

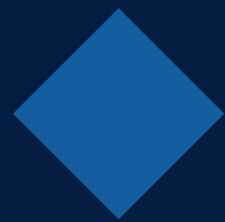
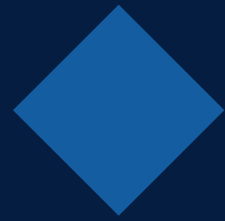
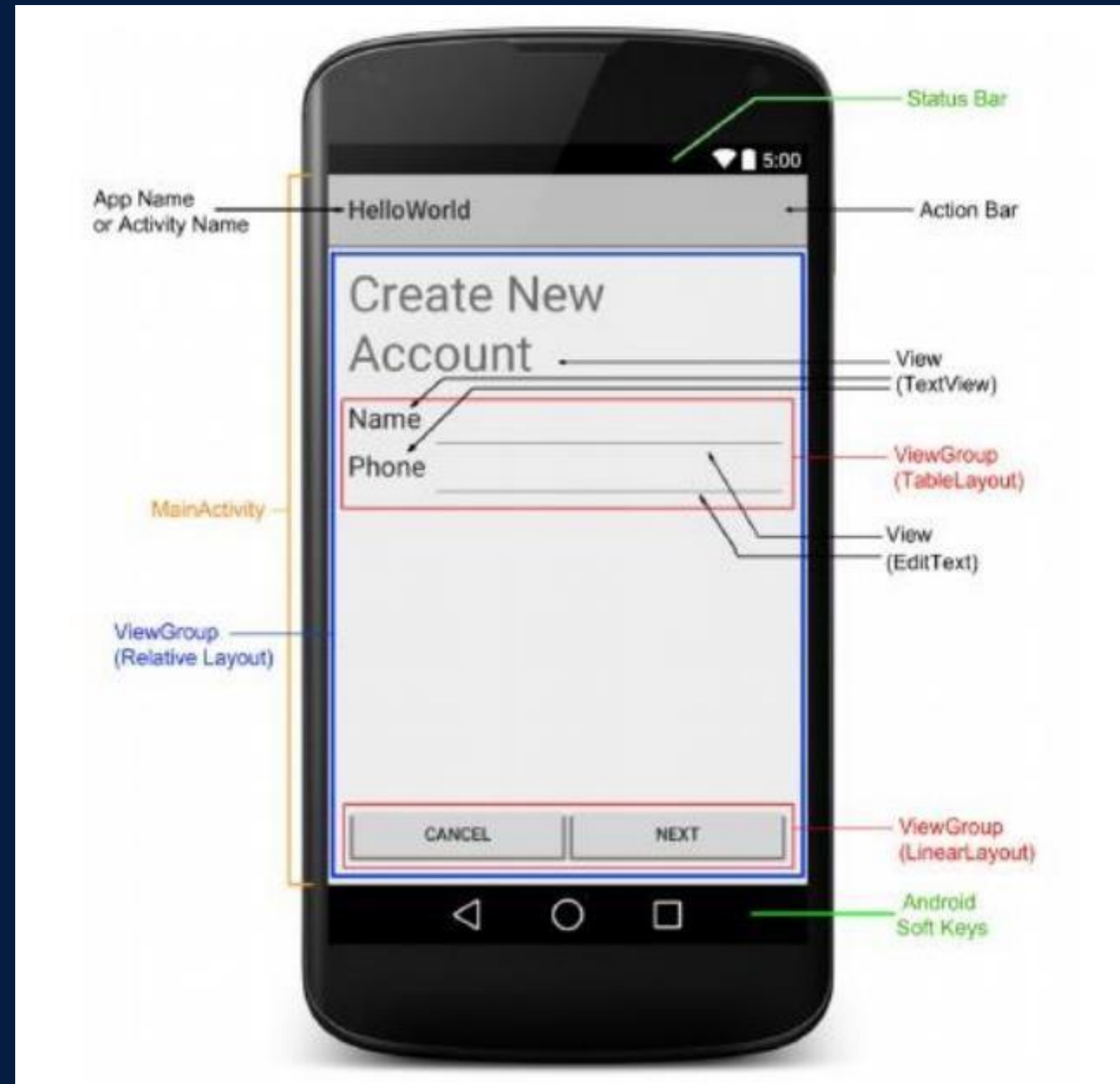


