

# ANDROID PROGRAMMING

## LESSON 3

Version 1.0

# Agenda

- Understanding Android Views, View Groups and Layouts
- A Guide to the Android Studio Layout Editor Tool
- A Guide to the Android ConstraintLayout
- An Android Studio Layout Editor ConstraintLayout Tutorial

# Understanding Android Views, View Groups and Layouts

- **Designing for Different Android Devices**
- **Views and View Groups**
  - Every item in a user interface is a subclass of the *android.view.View*
  - A view can also be comprised of multiple other views Such views are subclassed from the Android *ViewGroup* class (*android.view.ViewGroup*)

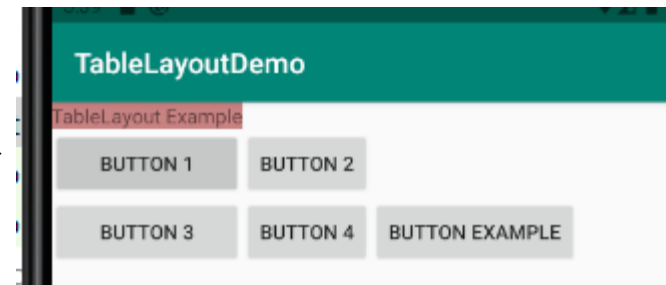
# Understanding Android Views, View Groups and Layouts

- **Android Layout Managers**

- ConstraintLayout
- LinearLayout
- TableLayout
- FrameLayout
- RelativeLayout
- AbsoluteLayout
- GridLayout
- CoordinatorLayout

# TableLayout and TableRow

```
<TableLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <TableRow>
        <TextView
            android:text="TableLayout Example"
            android:background="#c98282"
            android:gravity="center"/>
    </TableRow>
    <TableRow>
        <Button android:text="Button 1" />
        <Button android:text="Button 2"/>
    </TableRow>
    <TableRow>
        <Button android:text="Button 3" />
        <Button android:text="Button 4" />
        <Button android:text="Button example" />
    </TableRow>
</TableLayout>
```



- android:stretchColumns="col1, col2 ..." (\* for all)
- android:shrinkColumns="0,1,.. " (\* for all)
- `<Button android:text="Single row" />`
- `<Button android:text="Button 0" android:layout_column="1" />`
- android:layout\_span="3"

