Android Layout

Layouts

- An Android layout is a class that handles arranging the way its children appear on the screen.
- Anything that is a View (or inherits from View) can be a child of a layout.
- All of the layouts inherit from **ViewGroup** (which inherits from View) so you can nest layouts.

Types of Layouts

- The standard Layouts are:
  - FrameLayout
  - LinearLayout
  - RelativeLayout
  - TableLayout
  - AbsoluteLayout

Frame Layout

- **Frame Layout** is designed to display a single item at a time.
- You can have multiple elements within a Frame Layout but each element will be positioned based on the top left of the screen.
- Elements that overlap will be displayed overlapping.

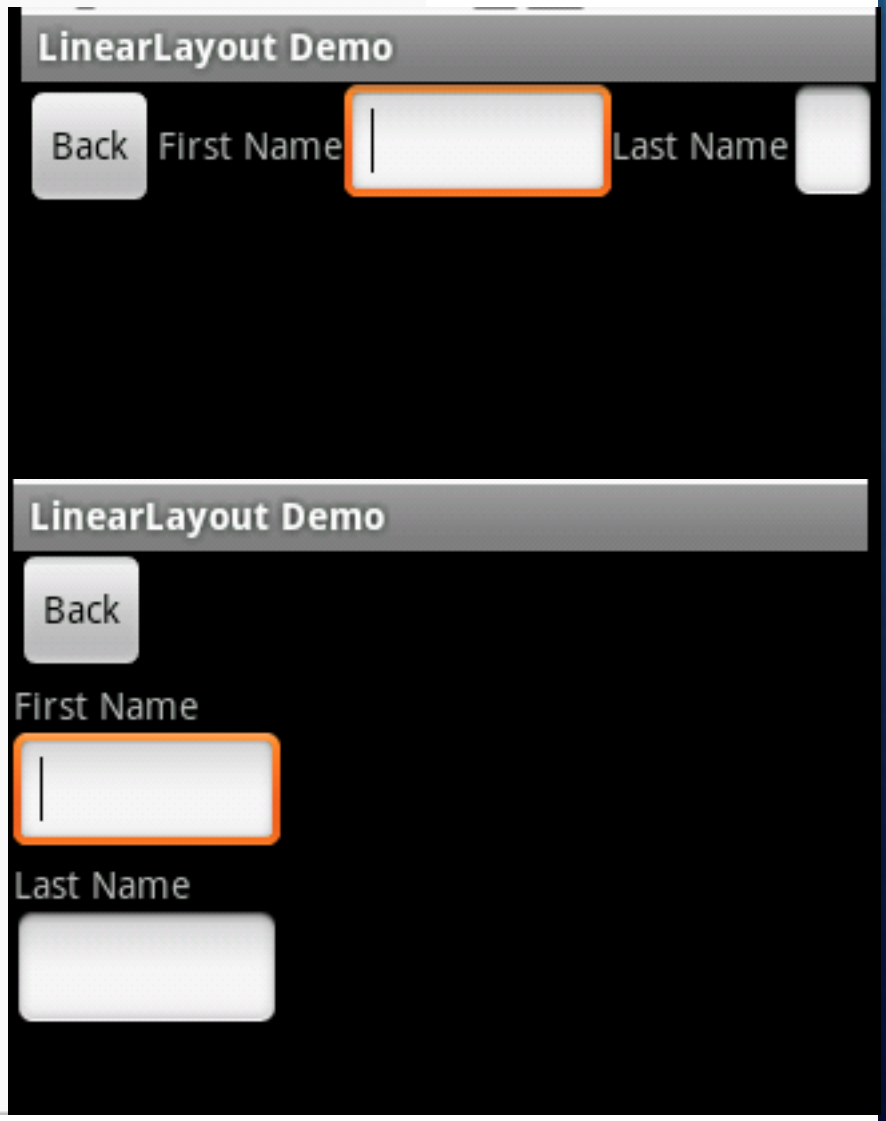
Uses of Framelayout

- In Game apps to display “Game Over” on top of existing view.
- To display play buttons to static images.
- To display texts on top of controls
- To Hide and show controls like Voting stars.