# UNIT 3

# UI COMPONENTS AND LAYOUTS



# ◆ COMPONENTS ON A SCREEN



# Directory Structure



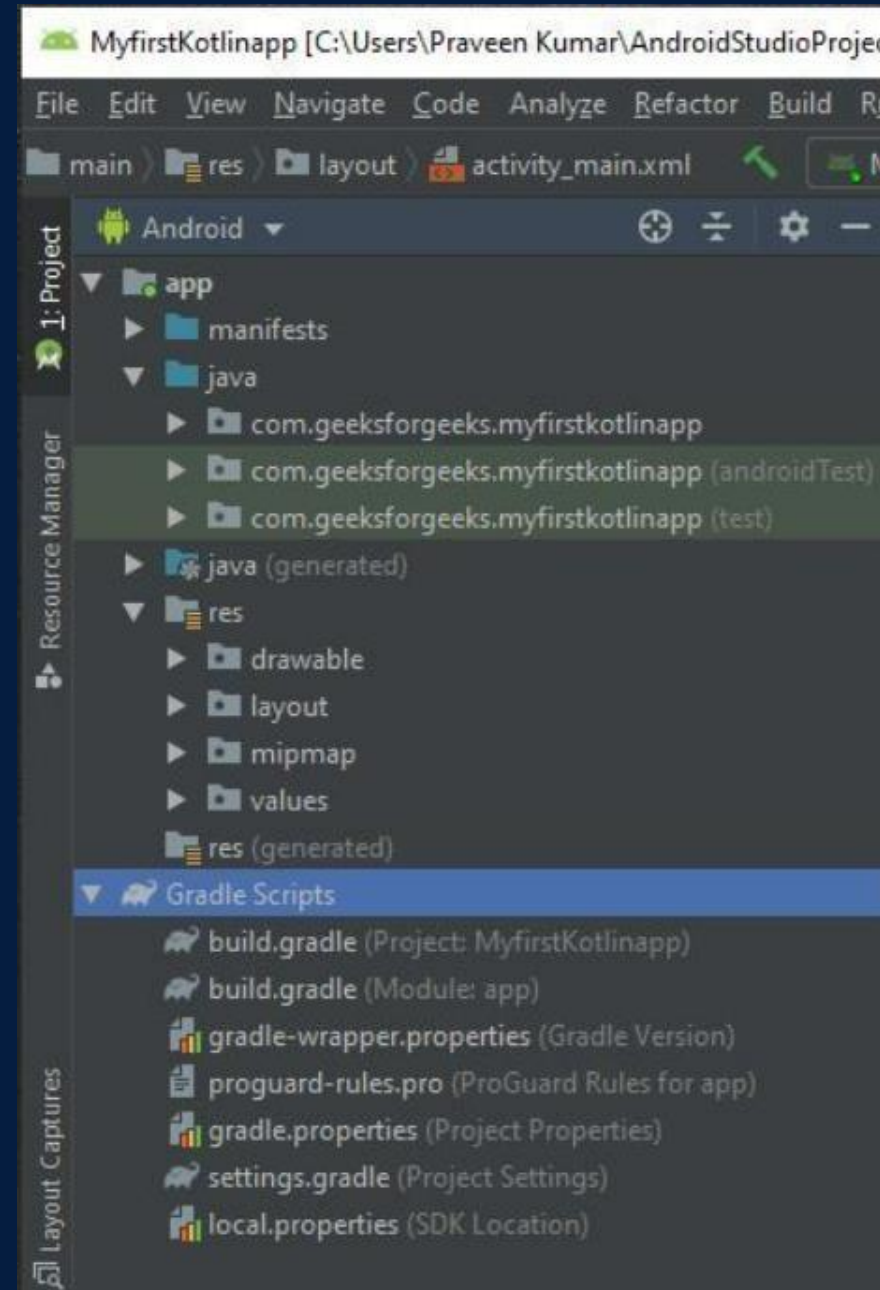
AndroidManifest.xml file is generated inside 'manifest' folder by Android Studio when we create a project which contains the configuration parameters of the project such as permissions, services and additional libraries.

will be storing all of the source code files written in Java us to

separate and sort the resources of our application. Resources basically mean all the needed files except the source code.

