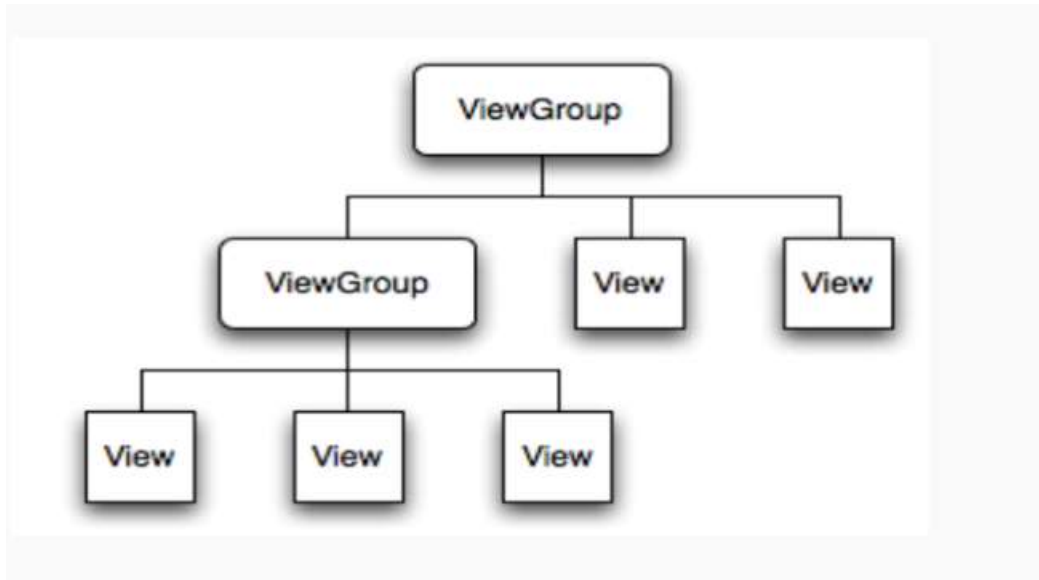


# Views And Layout

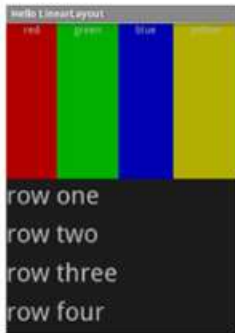
## 1. ViewGroup and View



View là các thành phần nhỏ lẻ tạo nên ứng dụng, ví dụ: TextView, Button, ListView

ViewGroup là tập hợp các View, View Group có thể chứa ViewGroup và các View

## 2. Linear Layout

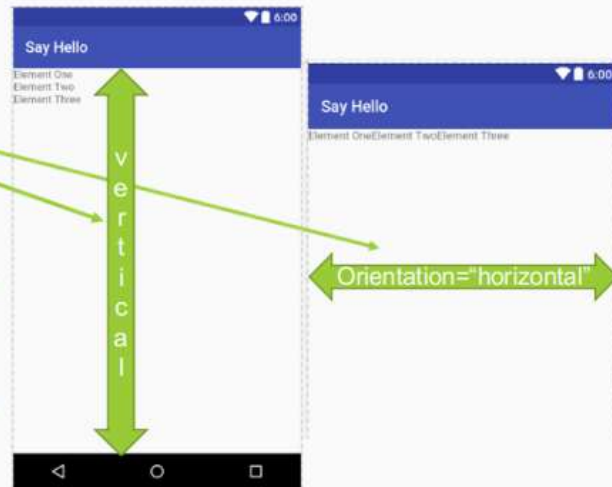


### Linear Layout

A LinearLayout is a GroupView that will lay child View elements vertically or horizontally.

Linear Layout là layout sắp xếp các thành phần con theo chiều ngang hoặc theo chiều dọc

```
<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>
```

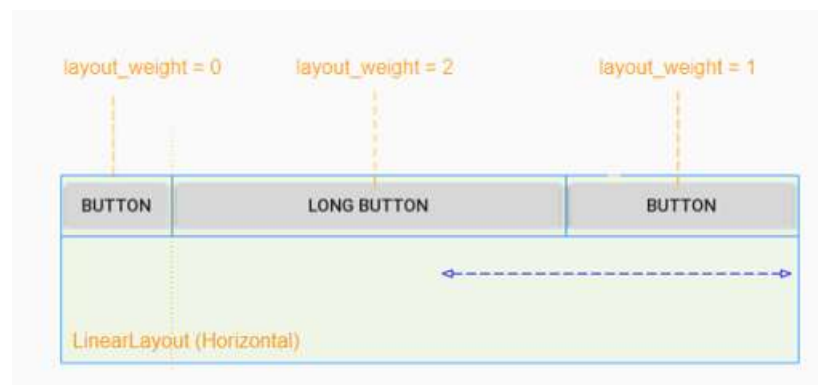


Nếu muốn các thành phần xếp theo cột, sử dụng orientation="vertical" hoặc "horizontal" với các thành phần theo chiều ngang. Ngoài ra cũng có thể thay đổi bằng setOrientation để set dynamic

Với các thành phần, kích thước của thành phần có các giá trị như sau

- Một giá trị cụ thể: 200dip
- Vừa đủ nội dung: wrap\_content
- Bằng độ lớn của paren: match\_parent

Weight: trọng số để xác định tỉ lệ giữa các thành phần.



