

Build Android  
App with Kotlin  
Bài 1

Android studio  
DỄ HIẾU - DỄ HỌC

## Android app with kotlin Bài 1

1

Install android studio

2

Kết nối ĐT android với Android studio

3

Cài đặt máy ảo hệ điều hành Android



1

## Install android studio

 **Lựa chọn 1 :**

- ✓ *Download trọn bộ cài đặt tại : <http://android.tuhoc.cc/>*

Name	Size
..	
1.android-studio-2021.2.1.15-windows.exe	974.402.544
2.genymotion-3.2.1-vbox.exe	146.897.016
3.Vysor-win-4.1.77.exe	115.651.304

≡ 0.bo cai dat android-tuhoc.cc.zip 

1, Công cụ soạn thảo, thiết kế app

2, Máy ảo android

3, Điều khiển DT thật, và view lên màn hình PC

TH1 : Bạn có điện thoại android ở ngoài đời, Chỉ cần xem hướng dẫn cài 1 và 3

TH2 : Bạn không có smartphone Android , Chỉ cần xem hướng dẫn cài 1 và 2

 **Lựa chọn 2 :**

- ✓ *Download trên trang chủ :*

1. <https://developer.android.com/studio>
2. <https://genymotion.com/download/>
3. <https://vysor.io/>





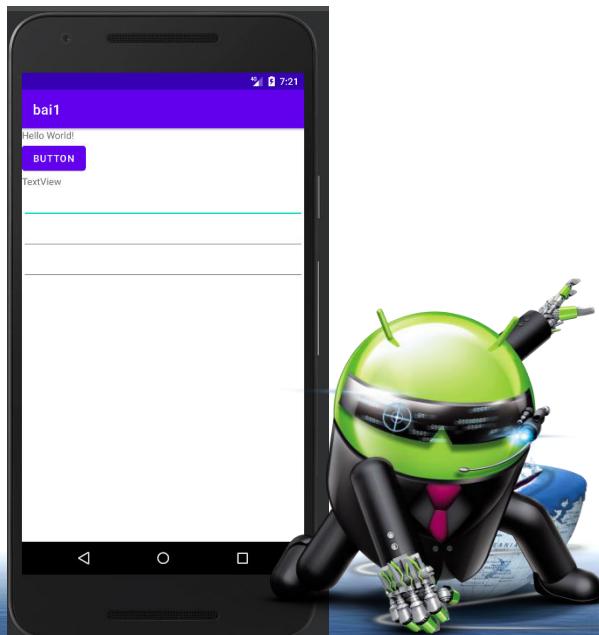
## Android app with kotlin Bài 2

1

Install Emulator android studio

2

Build first app with android studio



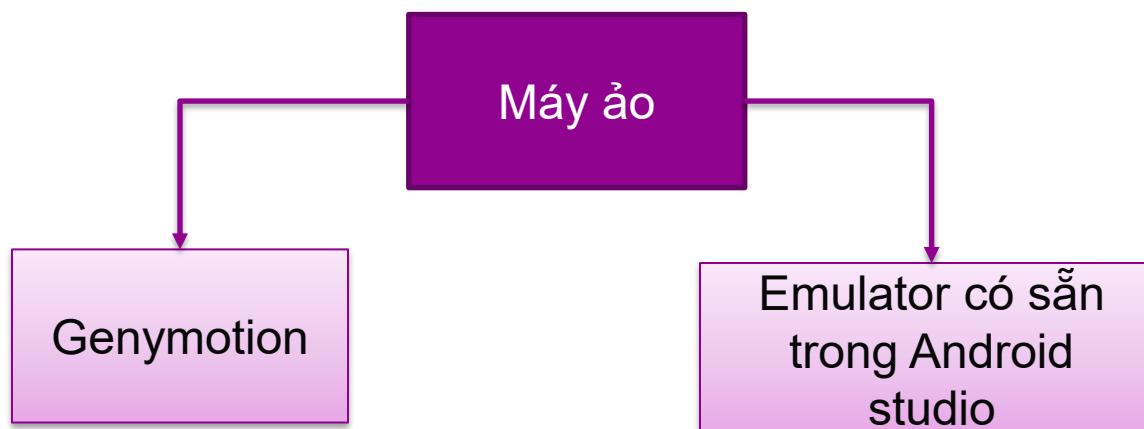
1

## Install android studio

 **Lựa chọn 1 :**

- ✓ *Download trọn bộ cài đặt tại : <http://android.tuhoc.cc/>*

Name	Size
..	
1.android-studio-2021.2.1.15-windows.exe	974.402.544
2.genymotion-3.2.1-vbox.exe	146.897.016
3.Vysor-win-4.1.77.exe	115.651.304

**1, Công cụ soạn thảo, thiết kế app****2, Máy ảo android****3, Điều khiển ĐT thật, và view lên màn hình PC**



## Android app with kotlin

### Bài 3

1

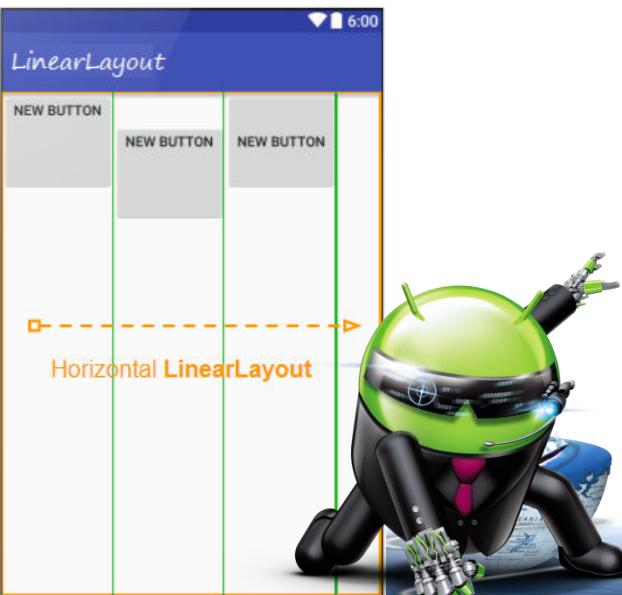
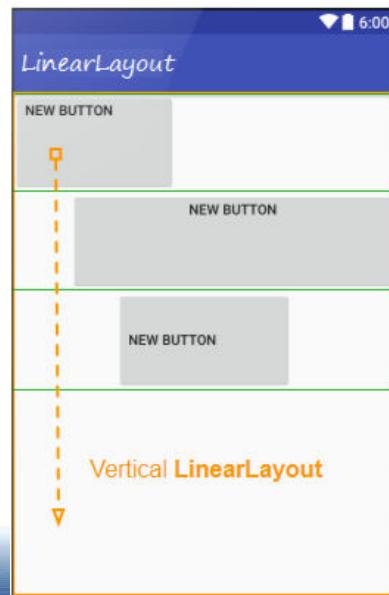
Làm quen android studio

2

LinearLayout

3

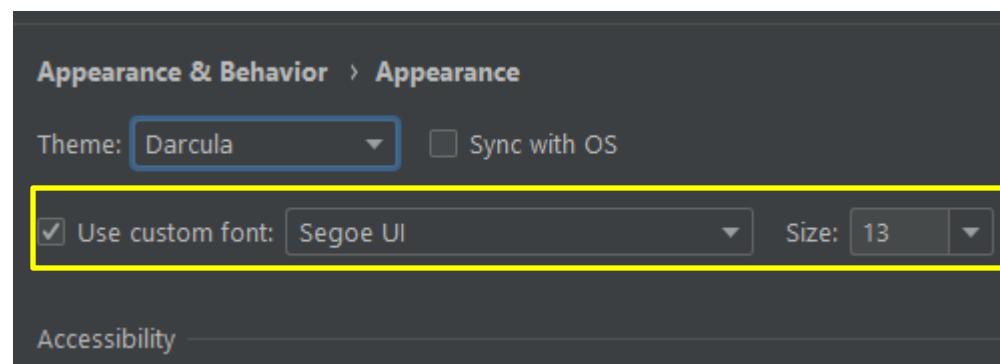
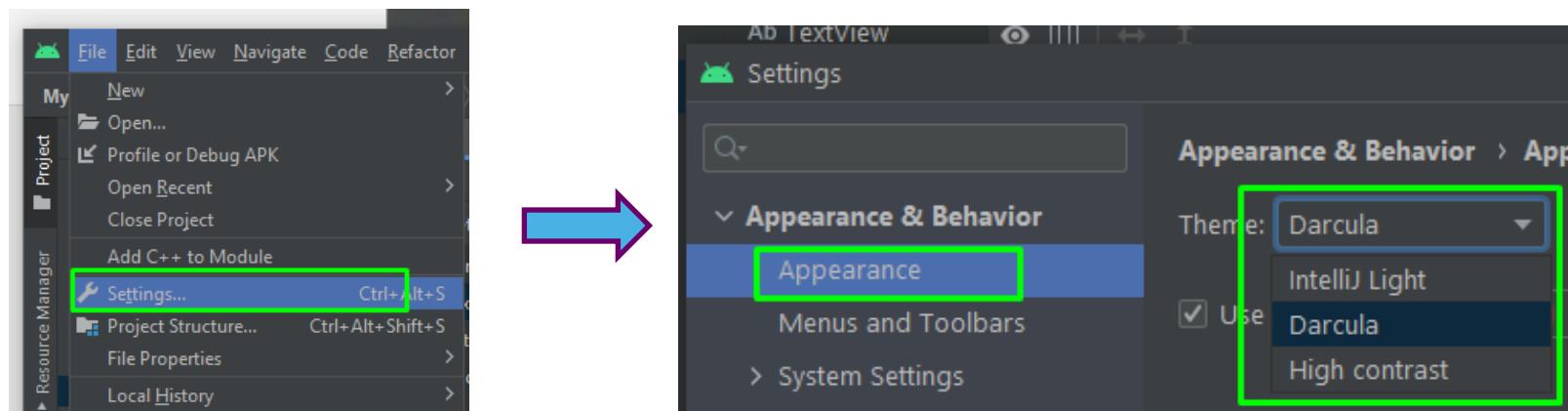
TableLayout



1

## Làm quen android studio

## □ 1. Chỉnh theme tối , sáng :



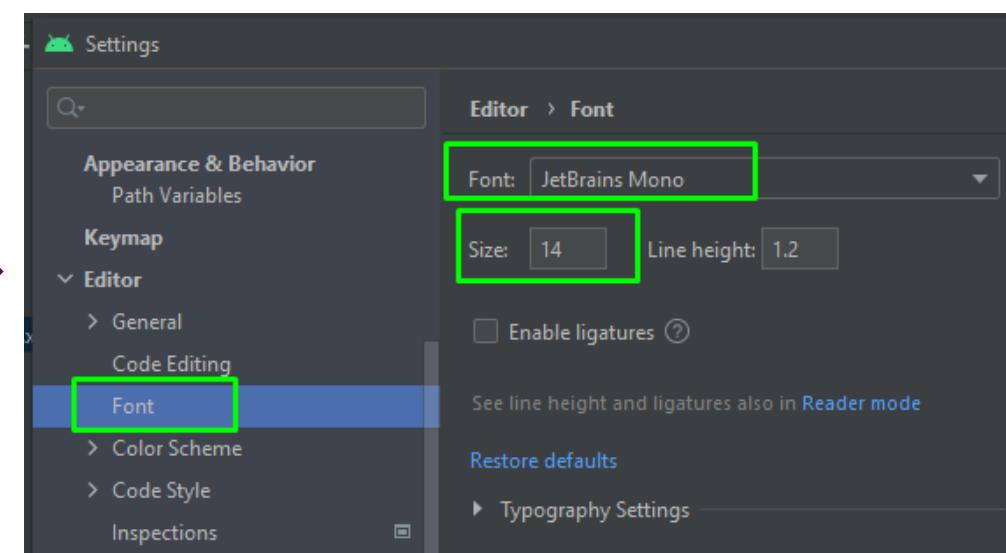
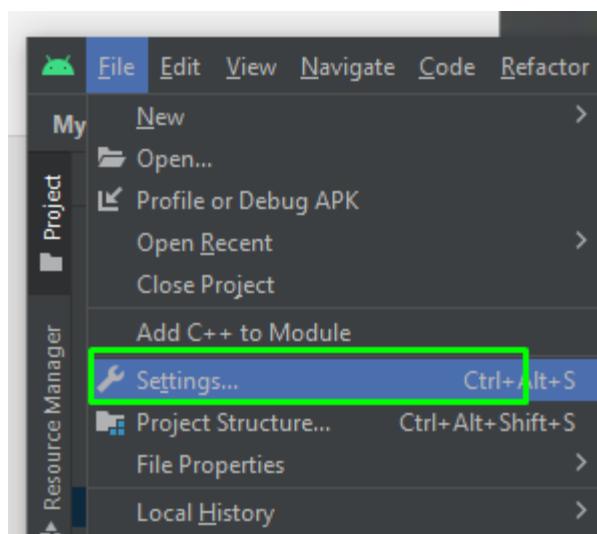
Font và cỡ chữ của android studio



