

# What is Android View?

A View is a simple building block of a user interface. It is a small rectangular box that can be TextView, EditText, or even a button. It occupies the area on the screen in a rectangular area and is responsible for drawing and event handling. View is a superclass of all the graphical user interface components.

## Why and How to use the View in Android?

Now you might be thinking what is the use of a View. So, the use of a view is to draw content on the screen of the user's Android device. A view can be easily implemented in an Application using the java code. Its creation is more easy in the XML layout file of the project. Like, the project for hello world that we had made initially.

## Types of Android Views

Another thing that might now come to your mind must be, "what are the available types of view in Android that we can use?"

For that, we'll see all these types one by one as follows:

- TextView
- EditText
- Button
- Image Button
- Date Picker
- RadioButton
- CheckBox buttons
- Image View

## What is Android View Group?

A View Group is a subclass of the ViewClass and can be considered as a superclass of Layouts. It provides an invisible container to hold the views or layouts. ViewGroup instances and views work together as a container for Layouts. To understand in simpler words it can be understood as a special view that can hold other views that are often known as a child view.

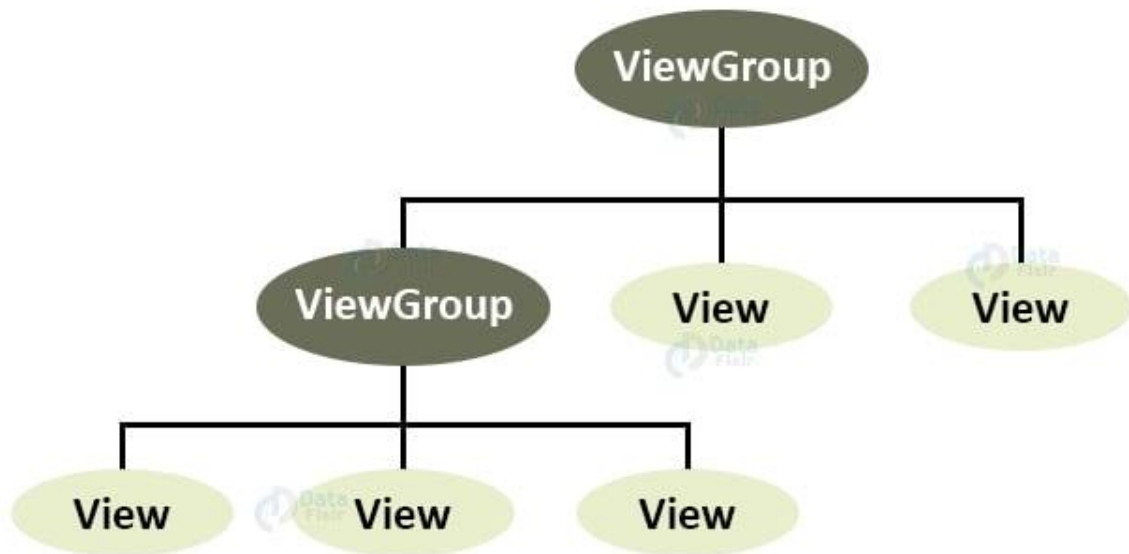
Following are certain commonly used subclasses for ViewGroup:

- LinearLayout

- RelativeLayout
- FrameLayout
- GridView
- ListView

Here is how Views and ViewGroups are linked:

## View Groups & Views



## What is Android Layout?

Layout basically refers to the arrangement of elements on a page these elements are likely to be images, texts or styles. These are a part of **Android Jetpack**. They define the structure of [android user interface](#) in the app, like in an activity. All elements in the layout are built with the help of Views and ViewGroups. These layouts can have various widgets like buttons, labels, textboxes, and many others.

We can define a Layout as follows:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/layout2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:background="#8ED3EB"
    android:gravity="center"
    android:orientation="vertical" >
    <TextView

```

```
android:id="@+id/textView4"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginLeft="10dp"
android:layout_marginTop="-40dp"
android:fontFamily="@font/almendra_bold"
android:text="This is a TextView" />
</LinearLayout>
```

# Attributes of Layout in Android

The following are the attributes for customizing a Layout while defining it:

- **android:id:** It uniquely identifies the Android Layout.
- **android:hint:** It shows the hint of what to fill inside the EditText.
- **android:layout\_height:** It sets the height of the layout.
- **android:layout\_width:** It sets the width of the layout.
- **android:layout\_gravity:** It sets the position of the child view.
- **android:layout\_marginTop:** It sets the margin of the from the top of the layout.
- **android:layout\_marginBottom:** It sets the margin of the from the bottom of the layout.
- **android:layout\_marginLeft:** It sets the margin of the from the left of the layout.
- **android:layout\_marginRight:** It sets the margin of the from the right of the layout.
- **android:layout\_x:** It specifies the x coordinates of the layout.
- **android:layout\_y:** It specifies the y coordinates of the layout.

# Types of Layouts in Android

Now that we've learned about the view and view groups and also somewhat about the layouts. Subsequently let us see the types of Layouts in Android, that are as follows:

- Linear Layout
- Relative Layout
- Constraint Layout
- Table Layout
- Frame Layout
- List View
- Grid View
- Absolute Layout
- WebView
- [ScrollView](#)

These are the types of layouts and out of them we'll learn about the two very important layouts:

## 1. Linear Layout

We use this layout to place the elements in a linear manner. A Linear manner means one element per line. This layout creates various kinds of forms on Android. In this, arrangement of the elements is in a top to bottom manner.

This can have two orientations:

**a. Vertical Orientation** – It is shown above where the widgets such as TextViews, and all in a vertical manner.

**b. Horizontal Orientation** – It is shown above where the widgets such as TextViews, and all in a horizontal manner.

## 2. Relative Layout

This layout is for specifying the position of the elements in relation to the other elements that are present there.

In the relative layout, alignment of the position of the elements to the parent container is possible. To define it in such a way, we write the following:

- `android:layout_alignParentTop= "true"`
- `android:layout_alignParentLeft= "true"`

If we write the above code, the element will get aligned on the top left of the parent container.

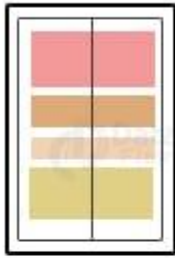
If we want to align it with some other element in the same container, it can be defined as follows:

- `android:layout_alignLeft= "@+id/element_name"`
- `android:layout_below= "@+id/element_name"`

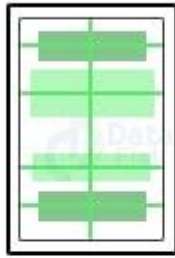
This will align the element below the other element to its left.

Here are the pictorial representations of different layouts-

# Types of Android Layouts



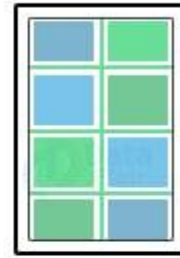
StackLayout



AbsoluteLayout



RelativeLayout



GridLayout



ContentView



ScrollView



Frame