

## Chương 02

# Xây dựng bố cục giao diện với Android Layout

1. Sử dụng *AbsoluteLayout*
2. Sử dụng *FrameLayout*
3. Sử dụng *LinearLayout*
4. So sánh *wrap\_content* và *match\_parent*
5. So sánh *gravity* và *layout\_gravity*
6. Sử dụng *weightSum* và *layout\_weight*
7. Xây dựng giao diện với *LinearLayout*
8. So sánh *padding* và *margin*
9. Xây dựng giao diện với *RelativeLayout*
10. Sử dụng *TableLayout*
11. *Android Screen Orientation*

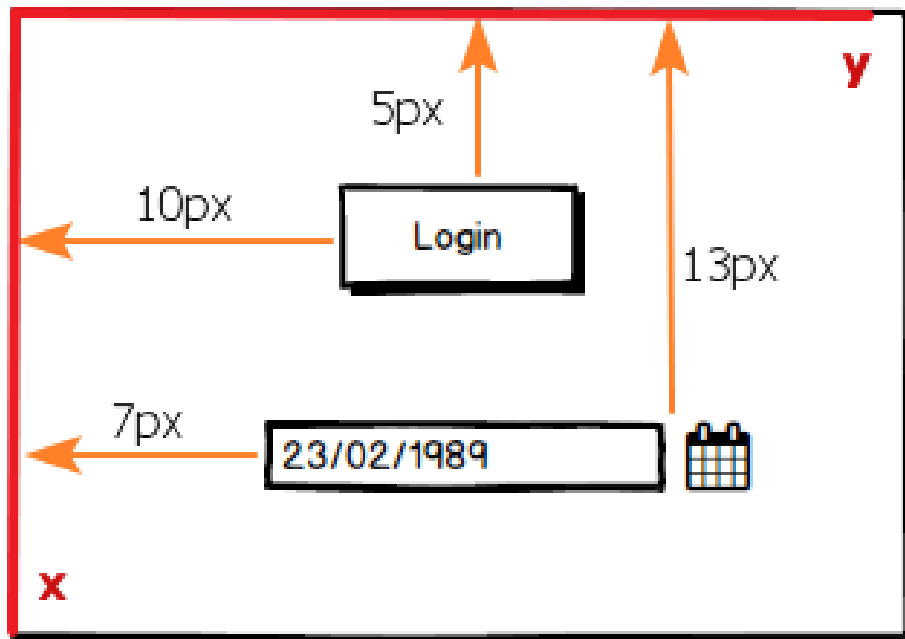
# Android Layout

*Layout định nghĩa cấu trúc và bố trí giao diện cho người dùng. Các layout phổ biến:*

- *AbsoluteLayout (không còn được sử dụng)*
- *FrameLayout*
- *LinearLayout*
- *RelativeLayout*
- *TableLayout*



# AbsoluteLayout



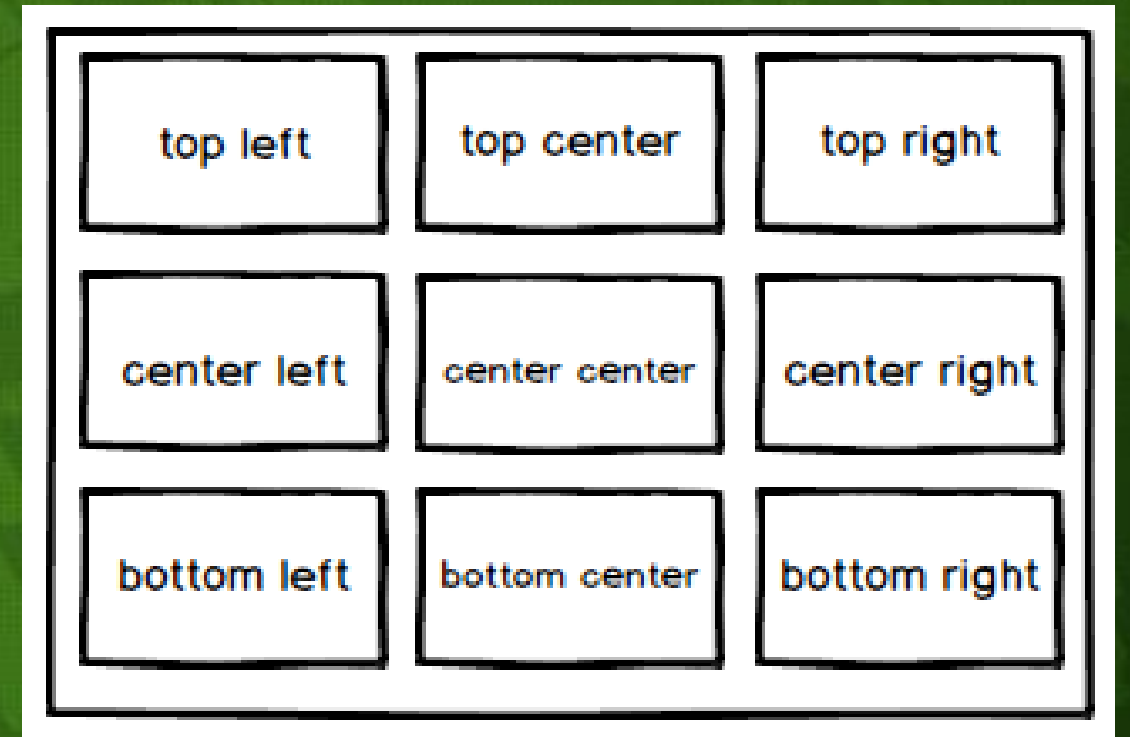
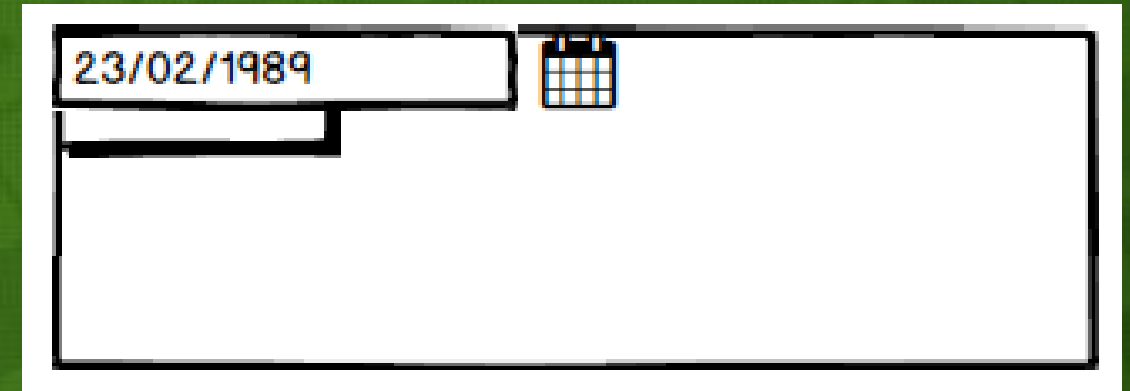
```
<AbsoluteLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:id="@+id/btnBack"
        android:text="Back"
        android:layout_x="10px"
        android:layout_y="5px"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</AbsoluteLayout>
```

- Thiết lập các control theo vị trí tùy thích. Phải xác định rõ vị trí tọa độ x và y trên màn hình làm cho việc xây dựng giao diện rất “cứng” và không còn linh hoạt
- Đang dần dần bị “deprecated”

# FrameLayout

- Phù hợp với giao diện có 1 control duy nhất, nếu có nhiều control các control sẽ nằm đè lên nhau tại góc trái màn hình
- Sử dụng thuộc tính `layout_gravity` để định lại vị trí các control

