from an external source and creates a View that represents each data entry.