Gravity để xác định cách căn lề: left, center, right, top, bottom

Margin, padding

### 3. Relative Layout



#### Relative Layout

A `RelativeLayout` is a `ViewGroup` that allows you to layout child elements in positions relative to the parent or siblings elements.

Relative Layout là layout mà các thành phần của nó ở vị trí tương đối so với thành phần cha hoặc các thành phần anh chị em

<u>Tên thuộc tính</u>	<u>Mô tả</u>
<u>android:layout_above</u>	Đặt phần tử hiện tại nằm kế sau phần tử có id được chỉ ra
<u>android:layout_alignBaseline</u>	Đặt phần tử này lên cùng dòng với phần tử có id được chỉ ra
<u>android:layout_alignBottom</u>	Canh sao cho đáy của phần tử hiện thời trùng với đáy của phần tử có id được chỉ ra
<u>android:layout_alignLeft</u>	Đặt cạnh trái của phần tử hiện thời trùng với cạnh trái của phần tử có id được chỉ ra
<u>android:layout_alignParentBottom</u>	Nếu thiết lập là true thì phần tử hiện thời sẽ được canh xuống đáy của phần tử chứa nó
<u>android:layout_alignParentLeft</u>	Nếu được thiết lập là true thì phần tử hiện thời sẽ canh trái so với phần tử chứa nó
<u>android:layout_alignParentRight</u>	Nếu được thiết lập là true thì phần tử hiện thời sẽ canh phải so với phần tử chứa nó
<u>android:layout_alignParentTop</u>	Nếu được thiết lập là true thì phần tử hiện thời sẽ canh lên đỉnh phần tử chứa nó
<u>android:layout_alignRight</u>	Canh cạnh phải của phần tử hiện thời trùng với cạnh phải của phần tử có id được chỉ ra
<u>android:layout_alignTop</u>	Canh đỉnh của phần tử hiện thời trùng với đỉnh của phần tử có id được chỉ ra
<u>android:layout_alignWithParentIfMissing</u>	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.
<u>android:layout_below</u>	Đặt phần tử hiện thời ngay sau phần tử có id được chỉ ra.
<u>android:layout_centerHorizontal</u>	Nếu thiết lập là true thì phần tử hiện thời sẽ được canh giữa theo chiều ngang phần tử chứa nó.

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

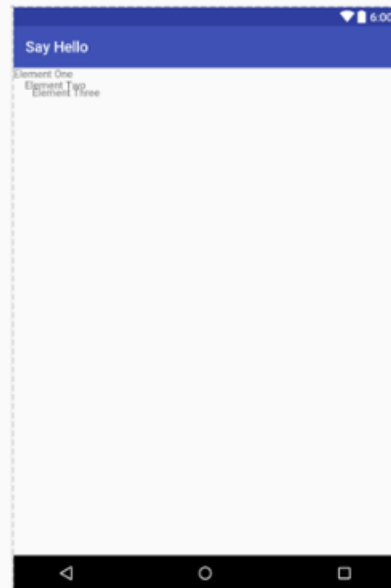
## 4. Table Layout



Là layout mà sắp xếp các thành phần của nó theo hàng hoặc cột

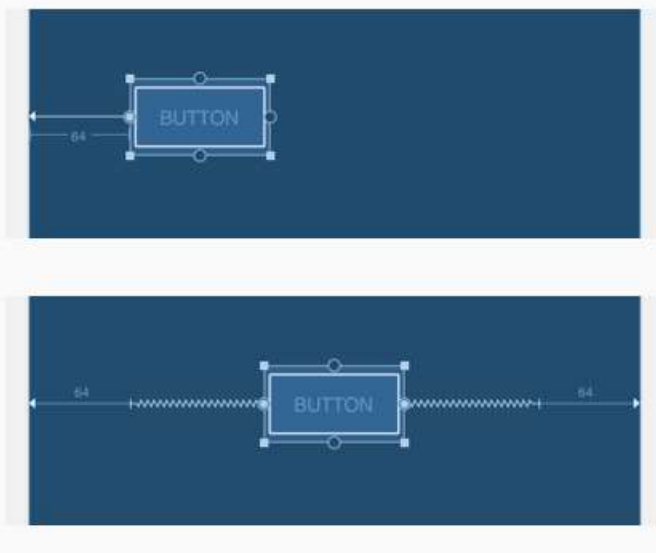
## 5. Absolute Layout

```
<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>
```



Chỉ định các vị trí con thông qua tọa độ x, y

## 6. ConstraintLayout



Tương tự như relative layout nhưng hỗ trợ kéo thả mạnh mẽ

Kích thước của một thành phần được xác định như sau:

- Fixed: chỉ định cứng,
- Match\_constraint: full width parent
- Wrap\_content