`android:layout_gravity="center"`

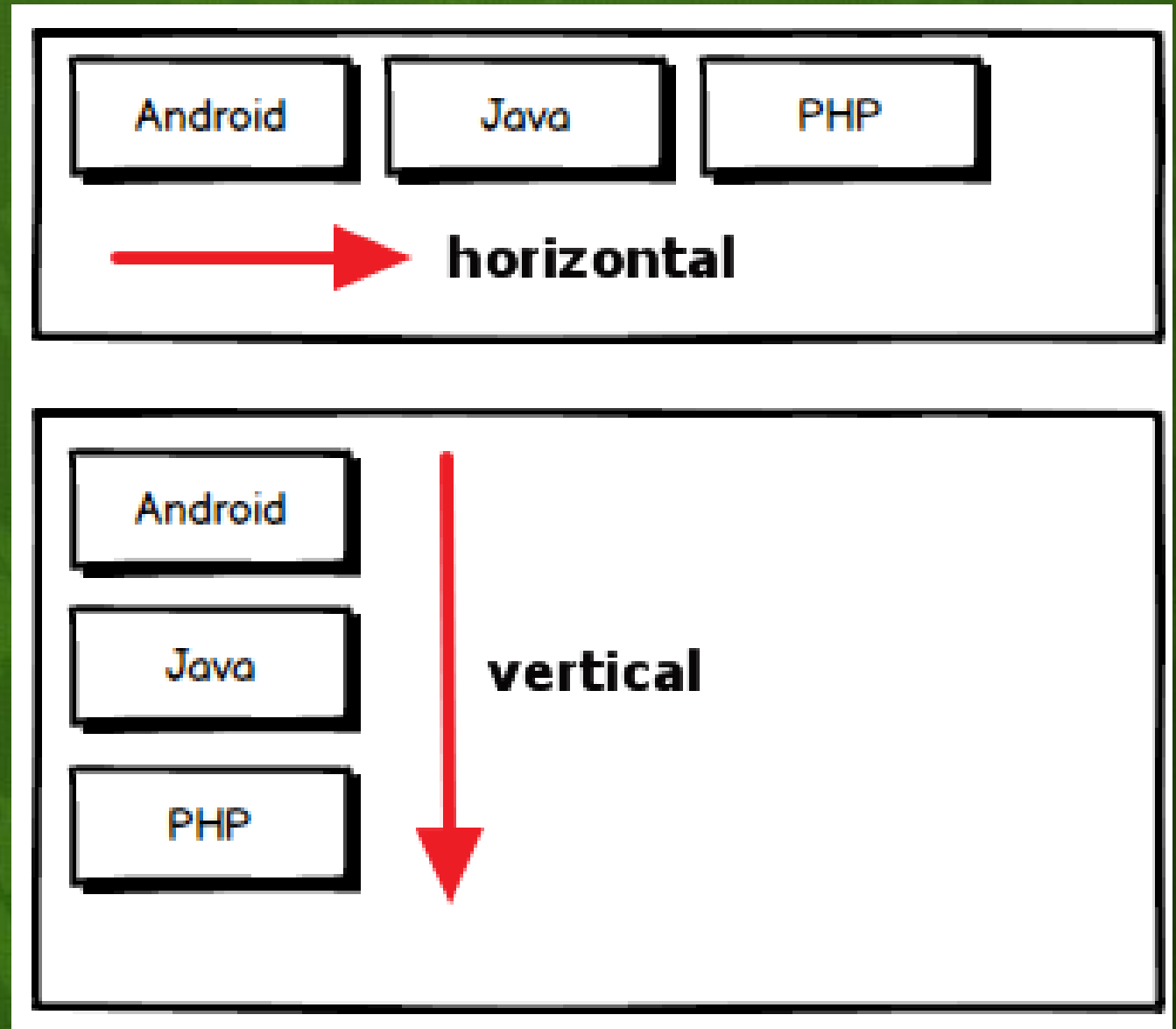


# LinearLayout

*Sắp xếp các control theo 2 hướng:*

- **Horizontal:** *Hướng từ trái qua phải (default)*
- **Vertical:** *Hướng từ trên xuống dưới*

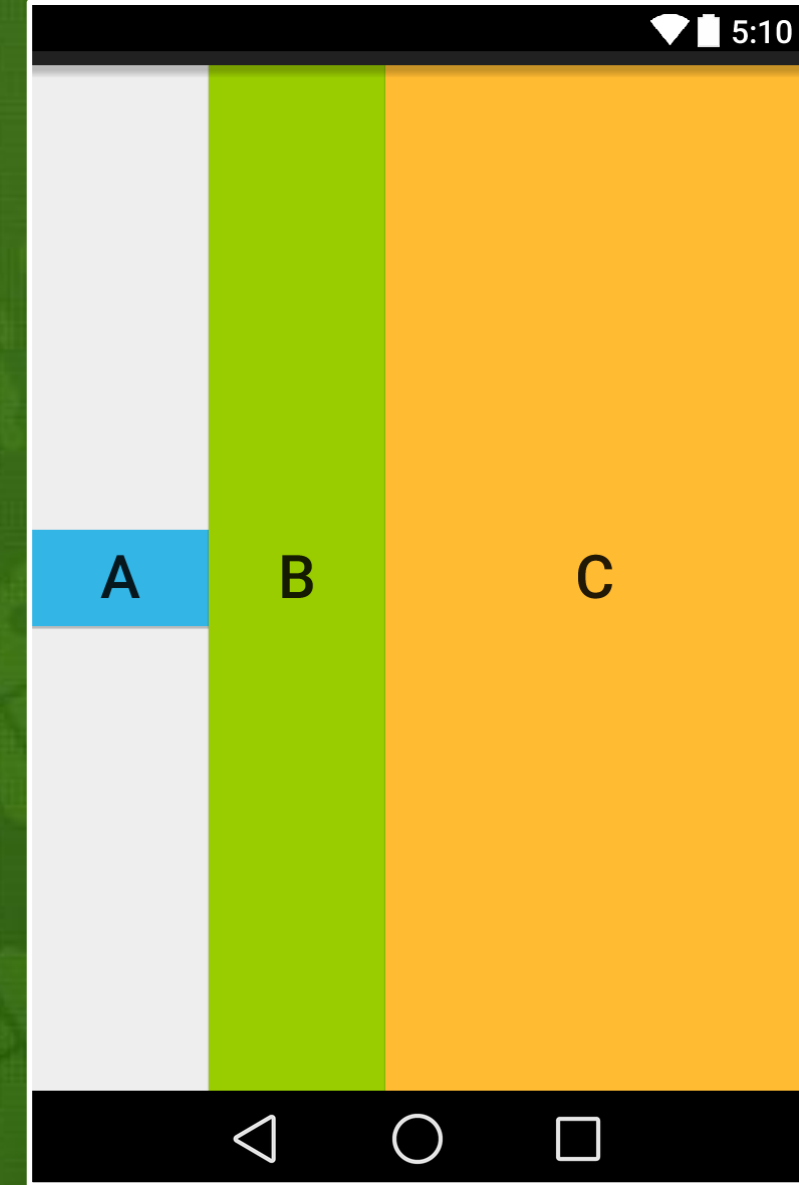
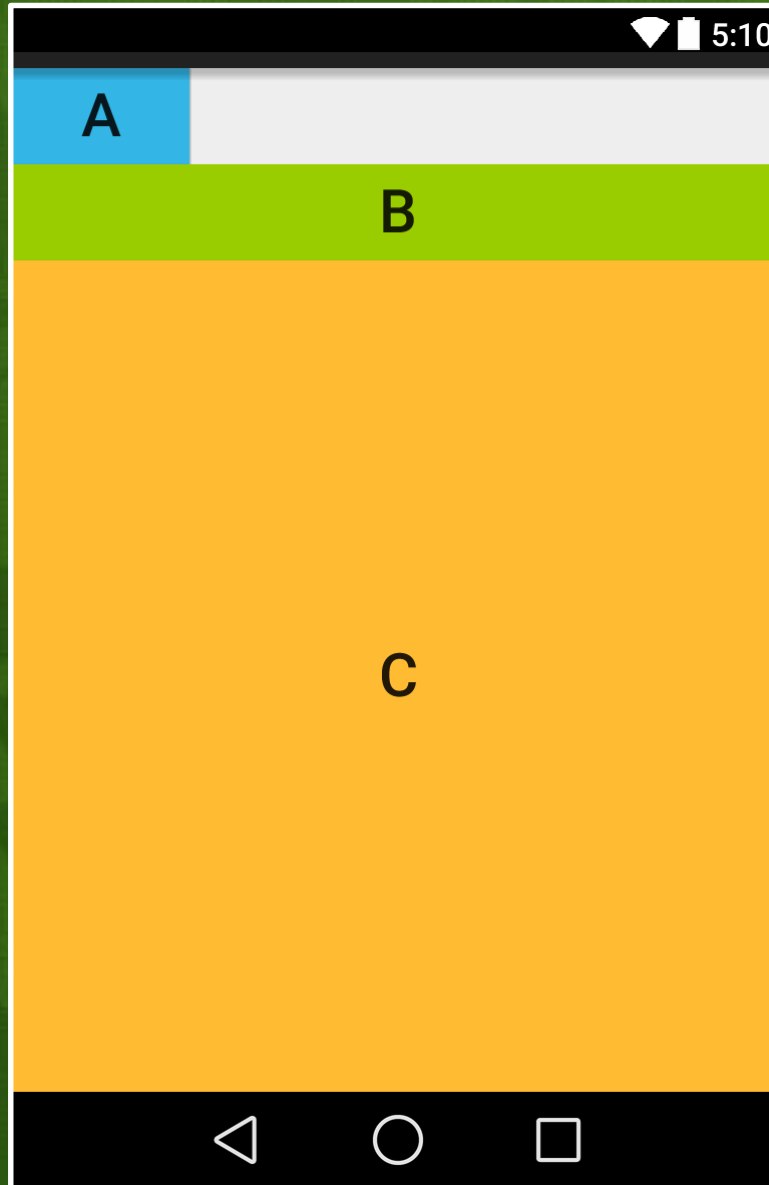
*Trong một giao diện chúng ta có thể kết hợp nhiều loại layout với nhau*



# wrap\_content vs match\_parent

*match\_parent* và *fill\_parent* thiết lập kích thước (width và height) của một view giống như view cha của nó

*wrap\_content* thiết lập kích thước của một view đúng như phần nội dung mà nó chứa





