

Build Android  
App with Kotlin

Android studio  
*Dễ Hiểu - Dễ học*

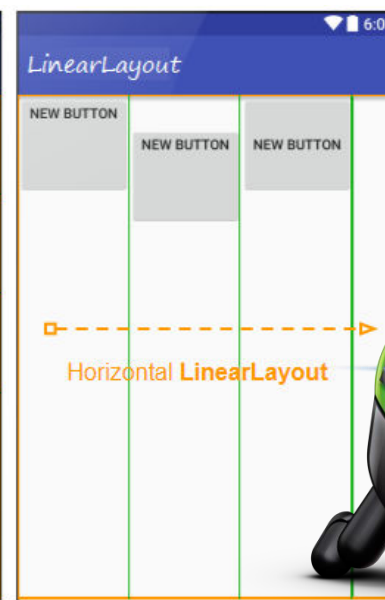
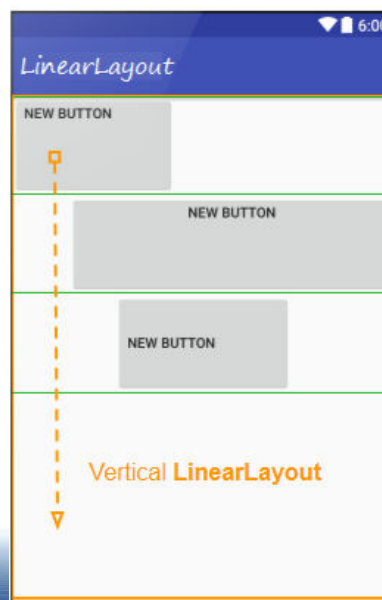


## Android app with kotlin Bài 3

1 Làm quen android studio

2 LinearLayout

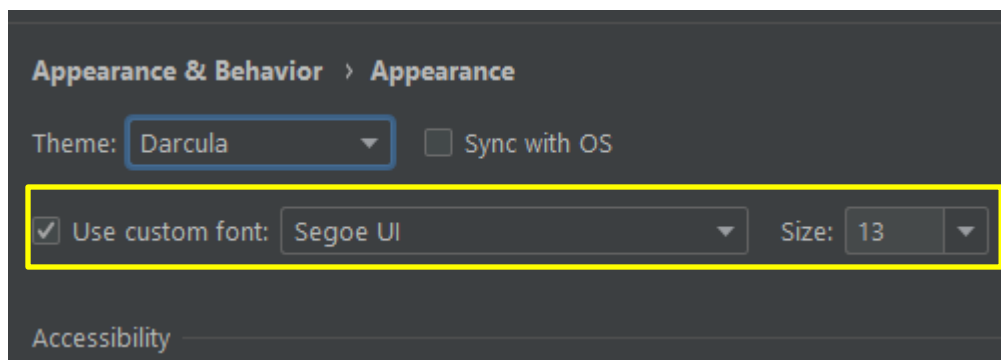
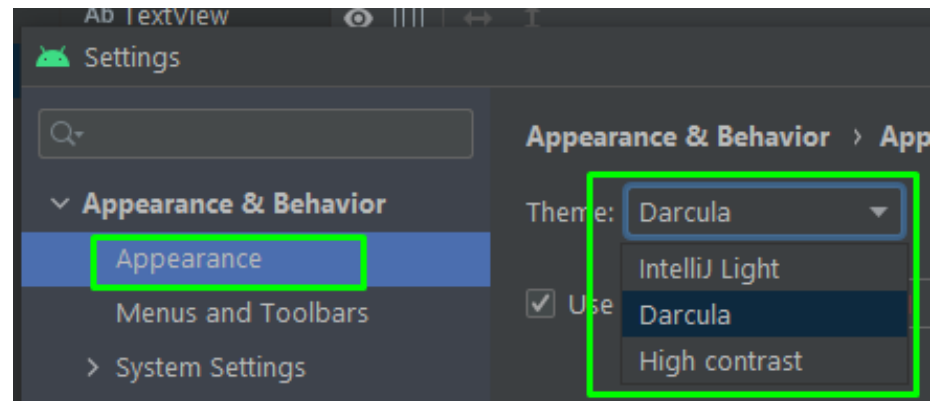
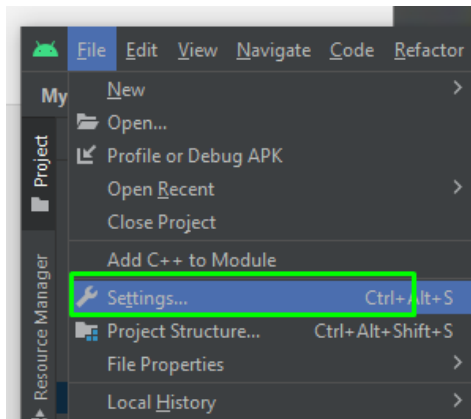
3 TableLayout



