**VIEW**

->A View is an object that draws something on the screen that the user can interact with.

It is responsible for measuring , layouting and drawings themselves and their child elements.

->There are number of Views (UI controls)provided by Android that allow us to build the graphical user interface for our app.

\*\*\*\*\*\*Some are:--

1)TextView:- TextView displays text to the user and allows them to edit programatically.

<**TextView**

**android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TEXT VIEW"  
 android:textSize="30dp"  
 android:textColor="#ffff"**/>

2)EditText:- This is a predefined subclass of TextView that permits rich editing capabilities.

<**EditText  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="hello!!"  
 android:hint="Type your name here"  
 android:textSize="10dp"**/>

3)Button:- Button represents a push button. A Push Button can be clicked or pressed by the user to perform an action.

There are different types of buttons used in android such as ImageButton, toggleButton, radiobutton.

<**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"   
 android:text="CLICK"**/>

4)CheckBox:- An on or off switch that can be toggled by the user and can be used to select multiple options from a group of selectable options.

<**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="MALE"**/>

<**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="FEMALE"**/>

**VIEWGROUP**

ViewGroup is an object which is responsible for arranging the components(View) in different manner.

It is a base class of some special UI class that can contain other view objects as children.

Android contains the following commonly used Viewgroup subclasses:-

1)LinearLayout:- Arrange components in Linear Form.

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"**

**android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 </**LinearLayout**>

2)Relative Layout:- Arrange the views w.r.t to its parent or its neighbours.

<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 </**RelativeLayout**>

3)Table Layout:- Arrange the views in tabular form.

<**TableLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 </**TableLayout**>

4)Grid Layout:- It is similar to table layout but here we can define number of rows

and number of columns by the Attribute rowcount and columnCount.

<**GridLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:columnCount="2"  
 android:rowCount="5"**>  
  
 </**GridLayout**>