# Exercise (02)

TableLayout Example		
	B1	B2
C1		C2
		C3

Login		
Username	enter id	
Password	enter password	
OK	CANCEL	FORGOT PASSWORD

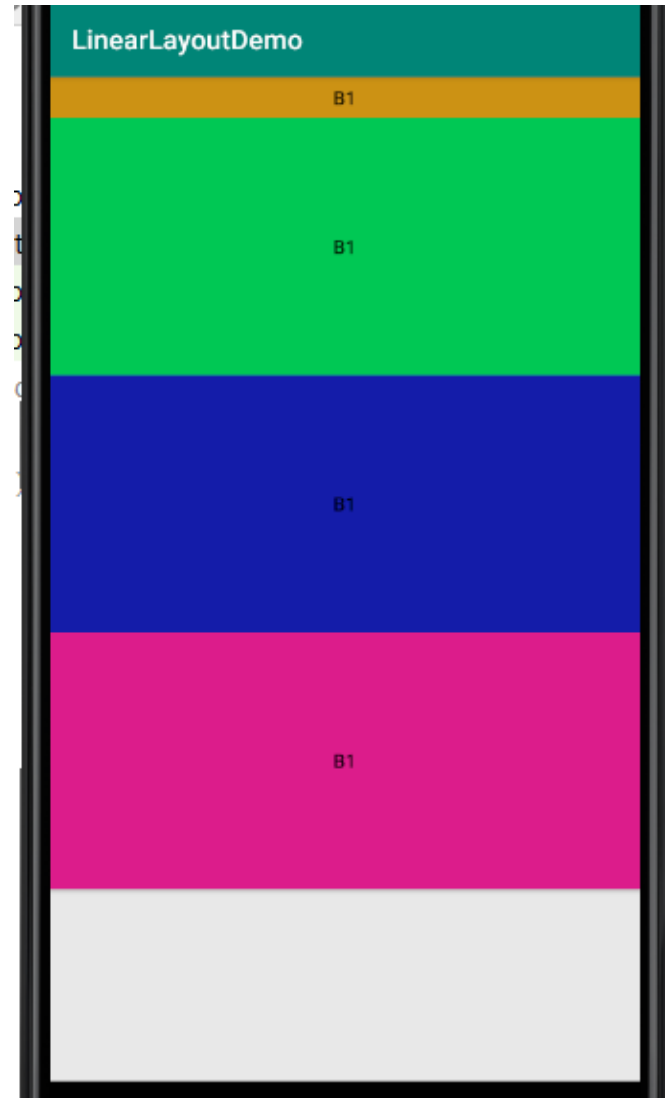
# LinearLayout (03)

```
<LinearLayout
    android:orientation="horizontal"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#a6dfdfff"
    android:layout_margin="10dp"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <Button
        android:background="#dc9e1a"
        android:text="B1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:background="#00c853"
        android:text="B1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```



# android:weightSum

- `android:weightSum="4"`
- `layout_weight="1"`



# RelativeLayout (04)

Thuộc tính	Ý nghĩa
<code>android:layout_alignParentBottom</code>	<code>true</code> căn thẳng cạnh dưới view con với cạnh dưới View cha
<code>android:layout_alignParentLeft</code>	<code>true</code> căn thẳng cạnh trái view con với cạnh trái View cha
<code>android:layout_alignParentRight</code>	<code>true</code> căn thẳng cạnh phải view con với cạnh phải View cha
<code>android:layout_alignParentTop</code>	<code>true</code> căn thẳng cạnh trên view con với cạnh trên View cha
<code>android:layout_centerInParent</code>	<code>true</code> căn view con vào giữa View cha
<code>android:layout_centerHorizontal</code>	<code>true</code> căn view con vào giữa View cha theo chiều ngang
<code>android:layout_centerVertical</code>	<code>true</code> căn view con vào giữa View cha theo chiều đứng

# Định vị View con bằng liên hệ giữa chúng với nhau

Thuộc tính	Ý nghĩa
<code>android:layout_below</code>	Nằm phía dưới View có ID được chỉ ra
<code>android:layout_above</code>	Nằm phía trên View có ID được chỉ ra
<code>android:layout_toLeftOf</code>	Nằm phía trái View có ID được chỉ ra
<code>android:layout_toRightOf</code>	Nằm phía phải View có ID được chỉ ra
<code>android:layout_alignBottom</code>	Căn thẳng cạnh dưới với cạnh dưới của View có ID được chỉ ra
<code>android:layout_alignLeft</code>	Căn thẳng cạnh trái với cạnh trái của View có ID được chỉ ra
<code>android:layout_alignRight</code>	Căn thẳng cạnh phải với cạnh phải của View có ID được chỉ ra
<code>android:layout_alignTop</code>	Căn thẳng cạnh trên với cạnh trên của View có ID được chỉ ra

<TextView

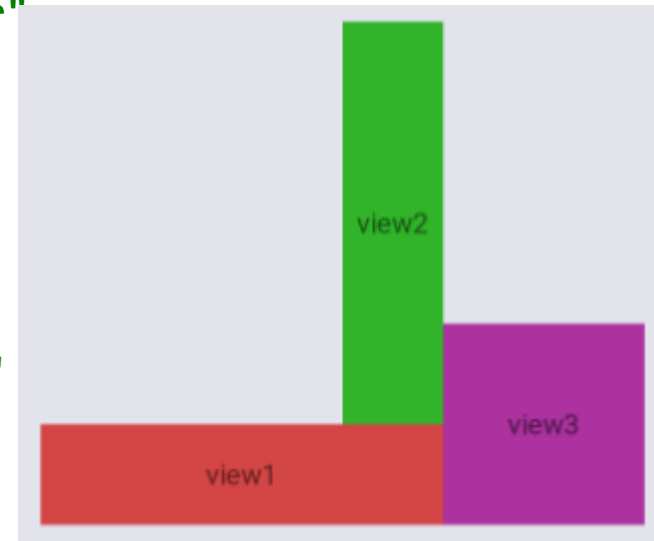
```
    android:id="@+id/view1"  
    android:text="view1" android:gravity="center"  
    android:layout_toLeftOf="@id/view3"  
    android:layout_alignBottom="@id/view3"  
    android:layout_width="200dp"  
    android:layout_height="50dp"  
    android:background="#e8d33636" />
```

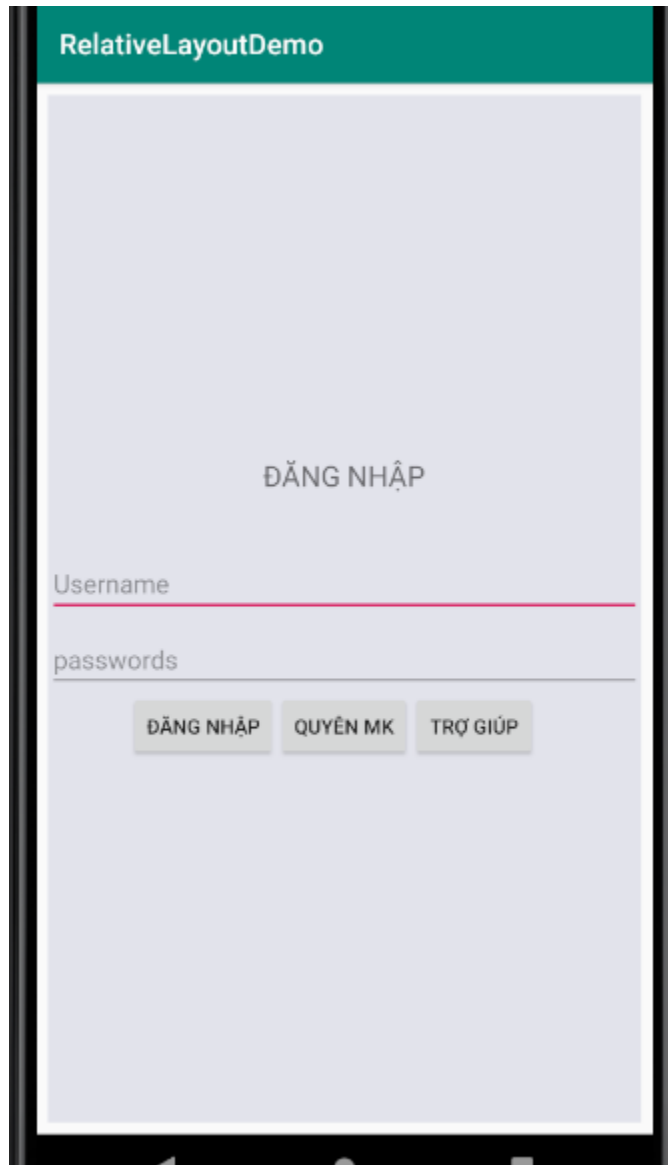
<TextView

