Week 3: Views and Layouts

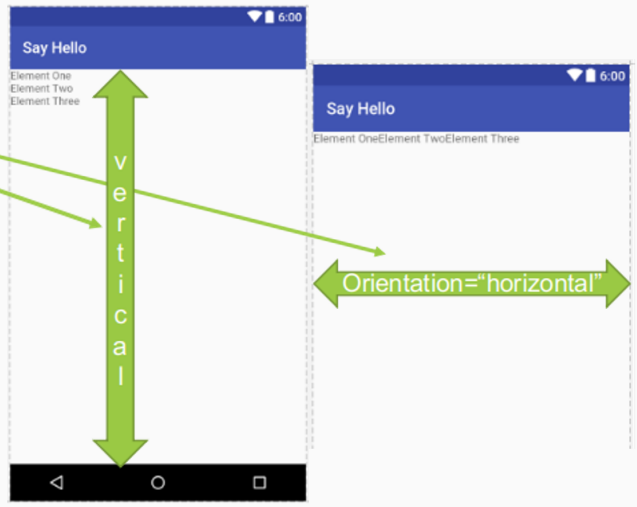
1. Layouts

- LinearLayout
- RelativeLayout
- ConstraintLayout
- TableLayout
- AbsoluteLayout

A. LinearLayout

- Definition: a ViewGroup that lays child View elements vertically or horizontally.

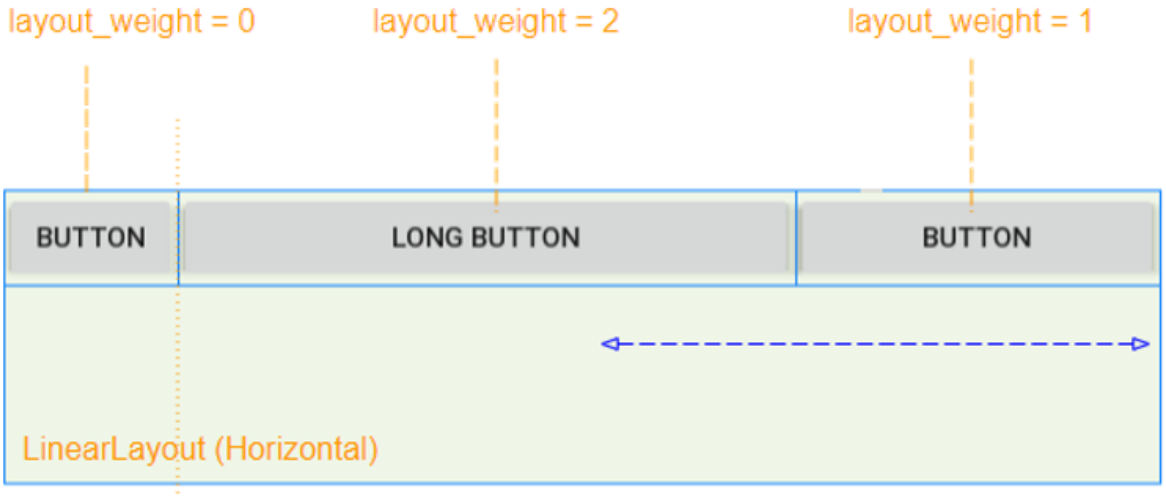
```
<LinearLayout xmlns:android=
    "http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Element One"
        />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Element Two"
        />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Element Three"
        />
</LinearLayout>
```



- Some important attributes:
  - orientation: vertical or horizontal

- layout_width and layout_height (required): determines the width and height of the layout. Can be set to:
  - A fixed value. Ex: 20dp, 30sp…
  - wrap_content: enough to fit all of its content
  - fill_parent or match_parent: equal to its parent's width or height

- weight (an attribute of child elements): determines how much space a child is allowed to take in ratio out of the remaining space (similar to flex in HTML / CSS). Default value is 0