1

## Làm quen android studio

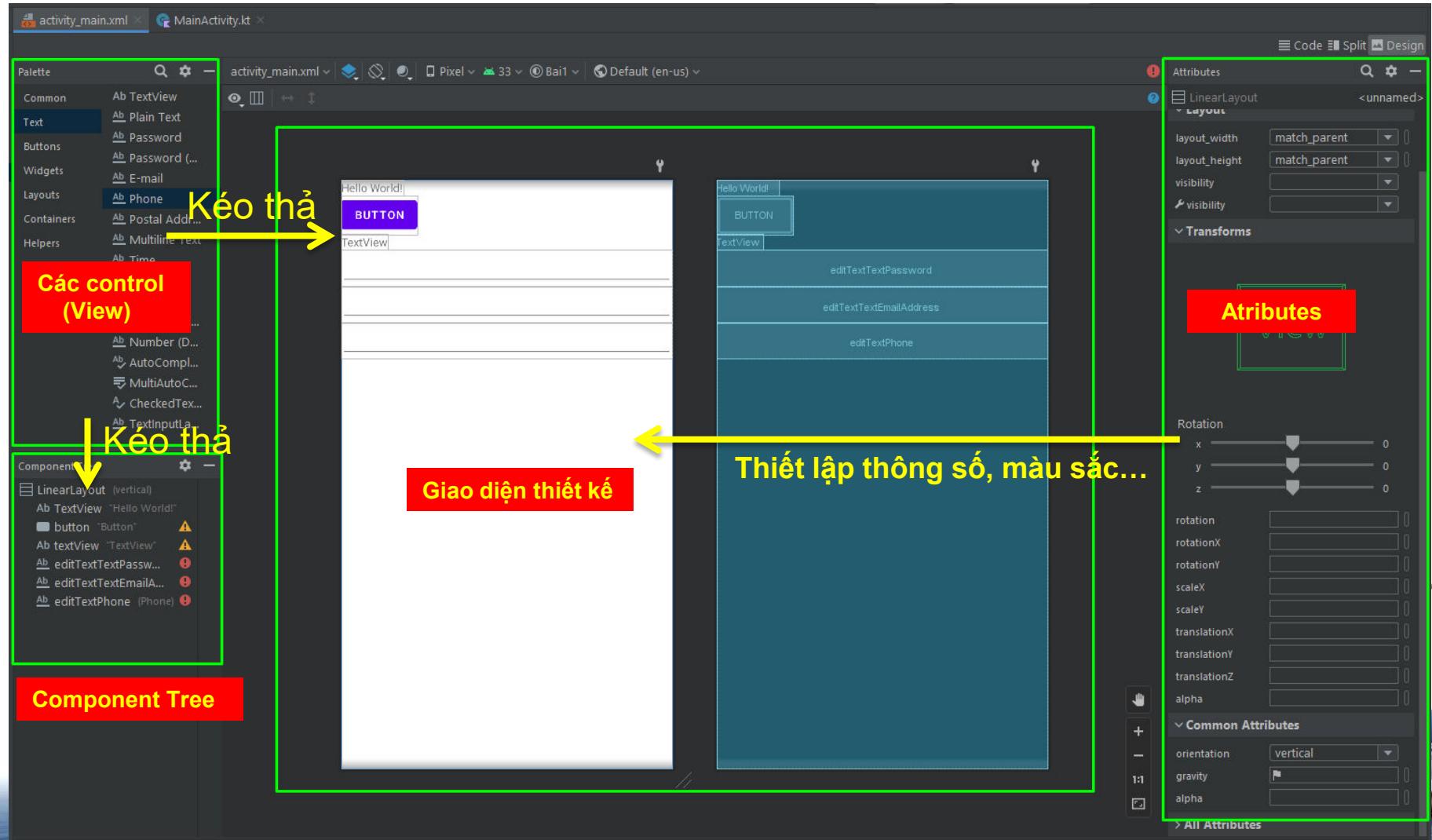
## □ 2. Chính font chữ, cỡ chữ coding :



1

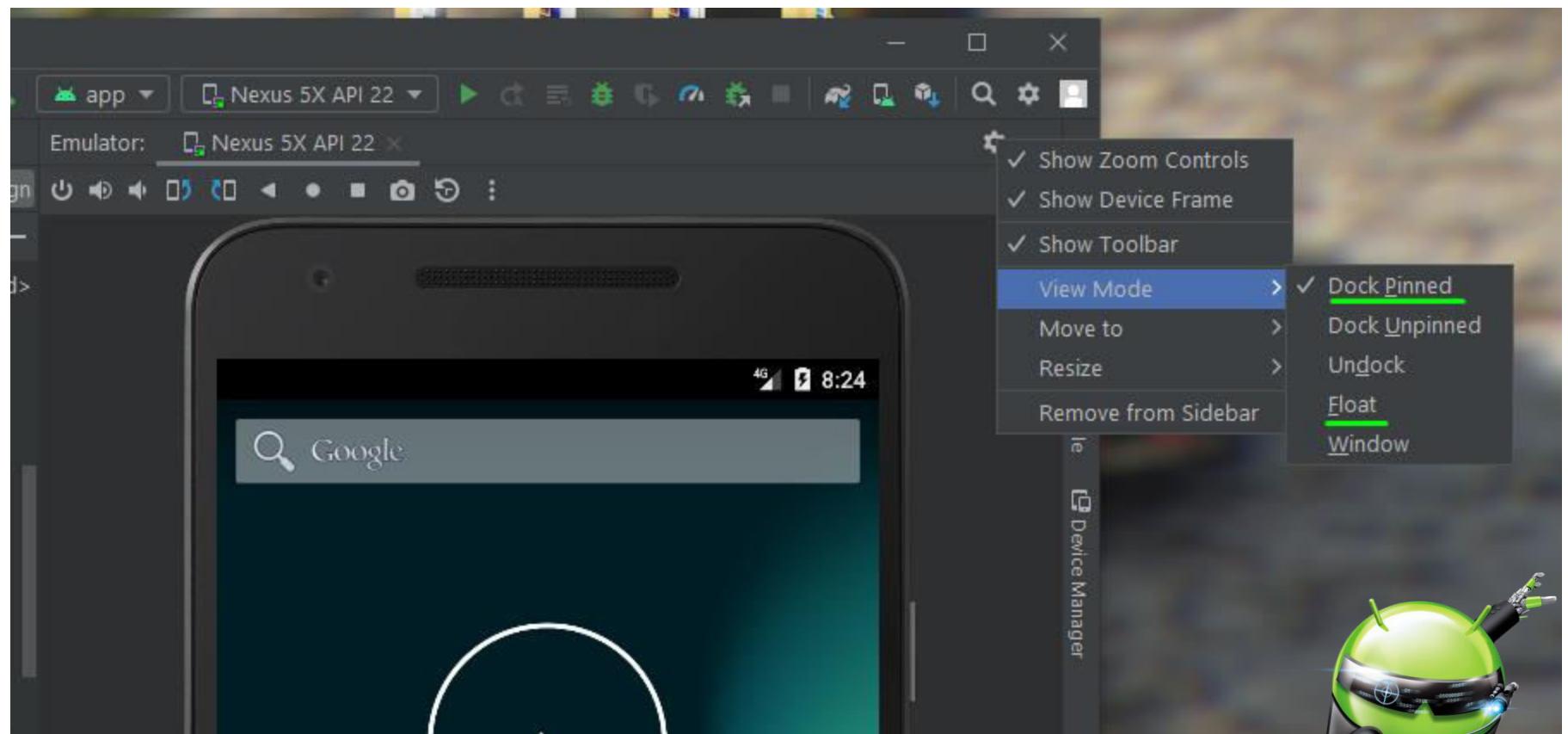
## Làm quen android studio

## □ 3. Giao diện thiết kế chính :



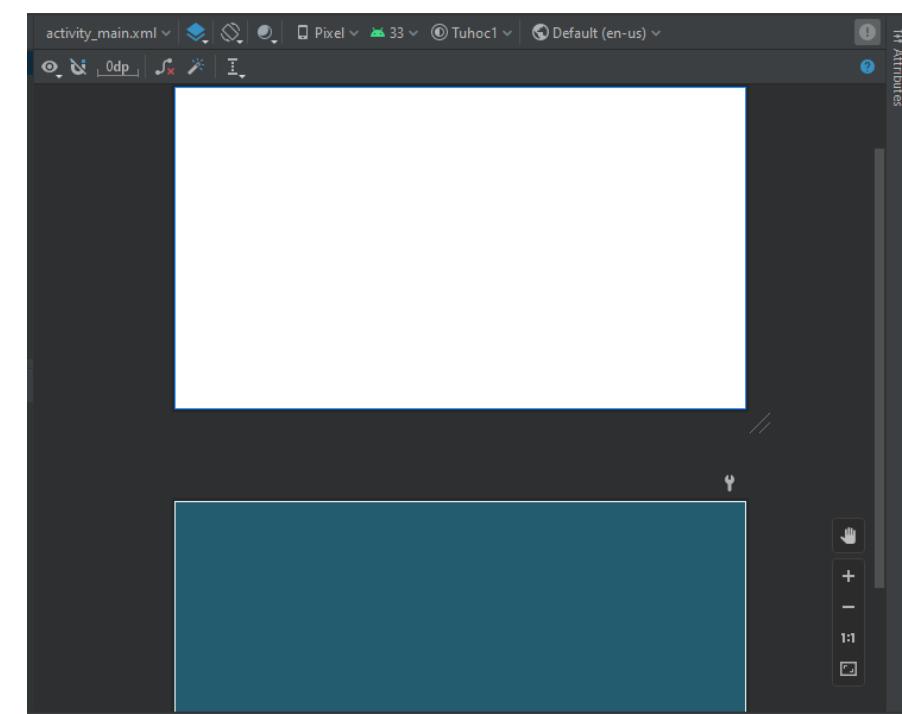
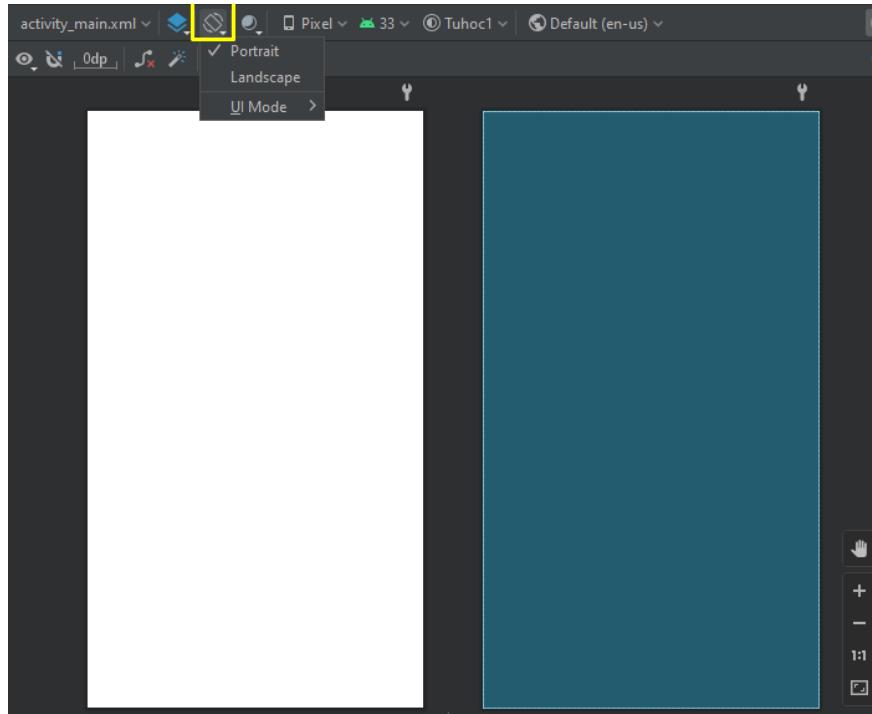
1

