Views & View Groups

An activity Contains views and view Groups. A view is a widget that has an appearance on Screen and Can be one of many subclasses. Such as Button or text view. The viewgroups objects are usually Called "Layouts".

~~A view~~

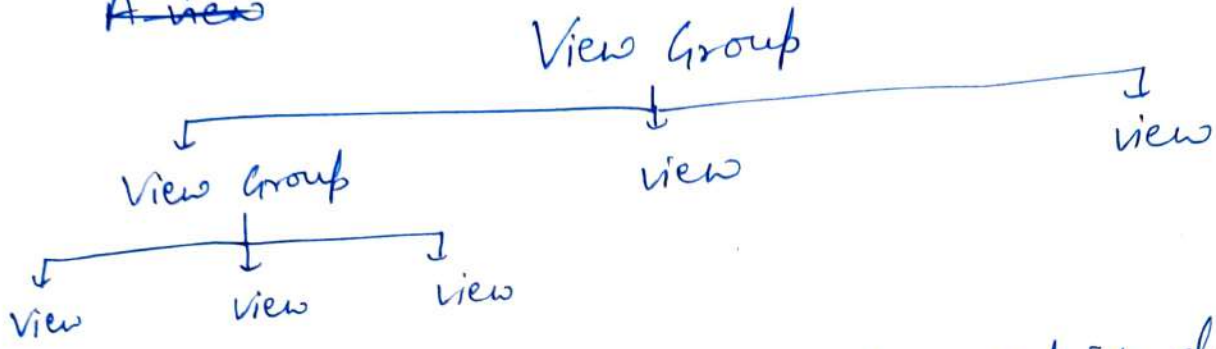


Illustration of a view hierarchy, which defines a UI (User interface) layouts.

One or more view Can be grouped together into a view Group. A view Group derives from the base class android.view.viewGroup.

Android Supports the following view Groups:—

i) Linear layout
ii) Absolute Layout
iii) Table layout
iv) Relative layout
v) Frame layout
vi) Scroll view

i) Linear layout :→ A layout that Organizes its children into a singl horizonal or Vertical row. It Creats a Scrollbar if the length of the window exceeds the length of the window exceeds the length of the screen.

```
<?xml version="1.0" encoding="UTF-8"?>
<Linear layout
  xmlns:android = "http://schemas.android.com/apk/
                   res/android"
  android:orientation = "Vertical"
  android:layout_width = "Fill_Parents"
  android:layout_height = "Fill_Parents"
                                    >
  <Text view
    android:layout_width = "Fill_Parent"
    android:layout_height = "Wrap_Content"
    android:text = "@string/Hello"
    />
</Linear layout>
```

Each view and view Group has a Set of Common attributes some of which are described below :—

Layout_width — Specifies the width of the view or view Group

Layout_height — Specifies the height of the view or view Group

Layout-marginTop - Specifies extra space on the top side of ..

Layout_marginBottom - Specifies extra space on the bottom ..

Layout_marginLeft - " " " Left "

Layout_marginRight - " " " Right "

Layout_gravity — Specifies how child views are positioned

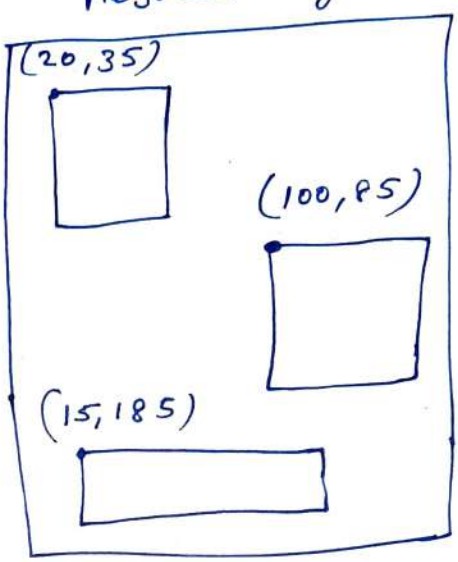
Layout_weight — Specifies how much of the extra space in the layout should be allocated to the view

Layout_x — Specifies the x-coordinate of the view or view Group.

Layout_y — Specifies the Y-Coordinate of the view or view Group

ii) Absolute Layout :- This layout enables you to specify the exact location of its children. Absolute layouts are less flexible and harder to maintain than other types of layouts without absolute positioning.