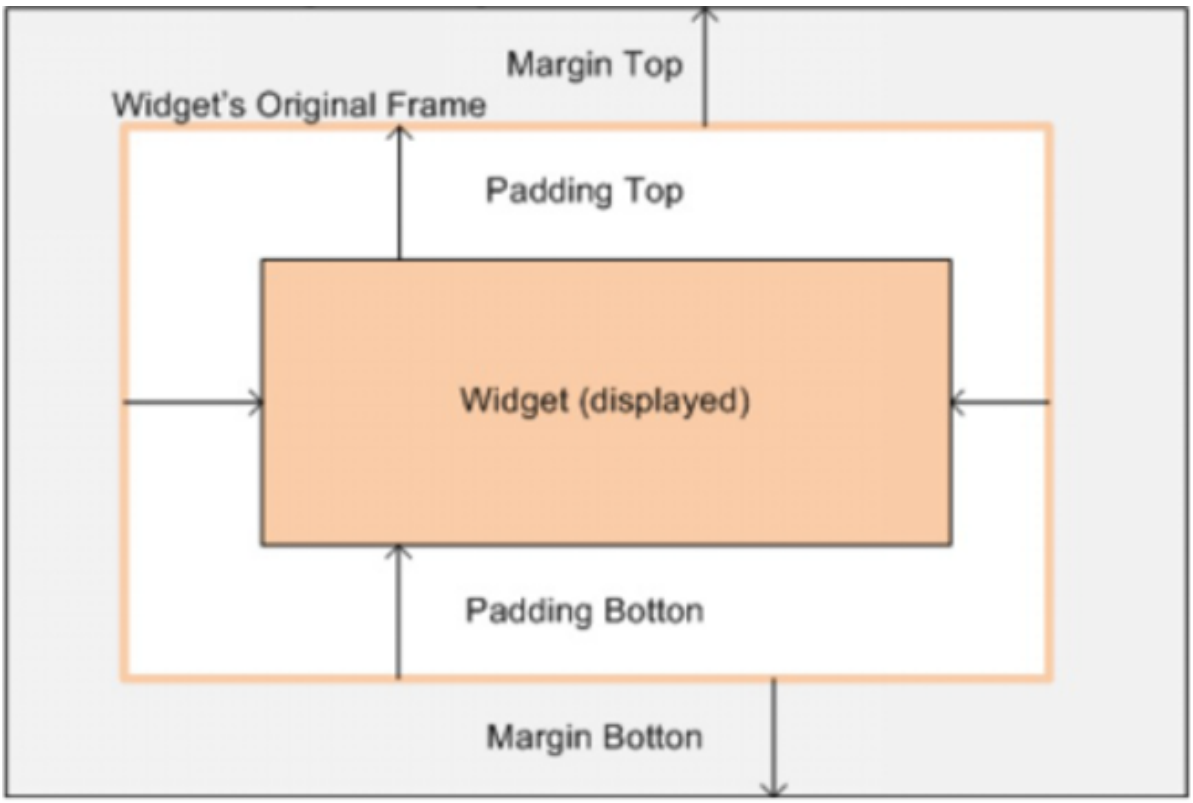


- layout_gravity and gravity: set the position of the layout and its content respectively. Takes values such as center, left, right, top, bottom

```
android:gravity="center"
```


```
android:layout_gravity="center"
```