Linear Layout

- **Linear Layout** organizes elements along a single line.
- You specify whether that line is vertical or horizontal using android:orientation.
- Can be used to display buttons one after the other.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="horizontal"
        android:layout_width="fill_parent"
    android:layout_height="fill_parent">
     <Button
        android:id="@+id/backbutton"
        android:text="Back"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:text="First Name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <EditText
        android:width="100px"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:text="Last Name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <EditText
        android:width="100px"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```



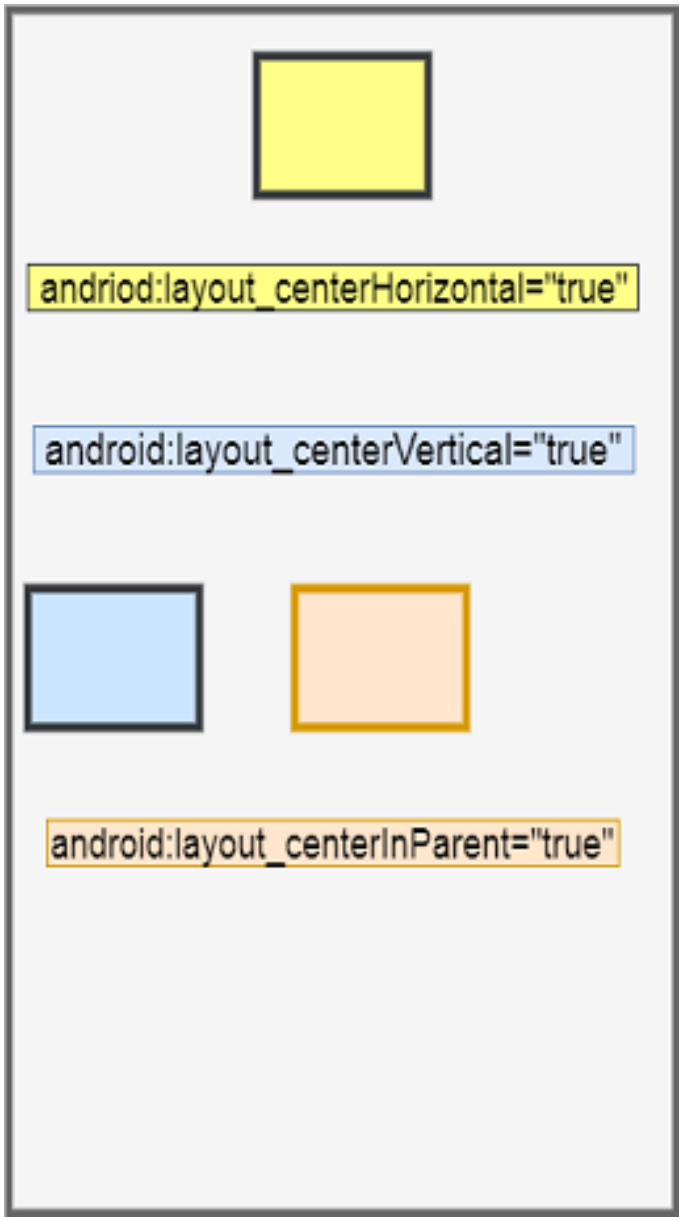
Relative Layout

- **Relative Layout** lays out elements based on their relationships with one another, and with the parent container.
- This is arguably the most complicated layout, and we need several properties to actually get the layout we want.
- There are properties that will layout elements relative to the parent container.