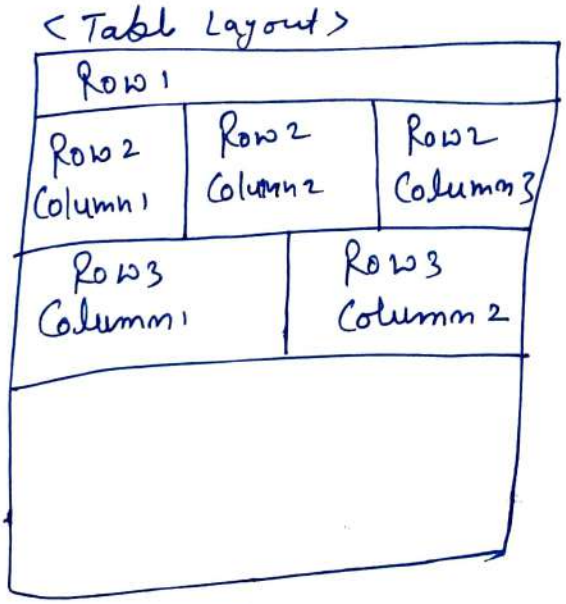
Absolute layout



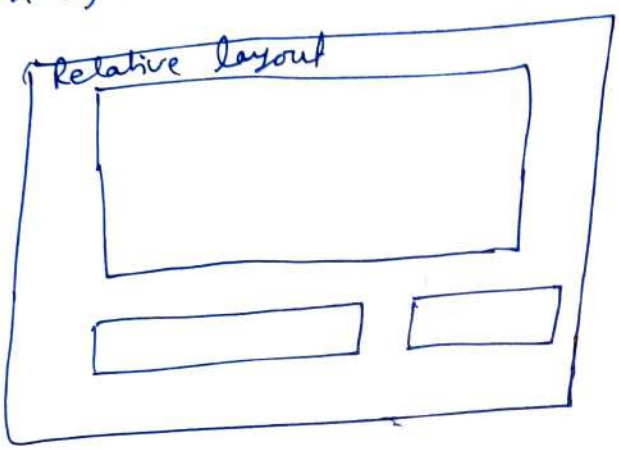
(iii) Table layout :-
A layout that arranges its children into rows and columns. A table layout consists of a number of TableRow. Each row has zero or more cells, each cell can hold one view objects.

Table layout Containers do not display border lines for their rows, columns or cells.

<Table Layout>



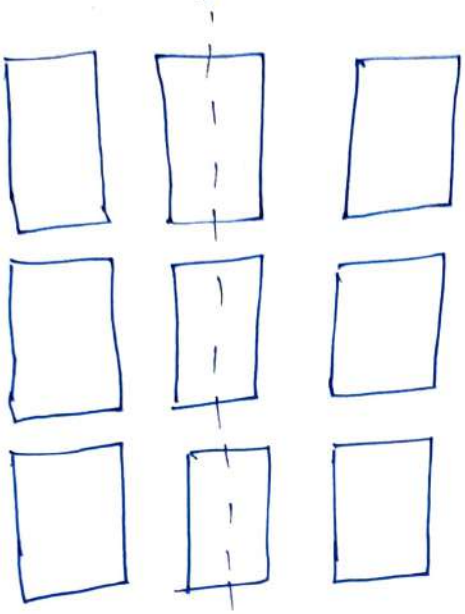
(iv) Relative layouts :- The relative layout enables you to specify the location of child objects relative to each other. The position of each view can be specified as relative to sibling elements (such as to the left of or below another view) or in positions relative to the parent RelativeLayout area (such as aligned to the bottom, left or center).



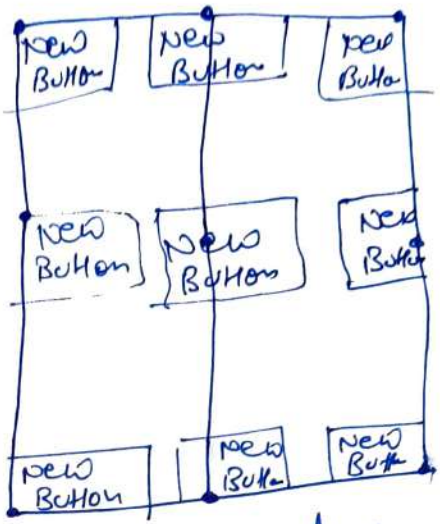
In RelativeLayout the position of child views relative to each other or relative to the parent. In case if didn't specify the position of child views, by default all child views are positioned to top-left of the layout.

(v) FrameLayout :- Frame layout is designed to block out an area on the screen to display a single item. Generally, FrameLayout should be used to hold a single child view, because it can be difficult to organize child views in a way that's scalable to different screen sizes without the children overlapping each other.

We can add multiple children to a FrameLayout and control their position by assigning gravity to each child, using the "android:layout_gravity" attributes.



FrameLayout.



Button Placed in FrameLayout.

(vi) Scroll view :- A ScrollView is a special type of FrameLayout in that it enables user to scroll through a list of views that occupy more space than the physical display. The ScrollView can contain only one child view or viewGroup, which normally is a LinearLayout.

Basic views :- Some of the basic views that you can use to design the UI of your Android applica

- → Text view
- → Edit Text
- → Button
- → Image Button
- → Check Box
- → Toggle Button
- → Radio Button
- → Radio Group

→ Textview Views :-

The Textview View is used to display text to the user. In some other platforms, the Textview is commonly known as the label view. Its sole purpose is to display text on the screen.

→ Button → Represents a push-button widget.

→ ImageButton → Similar to the Button views, except that it also displays an image.

→ EditText → A subclass of the Textview view, except that it allows user to edit its text content

→ CheckBox – A special type of button that has two states: checked or unchecked

→ RadioGroup and RadioButton :- The RadioButton has two states: either checked or unchecked. Once a RadioButton is checked, it can't be unchecked. A RadioGroup is used to group together one or more RadioButton views, thereby allowing only one RadioButton to be checked within the RadioGroup.

→ ToggleButton → Displays checked/unchecked states using a light indicator.

→ Listview view → Listviews are views that enables you to display a long list of items. There are two types of list view: List view & Spinner view.