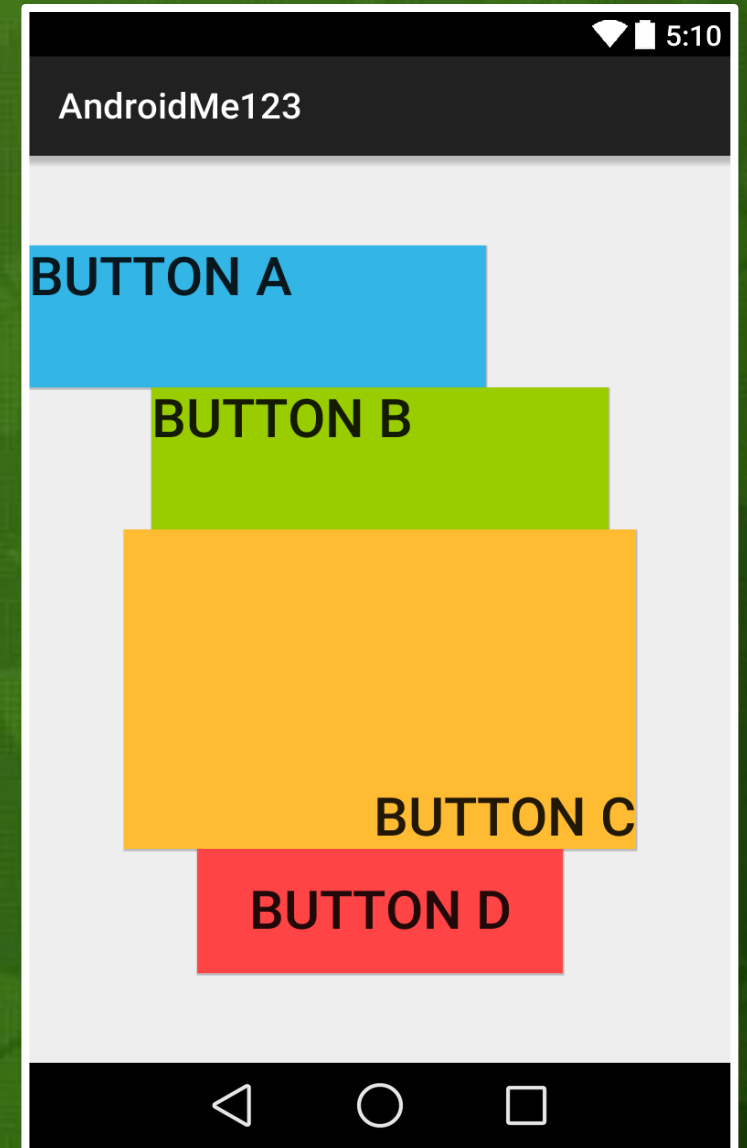
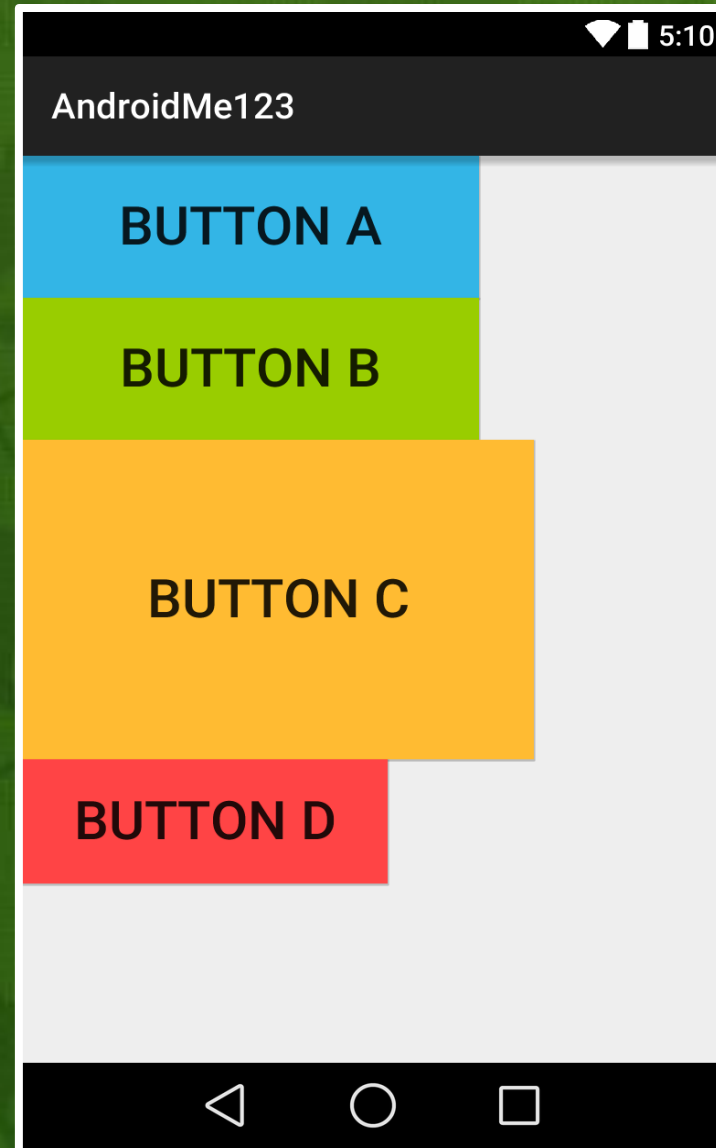
# gravity vs layout\_gravity

## *gravity*

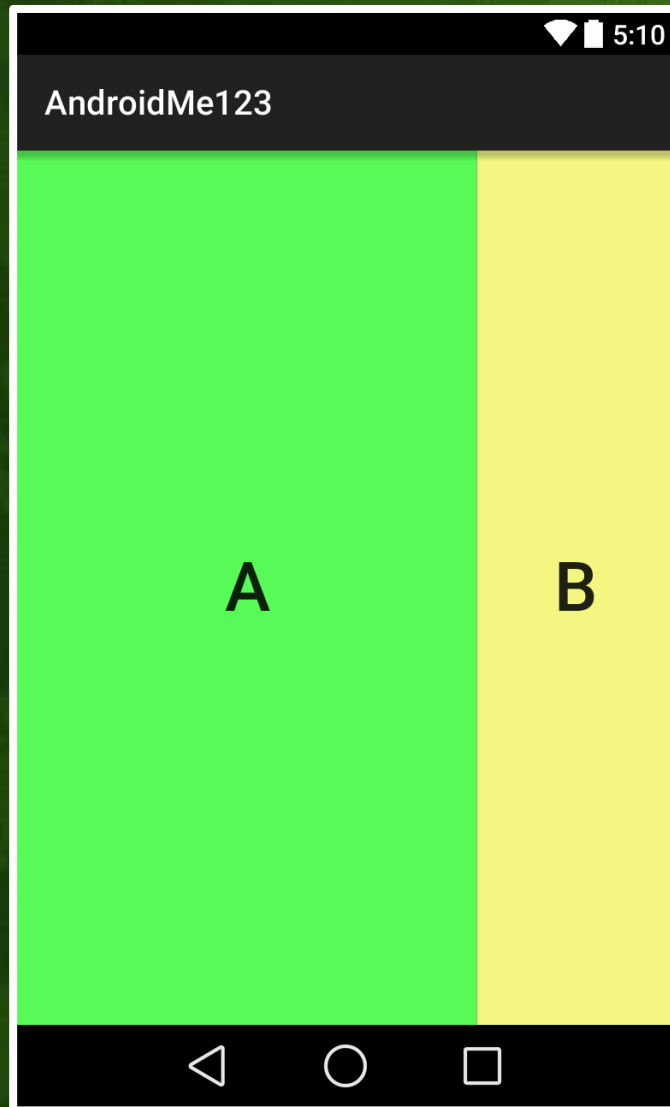
*Inside gravity of the View  
(text-align in CSS)*

## *layout\_gravity*

*Outside gravity of the  
View (float in CSS)*



# weightSum & layout\_weight (1)

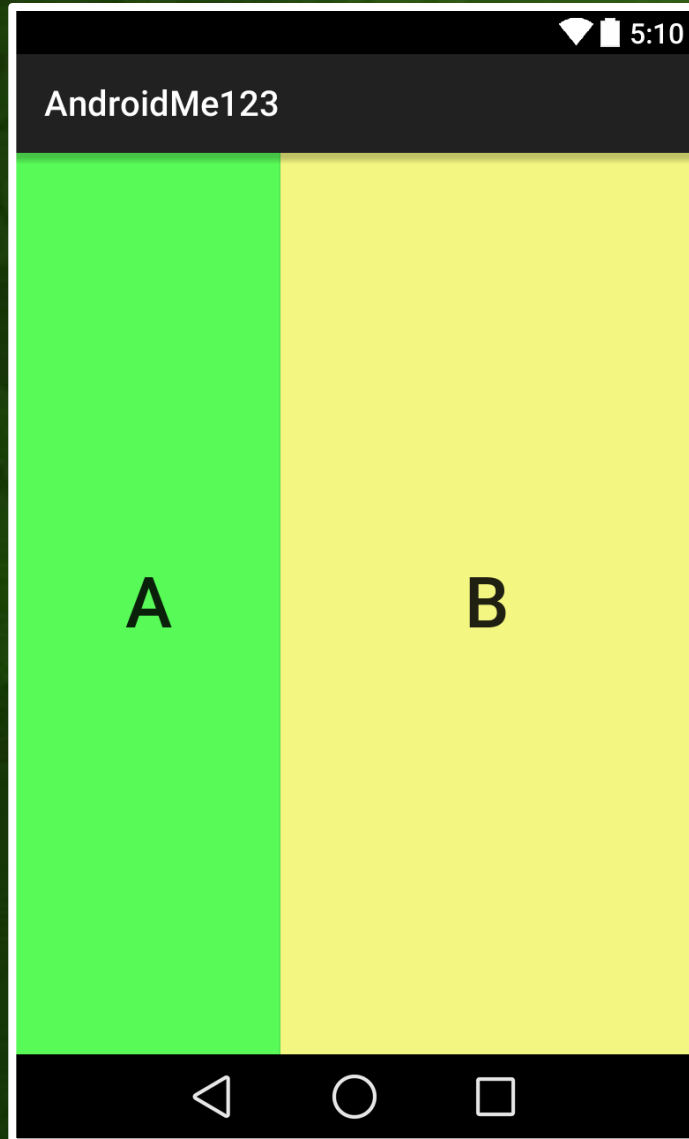


```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="100"
>
    <Button
        android:layout_weight="30"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
    />

    <Button
        android:layout_weight="70"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
    />
</LinearLayout>
```