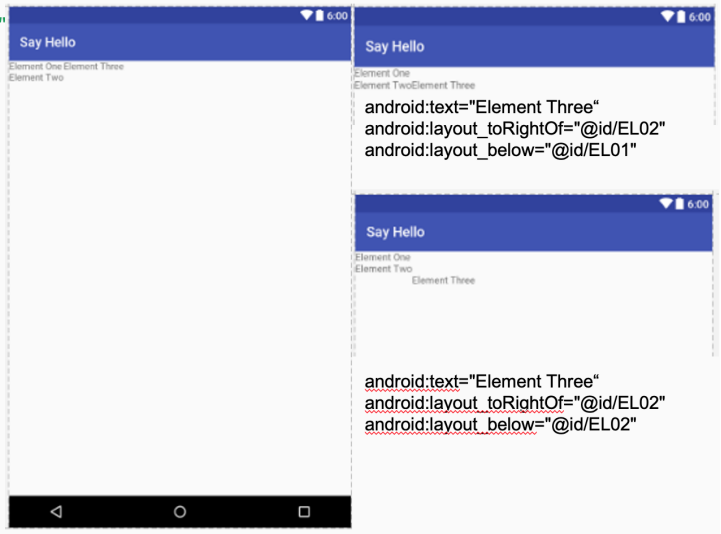
- layout_margin and padding: sets the margin and padding of the layout (similar to margin and padding of HTML / CSS)



B. RelativeLayout

- Definition: a ViewGroup that allows child elements to be laid out relative to its parent and siblings' positions.
- Idea: each child (or parent) element will have an assigned ID (assign with the id attribute), and each of the attributes of the RelativeLayout will use an id as an anchor for it to be positioned.

```
<RelativeLayout xmlns:android=
    "http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:id="@+id/EL01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Element One"
        />
    <TextView
        android:id="@+id/EL02"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Element Two"
        android:layout_below="@id/EL01"
        />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Element Three"
        android:layout_toRightOf="@id/EL02"
        />
</RelativeLayout>
```



- Some attributes:

Tên thuộc tính	Mô tả
android:layout_above	Đặt phần tử hiện tại nằm kế sau phần tử có id được chỉ ra
android:layout_alignBaseline	Đặt phần tử này lên cùng dòng với phần tử có id được chỉ ra
android:layout_alignBottom	Canh sao cho đáy của phần tử hiện thời trùng với đáy của phần tử có id được chỉ ra
android:layout_alignLeft	Đặt cạnh trái của phần tử hiện thời trùng với cạnh trái của phần tử có id được chỉ ra
android:layout_alignParentBottom	Nếu thiết lập là true thì phần tử hiện thời sẽ được canh xuống đáy của phần tử chứa nó
android:layout_alignParentLeft	Nếu được thiết lập là true thì phần tử hiện thời sẽ canh trái so với phần tử chứa nó

android:layout_alignParentRight	Nếu được thiết lập là true thì phần tử hiện thời sẽ canh phải so với phần tử chứa nó
android:layout_alignParentTop	Nếu được thiết lập là true thì phần tử hiện thời sẽ canh lên đỉnh phần tử chứa nó
android:layout_alignRight	Canh cạnh phải của phần tử hiện thời trùng với cạnh phải của phần tử có id được chỉ ra
android:layout_alignTop	Canh đỉnh của phần tử hiện thời trùng với đỉnh của phần tử có id được chỉ ra
android:layout_alignWithParentIfMissing	Nếu thiết lập là true, thì phần tử sẽ được canh theo phần tử chứa nó nếu các thuộc tính canh của phần tử không có.
android:layout_below	Đặt phần tử hiện thời ngay sau phần tử có id được chỉ ra.
android:layout_centerHorizontal	Nếu thiết lập là true thì phần tử hiện thời sẽ được canh giữa theo chiều ngang phần tử chứa nó.