APK files are built using the gradle build system. When we start an app, the gradle build scripts are automatically created. If we have special requirements for our project, we can specify these requirements here.

Android provides 4 important components



'java' is the folder in our project where we programming language. 'res' contains folders that help

to build an android application:

**01 Activities:** Activities are said to be the presentation layer of our applications. The UI of our application is build around one or more extensions of the Activity class. **Services:**

# Important Components to build an android app

These components run at backend, updating your data sources and

**02** Activities, triggering Notification and also broadcast Intents. They also perform some tasks when applications are not active.

— **Intents and Broadcast Receivers:**  
Android Intent is the message that is passed

**03** between components such as

— activities, content providers, broadcast receivers, services etc. Broadcast Receivers simply respond to broadcast

messages from other applications or from the system itself.

**04 Content Providers:** It is used to manage and persist the application data also typically interact with SQL database. They are also responsible for sharing the data beyond the application boundaries.



**Started:** After a service starts, it can run indefinitely and usually performs single operation. No result is returned to user.

**Bound:** In this case, a component is bound to a service so that a particular task can be completed.



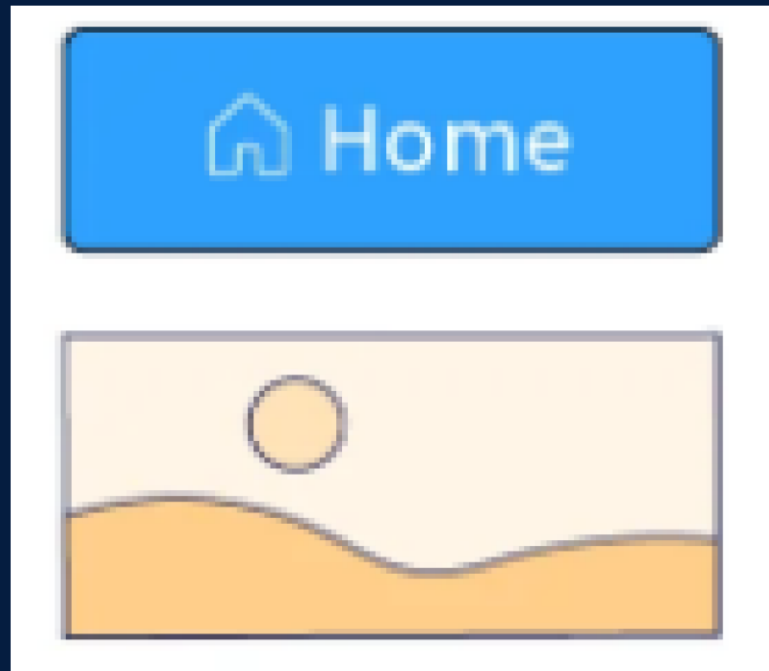
**Normal Broadcasts:** These are asynchronous in nature. Many receivers can be activated at the same time which doesn't have any defined order.