1

## Làm quen android studio

### ❑ 1. Chọn theme tối , sáng :



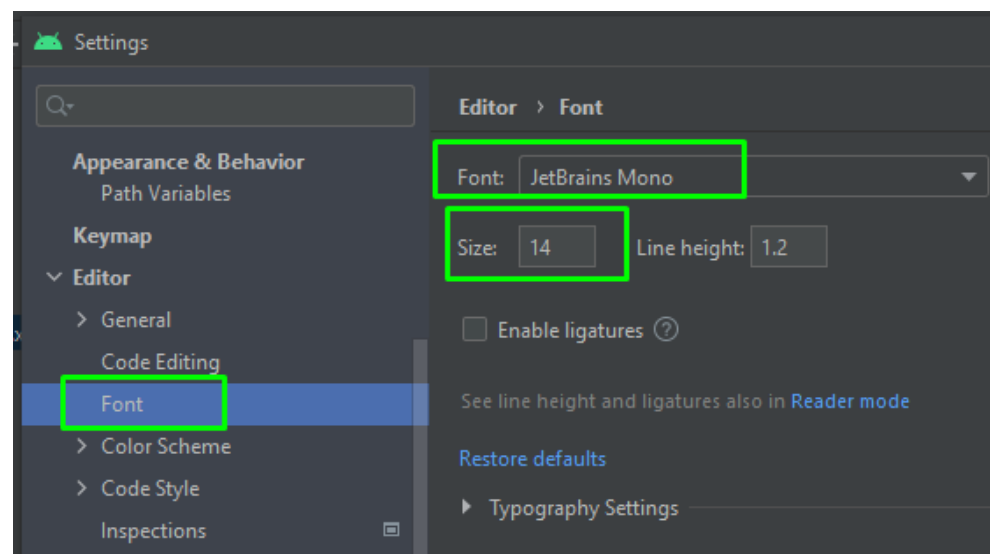
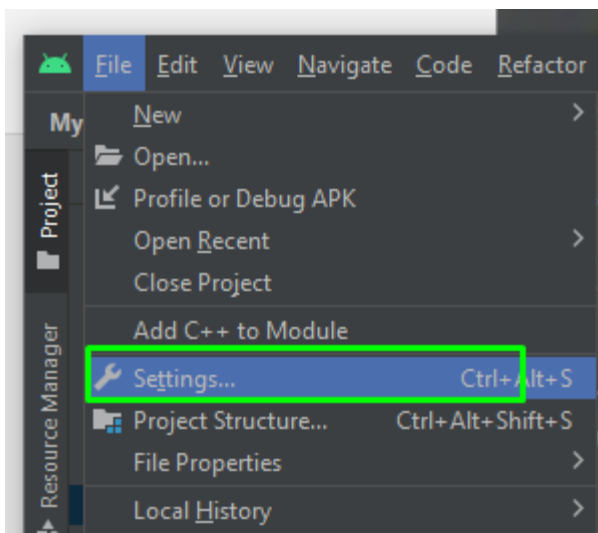
Font và cỡ chữ của android studio



1

## Làm quen android studio

### ❑ 2. Chỉnh font chữ, cỡ chữ coding :



1

## Làm quen android studio

### 3. Giao diện thiết kế chính :

**Các control (View)**

**Kéo thả**

**Kéo thả**

**Component Tree**

**Giao diện thiết kế**

**Thiết lập thông số, màu sắc...**

**Attributes**

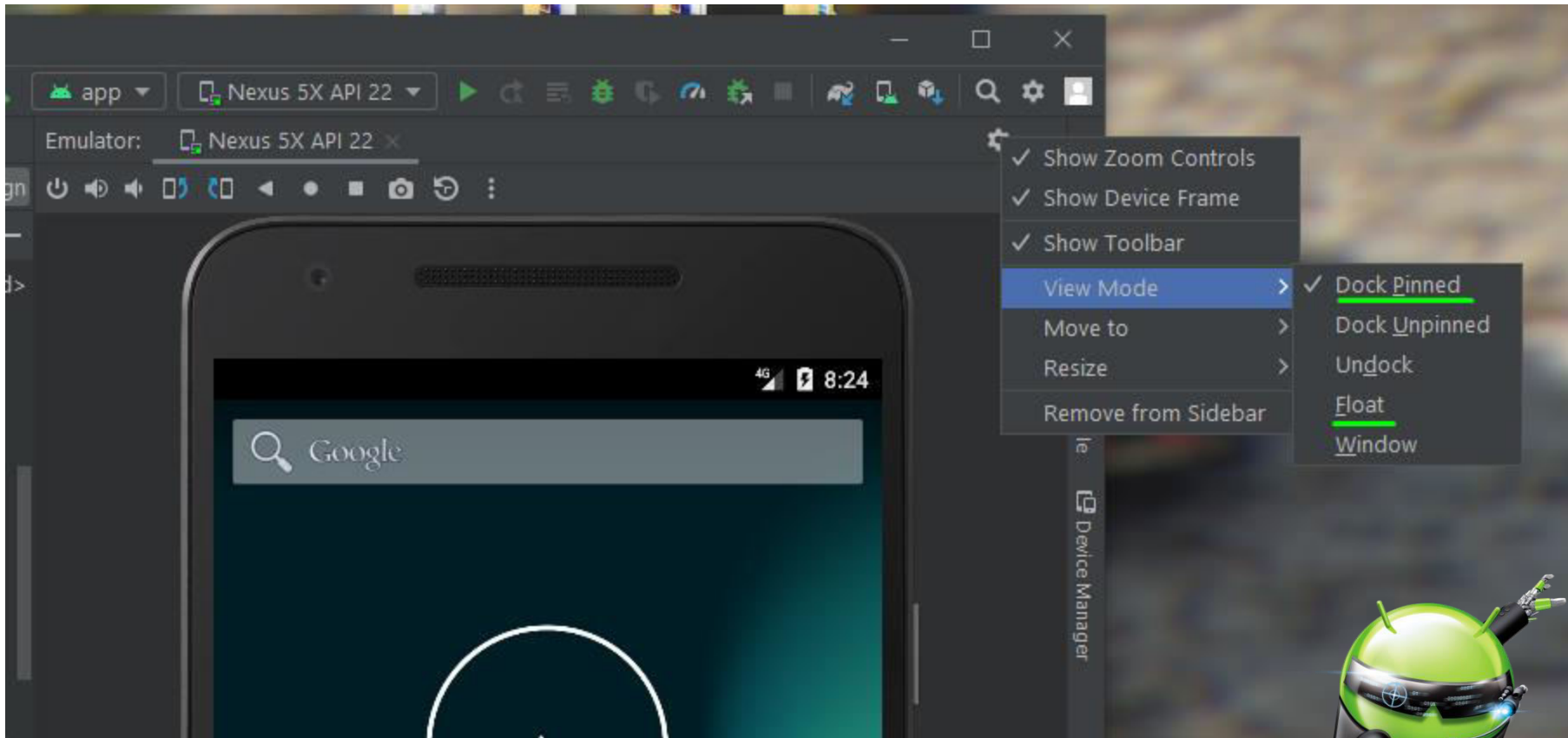
The screenshot shows the Android Studio IDE with the following components and annotations:

- Palette:** Located on the top left, it contains various UI controls categorized under 'Common', 'Text', 'Buttons', 'Widgets', 'Layouts', 'Containers', and 'Helpers'. A red box labeled 'Các control (View)' points to this area.
- Component Tree:** Located at the bottom left, it shows a hierarchical view of the UI components. A red box labeled 'Component Tree' points to this area.
- Design View:** The central area shows a visual representation of the UI. It includes a 'Hello World!' text view and a 'BUTTON'. A red box labeled 'Giao diện thiết kế' points to this area.
- Attributes:** Located on the right side, it shows the properties of the selected UI component. A red box labeled 'Attributes' points to this area.
- Annotations:**
  - A yellow arrow labeled 'Kéo thả' points from the 'BUTTON' in the Palette to the 'BUTTON' in the Design View.
  - A yellow arrow labeled 'Kéo thả' points from the 'LinearLayout' in the Component Tree to the Design View.
  - A yellow arrow labeled 'Thiết lập thông số, màu sắc...' points from the 'Attributes' panel to the Design View.

1

## Làm quen android studio

### ❑ 4. Ghim và tách Emulator ( máy ảo sẵn có trong Android studio ) :

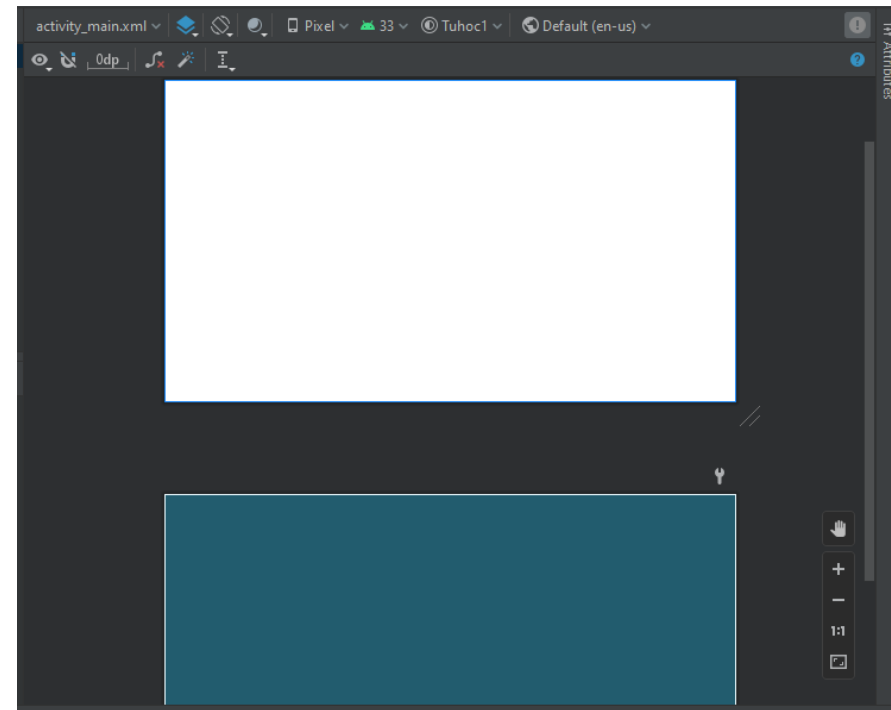
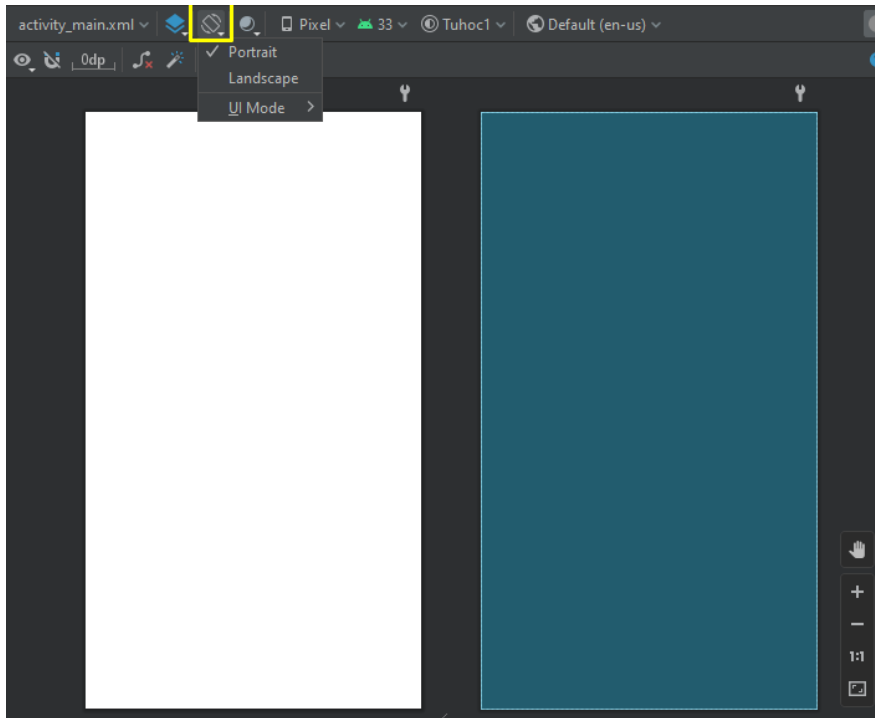