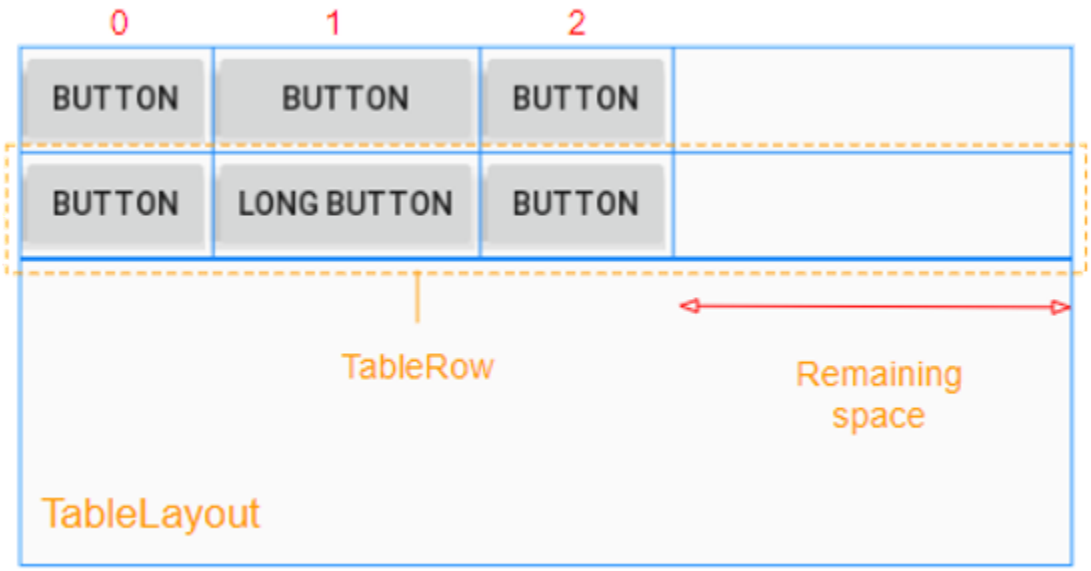
android:layout_centerInParent	Nếu thiết lập là true thì phần tử hiện thời sẽ được canh chính giữa theo chiều phải trái và trên dưới so với phần tử chứa nó.
android:layout_centerVertical	Nếu thiết lập là true thì phần tử hiện thời sẽ được canh chính giữa theo chiều dọc phần tử chứa nó.
android:layout_toLeftOf	Đặt cạnh phải của phần tử hiện thời trùng với cạnh trái của phần tử có id được chỉ ra.
android:layout_toRightOf	Đặt cạnh trái của phần tử hiện thời trùng với cạnh phải của phần tử có id được chỉ ra.

C. ConstraintLayout

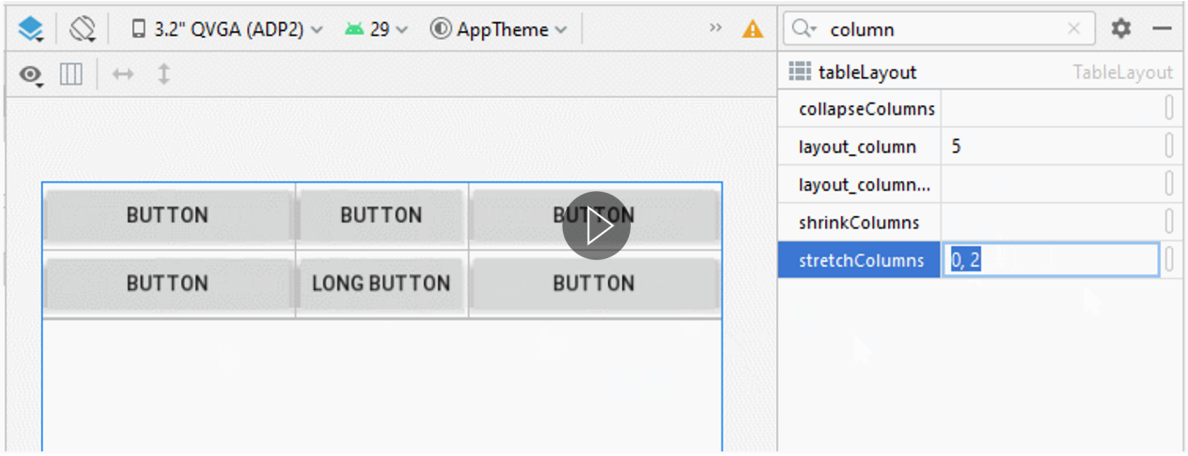
- Similar to RelativeLayout, but with powerful drag and drop ability (if you use it).
- Some important attributes:
  - fixed: adds a fixed constraint to the parent or its siblings.
  - match_constraint: similar to match_parent of RelativeLayout.
  - wrap_content: similar to RelativeLayout.