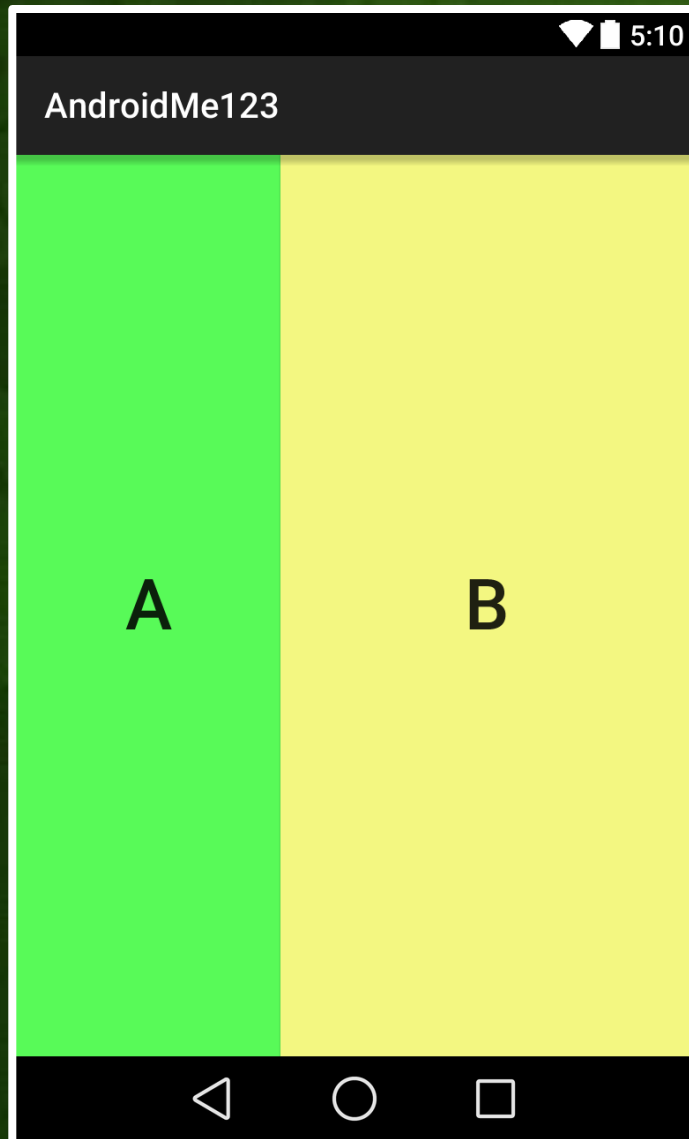
# weightSum & layout\_weight (2)



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:weightSum="100"
>
    <Button
        android:layout_weight="30"
        android:layout_width="wrap_content"
        android:layout_height="match_parent" />

    <Button
        android:layout_weight="70"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"/>
</LinearLayout>
```

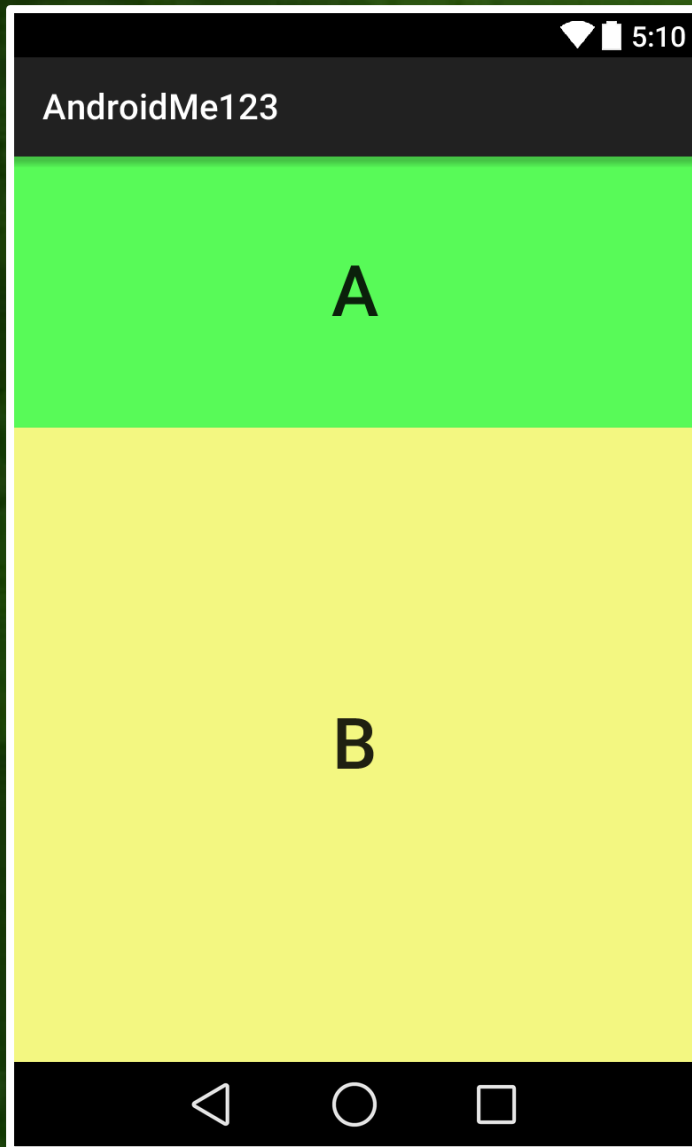
# weightSum & layout\_weight (3)



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:weightSum="100"
>
    <Button
        android:layout_weight="30"
        android:layout_width="0dp"
        android:layout_height="match_parent" />