**Ordered Broadcasts:** Broadcasts are delivered to receiver on one-to-one and sequential basis.

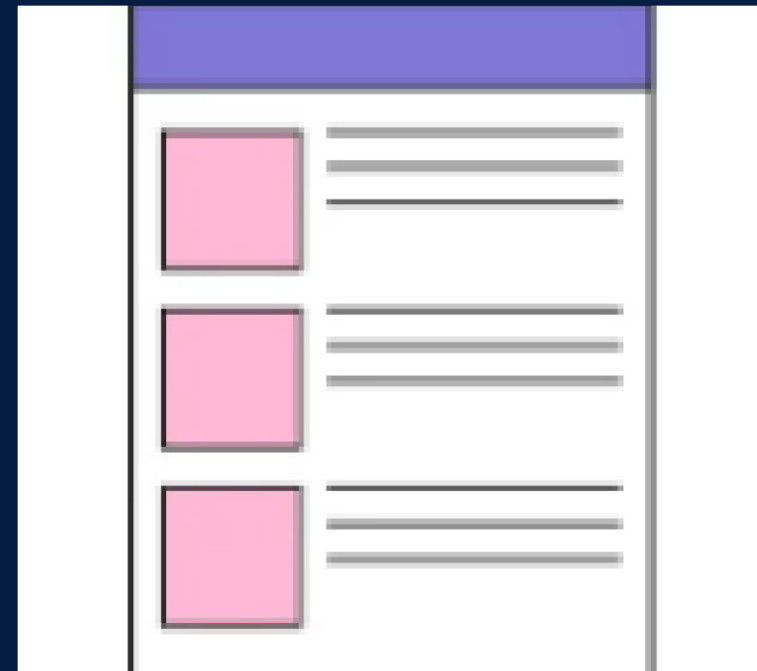
# Fundamentals of UI design



View



ViewGroup



Fragment



Activity



A **View** is an object/widget that draws something on the screen by the help of user interact.

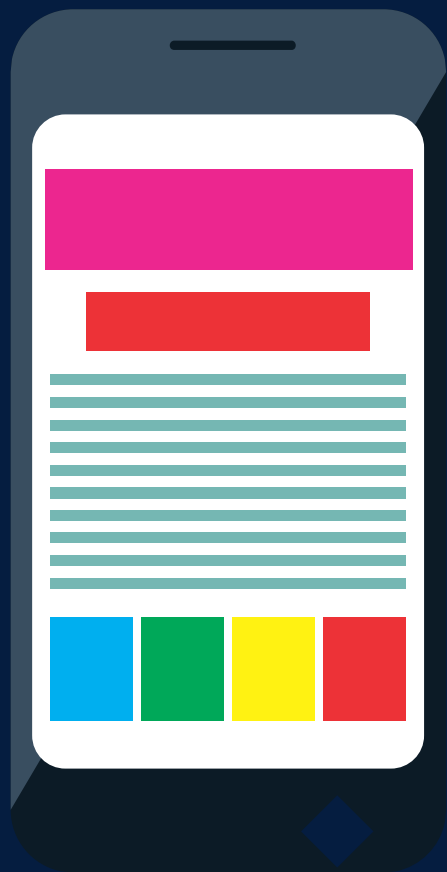
The **ViewGroup** is a subclass of View and provides invisible container that hold other Views or ViewGroups.

