Day 7: Relative Layout

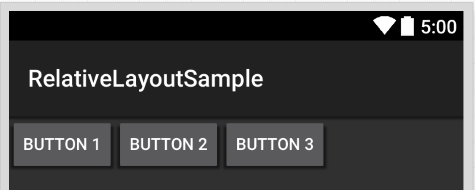
Relative Layout is one of the more popular layouts in Android and is used to put Views on a screen in relative positions to child and parent views and view groups.

Let’s start digging into the XML code that is used to create RelativeLayout with a really basic example.

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent">  <Button  android:id="@+id/btnButton1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Button 1" />  <Button  android:id="@+id/btnButton2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Button 2"  android:layout\_toRightOf="@+id/btnButton1" />  <Button  android:id="@+id/btnButton3"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Button 3"  android:layout\_toRightOf="@+id/btnButton2" />  </RelativeLayout> |

Gist file: <https://gist.github.com/vkoppaka/4f177de27d9c237565cd>

In the above snippet a new attribute “**android:layout\_toRightOf**” is introduced. This attribute is specific to the Views that are placed inside the Relative Layout. What this attribute says is that a view should be placed to the right of the view that is defined as its value. So in the above example, Button 2 will be placed to the right of Button 1 and Button 3 will be placed to the right of Button 2. If you were to see the above code in the designer or run on the device you should see the three buttons line up next to each other.



There are lot more properties that belong to the Relative Layout and they are (few of them) –

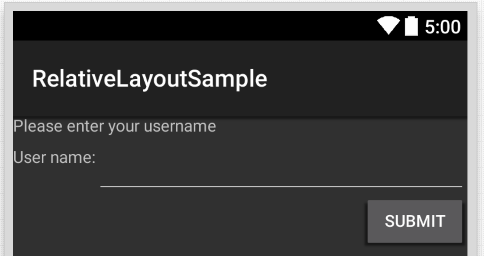
1. layout\_toLeftOf - Puts the view to the left of whatever view you mention in the value.
2. layout\_toRightOf - Puts the view to the right of whatever view you mention in the value.
3. layout\_above - Puts the view above whatever view you mention in the value.
4. layout\_below - Puts the view below whatever view you mention in the value.
5. layout\_alignLeft - Makes the left edge of your view match the left edge of the view whose ID you mention in the value.
6. layout\_alignRight - Makes the right edge of your view match the left edge of the view whose ID you mention in the value.
7. layout\_alignBottom - Makes the bottom edge of your view match the left edge of the view whose ID you mention in the value.
8. layout\_alignParentLeft - Makes the left edge of your view match the left edge of the parent view.
9. layout\_alignParentTop - Makes the right edge of your view match the right edge of the parent view.
10. layout\_centerHorizontal - Horizontally centers the view in its parent.
11. layout\_centerVertical - Vertically centers the element in its parent.

Let’s look at an example that uses some of these newly learned layout properties

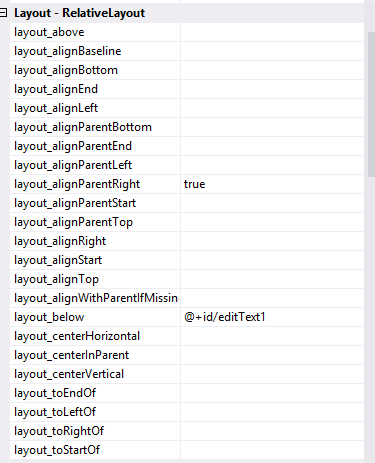
|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent">  <TextView  android:id="@+id/textView1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentTop="true"  android:layout\_marginBottom="10dp"  android:text="Please enter your username" />  <TextView  android:id="@+id/textView2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_below="@+id/textView1"  android:text="User name:" />  <EditText  android:id="@+id/editText1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentRight="true"  android:layout\_alignTop="@+id/textView2"  android:layout\_toRightOf="@+id/textView2" />  <Button  android:id="@+id/btnSubmit"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentRight="true"  android:layout\_below="@+id/editText1"  android:text="Submit" />  </RelativeLayout> |

Gist file: <https://gist.github.com/vkoppaka/cce454f3ce283d96272f>

And the UI would look like:



For a full list of all possible Relative Layout Options, we can explore properties of a view in Visual Studio.



That’s it for today. Tomorrow we will explore more Layout options.

Additional Note:

If you are having any trouble getting XML and Designer preview to match, please build your solution once.

Venkata