## Làm quen android studio

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

1

## Làm quen android studio

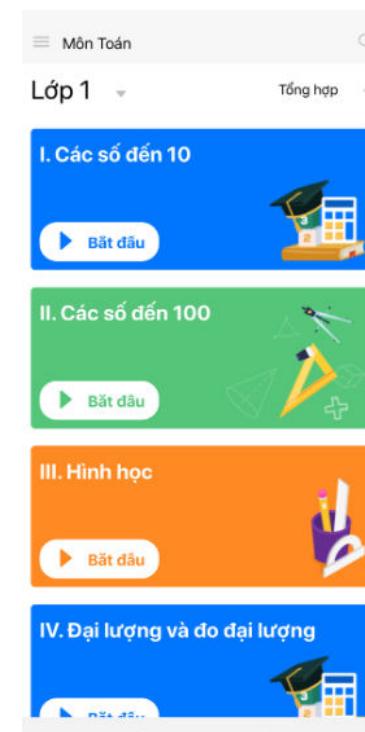
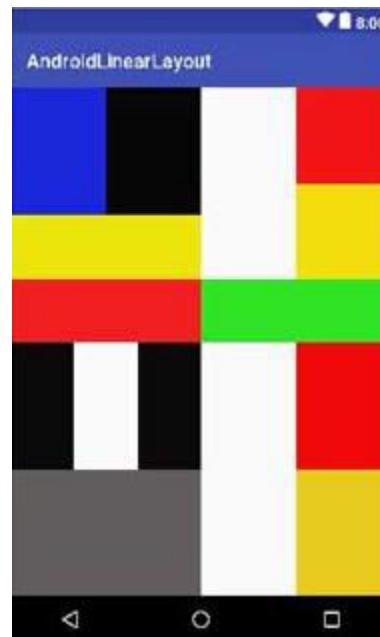
 5. Xoay ngang , dọc màn hình giao diện:

2

## LinearLayout

### □ 6. Khái quát LinearLayout :

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



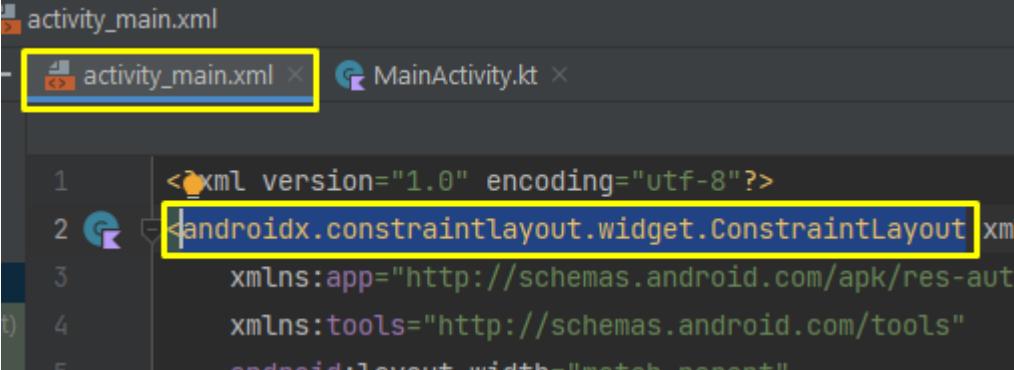
Section	Content	Icon
I. Các số đến 10	Các số từ 1 đến 10	Graduation cap icon
II. Các số đến 100	Các số từ 1 đến 100	Geometry tools icon
III. Hình học	Hình học cơ bản	Pencil icon
IV. Đại lượng và đo đại lượng	Đại lượng và đo lường	Graduation cap icon



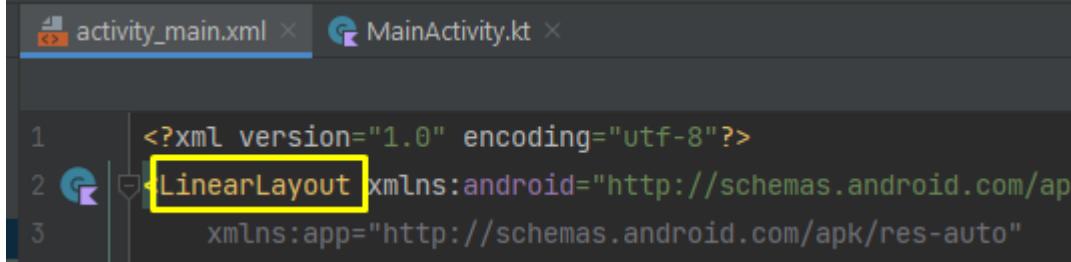
2

## LinearLayout

## □ 7. Set LinearLayout ngay sau khi tạo project :



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout>
```



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout>
```



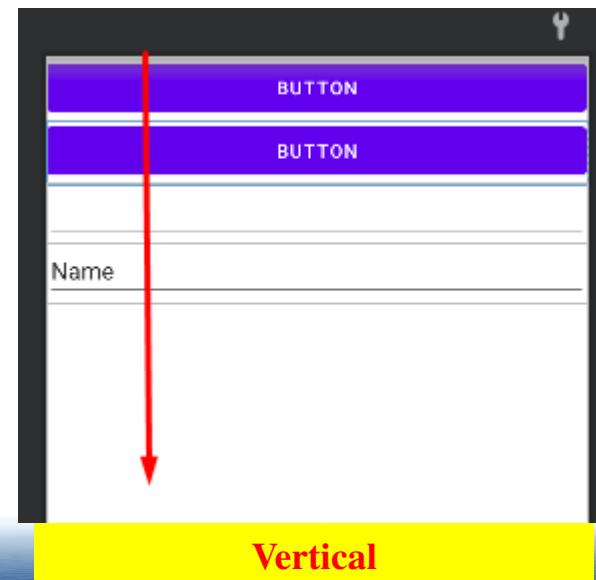
2

## LinearLayout

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

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

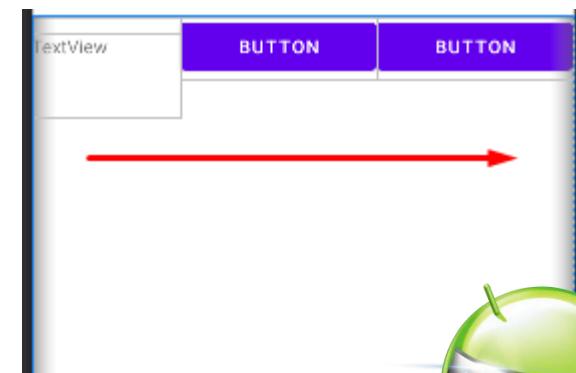
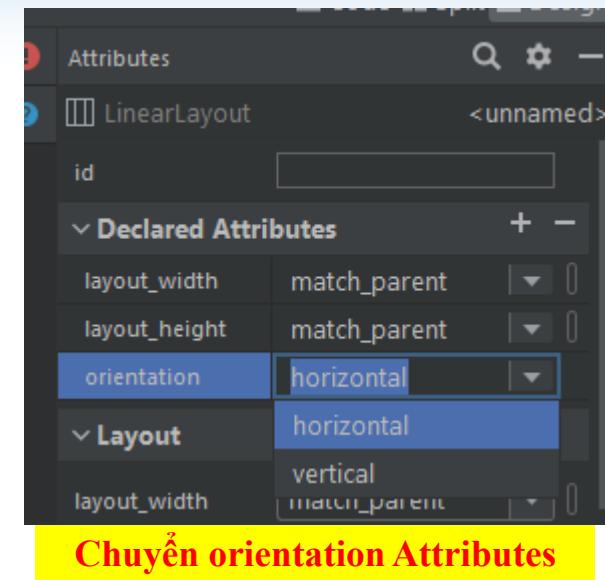
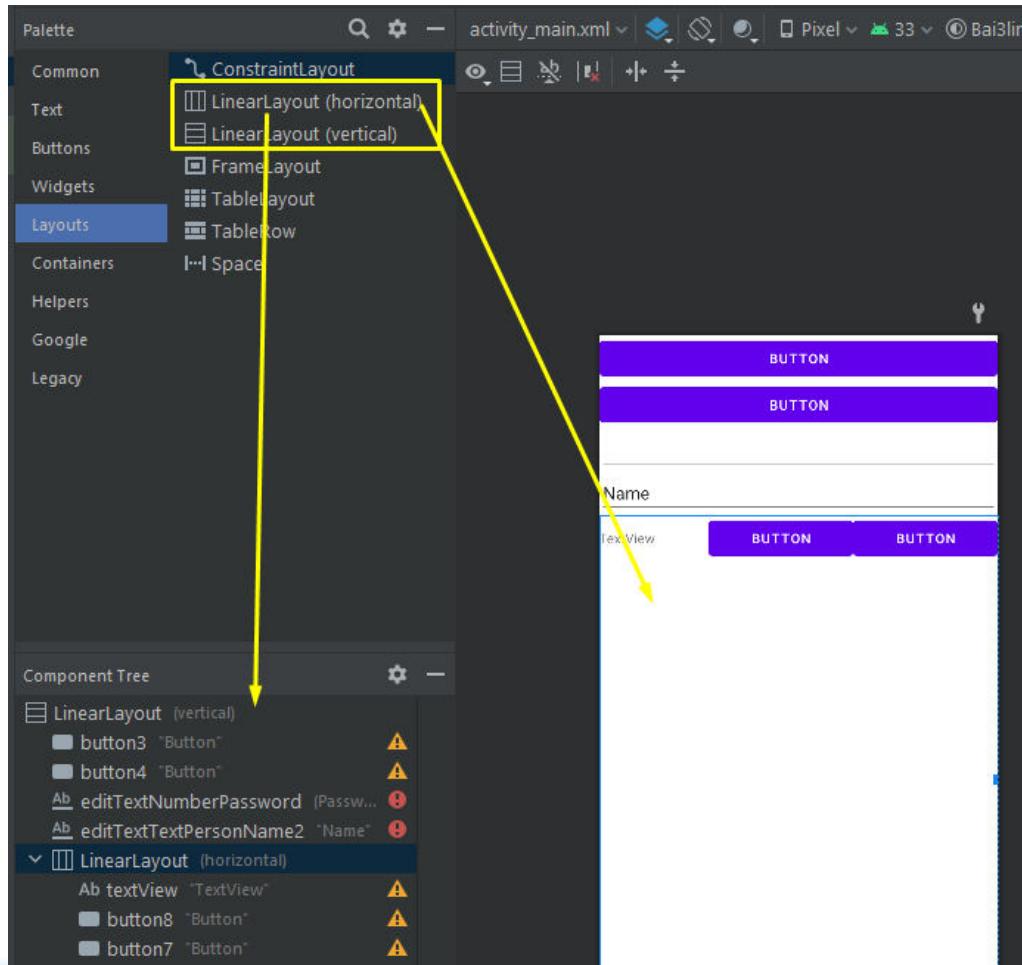
```
android:orientation=""  
    android:layout_height="vertical"  
    tools:context=".Main" horizontal  
  
    Ctrl+Down and Ctrl+Up will move caret
```