**Fragments** represent a portion of user interface in an Activity.

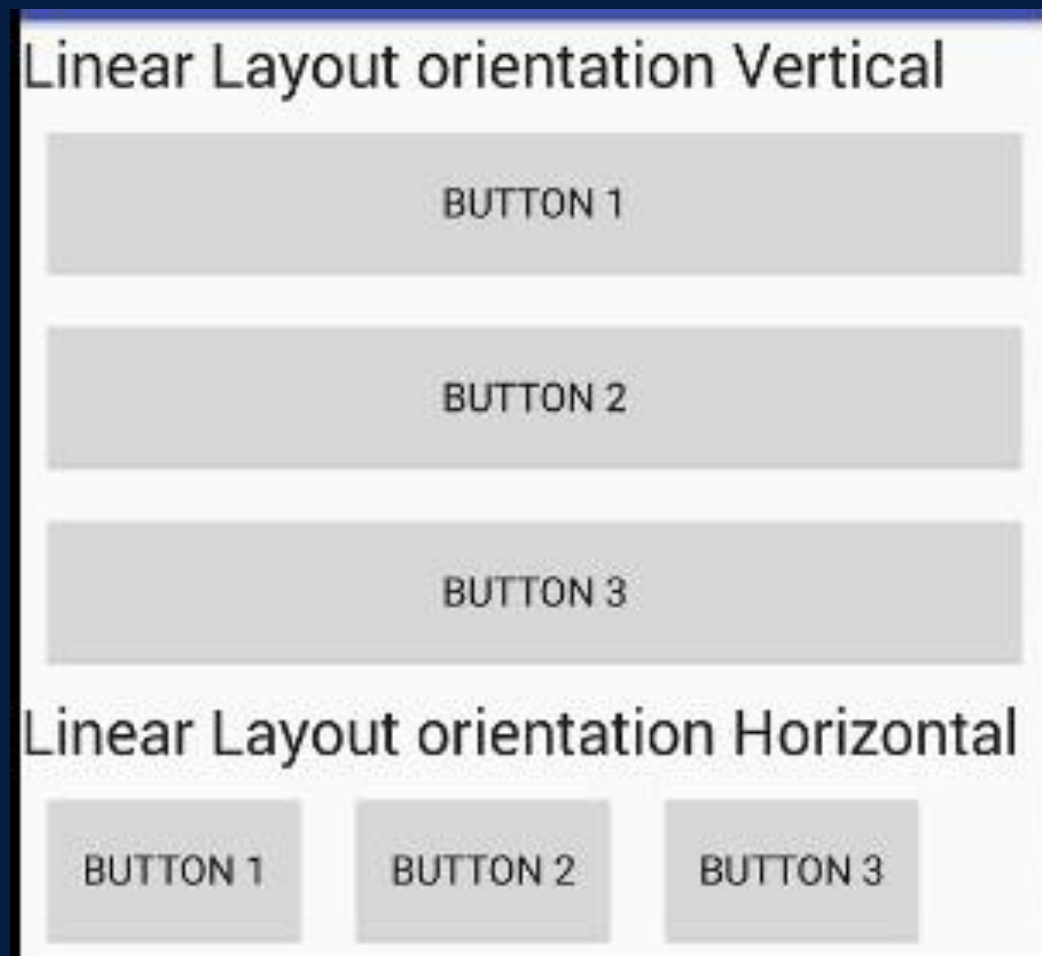
**Activities** represent a single screen that user interact. To display a UI we assign a View to an Activity.

## UI Layouts

The views or the controls that we want to display in an application are arranged in an order or sequence by placing them in the desired layout. The layouts also known as Containers or ViewGroups.