1

## Làm quen android studio

### ❑ 5. Xoay ngang , dọc màn hình giao diện:

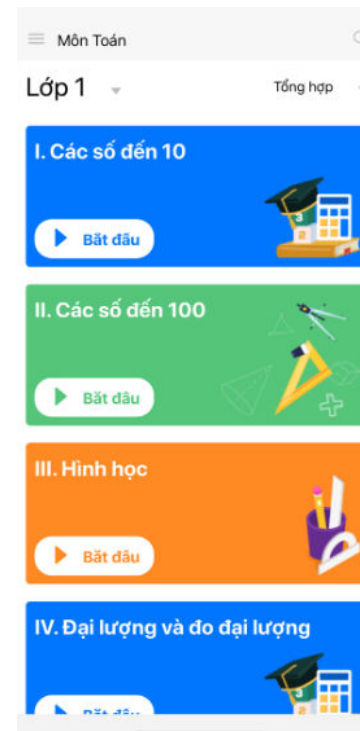
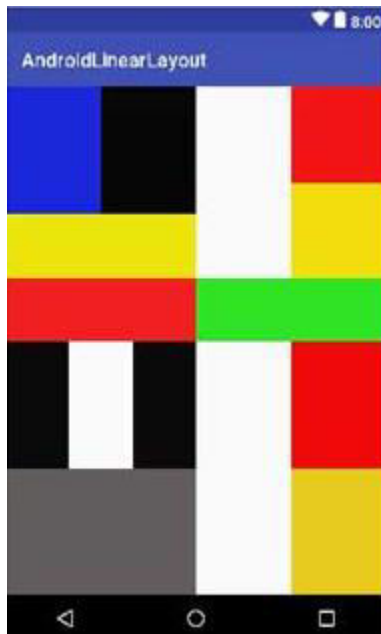


2

## LinearLayout

### ❑ 6. Khái quát LinearLayout :

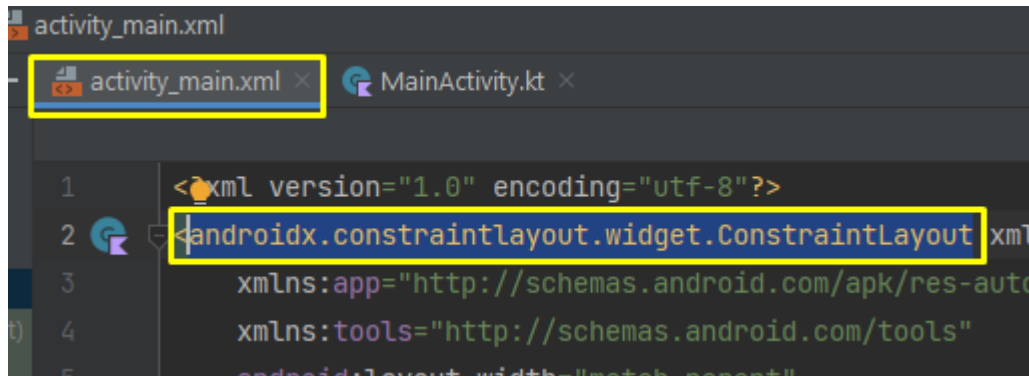
**LinearLayout:** Cho phép chúng ta sắp xếp các view trên giao diện theo chiều ngang hoặc chiều dọc.