2

## LinearLayout

### ☐ Kéo thả LinearLayout :



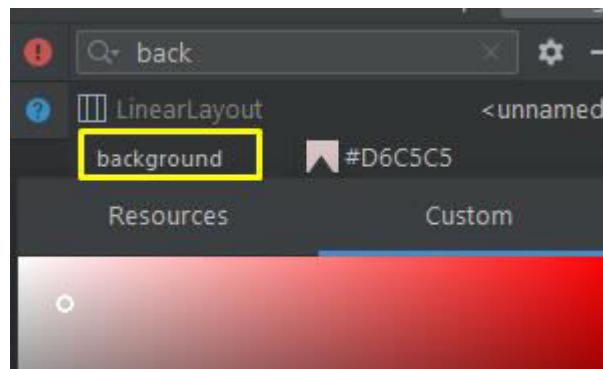
**Horizontal**



2

## LinearLayout

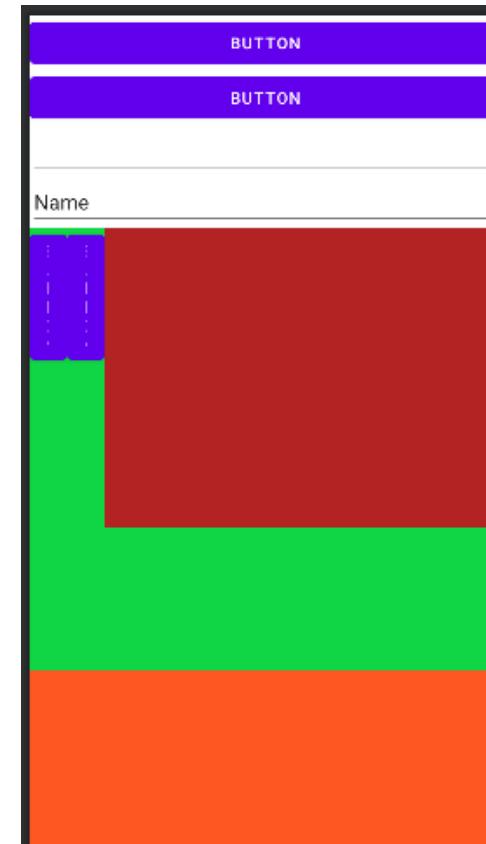
### □ 8.2 background: Màu nền



C1: Edit Attributes

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

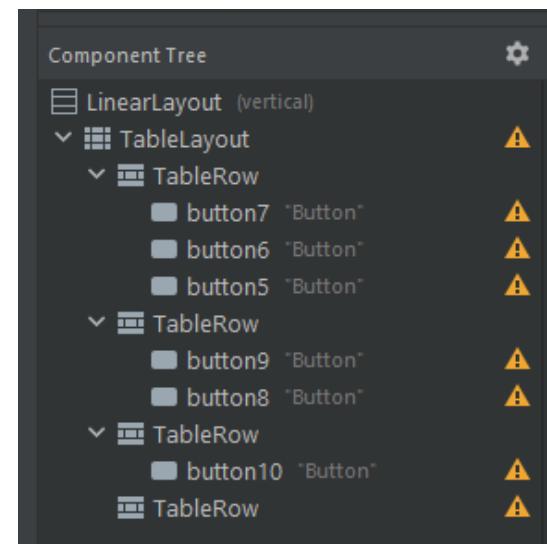
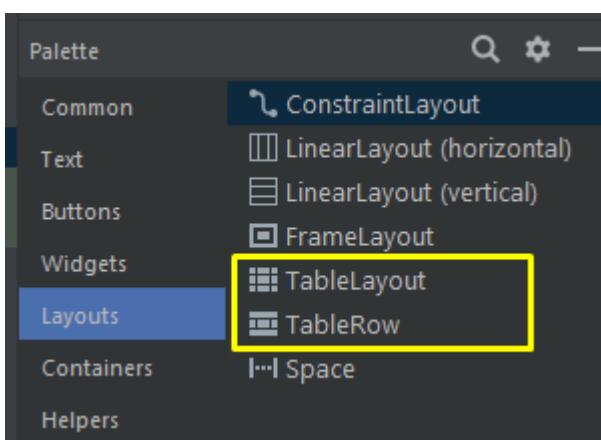
C2 : Sử dụng coding



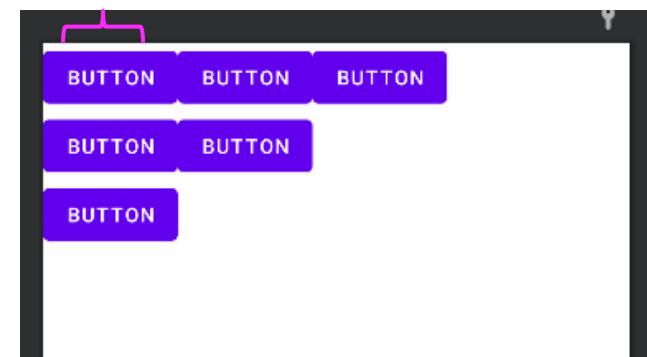
3

## 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



Cột 0    Cột 1    Cột 2



3

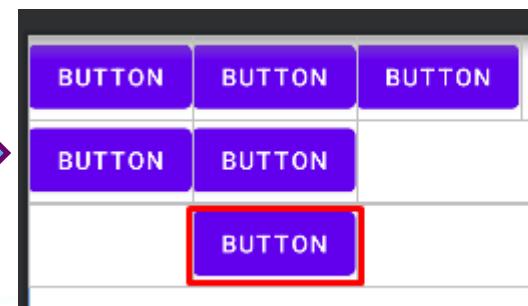
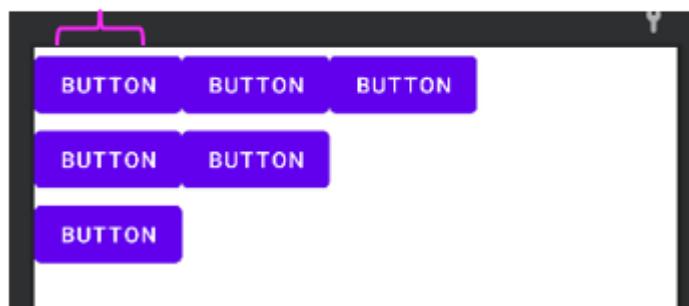
## TableLayout

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

STT	Thuộc tính	Nội dung
1	layout_column	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" />
```

Cột 0   Cột 1   Cột 2



3

## TableLayout

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

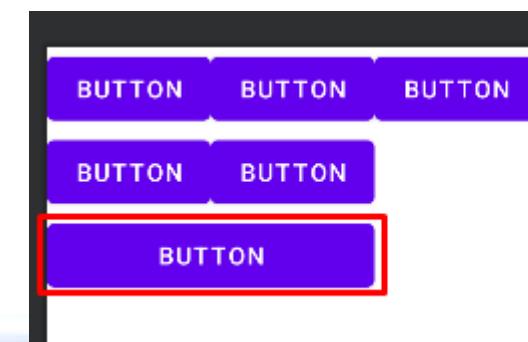
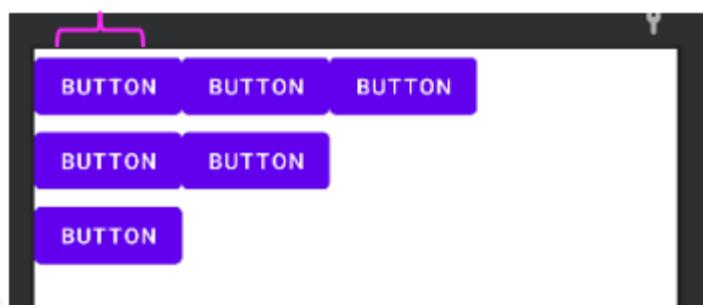
STT	Thuộc tính	Nội dung
2	layout_span="x"	Set vị trí của view nằm trên bao nhiêu cột

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

Cột 0

Cột 1

Cột 2

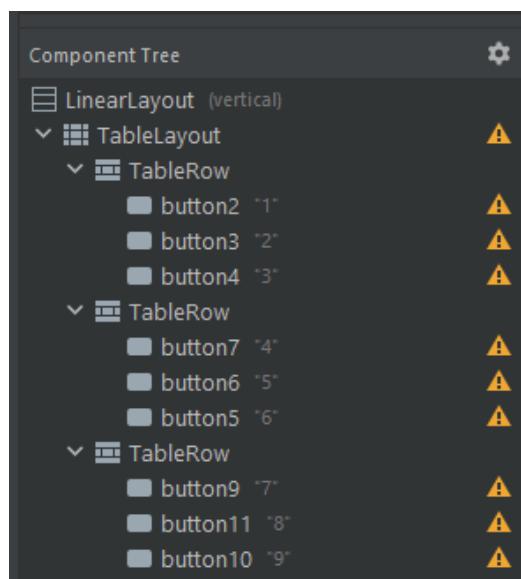


3

## 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



Code snippet showing the application of the `stretchColumns="*"` attribute:

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

The image illustrates the transformation of the layout. On the left, labeled "Trước" (Before), the layout consists of three separate rows of three buttons each. An arrow points to the right, labeled "Sau" (After), where the three rows have been merged into a single row, demonstrating the effect of the `stretchColumns="*"` attribute.



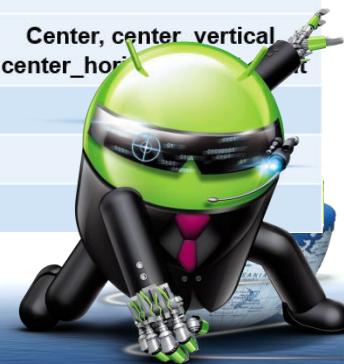


## Android app with kotlin

### Bài 3,2

Các thuộc tính cơ bản (P2)

STT	Thuộc tính	Nội dung	Chi tiết
1	<b>orientation</b>	Vertical: theo chiều dọc	
		Horizontal: theo chiều ngang	
2	<b>background</b>	Set màu nền	
3	<b>Layout_width</b>	Chiều rộng layout	Wrap_content, match_parent, px, dp
4	<b>Layout_height</b>	Chiều cao layout	
5	<b>Gravity</b>	Sắp xếp các thành phần nằm trên view, view group	Center, center_vertical, center_horizontal
6	<b>Margin</b>	Căn lề layout so với view chứa nó	
7	<b>padding</b>	Căn lề nội dung trong nó	
8	<b>Weight</b>	Chia theo trọng số	



2

## LinearLayout

### ❑ Các thuộc tính quan trọng thường gặp :

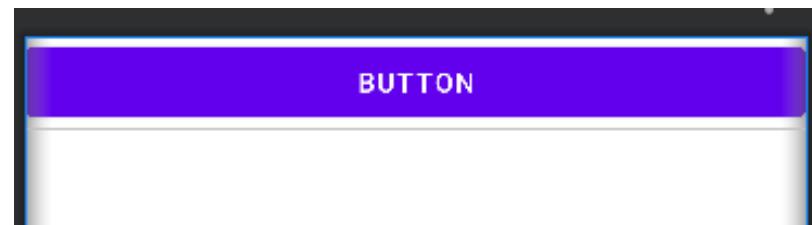
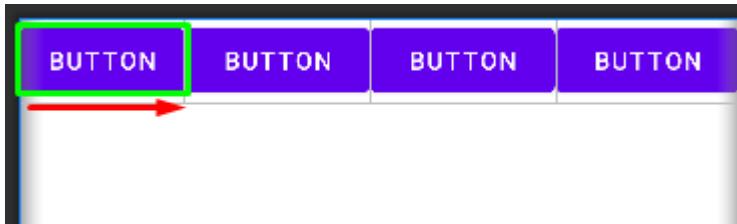
STT	Thuộc tính	Nội dung	Chi tiết
1	<b>orientation</b>	<b>Vertical:</b> theo chiều dọc	
		<b>Horizontal:</b> theo chiều ngang	
2	<b>background</b>	<b>Set màu nền</b>	
3	<b>Layout_width</b>	<b>Chiều rộng layout</b>	
4	<b>Layout_height</b>	<b>Chiều cao layout</b>	<b>Wrap_content,</b> <b>match_parent,px, pd</b>
5	<b>Gravity</b>	<b>Sắp xếp các thành phần nằm trên view, view group</b>	<b>Center, center_vertical,</b> <b>center_horizontal, left, right</b>
6	<b>Margin</b>	<b>Căn lề layout so với view chứa nó</b>	
7	<b>padding</b>	<b>Căn lề nội dung trong nó</b>	
8	<b>Weight</b>	<b>Chia theo trọng số</b>	



## 2

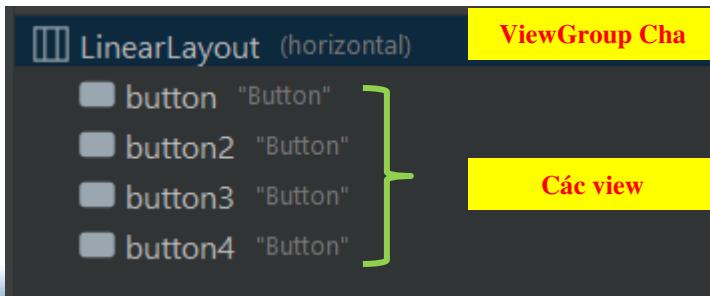
## LinearLayout

STT	Thuộc tính	Nội dung	Chi tiết
3	Layout_width	Chiều rộng layout	Wrap_content, match_parent,px, pd
4	Layout_height	Chiều cao layout	



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

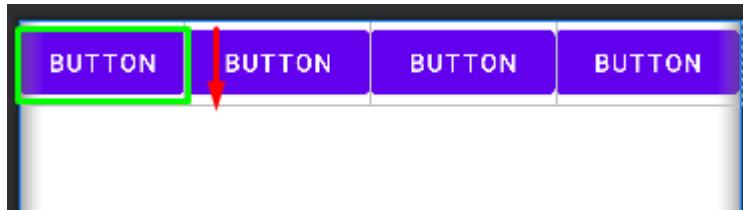
```
<Button
    android:id="@+id/button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button" />
```



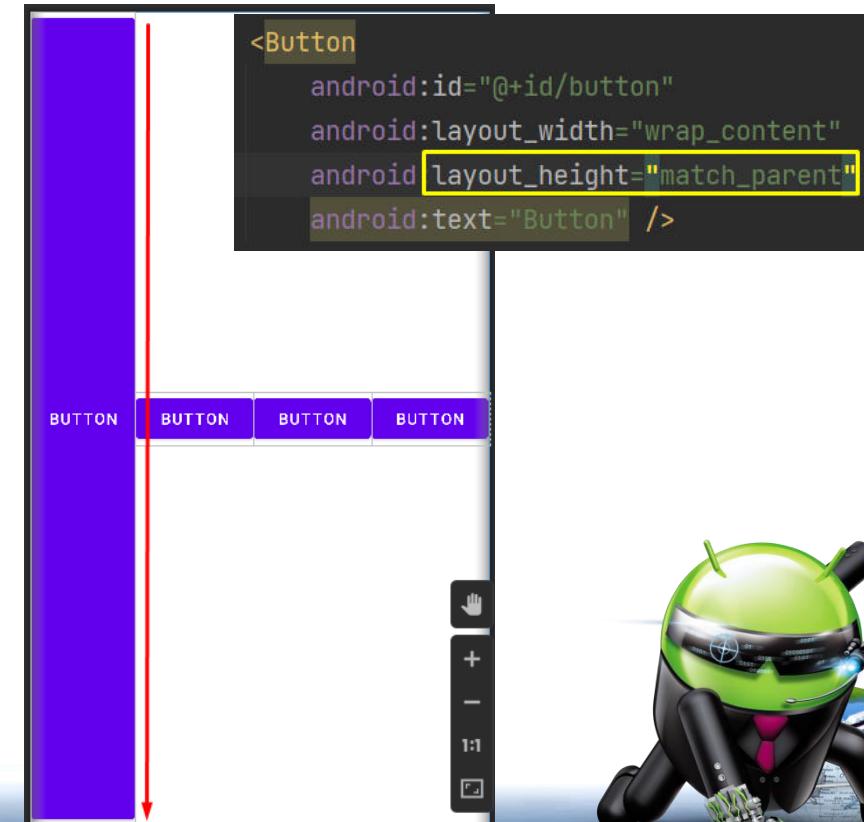
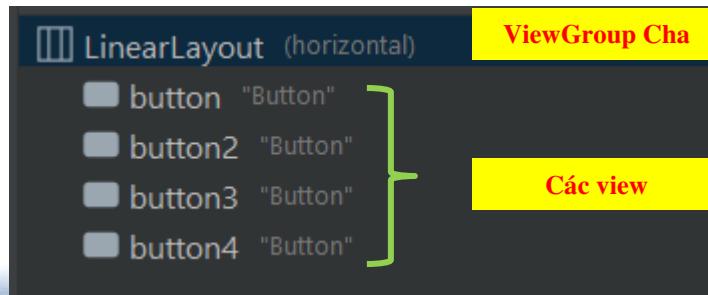
## 2

## LinearLayout

STT	Thuộc tính	Nội dung	Chi tiết
3	Layout_width	Chiều rộng layout	Wrap_content, match_parent, px, pd
4	Layout_height	Chiều cao layout	



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

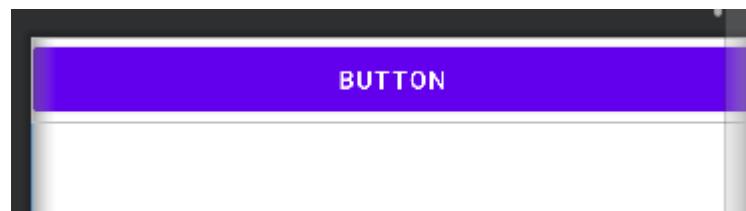


2

## LinearLayout

STT	Thuộc tính	Nội dung	Chi tiết
5	<b>Gravity</b>	Sắp xếp các thành phần nằm trên view, view group	Center, center_vertical, center_horizontal, left, right