```
    android:id="@+id/view2" android:text="view2"  
    android:layout_alignRight="@id/view1"  
    android:layout_above="@id/view1"  
    android:gravity="center"  
    android:layout_width="50dp" android:layout_height="200dp"  
    android:background="#e71faf15" />
```

<TextView

```
    android:id="@+id/view3"  
    android:layout_centerInParent="true"  
    android:layout_alignParentRight="true"  
    android:layout_marginRight="30dp"  
    android:text="view3" android:gravity="center"  
    android:layout_width="100dp"  
    android:layout_height="100dp"  
    android:background="#d4a00f8f" />
```





# Exercise at class

# A Guide to the Android ConstraintLayout

- **ConstraintLayout**

- Introduced in Android 7
- use of this layout manager is recommended for most layout requirements.

# A Guide to the Android ConstraintLayout (05)

- **How ConstraintLayout Works**
  - Constraints
  - Margins
  - Opposing Constraints
  - Constraint Bias
  - Chains
  - Chain Styles

# A Guide to the Android ConstraintLayout

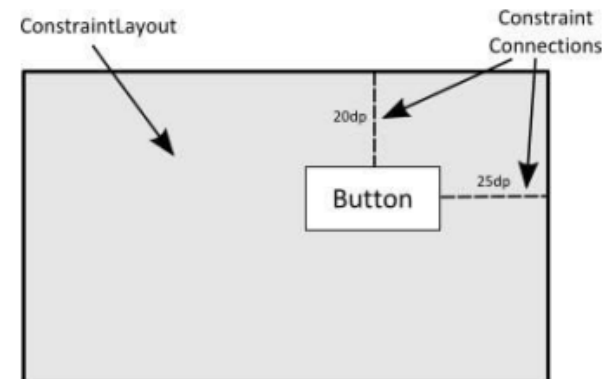
- **Constraints**

Constraints are essentially sets of rules that dictate the way in which a widget is aligned and distanced in relation to other widgets

- **Margins**

A margin is a form of constraint that specifies a fixed distance