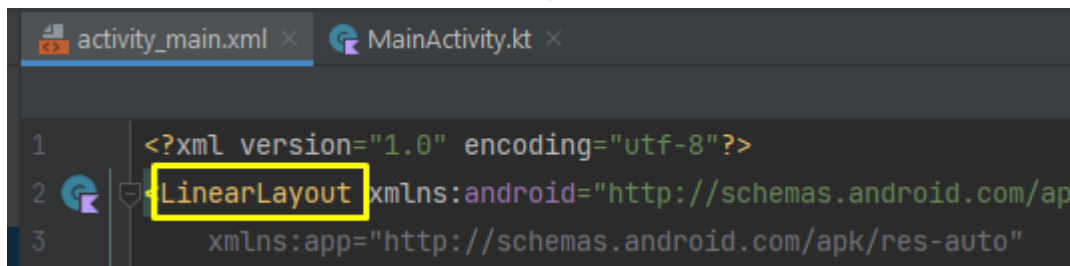
2

## LinearLayout

❑ 7. Set LinearLayout ngay sau khi tạo project :



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     >
```



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     >
```





2

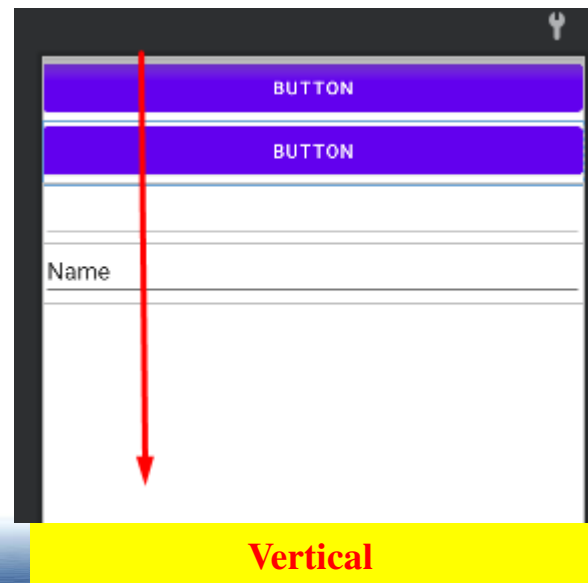
## LinearLayout

### ❑ 8. Các thuộc tính quan trọng của LinearLayout :

STT	Thuộc tính	Nội dung
1	<b>orientation</b>	Vertical: theo chiều dọc
		Horizontal: theo chiều ngang
2	<b>background</b>	Set màu nền
3	<b>id</b>	Giống tên biến, sử dụng khi code

```
android:orientation="vertical"
android:layout_height="vertical"
tools:context=".Main"
horizontal
```

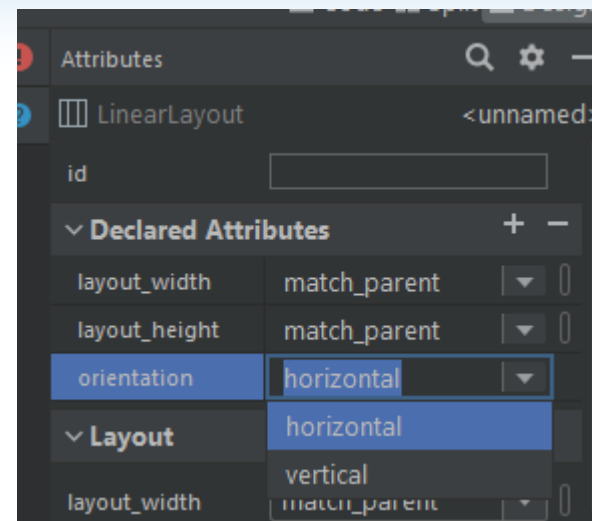
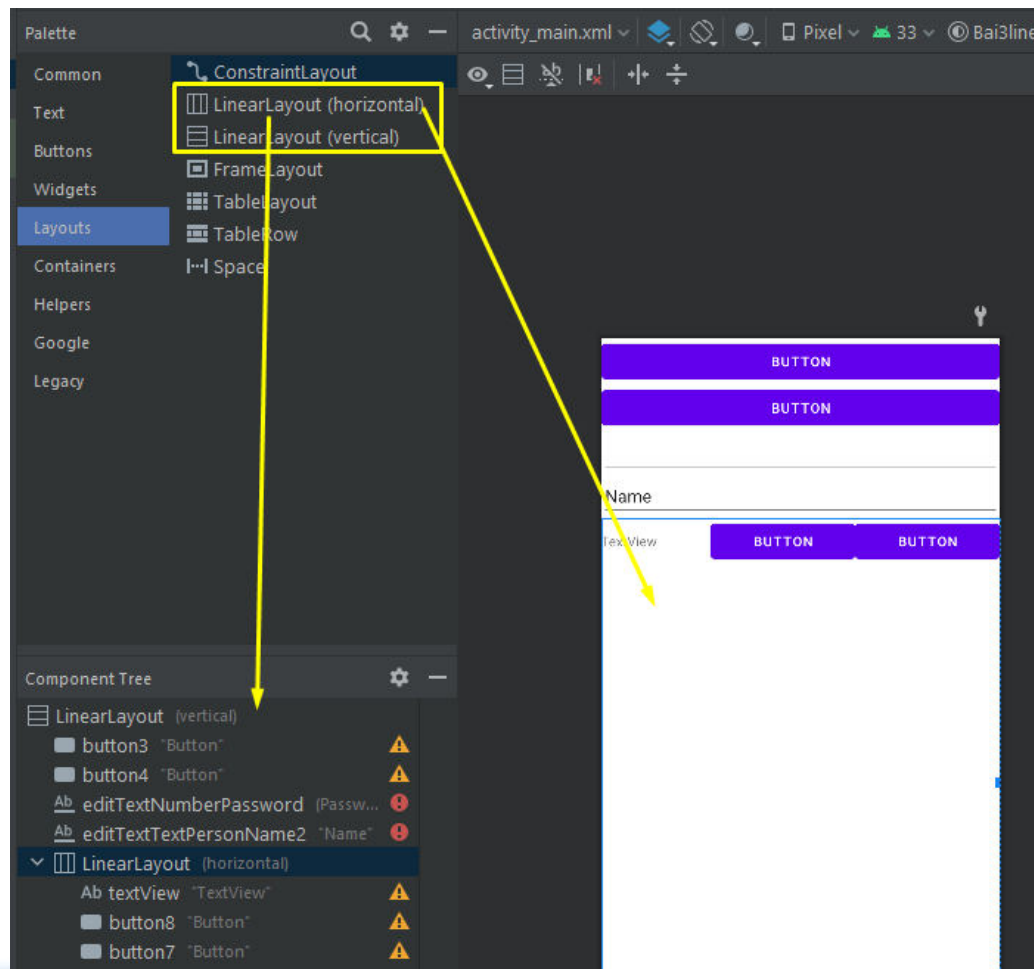
Ctrl+Down and Ctrl+Up will move caret