```
<Button
    android:id="@+id/button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Button" />
```



```
    android:gravity=""
    android:layout_h[center
    android:text="Bu[fill_vertical
    fill
    clip_vertical
    center_vertical
    center_horizontal]
```



android:gravity="left"



android:gravity="right"

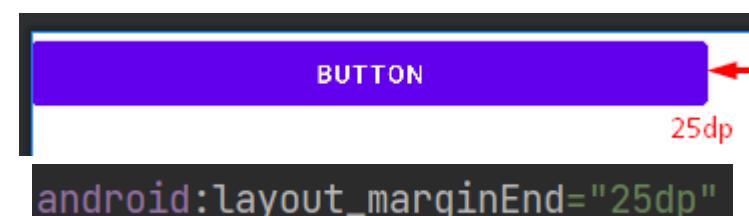
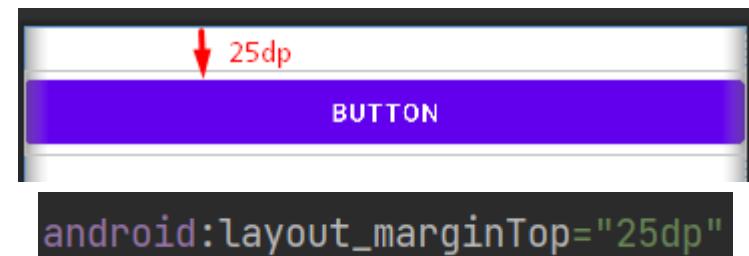


2

## LinearLayout

STT	Thuộc tính	Nội dung	Chi tiết
6	Margin	Căn lè layout so với view chứa nó	

```
<Button
    android:id="@+id/button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Button" />
```



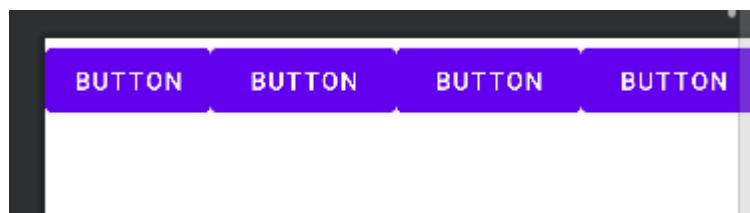
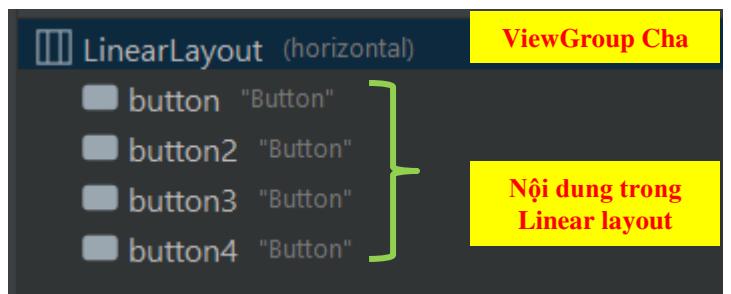
```
ma
android:layout_marginTop
android:layout_margin
android:layout_marginRight
android:layout_marginLeft
android:layout_marginStart
android:layout_marginBottom
```



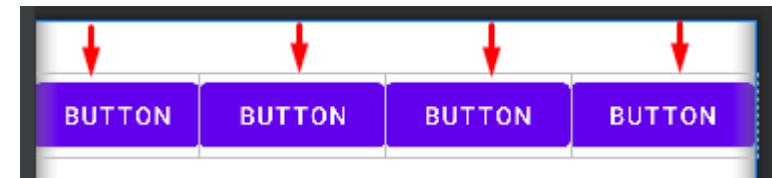
## 2

## LinearLayout

STT	Thuộc tính	Nội dung	Chi tiết
7	padding	Căn lề nội dung trong nó	