```
<Button  
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="25dp"  
    android:layout_marginEnd="25dp"  
>
```

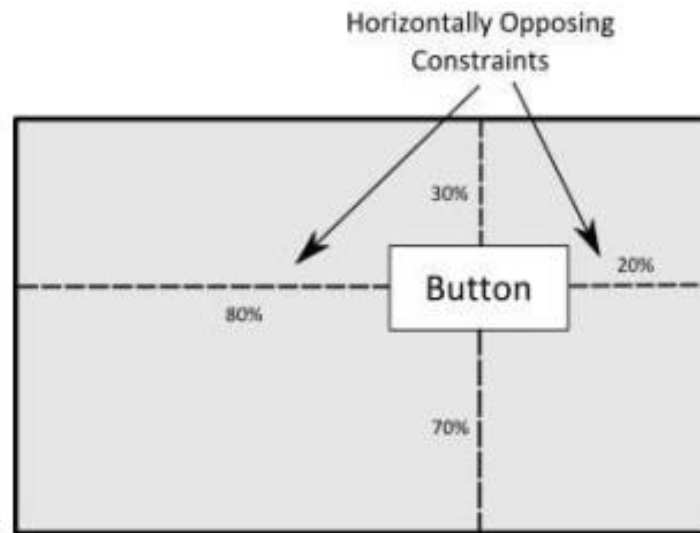




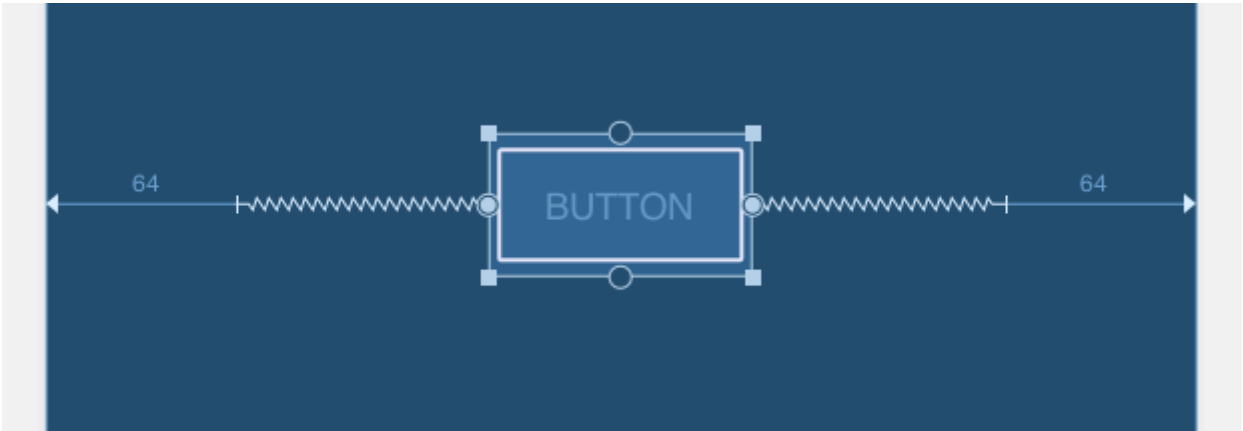
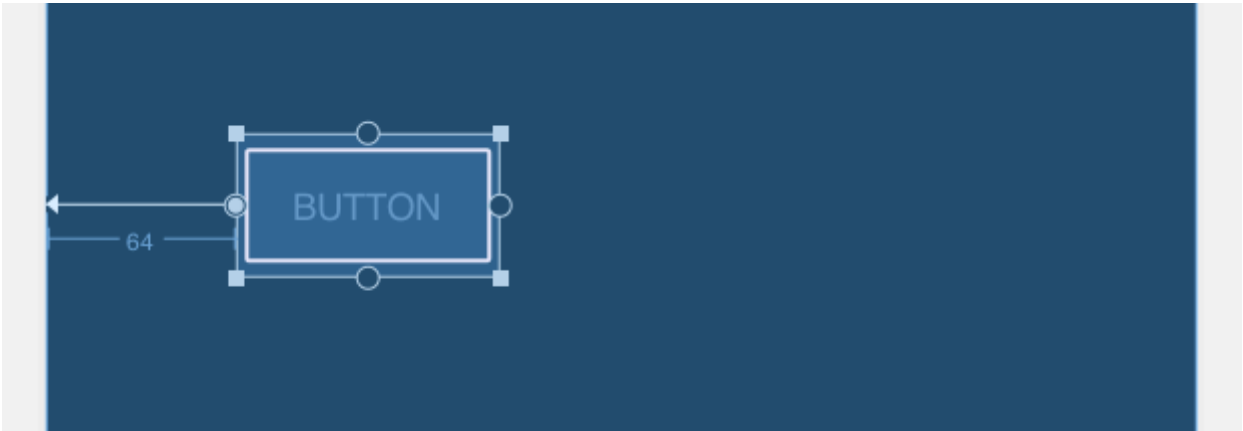
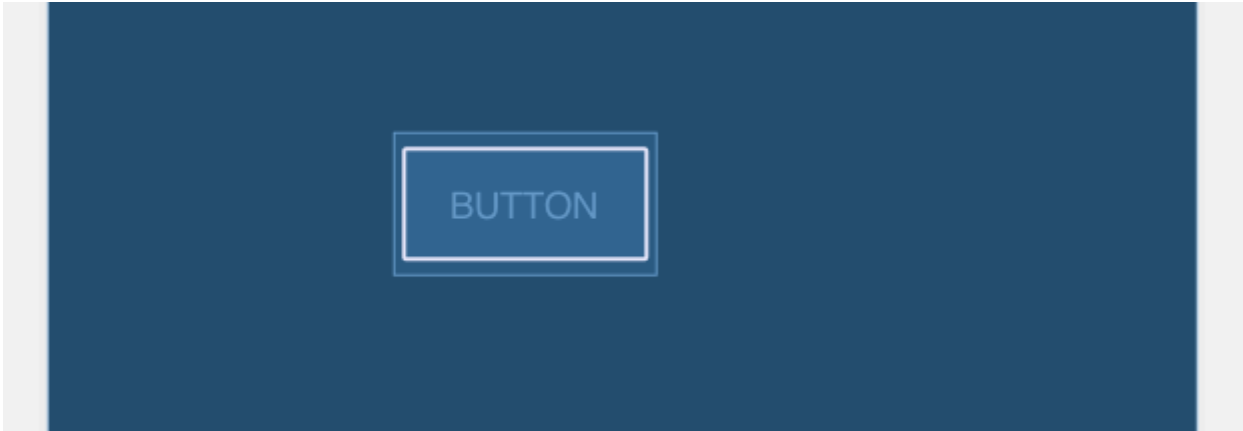
# A Guide to the Android ConstraintLayout

- **Opposing Constraints**

Two constraints operating along the same axis on a single widget are referred to as *opposing constraints*

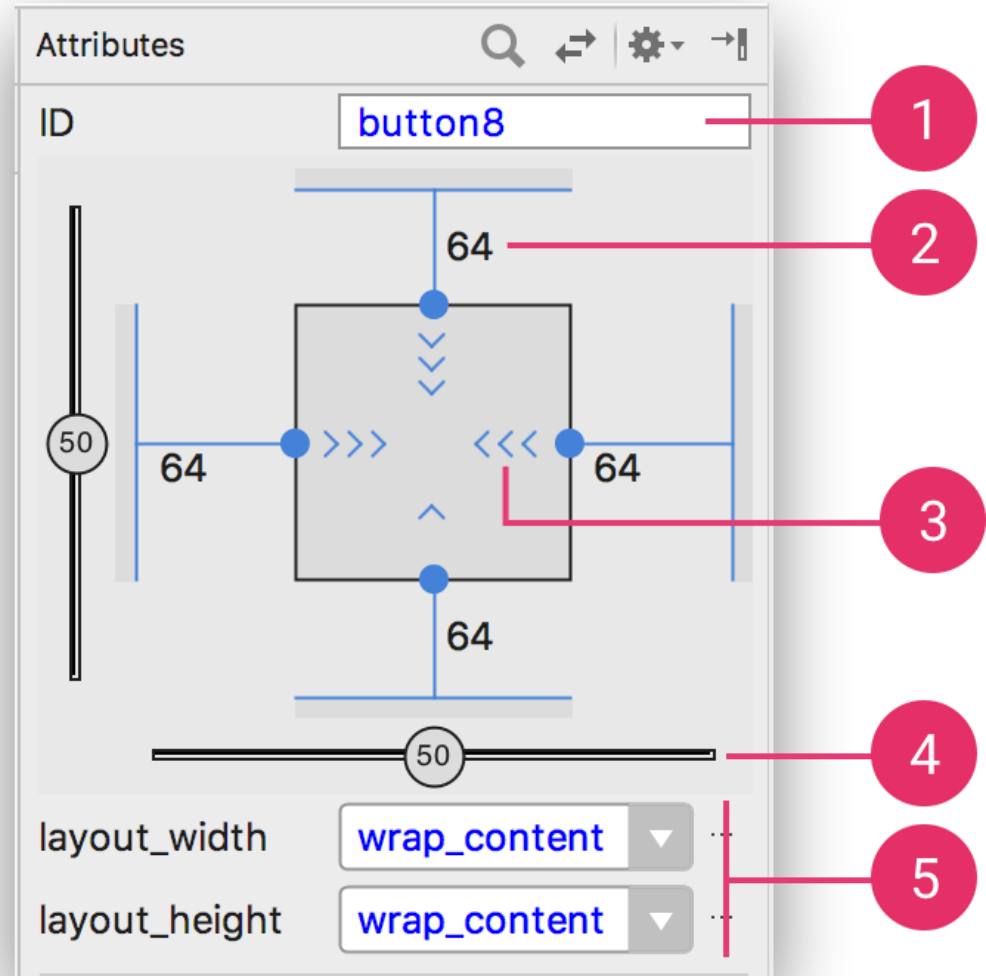
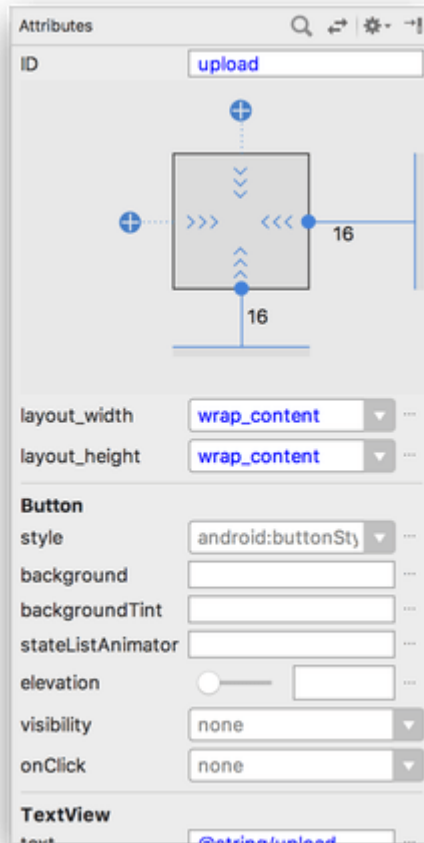


Ràng buộc	Ý nghĩa ràng buộc
<code>layout_constraintLeft_toLeftOf</code>	Ràng buộc cạnh trái của phần tử tới phần tử chỉ ra trong giá trị (gán ID)
<code>layout_constraintLeft_toRightOf</code>	Bên trái với bên phải của phần tử chỉ ra
<code>layout_constraintRight_toLeftOf</code>	Bên phải với bên trái
<code>layout_constraintRight_toRightOf</code>	Phải với phải
<code>layout_constraintTop_toTopOf</code>	Cạnh trên với cạnh trên