D. TableLayout

- Definition: a ViewGroup that lays child elements into rows and columns.
- Inside nests a bunch of TableRow elements, and inside each TableRow is a bunch of child View elements being spanned horizontally.

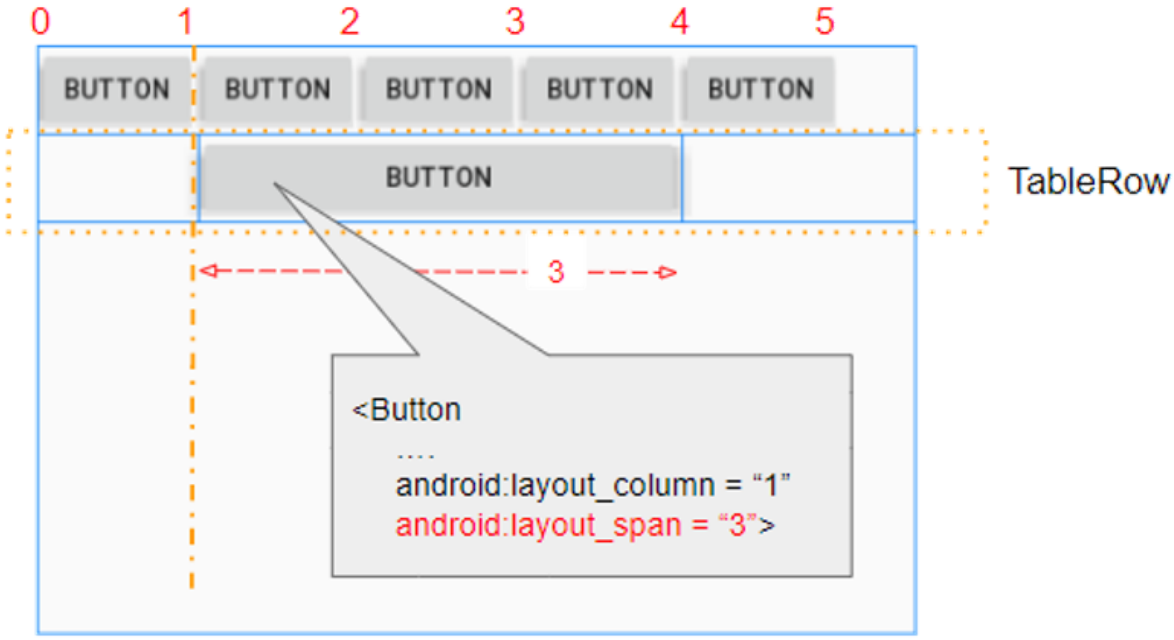


- Some important attributes:
  - stretchColumns: takes the value of a string representing the columns that will be stretched to fill the remaining space. Ex: android.stretchColumns=”0, 2” ⇒ stretches the first and third column to fill the remaining space.



  - shrinkColumns: syntax is similar to stretchColumns. Shrinks the assigned column to prevent overflowing (if exists).

- layout_span and layout_column:
  - layout_span: apply to the child View to span it across x columns (similar to merging cells in Excel)
  - layout_column: apply to the child View to put it in column x



E. AbsoluteLayout

- Positions the child elements using x, y coordinates with attributes layout_x and layout_y. These attributes are applied to the child elements themselves.
- The x and y coordinates are relative to its parent.

```
<AbsoluteLayout xmlns:android=
    "http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Element One"
        />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Element Two"
        android:layout_x="30px"
        android:layout_y="30px"
        />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Element Three"
        android:layout_x="50px"
        android:layout_y="50px"
        />
</AbsoluteLayout>
```