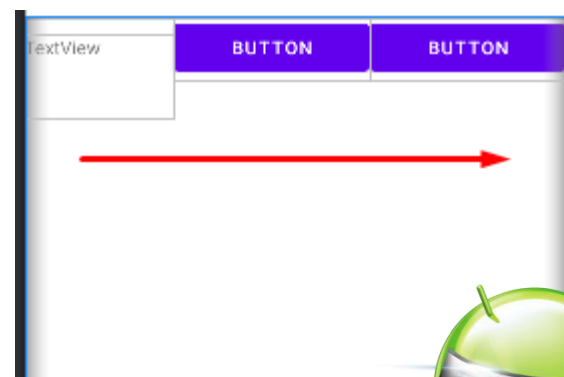
2

## LinearLayout

❑ Kéo thả linearlayout :



Chuyển orientation Attributes



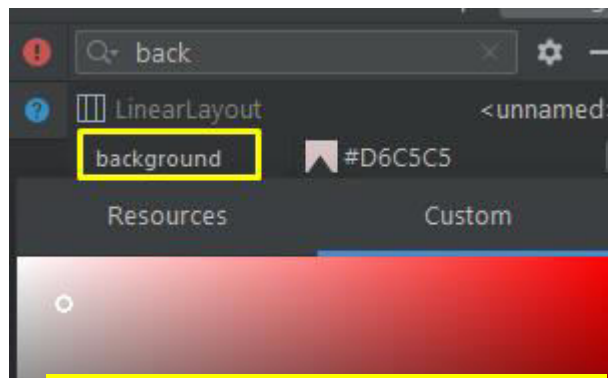
Horizontal



2

## LinearLayout

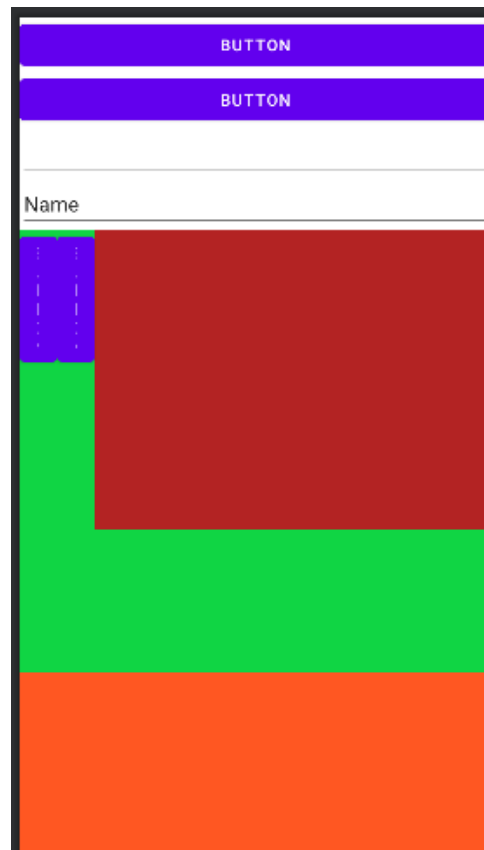
### ❑ 8.2 background: *Màu nền*



C1: Edit Attributes

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#10D544"
    android:orientation="horizontal">
```

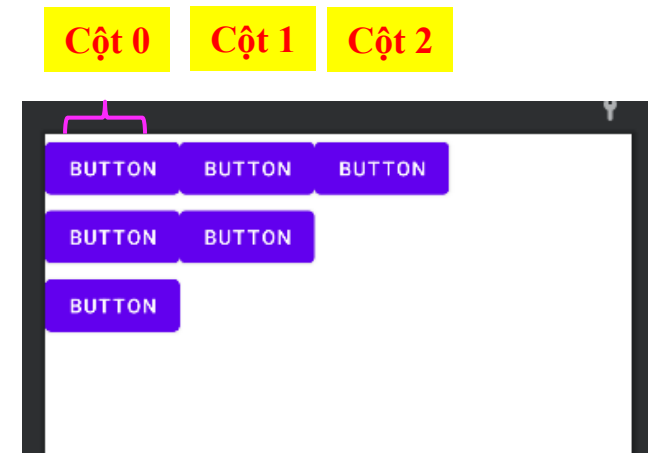
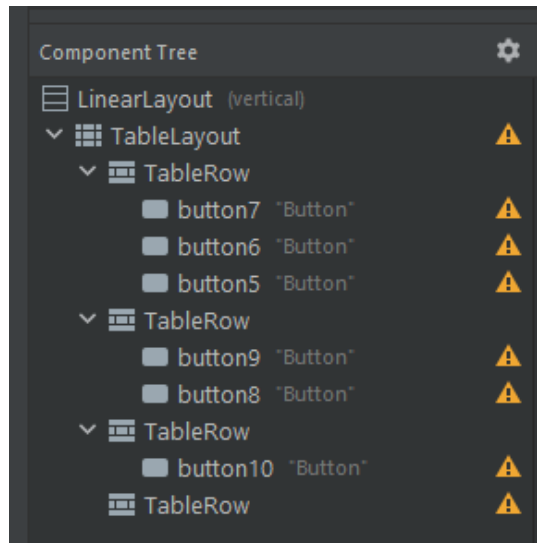