<code>layout_constraintTop_toBottomOf</code>	Cạnh trên nối với cạnh dưới
<code>layout_constraintBottom_toTopOf</code>	Dưới với trên
<code>layout_constraintBottom_toBottomOf</code>	Dưới với dưới
<code>layout_constraintBaseline_toBaselineOf</code>	Trùng Baseline
<code>layout_constraintStart_toEndOf</code>	Bắt đầu - Kết thúc
<code>layout_constraintStart_toStartOf</code>	Bắt đầu - Bắt đầu
<code>layout_constraintEnd_toStartOf</code>	Cuối với bắt đầu
<code>layout_constraintEnd_toEndOf</code>	Cuối với cuối

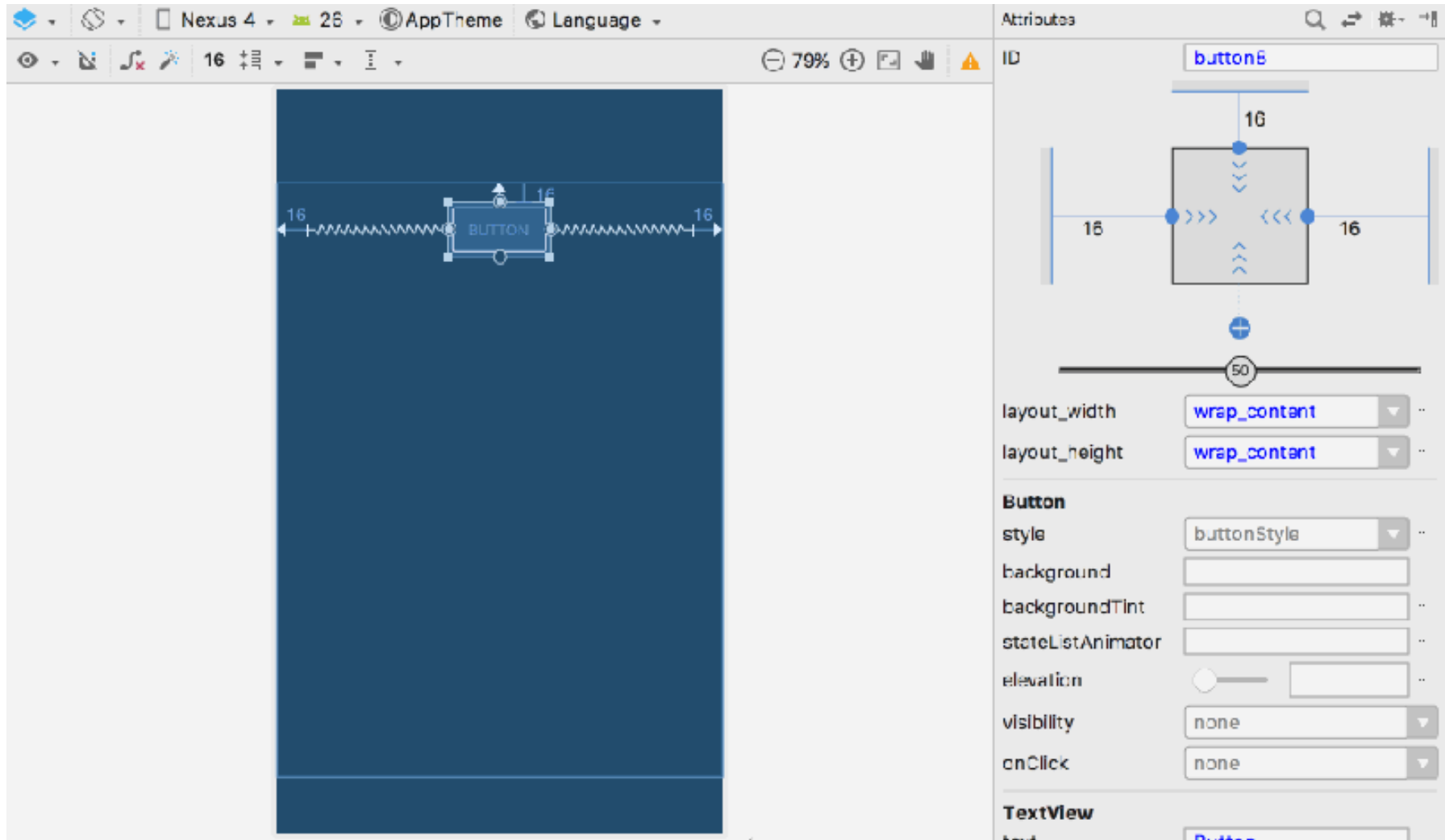




- Attribute



# Bias (*phần trăm của trọng số*)



<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="TextView 1"

app:layout\_constraintLeft\_toRightOf="@+id/textView2"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="TextView 2"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toLeftOf="@+id/textView1"

app:layout\_constraintTop\_toTopOf="parent" />

# A Guide to the Android ConstraintLayout

- **Constraint Bias**