    <Button
        android:layout_weight="70"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
</LinearLayout>
```

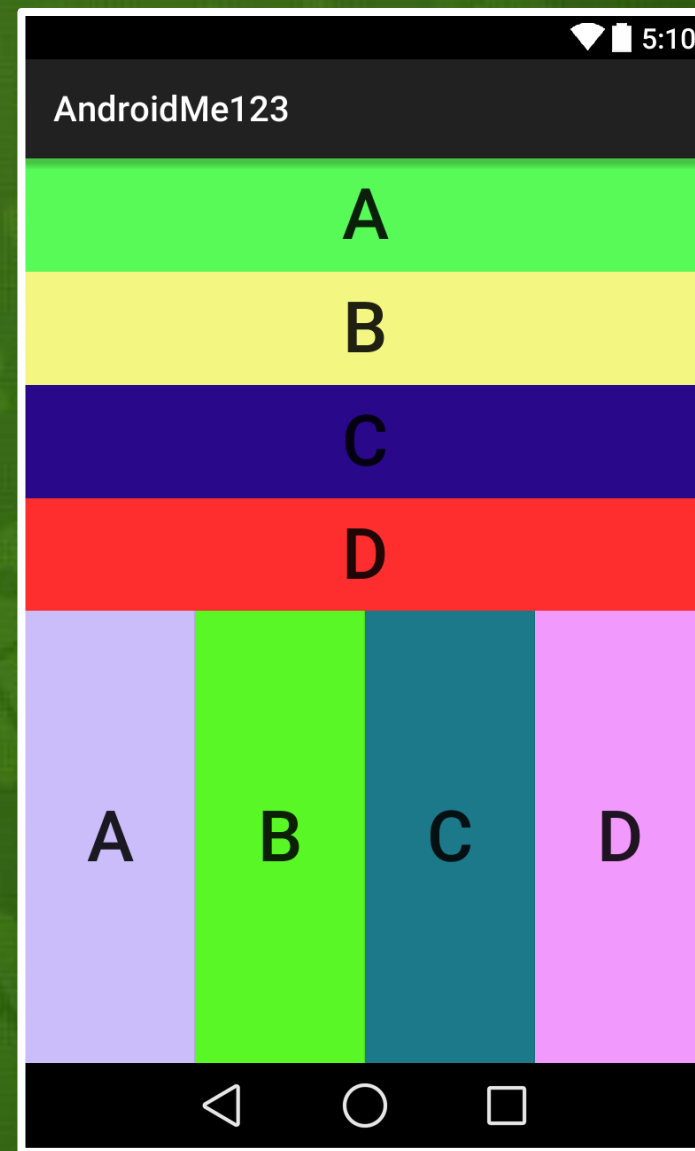
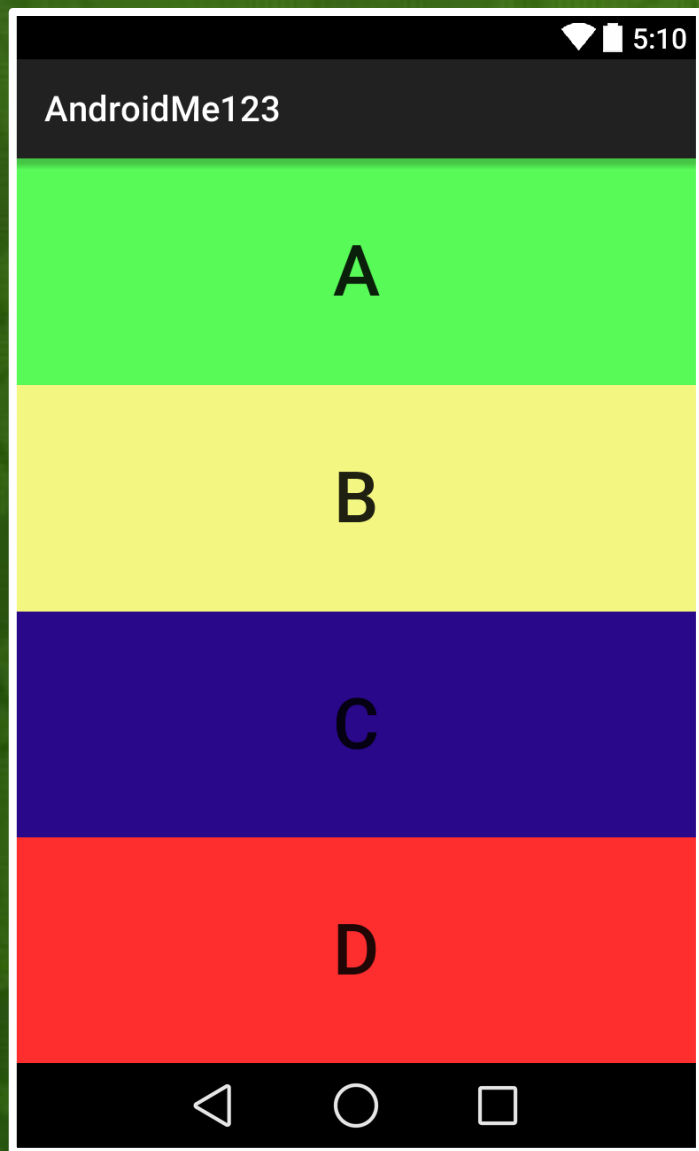
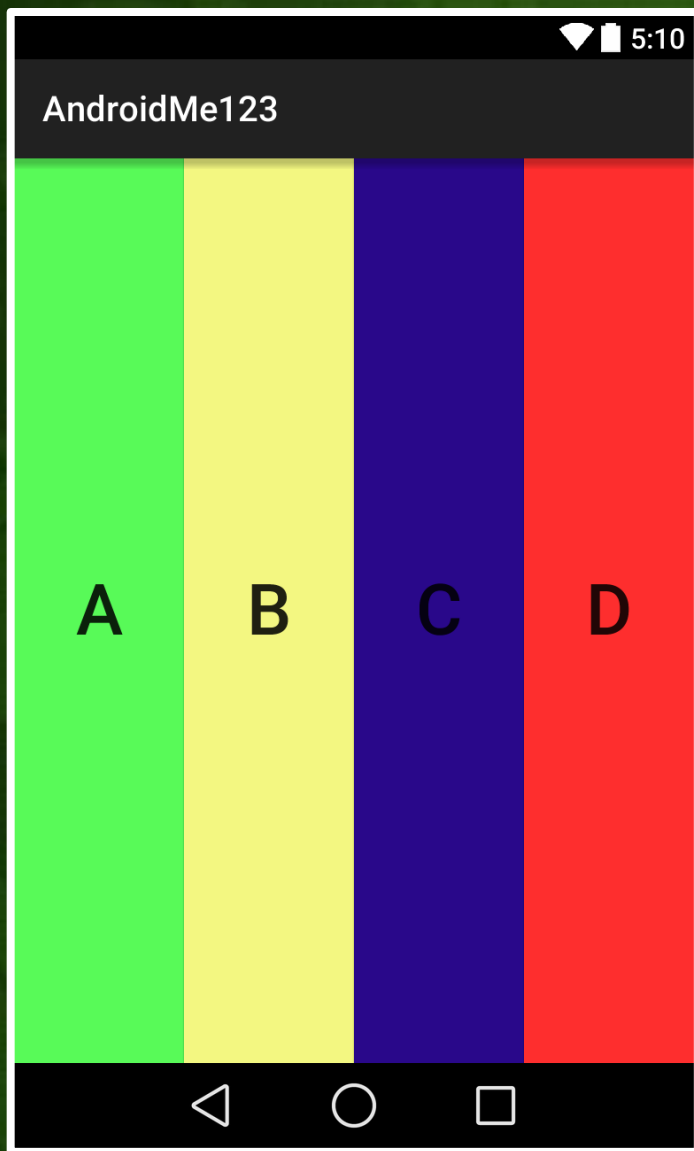
# weightSum & layout\_weight (4)



```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    android:weightSum="100" >  
    <Button  
        android:layout_weight="30"  
        android:layout_width="match_parent"  
        android:layout_height="0dp"/>  
  
    <Button  
        android:layout_weight="70"  
        android:layout_width="match_parent"  
        android:layout_height="0dp" />  
</LinearLayout>
```