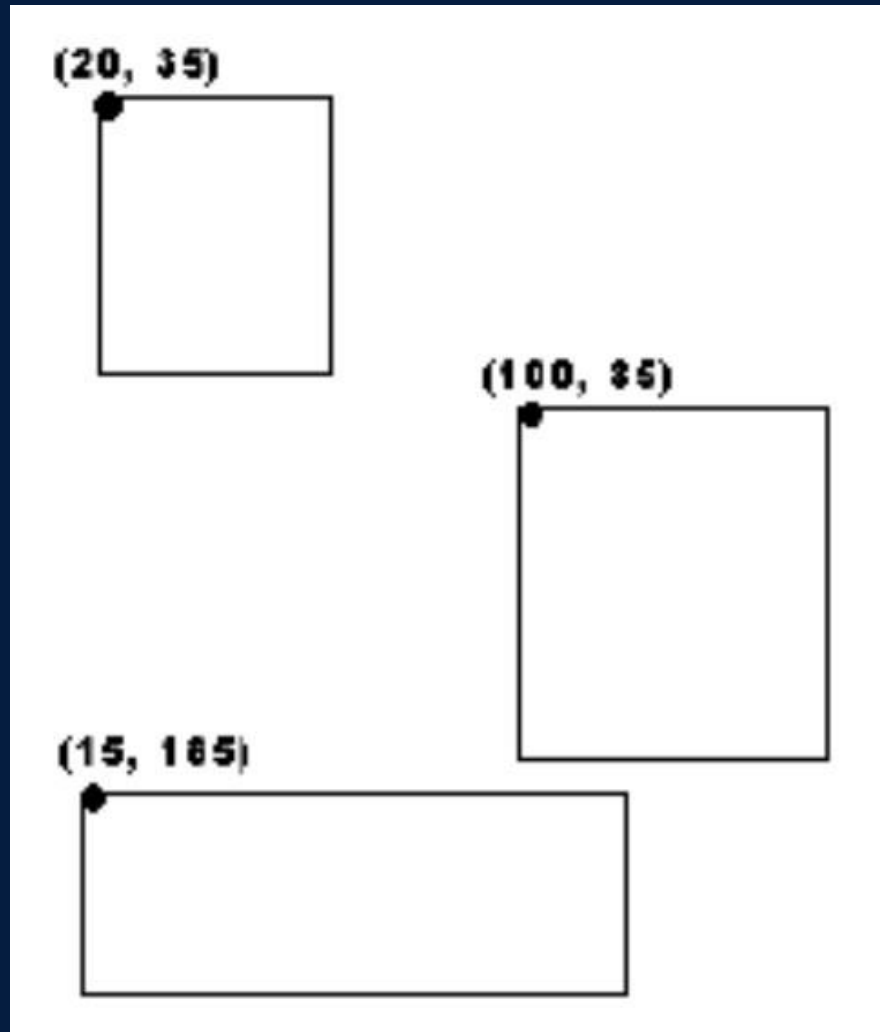
## LINEAR LAYOUT

All elements are arranged in a Each child element is laid out in relation to descending column from top to bottom other child elements. That is, a child or left to right.element appears in relation to the previous child.