It has now been established that a widget in a ConstraintLayout can potentially be subject to opposing constraint connections

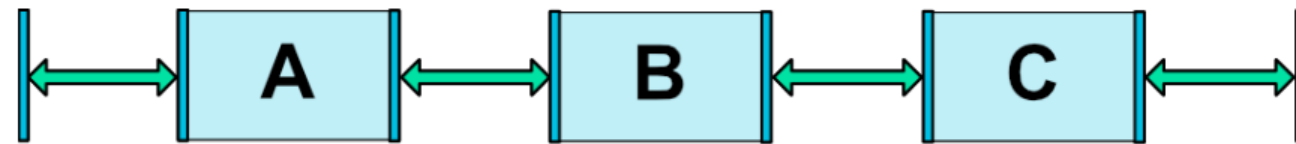
```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onClick"
    android:text="Test Button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.501"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0" />
```

# Chains trong Constraint layout

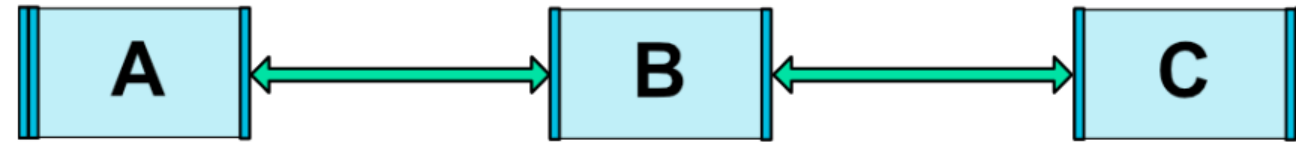
- Các Chain (chuỗi) cung cấp hành vi giống như nhóm theo một trục đơn (theo chiều ngang hoặc chiều dọc)
- Để tạo một chain, ta kết nối các View với nhau theo cả 2 hướng (bi-directional connection)
- Hai hướng ở đây có nghĩa là đuôi của view 1 neo vào đầu của view 2 và đầu của view 2 cũng cần được neo vào đuôi của view 1. ( nếu không sẽ không tính là chain)



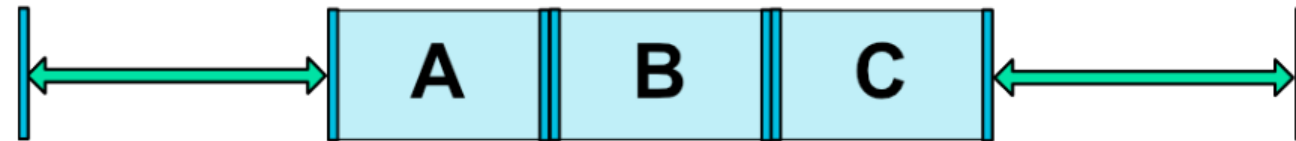
# chainStyle của Chain



Spread Chain



Spread Inside Chain



Packed Chain

<Button