```
android:orientation="horizontal"
        android:paddingTop="30dp"
```



Tất cả các view sẽ căn lề trên (Top) 30dp

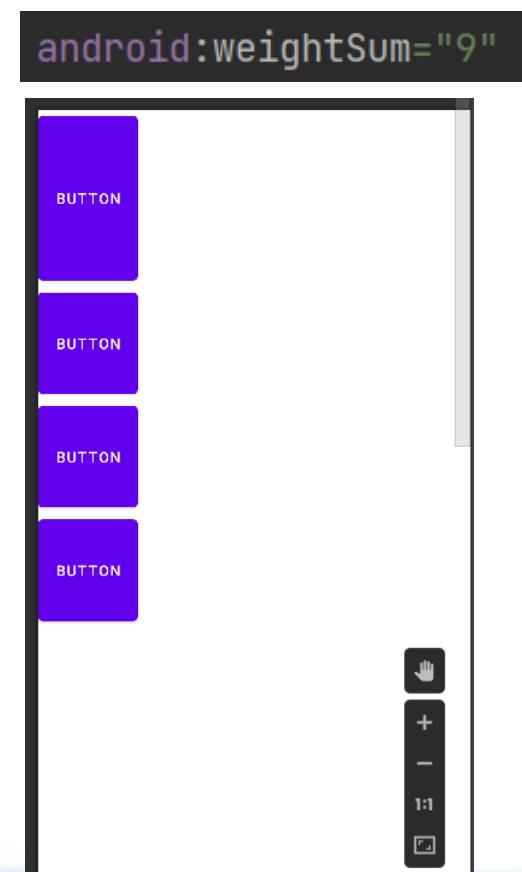
```
android:orientation="horizontal"
pa...
android:padding
        android:paddingTop
        android:paddingLeft
        android:paddingStart
```



2

## LinearLayout

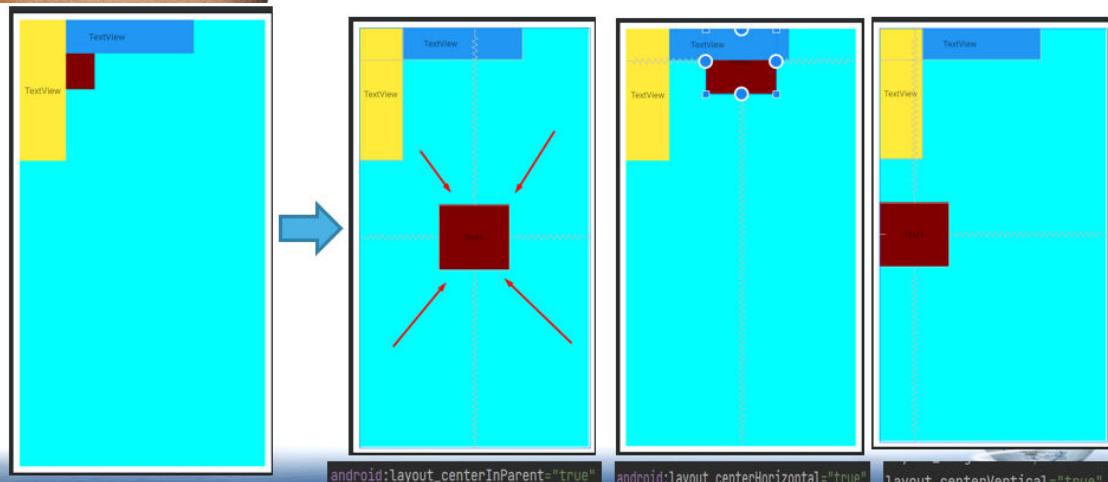
STT	Thuộc tính	Nội dung	Chi tiết
8	Weight	Chia theo trọng số	





## Android app with kotlin Bài 3,3

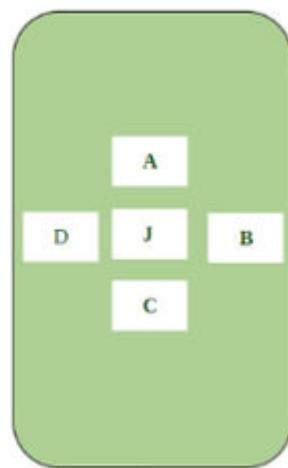
### Relative Layout



Relative View 1



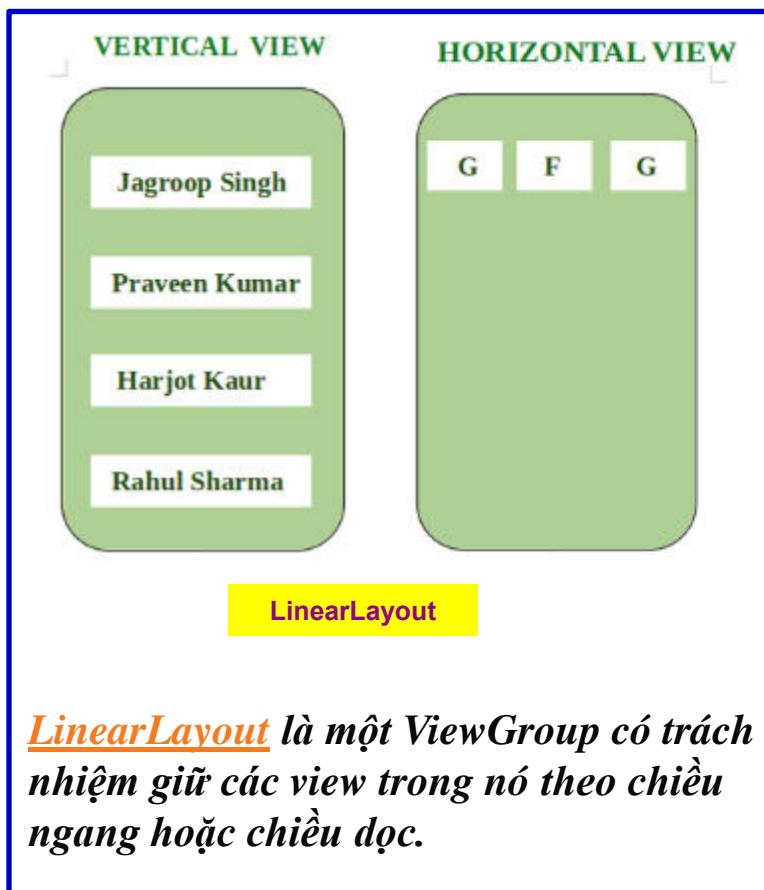
Relative View 2



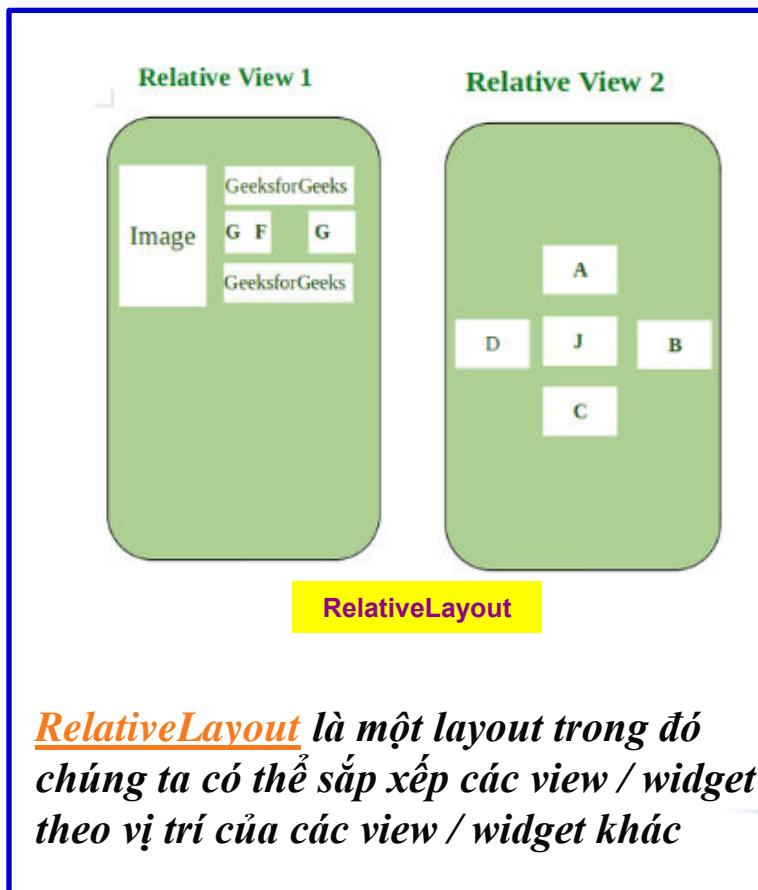
RelativeLayout

## Relative Layout

### □ 1. LinearLayout & RelativeLayout :



**LinearLayout** là một **ViewGroup** có trách nhiệm giữ các view trong nó theo chiều ngang hoặc chiều dọc.

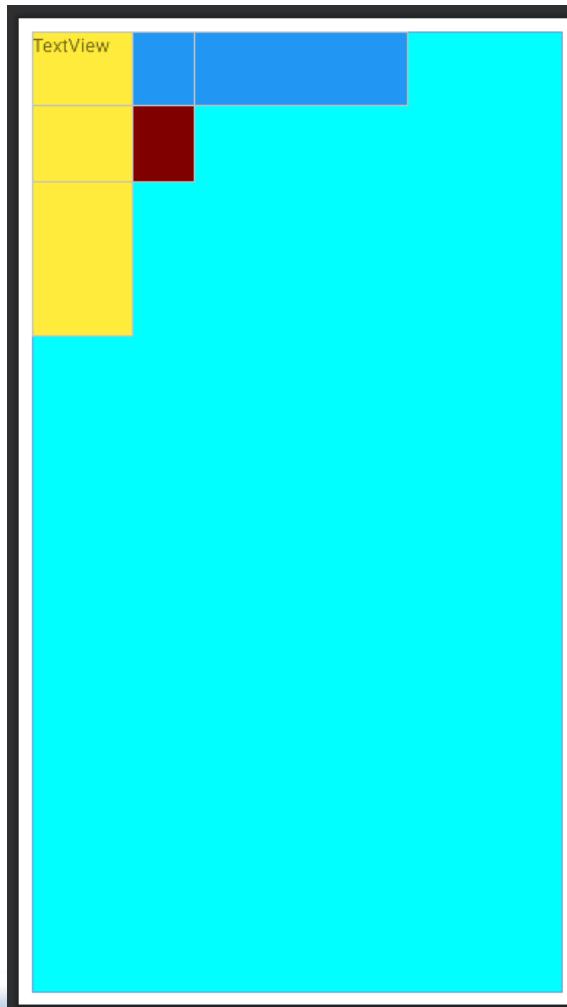


**RelativeLayout** là một layout trong đó chúng ta có thể sắp xếp các view / widget theo vị trí của các view / widget khác



## Relative Layout

### ❑ RelativeLayout :



**\*\* Khi chưa thiết lập mối ràng buộc vị trí với ViewGroup cha, hoặc với các view con khác thì mặc định các view sẽ được đặt tại gốc 0,0 ( góc trên, bên trái màn hình )**

**\*\* View khai báo sau sẽ chồng lên view khai báo trước nó**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#00FFFF"
    android:layout_margin="10dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/txt1"
        android:layout_width="120dp"
        android:layout_height="111dp"
        android:background="#800000"
        android:text="TextView" />

    <TextView
        android:id="@+id/txt2"
        android:layout_width="278dp"
        android:layout_height="54dp"
        android:background="#2196F3"
        android:text="TextView" />

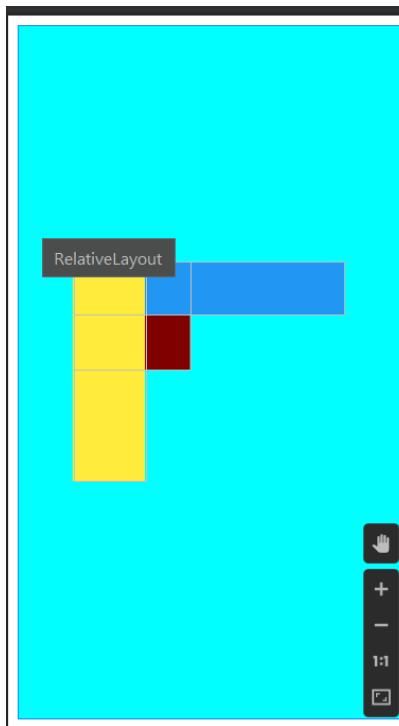
    <TextView
        android:id="@+id/txt3"
        android:layout_width="74dp"
        android:layout_height="225dp"
        android:background="#FFEB3B"
        android:text="TextView" />

</RelativeLayout>
```

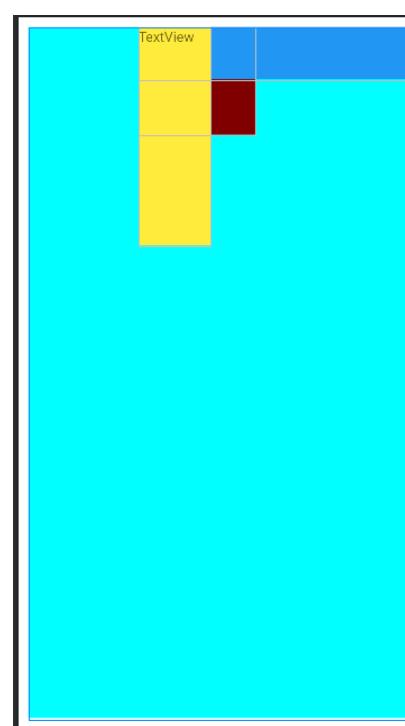


## Relative Layout

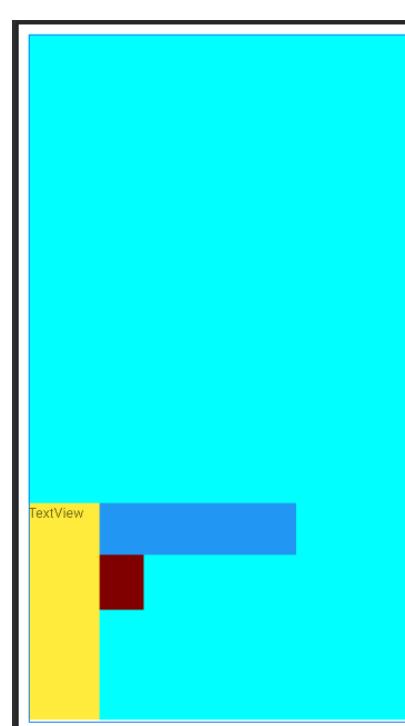
STT	Thuộc tính	Nội dung	Chi tiết
1	Gravity	Sắp xếp các thành phần nằm trên view, view group	Center, center_vertical, center_horizontal, left, right