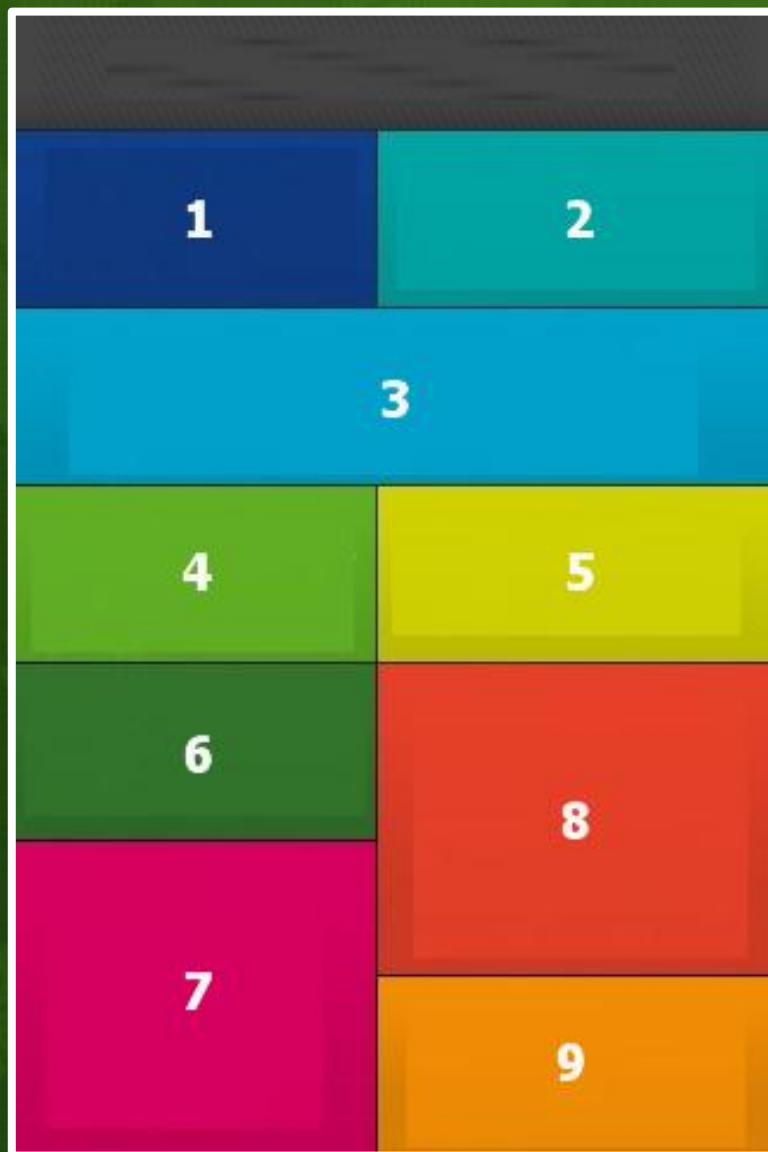


# Thực hành LinearLayout (1)



## Thực hành LinearLayout (2)

*LinearLayout*  
*vertical*  
*weightSum = 6*



$\text{LinearLayout (H)} = B + B$

Button

$\text{LinearLayout (H)} = B + B$

$\text{LinearLayout (H)}$

- $\text{LinearLayout (V)} = B + B$
- $\text{LinearLayout (V)} = B + B$

# Thực hành LinearLayout (3)

*LinearLayout*  
*vertical*  
*weightSum = 6*



$\text{LinearLayout (H)} = B + B$

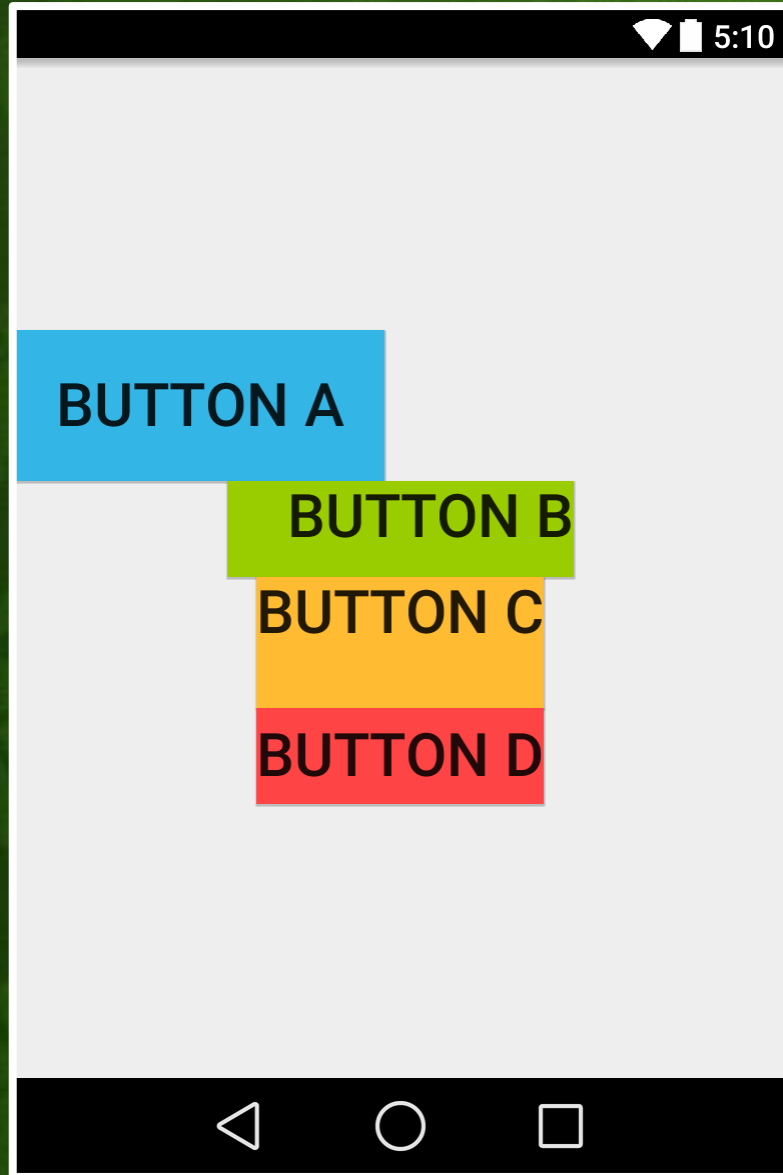
Button

$\text{LinearLayout (H)}$

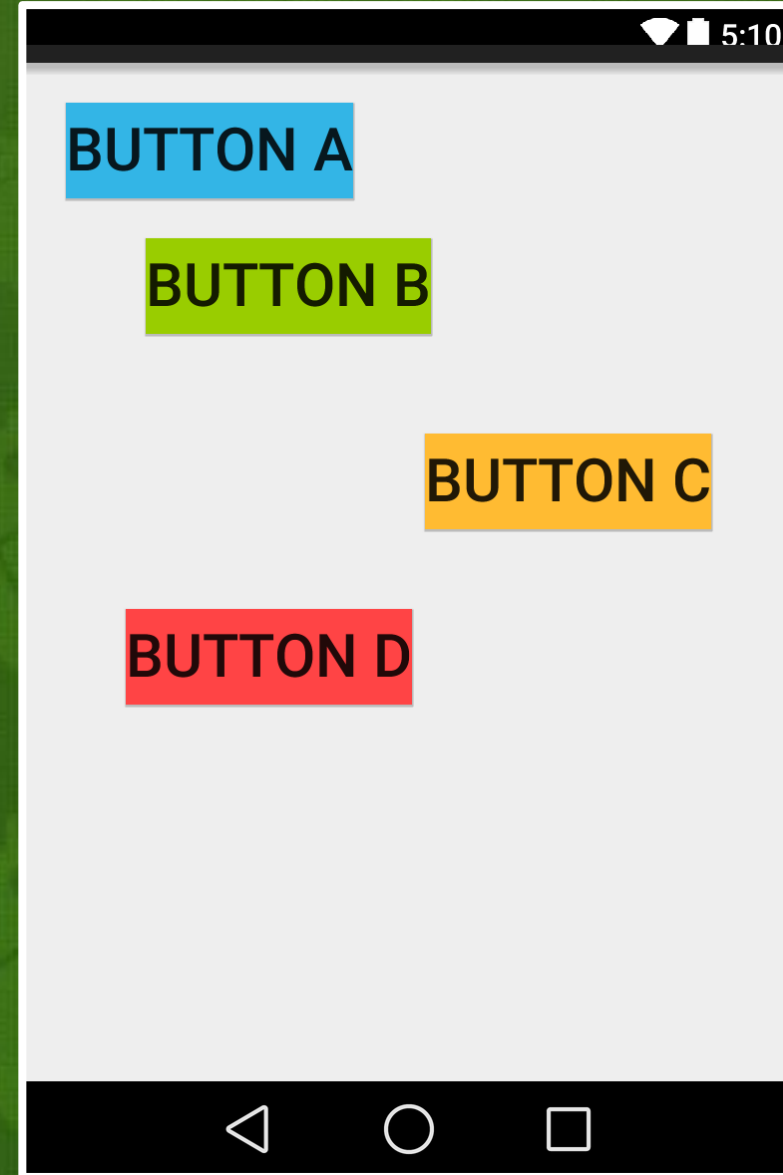
- $\text{LinearLayout (V)} = B + B + B$
- $\text{LinearLayout (V)} = B + B + B$



# Margin vs Padding



padding

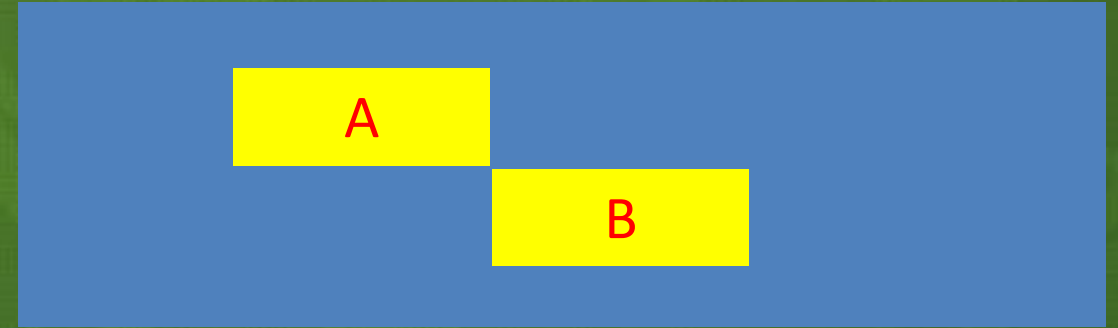


margin

# RelativeLayout

*Sắp xếp các View theo những vị trí tương đối với nhau. Vị trí mỗi View được sắp xếp dựa vào 2 yếu tố:*

- Vị trí so với anh em của nó*
- Vị trí so với thành phần chứa nó*

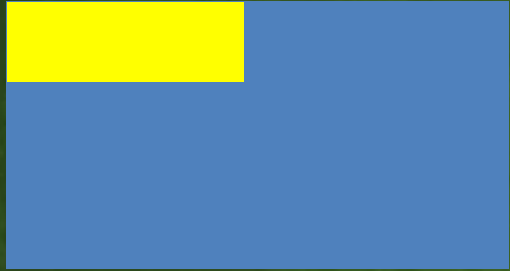


A định vị trí so với B (anh em)

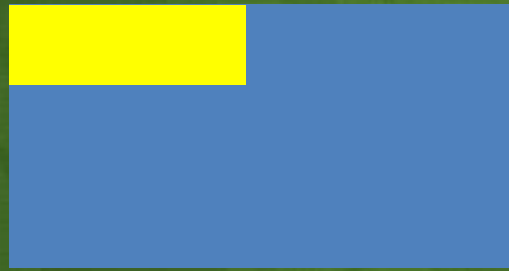


A định vị trí so với khối chứa nó

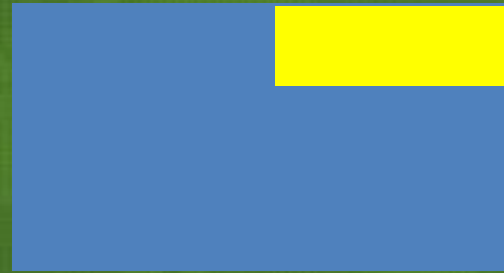
# RelativeLayout – Vị trí trong mối quan hệ cha con