```
    android:id="@+id/label_a"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:gravity="center"  
    android:text="A"  
    android:textSize="22sp"  
    app:layout_constraintEnd_toStartOf="@+id/label_b"  
    app:layout_constraintHorizontal_chainStyle="packed"  
    app:layout_constraintStart_toStartOf="parent"/>
```


<Button

```
    android:id="@+id/label_b"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:gravity="center"  
    android:text="B"  
    android:textSize="22sp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toEndOf="@id/label_a"/>
```

# Some Widgets (06)

5:39

06SomeWidgets



☐ iPhone

☒ Android

☐ Window Mobile



DISPLAY

☒ Male

☐ Female

SHOW

Rate Me

SUBMIT

Result :

Malaysia ▼

PTIT ▼

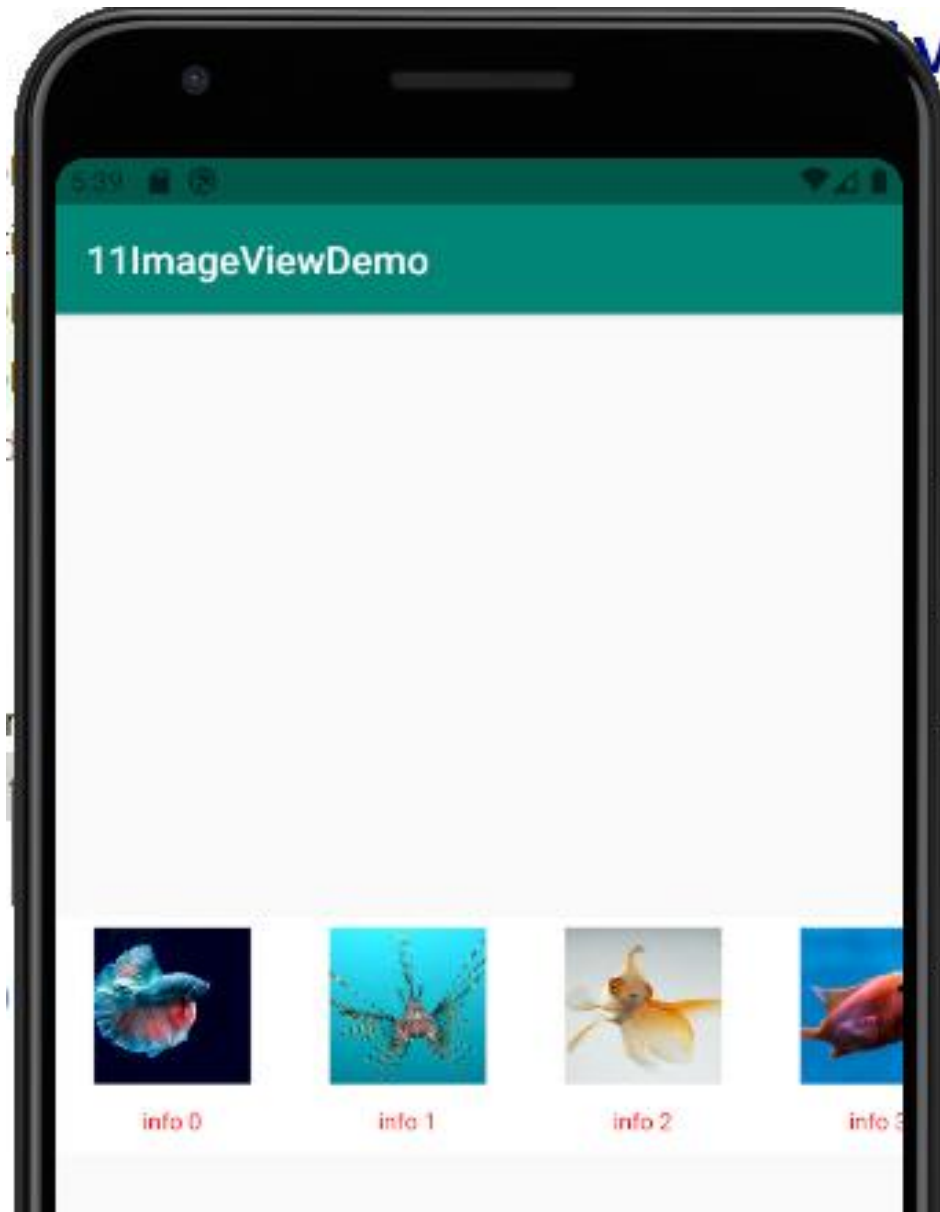
SUBMIT

# ScrollView, ImageView

- ScrollView (đứng), HorizontalScrollView (ngang) ta **buộc phải** cho tất cả những cái cần cuộn vào trong một **LinearLayout** có hướng orientation là đứng hoặc ngang
- Muốn vừa cuộn ngang lại cuộn đứng được thì ta phải lồng hai loại với nhau.