## RELATIVE LAYOUT



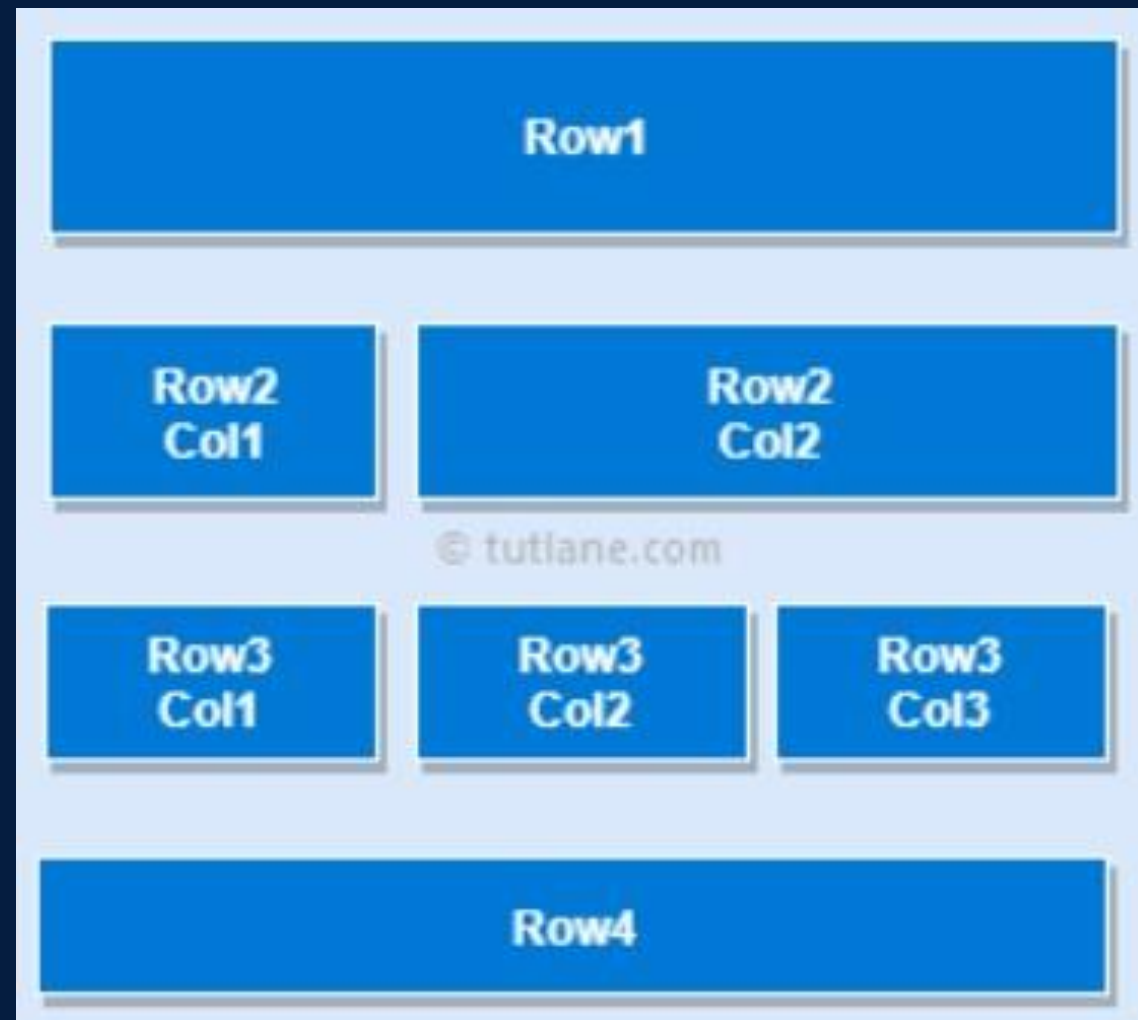


## ABSOLUTE LAYOUTFRAME LAYOUT

Each child is given a specific location within the bounds of the parent layout object. This layout is not suitable for devices with different screen sizes and hence is deprecated.

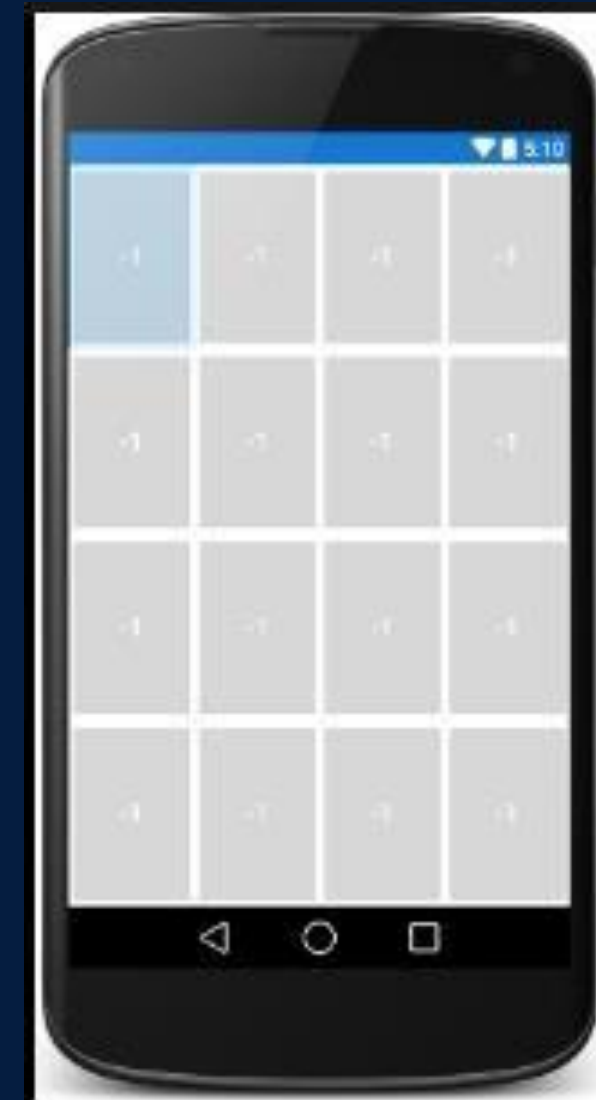
Views added to this are always placed at the top left of the layout. Any other view that is added to the FrameLayout overlaps the previous view; that is, each view stacks on top of the previous one.





### TABLE LAYOUT

The screen is assumed to be divided in table rows, and each of the child elements is arranged in a specific row and column. Child views are arranged in a grid format, that is, in the rows and columns pattern. The



### GRID LAYOUT

views can be placed at the specified row and column location.





# Thank You