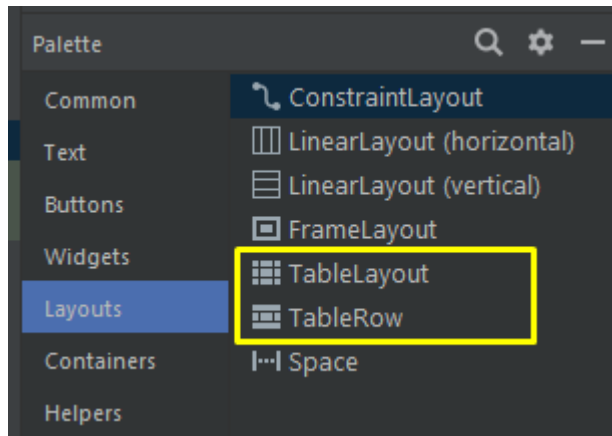
C2 : Sử dụng coding



3

## TableLayout

- ❑ **9, TableLayout:** Cho phép chúng ta sắp xếp các view trên giao diện dòng và cột  
 -- Cột sẽ tự động lấy theo dòng có số lượng nhiều view (control) nhất



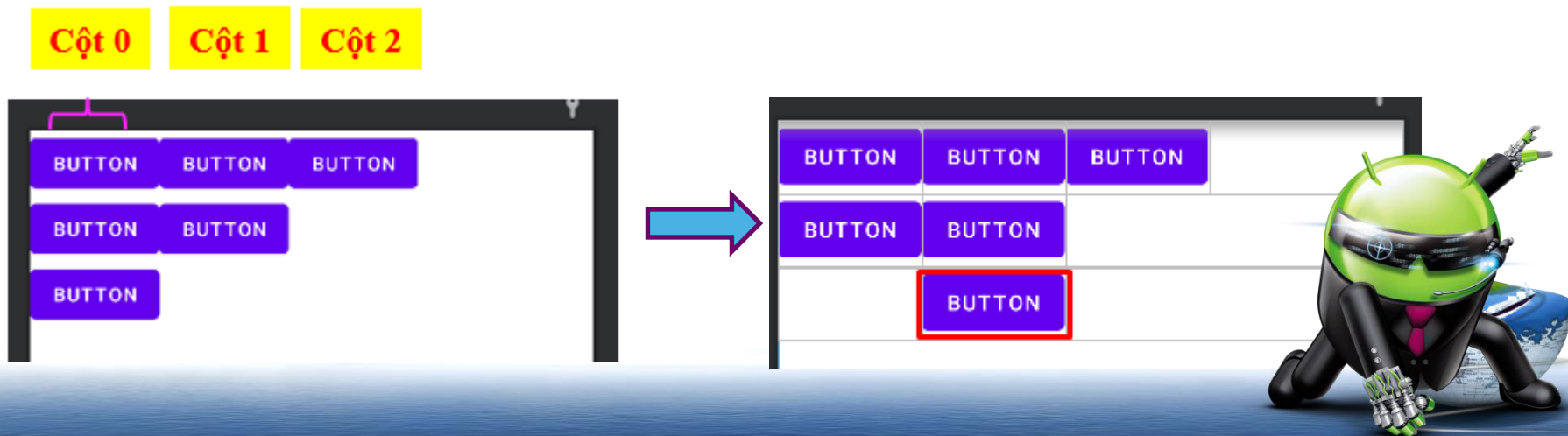
3

## TableLayout

### ❑ 10. Các thuộc tính quan trọng của TableLayout :

STT	Thuộc tính	Nội dung
1	<code>layout_column</code>	Set vị trí của view trên cột nào đó