Relative To Container

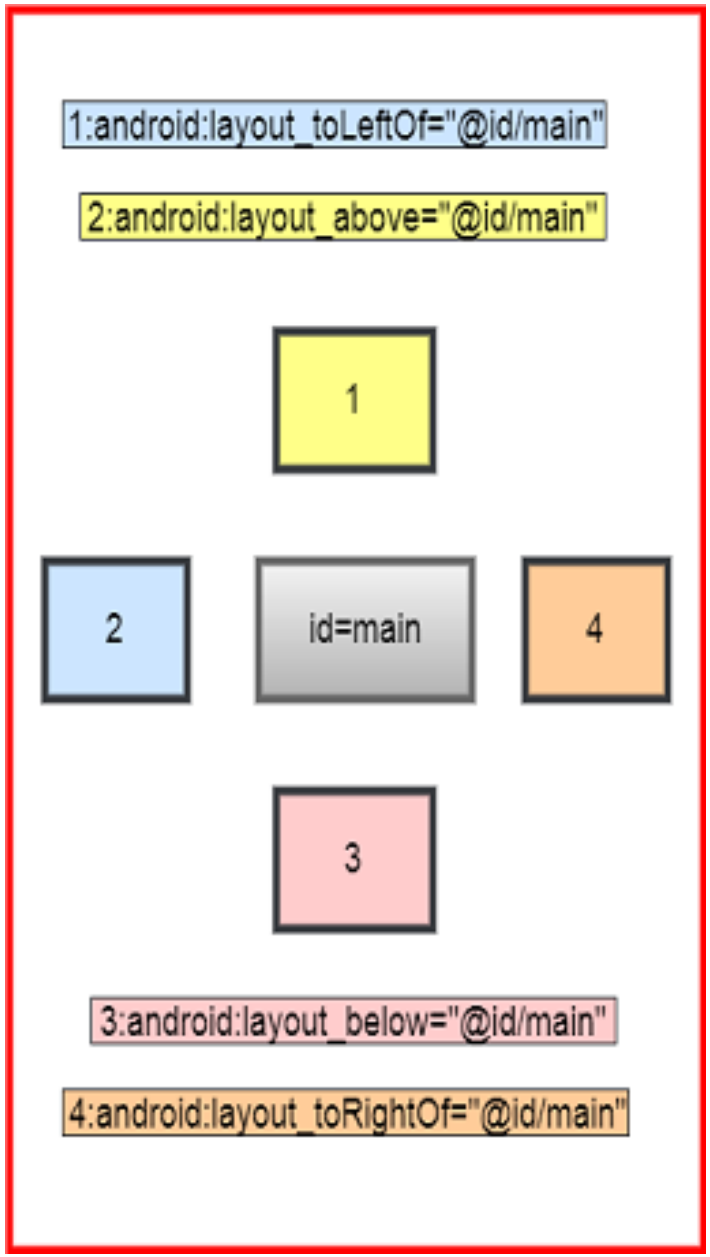
- android:layout_alignParentBottom – Places the bottom of the element on the bottom of the container
- android:layout_alignParentLeft – Places the left of the element on the left side of the container
- android:layout_alignParentRight – Places the right of the element on the right side of the container
- android:layout_alignParentTop – Places the element at the top of the container
- android:layout_centerHorizontal – Centers the element horizontally within its parent container
- android:layout_centerInParent – Centers the element both horizontally and vertically within its container
- android:layout_centerVertical – Centers the element vertically within its parent container


View 1
android:layout_alignParentTop="true"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_alignParentLeft="true"
android:layout_alignParentBottom="true"
View 2


andriod:layout_centerHorizontal="true"
android:layout_centerVertical="true"
android:layout_centerInParent="true"

