1. Linear Layout

In a linear layout, like the name suggests, all the elements are displayed in a linear fashion(below is an example of the linear layouts), either **Horizontally** or **Vertically** and this behavior is set in **android:orientation**which is an attribute of the node LinearLayout.

Example of Vertical layout snippet

|  |
| --- |
| <LinearLayout android:orientation="vertical"> .... </LinearLayout> |

Example of Horizontal layout snippet

|  |
| --- |
| <LinearLayout android:orientation="horizontal"> .... </LinearLayout> |

Now that we know the two types of linear layouts, here are the steps you need to follow to create them

**1**. Create a new project **File -> New -> Android Project**  
**2**. In Package Explorer right click on **res/layout** folder and create a new Android XML File and name it as you wish. I am naming it as “**linear\_layout.xml**”  
**res/layout -> Right Click -> New -> Android XML File**  
**3**. Now open newly created xml file (in my case “**linear\_layout.xml**”) and type the following code.

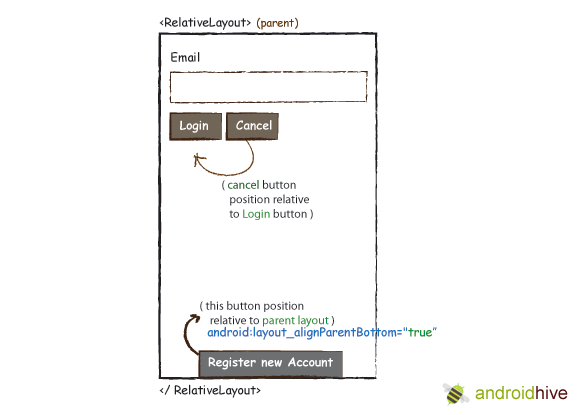
|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <!-- Parent linear layout with vertical orientation -->  <LinearLayout    xmlns:android="http://schemas.android.com/apk/res/android"    android:orientation="vertical"    android:layout\_width="match\_parent"    android:layout\_height="match\_parent">      <TextView android:layout\_width="fill\_parent" android:layout\_height="wrap\_content"              android:text="Email:" android:padding="5dip"/>      <EditText android:layout\_width="fill\_parent" android:layout\_height="wrap\_content"              android:layout\_marginBottom="10dip"/>      <Button android:layout\_width="fill\_parent" android:layout\_height="wrap\_content"              android:text="Login"/>      <!-- Child linear layout with horizontal orientation -->    <LinearLayout android:layout\_width="fill\_parent"                        android:layout\_height="wrap\_content"                android:orientation="horizontal" android:background="#2a2a2a"                android:layout\_marginTop="25dip">      <TextView android:layout\_width="fill\_parent" android:layout\_height="wrap\_content"           android:text="Home" android:padding="15dip" android:layout\_weight="1"           android:gravity="center"/>      <TextView android:layout\_width="fill\_parent" android:layout\_height="wrap\_content"           android:text="About" android:padding="15dip" android:layout\_weight="1"           android:gravity="center"/>      </LinearLayout>    </LinearLayout> |

2. Relative Layout

In a relative layout every element arranges itself relative to other elements or a parent element.  
As an example, lets consider the layout defined below. The “**Cancel**” button is placed relatively, to the **right of**the “**Login**” button **parallely**. Here is the code snippet that achieves the mentioned alignment (Right of Login button parallely)

Example code snippet

|  |
| --- |
| <Button android:id="@+id/btnLogin" ..></Button>    <Button android:layout\_toRightOf="@id/btnLogin"              android:layout\_alignTop="@id/btnLogin" ..></Button> |



Here are the steps to create a relative layout

**1**. Create a new project File -> New -> Android Project  
**2**. In Package Explorer right click on **res/layout** folder and create a new Android XML File and name it as you wish. I am naming it as “relative\_layout.xml”  
**res/layout -> Right Click -> New -> Android XML File**  
**3**. Now open newly created xml file (in my case “**relative\_layout.xml**”) and type the following code.

|  |
| --- |
| <?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="wrap\_content">        <TextView android:id="@+id/label" android:layout\_width="fill\_parent"                android:layout\_height="wrap\_content" android:text="Email" />        <EditText android:id="@+id/inputEmail" android:layout\_width="fill\_parent"                android:layout\_height="wrap\_content" android:layout\_below="@id/label" />        <Button android:id="@+id/btnLogin" android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content" android:layout\_below="@id/inputEmail"              android:layout\_alignParentLeft="true" android:layout\_marginRight="10px"              android:text="Login" />        <Button android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"              android:layout\_toRightOf="@id/btnLogin"              android:layout\_alignTop="@id/btnLogin"  android:text="Cancel" />        <Button android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"              android:layout\_alignParentBottom="true" android:text="Register new Account"              android:layout\_centerHorizontal="true"/>  </RelativeLayout> |

4. Same like before open your MainActivity.java file and set the layout to your newly created relative layout file. In my case its **R.layout.relative\_layout**

|  |
| --- |
| setContentView(R.layout.relative\_layout); |

5. To run the application, **right click on the project -> Run As -> 1. Android Application**. You should see your newly created relative layout in the emulator.

