**ConstraintLayout in Android**

ConstraintLayout is similar to that of other View Groups which we have seen in Android such as RelativeLayout, LinearLayout, and many more. In this article, we will take a look at using ConstraintLayout in Android.

### **Advantages of using ConstraintLayout in Android**

* ConstraintLayout provides you the ability to completely design your UI with the drag and drop feature provided by the Android Studio design editor.
* It helps to improve the UI performance over other layouts.
* With the help of ConstraintLayout, we can control the group of widgets through a single line of code.
* With the help of ConstraintLayout, we can easily add animations to the UI components which we used in our app.

### **Disadvantages of using ConstraintLayout**

* When we use the Constraint Layout in our app, the XML code generated becomes a bit difficult to understand.
* In most of the cases, the result obtain will not be the same as we got to see in the design editor.
* Sometimes we have to create a separate layout file for handling the UI for the landscape mode.

### **How ConstraintLayout differs from Linear Layout?**

* With the help of ConstraintLayout, we can position our UI components in any sort of order whether it may be horizontal or vertical. But in the case of Linear Layout, we can only arrange our UI components either in a horizontal or in a vertical manner.
* Linear Layout provides usability with which we can equally divide all the UI components in a horizontal or vertical manner using weight sum but in Constraint Layout, we have to arrange this UI component manually.
* In Linear Layout the UI which is actually seen in the Design editor of Android Studio will be the same as that we will get to see in the app, but in the case of Constraint layout if the UI component is not Constrained then the UI will not look the same as that of in design editor.

### **How ConstraintLayout differs from RelativeLayout?**

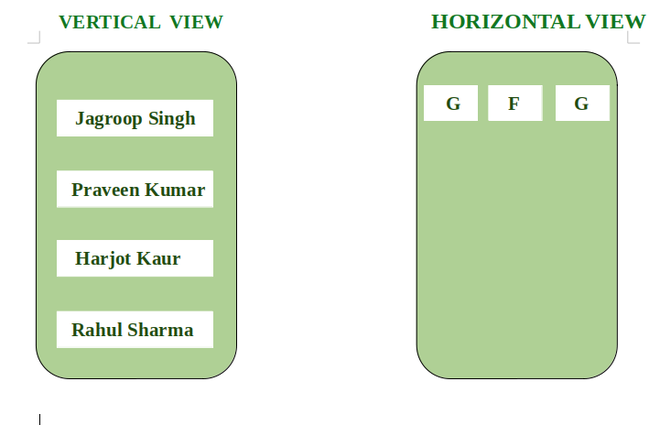
* In Constraint Layout, we have to add constraints to the view on all four sides whereas in Relative Layout we can simply align our UI component relative to its ID using the ids of UI components.
* In Relative Layout, the UI which is actually seen in the Design editor of Android Studio will be the same as that we will get to see in the app, but in the case of Constraint layout if the UI component is not Constrained then the UI will not look same as that of in design editor.

**Difference Between LinearLayout and RelativeLayout in Android**

### **LinearLayout**

[LinearLayout](https://www.geeksforgeeks.org/linearlayout-and-its-important-attributes-with-examples-in-android/) is a type of view group which is responsible for holding views in it either Horizontally or vertically. It is a type of Layout where one can arrange groups either Horizontally or Vertically.

**Example Diagram:**



### **RelativeLayout**

[RelativeLayout](https://www.geeksforgeeks.org/android-relativelayout-in-kotlin/) is a layout in which we can arrange views/widgets according to the position of other view/widgets. It is independent of horizontal and vertical view and we can arrange it according to one’s satisfaction.

**Example Diagram:**