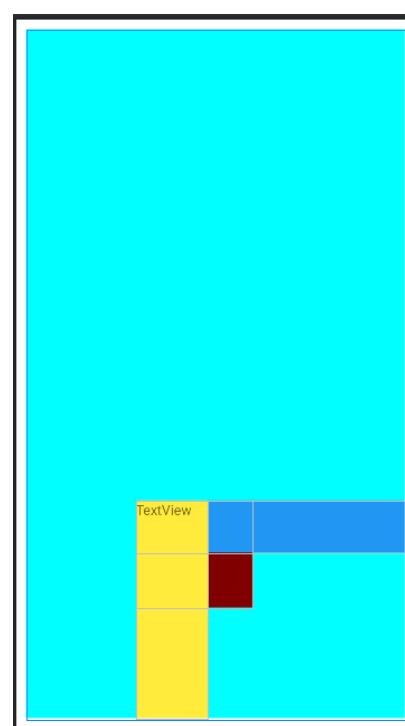
`android:gravity="center"`



`android:gravity="right"`



`android:gravity="bottom"`



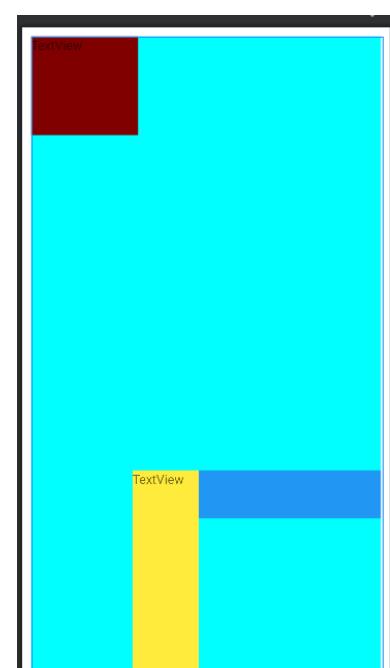
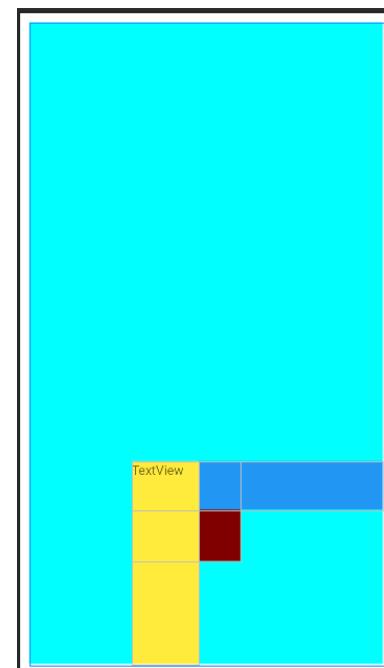
`android:gravity="bottom|right"`

## Relative Layout

STT	Thuộc tính	Nội dung	Chi tiết
1	<b>Gravity</b>	Sắp xếp các thành phần nằm trên view, view group	Center, center_vertical, center_horizontal, left, right

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#00FFFFFF"
    android:gravity="bottom|right"
    android:ignoreGravity="@+id/txt1"
    android:layout_margin="10dp"
    tools:context=".MainActivity">
```

*ignoreGravity : Bỏ qua áp dụng Gravity với 1 view nào đó*



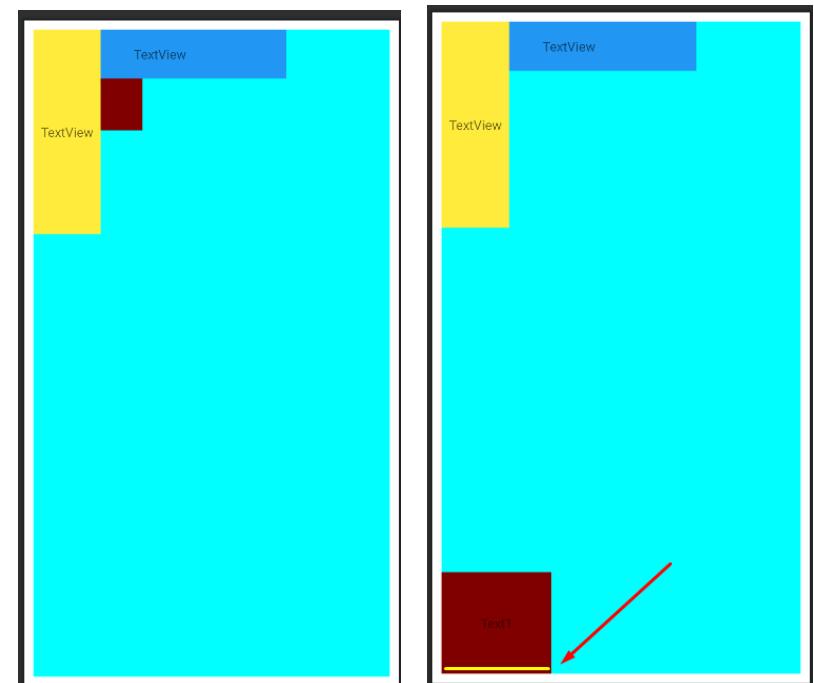
## Relative Layout

### □ 2.1 Thiết lập vị trí giữa view con và view cha : Dựa trên align

```
android:layout_alignParentBottom  
android:layout_alignParentEnd  
android:layout_alignParentLeft  
android:layout_alignParentRight  
android:layout_alignParentStart  
android:layout_alignParentTop
```

#### Thiết lập tại view con

```
<TextView  
    android:id="@+id/txt1"  
    android:layout_width="120dp"  
    android:layout_height="111dp"  
    android:background="#800000"  
    android:layout_alignParentBottom="true"  
    android:gravity="center"  
    android:text="Text1" />
```



```
android:layout_alignParentBottom="true"
```

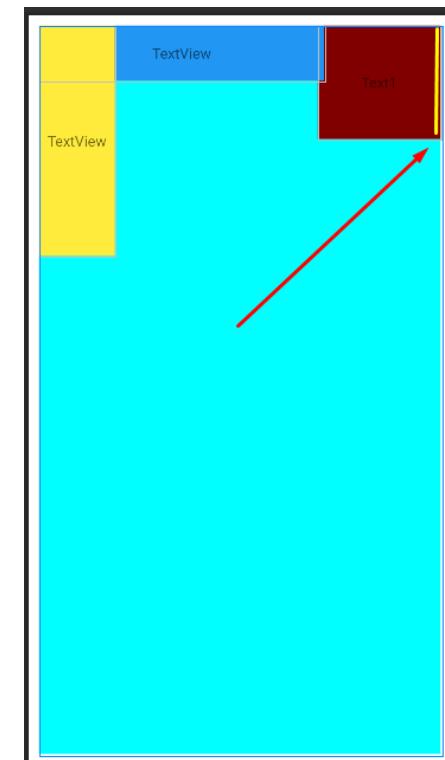
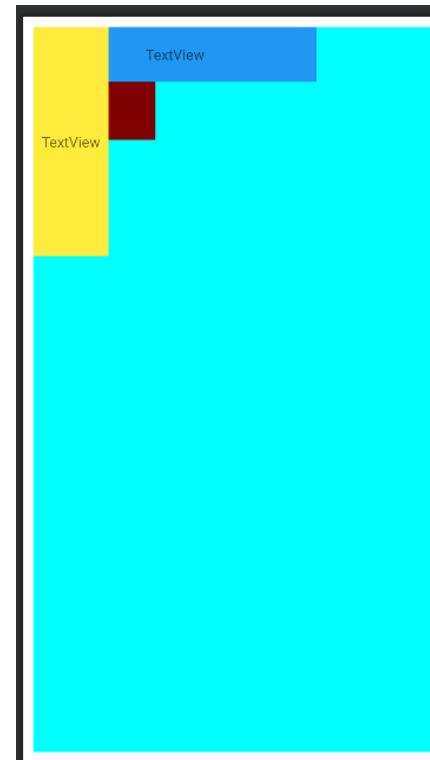
Cạnh dưới view con trùng với  
cạnh dưới view cha



## Relative Layout

### □ 2.1 Thiết lập vị trí giữa view con và view cha : Dựa trên align

```
<TextView  
    android:id="@+id/txt1"  
    android:layout_width="120dp"  
    android:layout_height="111dp"  
    android:background="#800000"  
    android:layout_alignParentEnd="true"  
    android:gravity="center"  
    android:text="Text1" />
```



android:layout\_alignParentEnd="true"

Cạnh trái view con trùng với  
cạnh trái view cha

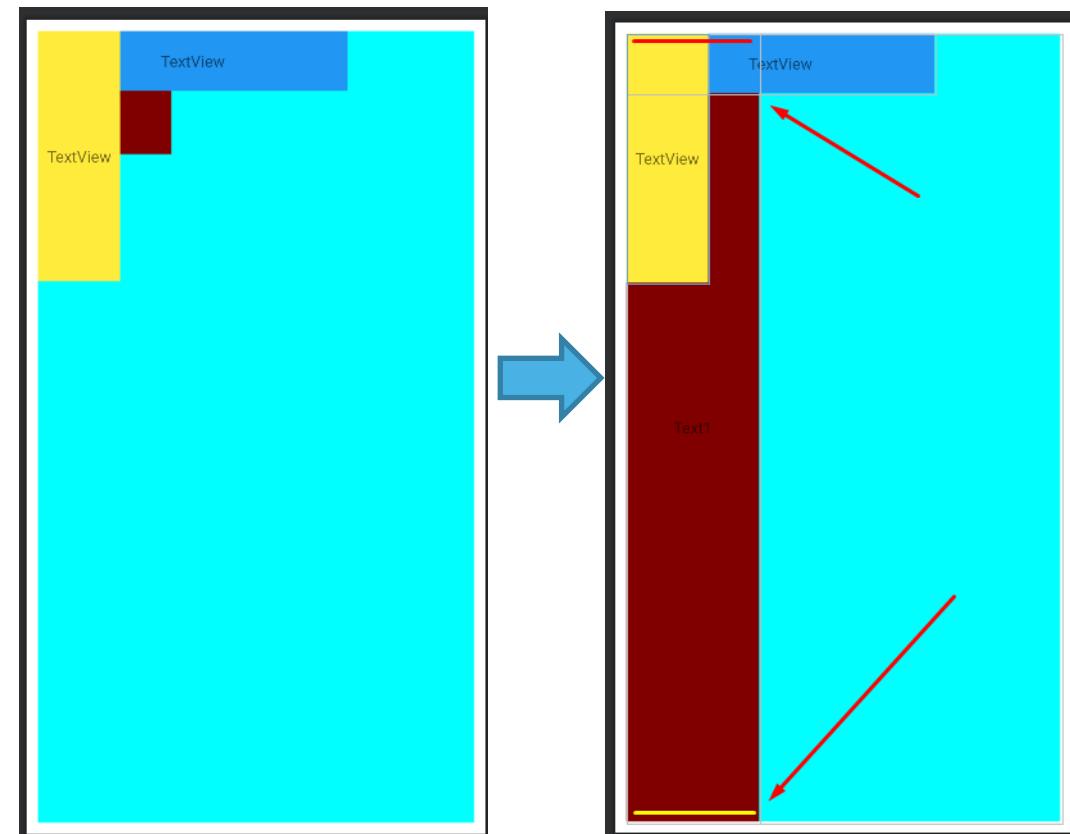


## Relative Layout

### □ 2.1 Thiết lập vị trí giữa view con và view cha : Dựa trên align