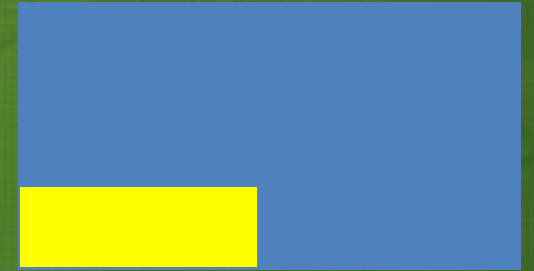
layout\_alignParentLeft



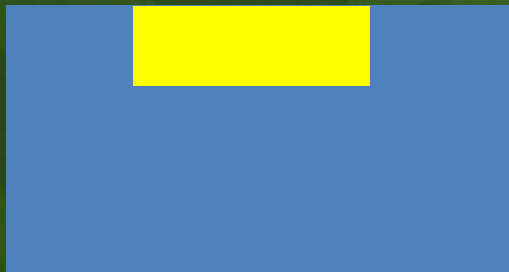
layout\_alignParentTop



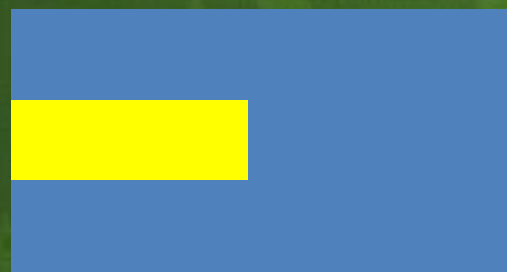
layout\_alignParentRight



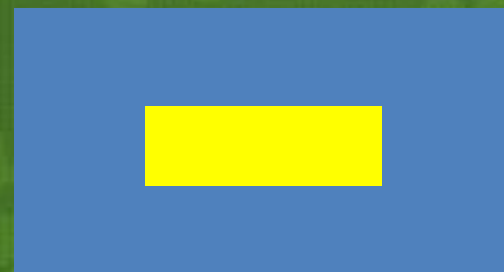
layout\_alignParentBottom



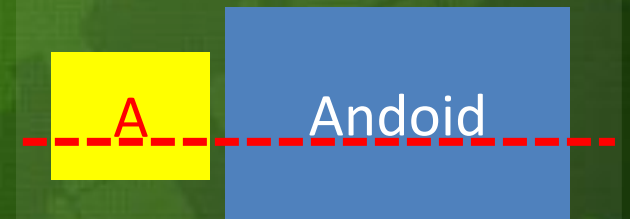
layout\_centerHorizontal



layout\_centerVertical



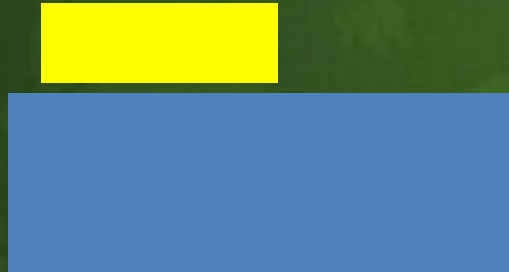
layout\_centerInParent



layout\_alignBaseline



# RelativeLayout – Vị trí trong mối quan hệ anh em



layout\_above



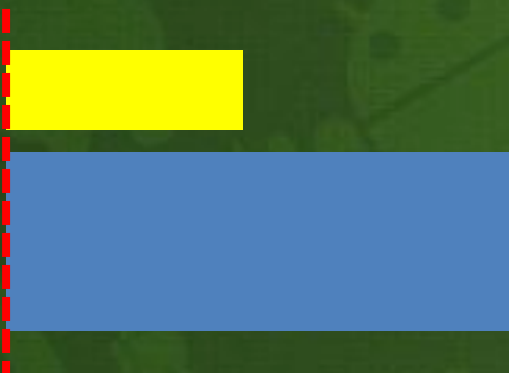
layout\_below



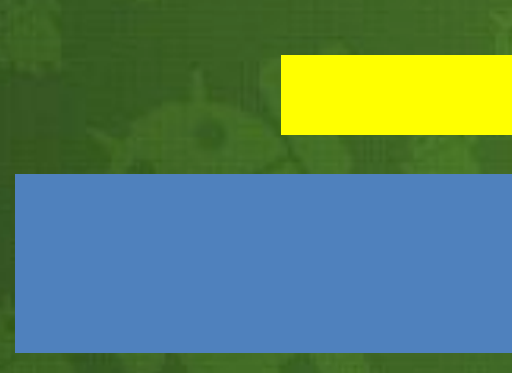
layout\_toLeftOf



layout\_toRightOf



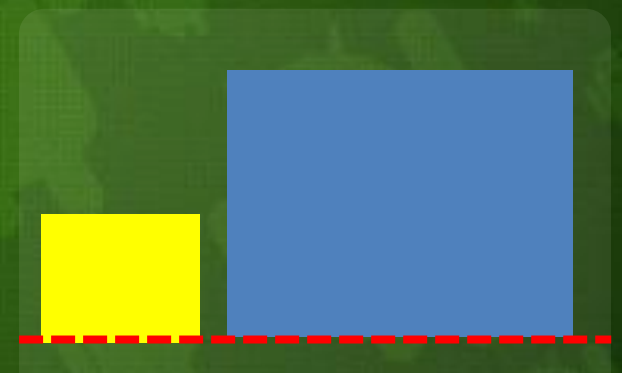
layout\_alignLeft



layout\_alignRight

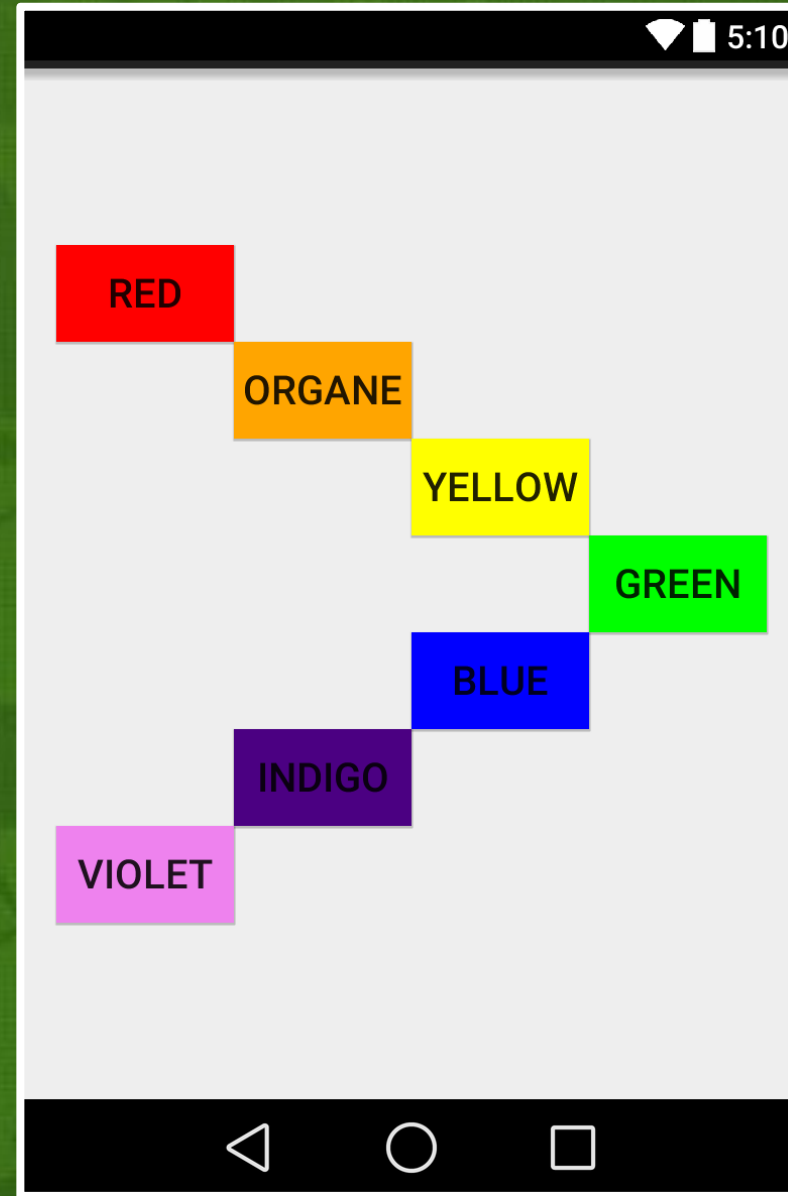
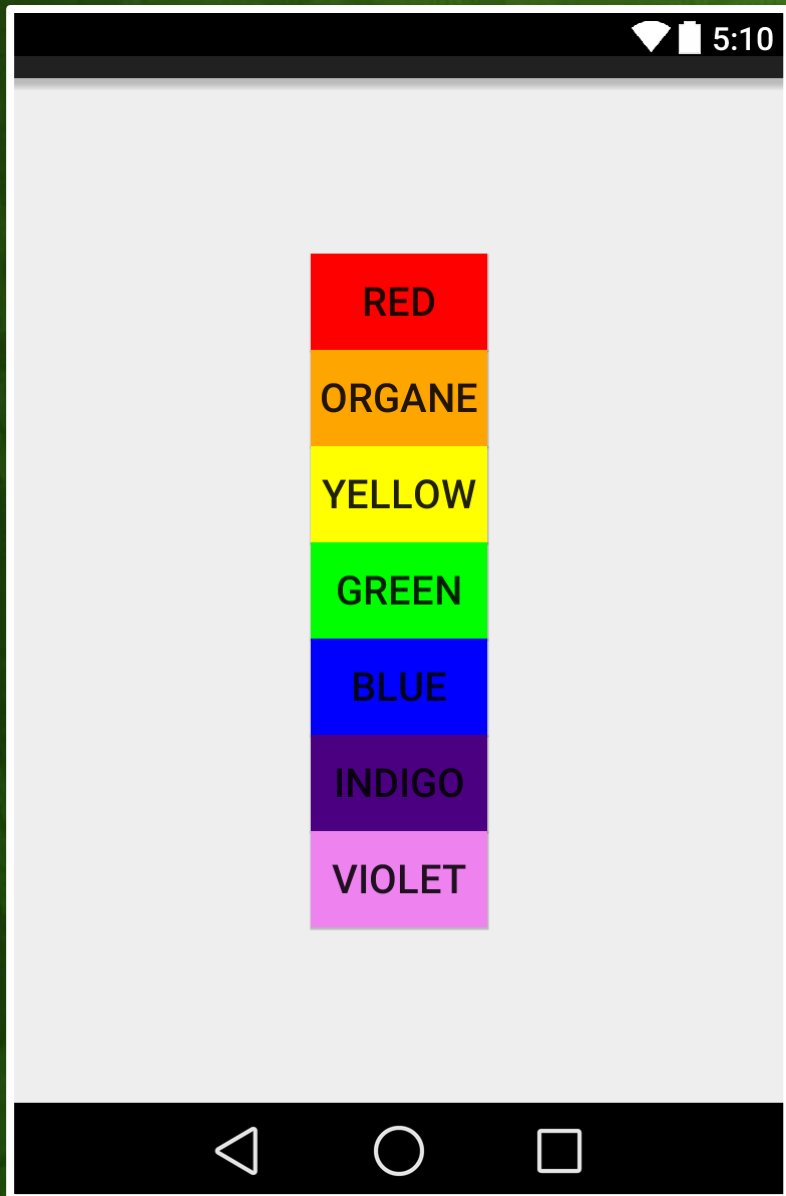