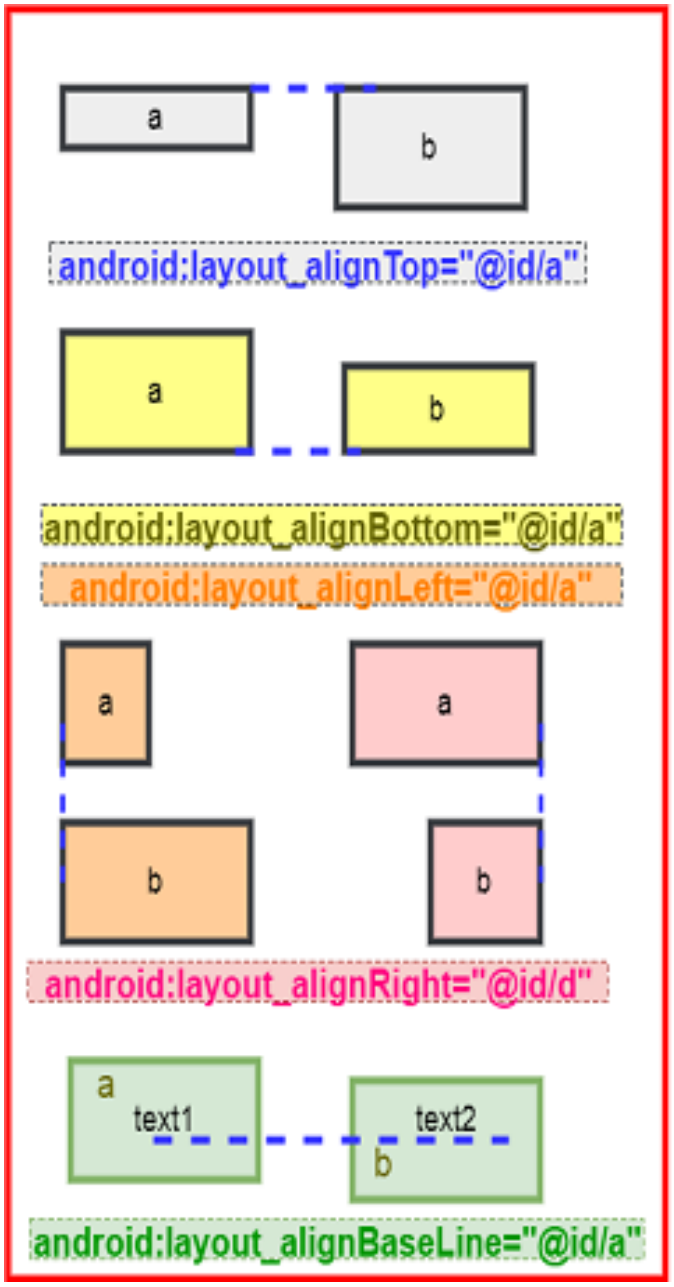
Relative To Other Elements

- android:layout_above
- android:layout_below
- android:layout_toLeftOf
- android:layout_toRightOf
- Each element that is used in this way must have an ID defined using android:id="@+id/XXXXX" where XXXXX is replaced with the desired id.
- You use "@id/XXXXX" to reference an element by its id.


1:android:layout_toLeftOf="@id/main"
2:android:layout_above="@id/main"
1
2
id=main
4
3
3:android:layout_below="@id/main"
4:android:layout_toRightOf="@id/main"

Alignment With Other Elements

- android:layout_alignBaseline
- android:layout_alignBottom
- android:layout_alignLeft
- android:layout_alignRight
- android:layout_alignTop


a
b
android:layout_alignTop="@id/a"
a
b
android:layout_alignBottom="@id/a"
android:layout_alignLeft="@id/a"
a
a
b
b
android:layout_alignRight="@id/d"
a
text1
text2
b
android:layout_alignBaseLine="@id/a"

```
<RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        xmlns:android="http://schemas.android.com/apk/res/android">
        <Button
                android:id="@+id/backbutton"
                android:text="Back"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
        <TextView
                android:id="@+id/firstName"
                android:text="First Name"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@id/backbutton" />
        <EditText
                android:width="100px"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_toRightOf="@id/firstName"
                android:layout_alignBaseline="@id/firstName" />
        <TextView
                android:id="@+id/lastName"
                android:text="Last Name"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@id/firstName" />
        <EditText
                android:width="100px"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_toRightOf="@id/lastName"
                android:layout_alignBaseline="@id/lastName" />
</RelativeLayout>
```

Absolute Layout

- **AbsoluteLayout** is based on the simple idea of placing each control at an absolute position.
- You specify the exact x and y coordinates on the screen for each control.
- This is not recommended for most UI development (in fact AbsoluteLayout is currently deprecated)

```
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button
        android:id="@+id/backbutton"
        android:text="Back"
        android:layout_x="10px"
        android:layout_y="5px"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:layout_x="10px"
        android:layout_y="110px"
        android:text="First Name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <EditText
        android:layout_x="150px"
        android:layout_y="100px"
        android:width="100px"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:layout_x="10px"
        android:layout_y="160px"
        android:text="Last Name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
        <EditText
        android:layout_x="150px"
        android:layout_y="150px"
        android:width="100px"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</AbsoluteLayout>
```

