Desiging User Interfaces with Layouts

* Explain Different Types of Layouts.

=> Layouts are used to define the structure and organization of UI element.

Layouts allow to user to arrange User Interface components.

Layouts provides a way to group multiple User Interface components together.

This are the basic types of Layouts.

1) Linear Layout
2) Relative Layout
3) Constraint Layout
4) Table Layout
5) Frame Layout etc.

1 Linear Layout: Linear Layout is one of the basic Layout classes in Android.

It is used to organizes UI elements either horizontally or Vertically based on orientation.

There are main Two Types of Orientation in Layout.

(a) Horizontal
(b) Vertical

a Horizontal: In this orientation, UI elements are arranged From left to right.

Syntax: android:orientation="horizontal"

In this Orientation, every compohents are arranged in horizontally.

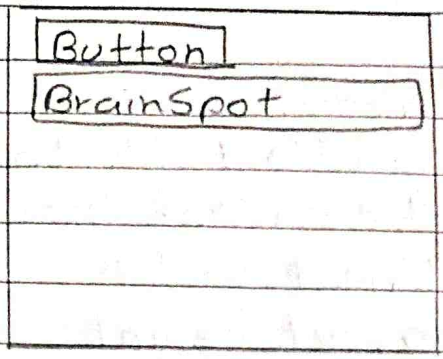
b Vertical: In this orientation, UI elements are arranged from top to bottom.

Syntax: android:orientation="Vertical"

→ Example:

```
<LinearLayout
   xmlns:android:"http://schema.
     android.com/apk/res/android"
  android:layout_width="match-parent"
  android:layout_height="wrap-content"
  android:orientation="vertical">

  <LinearLayout
     android:orientation="horizontal">
     <Button
        android:id="@id/but1"
        android:text="First"/>
  </LinearLayout>

  <LinearLayout
     android:orientation="horizontal">
     <TextView
        android:id="@id/tv"
        android:text="BrainSpot"/>
  </LinearLayout>

</LinearLayout>
```

We can use Two Linear Layouts together with different orientation.



2 Relative Layout: This Layout is allow user to position child views relative to each other or to the parent Layout.

It enables user to create complex and flexible user interface.

In this Layout, Using such attribute we can create complex design.

Attribute : layout_below
layout_above
layout_alignParentLeft
layout_alignParentBottom

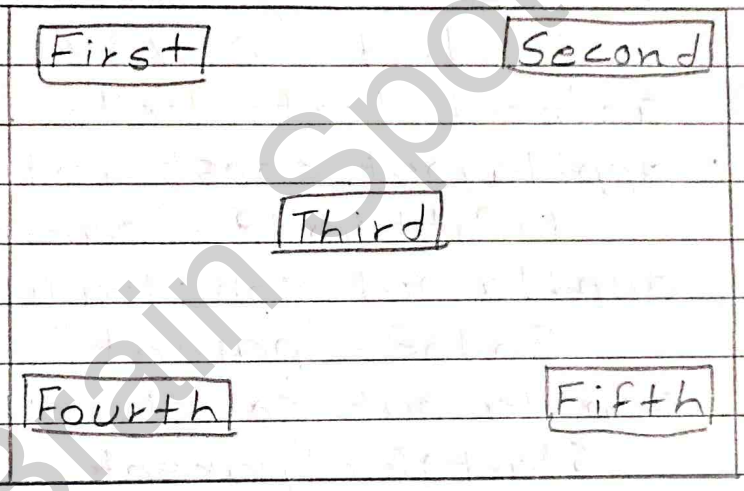
In Relative Layout, we can use multiple Linear Layout.

This Layout provides a lot of Flexibility in designing complex UIs.

-> Example:

```
<RelativeLayout
    xmlns:android="http://schemas.
    android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap-content">

    <Button
        android:layout_centerHorizontal
            ="true"
        android:layout_alignParentLeft
            ="true"
        android:text="First" />

    <Button
        android:layout_alignParent
            Bottom="true"
        android:text="Fourth" />

    <Button
        android:layout_alignParent
            Right="true"
        android:text="Second" />

    <Button
        android:layout_centerIn
            Parent="true"
        android:text="Third" />
```

```
<Button
    android:layout_alignParent
        Right="true"
    android:layout_alignParent
        Bottom="true"
    android:text="Fifth" />

</RelativeLayout>
```



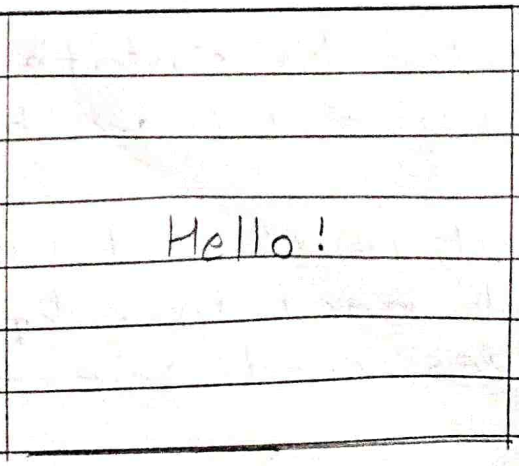
3 Constraint Layout: This Layout is a Flexible and poweful layout in Android.

It allow user to create complex and responsive UI with flat view.

This Layout is useful when designing layout that need to adapt to various screen sizes and orientations.

-> Example:

```
<android.x.constraintlayout.widget.
  ConstraintLayout
  xmlns:android="http://schemas.
    android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  >

  <TextView
      android:id="@id/tv"
      android:text="Hello!"
      app:layout_constraintBottom.
          toBottomOF="parent"
      app:layout_constraintEnd_to
          EndOf="parent"
      app:layout_constraintStart_to
          StartOF="parent"
      app:layoutTop_to TopOf="parent" />

</android.x.constraintlayout.widget.
  constraintLayout>
```



* Difference between View and ViewGroup.

=>

	View	ViewGroup
1	A View is the basic building block for UI element.	A ViewGroup is also a subclass of View.
2	View is a Single UI element.	ViewGroup is a container for multiple View elements.
3	View subclasses include 'TextView', 'Button' etc.	ViewGroup is the base class for layouts.