layout\_alignTop

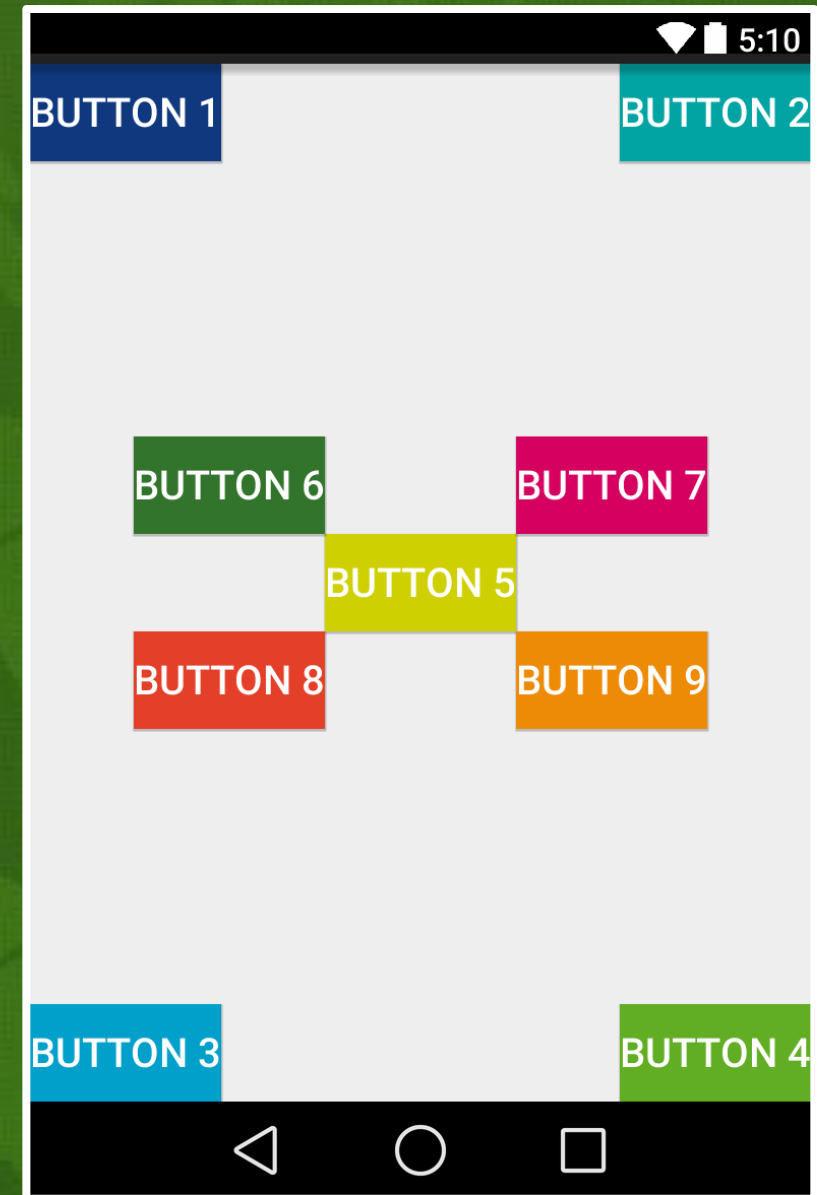
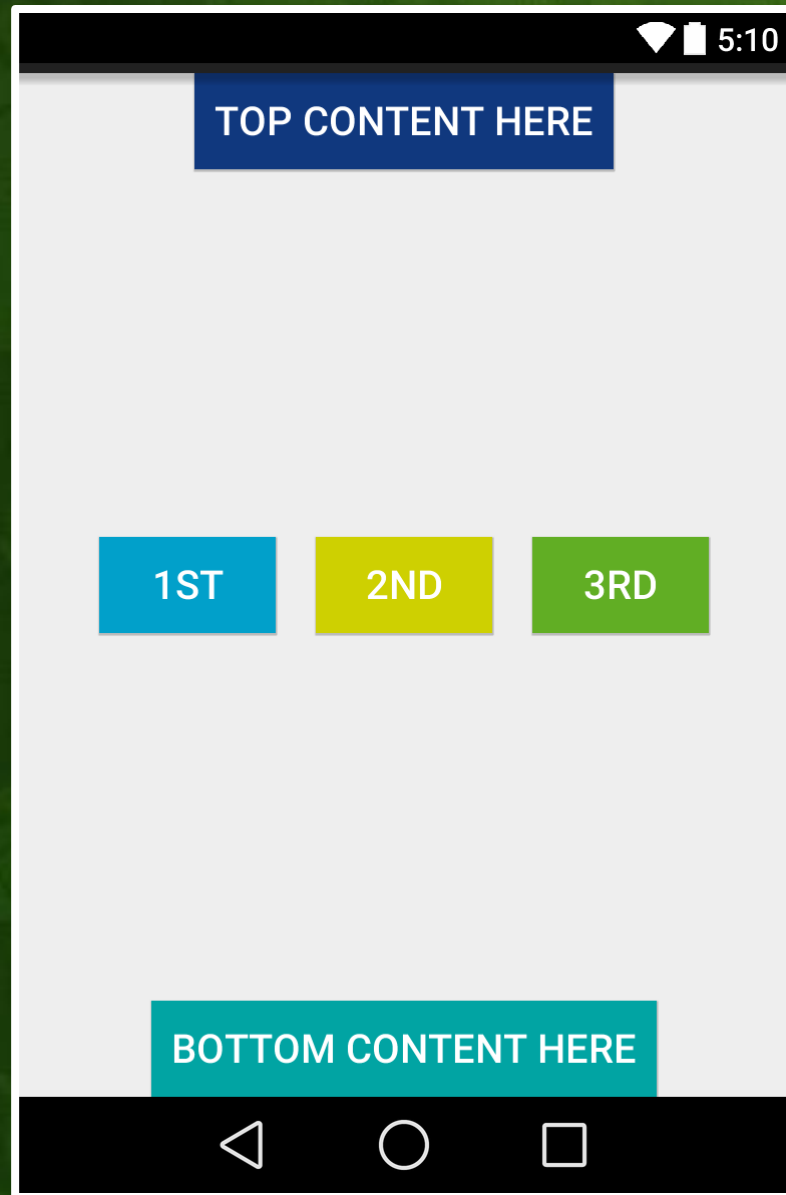


layout\_alignBottom

# RelativeLayout – Exercise (1)



# RelativeLayout – Exercise (2)





# TableLayout

- *Sắp xếp các view theo hàng (row) và cột (column)*
- *Sử dụng <TableRow> để định nghĩa một hàng nào đó*
- *Số View nhiều nhất trong TableRow dùng để định nghĩa số cột cho toàn table*
- *Xây dựng ví dụ và quan sát các thuộc tính width, height của TableRow và View*

R1 - C1

R1 - C2R1 - C3

R2 - C1

R2 - C2

# TableLayout - stretchColumns

R1 - C1	R1 - C2	R1 - C3	
R2 - C1	R2 - C2		

```
<TableLayout  
    android:stretchColumns="1">
```

R1 - C1	R1 - C2	R1 - C3
R2 - C1	R2 - C2	

```
<TableLayout  
    android:stretchColumns="0,2">
```

R1 - C1	R1 - C2	R1 - C3
R2 - C1	R2 - C2	

```
<TableLayout  
    android:stretchColumns="*">
```

R1 - C1	R1 - C2	R1 - C3
R2 - C1	R2 - C2	

# TableLayout – layout\_span & layout\_column

R1 - C1	R1 - C2	R1 - C3
R2 - C1	Row 2 - Column 2	

*android:layout\_column="2">*

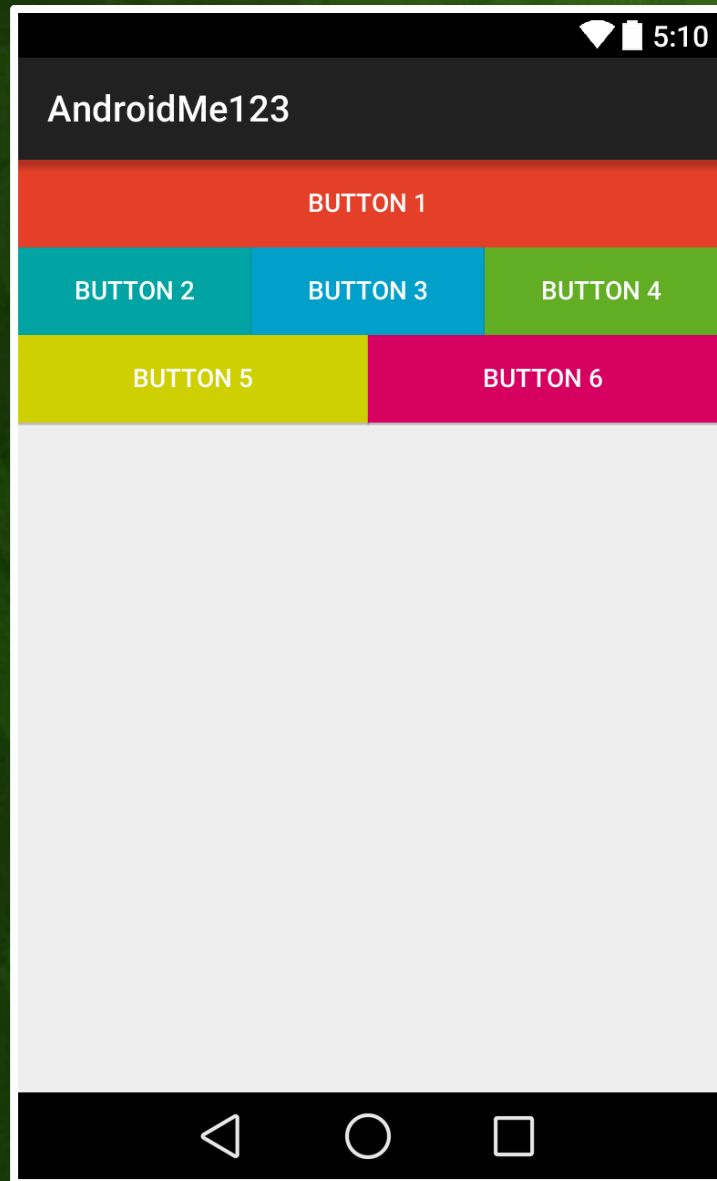
R1 - C1	R1 - C2	R1 - C3
R2 - C1		Row 2 - Column 2

*android:layout\_span="2">*

R1 - C1	R1 - C2	R1 - C3
R2 - C1	Row 2 - Column 2	



# TableLayout – Exercise





# Android Screen Orientation