```
<HorizontalScrollView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fillViewport="true" >
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:orientation="horizontal" >
<ScrollView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fillViewport="true" >
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical" >
<!-- Noi dung -->
</LinearLayout>
</ScrollView>
</LinearLayout>
</HorizontalScrollView>
```

At class (11)




**Exercise submitted  
on classroom google**

# Exercise 1 (06)

6:02

06SomeWidgets



Select the platform you like:

☐ iPhone

☒ Android

☐ Window Mobile

Gender:

☒ Male

☐ Female

Rate your girl/guy in love with you:

★★★★★★

Your country:

Malaysia

Your university:

PTIT

Your favourite:

☒ Tennis

☐ Running

☒ Swimming

☐ Sleeping

☐ reading

SUBMIT

Platform:Android  
Gender: Male  
Rate:3  
Country: Malaysia  
My unviversity:PTIT  
My favourite: tennis, swimming

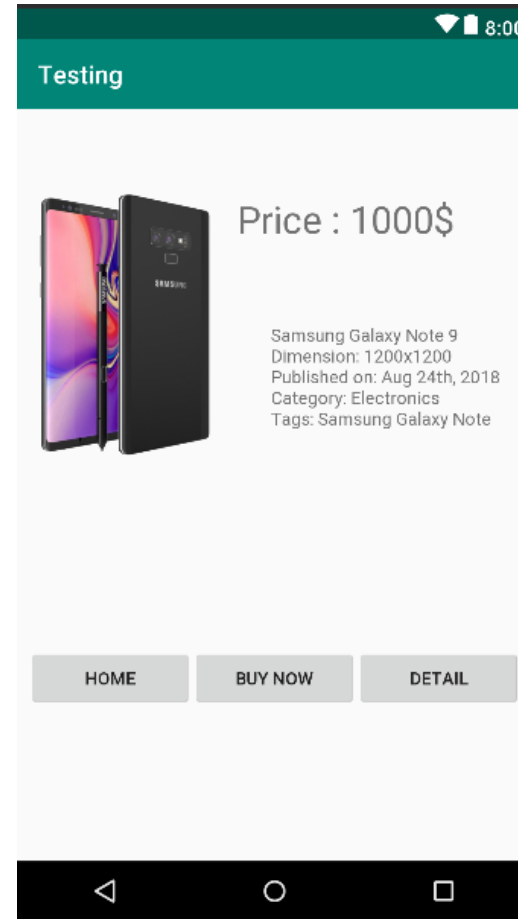


# An Android Studio Layout Editor

## **ConstraintLayout** Tutorial

### Exercise 2

Let's design !




# Giao diện đăng ký, đăng nhập tài khoản

## Exercise 3

8:00

**Dsport Team**






test@gmail.com

.....

[Quên mật khẩu](#)

**Đăng nhập**

Lựa chọn khác

[Tạo tài khoản](#)

8:01

Email

Tên người dùng

Mật khẩu

nhập lại mật khẩu

**Tạo tài khoản**

[Quay về trang đăng nhập](#)

- End of Lesson 3



Thank you!