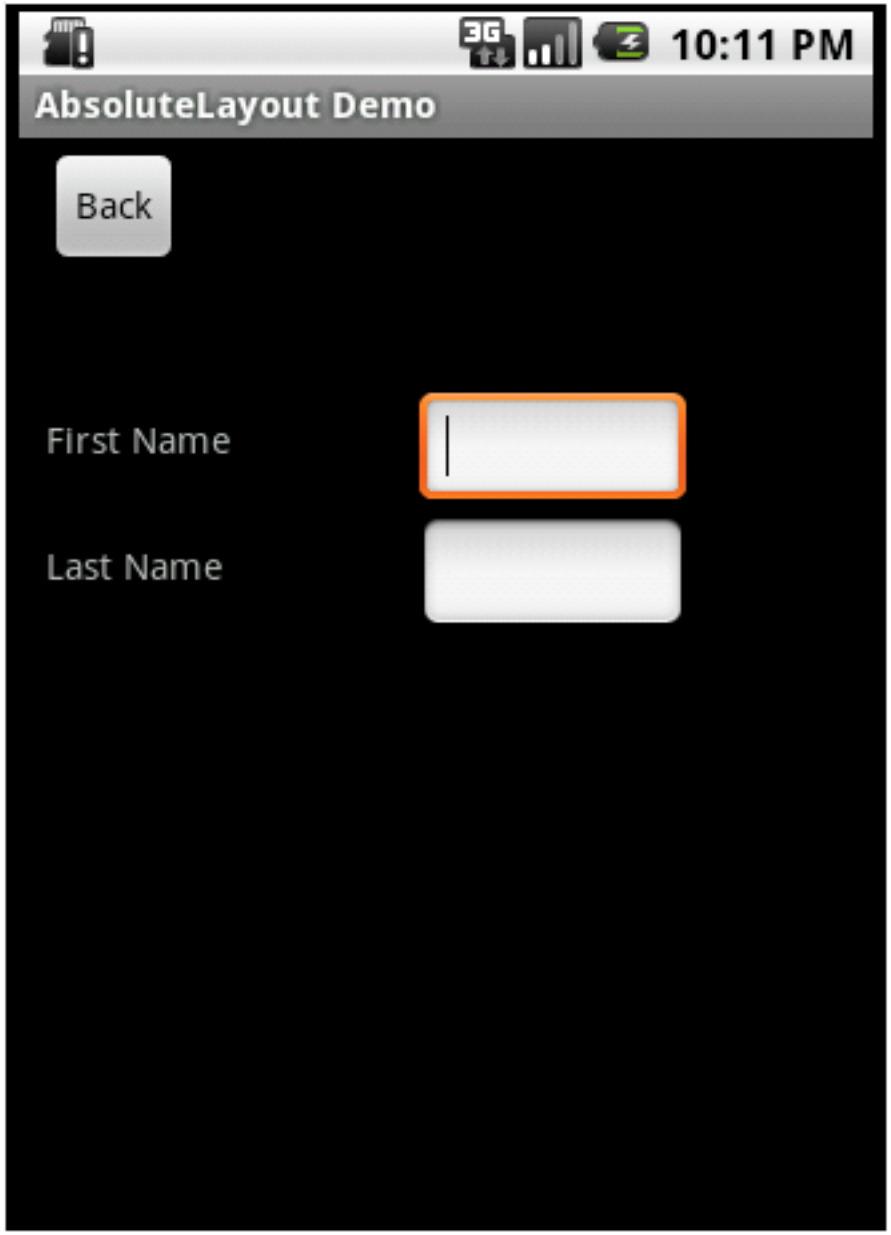


Table Layout

- **Table Layout** organizes content into rows and columns.
- The rows are defined in the layout XML, and the columns are determined automatically by Android.
- To specify columns manually android:layout_span="3" can be called.

```
<TableLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        xmlns:android="http://schemas.android.com/apk/res/android">
        <TableRow>
                <Button
                android:id="@+id/backbutton"
                android:text="Back"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
        </TableRow>
        <TableRow>
                <TextView
                android:text="First Name"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_column="1" />
                <EditText
                android:width="100px"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
        </TableRow>
        <TableRow>
                <TextView
                android:text="Last Name"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_column="1" />
                <EditText
                android:width="100px"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
        </TableRow>
</TableLayout>
```