```
<Button
    android:id="@+id/button10"
    android:layout_column="1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button" />
```



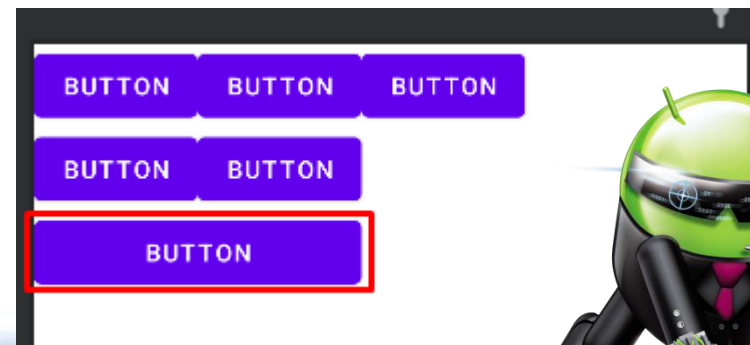
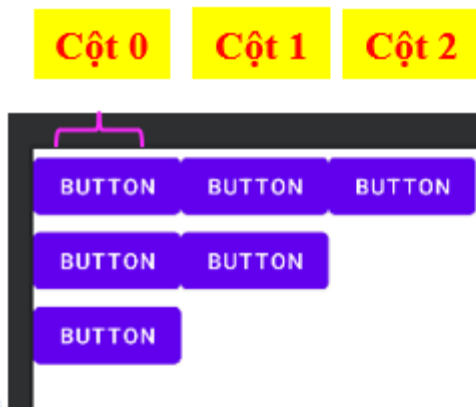
3

## TableLayout

### ❑ 10. Các thuộc tính quan trọng của TableLayout :

STT	Thuộc tính	Nội dung
2	<code>layout_span="x"</code>	Set vị trí của view <b>nằm trên bao nhiêu cột</b>

```
<Button
    android:id="@+id/button10"
    android:layout_span="2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button" />
```



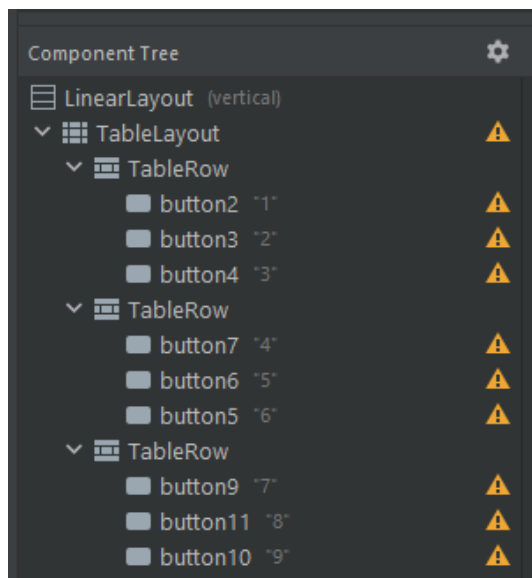


3

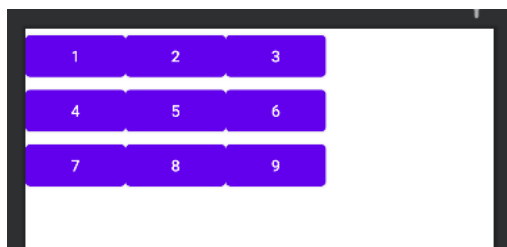
## TableLayout

### ❑ 10. Các thuộc tính quan trọng của TableLayout :

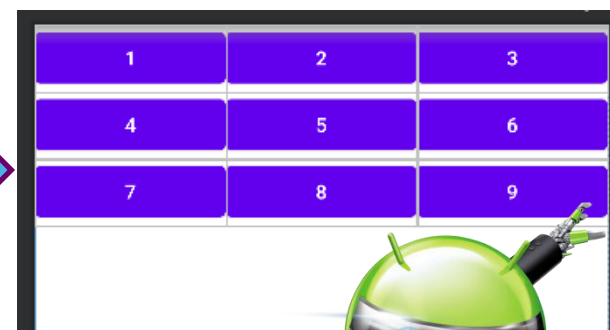
STT	Thuộc tính	Nội dung
3	<code>stretchColumns="*"</code>	Dàn đều các view vừa màn hình



```
<TableLayout
    android:layout_width="match_parent"
    android:stretchColumns="*"
    android:layout_height="match_parent">
```



Trước



Sau