```
<TextView  
    android:id="@+id/txt1"  
    android:layout_width="120dp"  
    android:layout_height="111dp"  
    android:background="#800000"  
    android:layout_alignParentBottom="true"  
    android:layout_alignParentTop="true"  
    android:gravity="center"  
    android:text="Text1" />
```

Nếu có nhiều ràng buộc vị trí



Cạnh dưới và cạnh trên của view con trùng cạnh dưới và trên của view cha

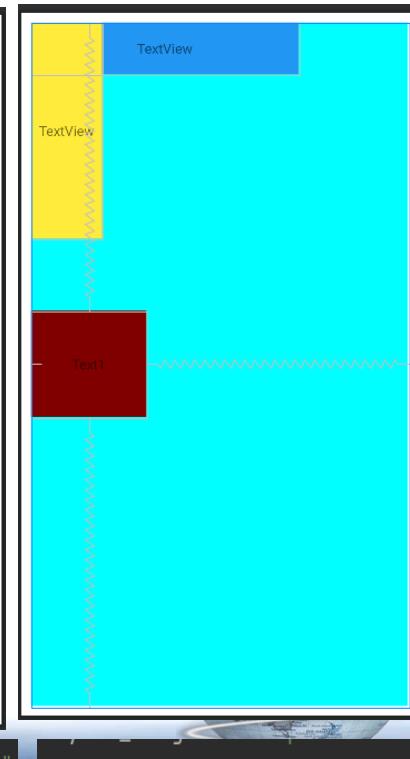
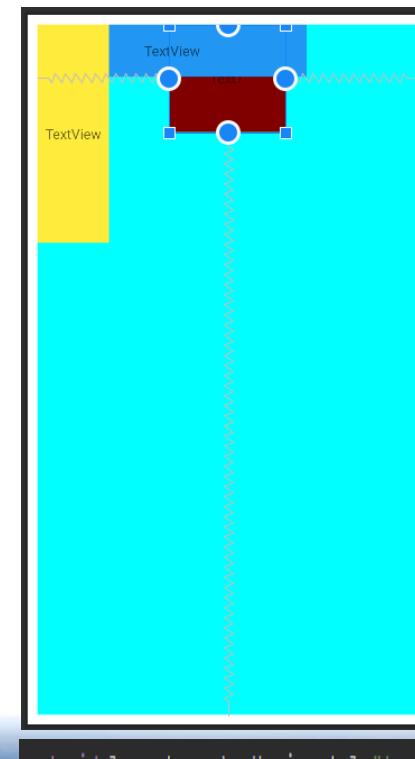
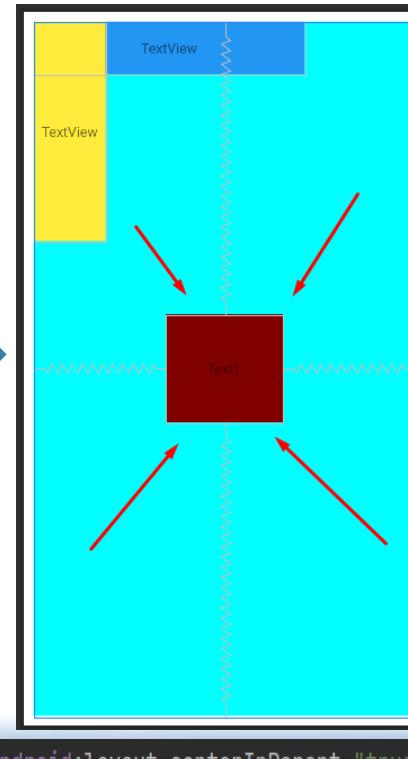
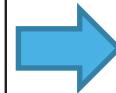
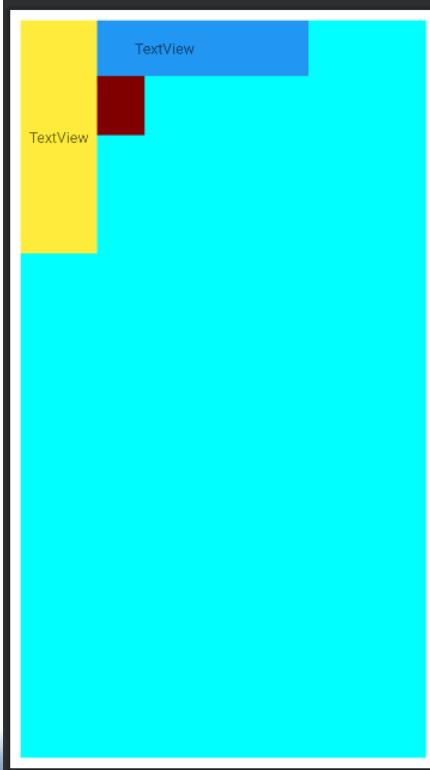


## Relative Layout

### ❑ 2.2 Thiết lập vị trí giữa view con và view cha : Căn giữa center

```
android:layout_centerInParent  
android:layout_centerHorizontal  
android:layout_centerVertical
```

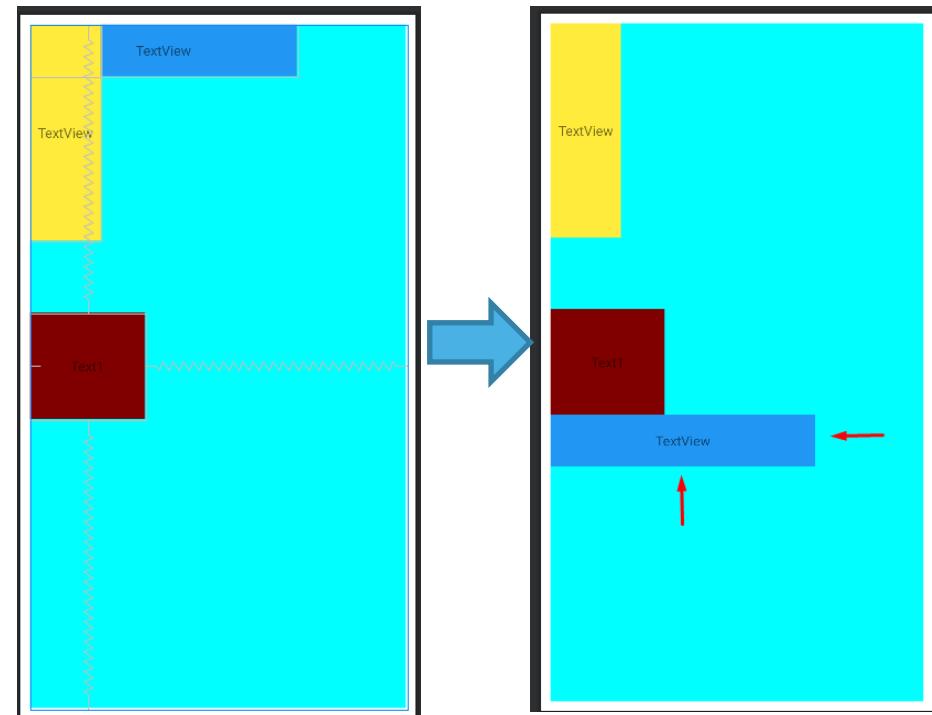
Press Ctrl+Space to view tags from other namespaces [Next Tip](#)



## Relative Layout

### □ 3. Thiết lập vị trí giữa view con với view con : Trên, dưới, trái, phải

```
<TextView  
    android:id="@+id/txt1"  
    android:layout_width="120dp"  
    android:layout_height="111dp"  
    android:layout_centerVertical="true"  
    android:background="#800000"  
    android:gravity="center"  
    android:text="Text1" />  
  
<TextView  
    android:id="@+id/txt2"  
    android:layout_width="278dp"  
    android:layout_height="54dp"  
    android:background="#2196F3"  
    android:layout_below="@+id/txt1"  
    android:gravity="center"  
    android:text="TextView" />
```



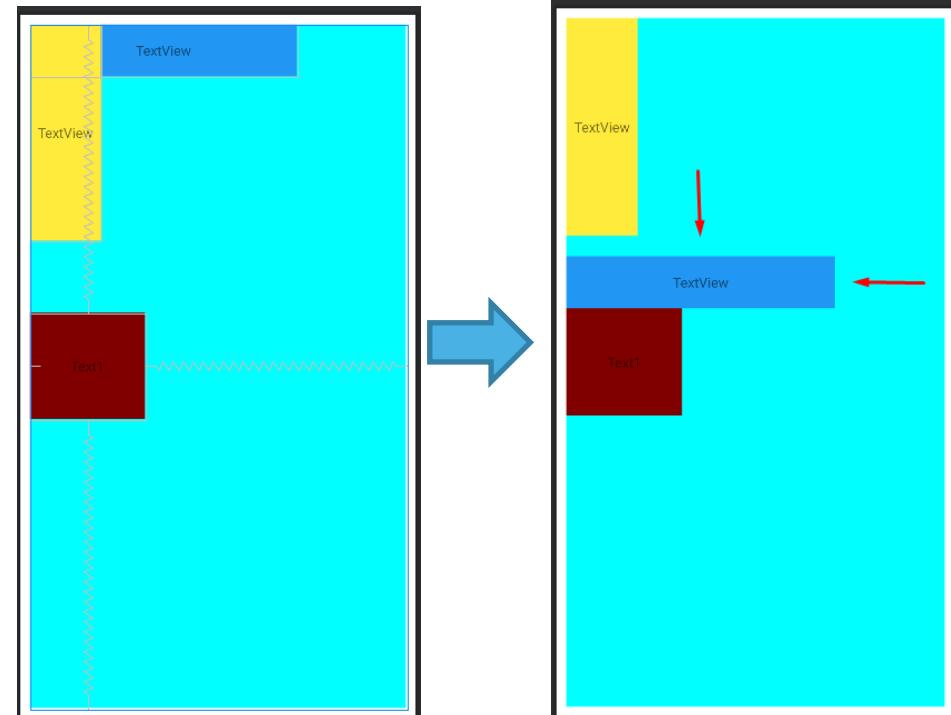
***Layout\_below = bên dưới***



## Relative Layout

### □ 3. Thiết lập vị trí giữa view con với view con : Trên, dưới, trái, phải

```
<TextView  
    android:id="@+id/txt1"  
    android:layout_width="120dp"  
    android:layout_height="111dp"  
    android:layout_centerVertical="true"  
    android:background="#800000"  
    android:gravity="center"  
    android:text="Text1" />  
  
<TextView  
    android:id="@+id/txt2"  
    android:layout_width="278dp"  
    android:layout_height="54dp"  
    android:background="#2196F3"  
    android:layout_above="@+id/txt1"  
    android:gravity="center"  
    android:text="TextView" />
```



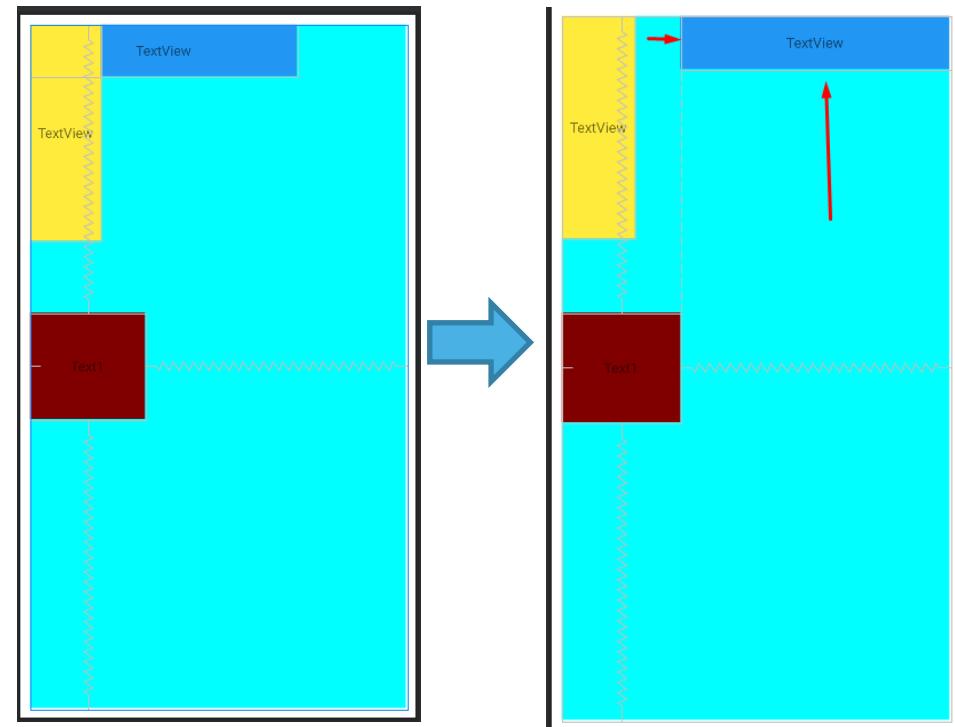
*Layout\_above = bên trên*



## Relative Layout

### □ 3.1 Thiết lập vị trí giữa view con với view con : Trên, dưới, trái, phải

```
<TextView  
    android:id="@+id/txt1"  
    android:layout_width="120dp"  
    android:layout_height="111dp"  
    android:layout_centerVertical="true"  
    android:background="#800000"  
    android:gravity="center"  
    android:text="Text1" />  
  
<TextView  
    android:id="@+id/txt2"  
    android:layout_width="278dp"  
    android:layout_height="54dp"  
    android:background="#2196F3"  
    android:layout_toRightOf="@+id/txt1"  
    android:gravity="center"  
    android:text="TextView" />
```

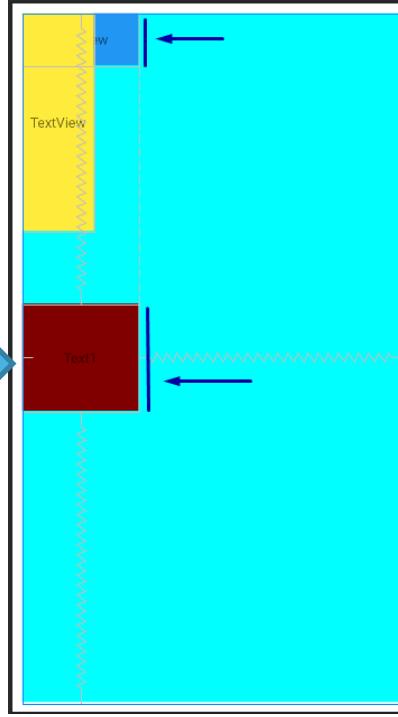
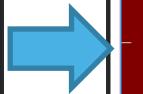
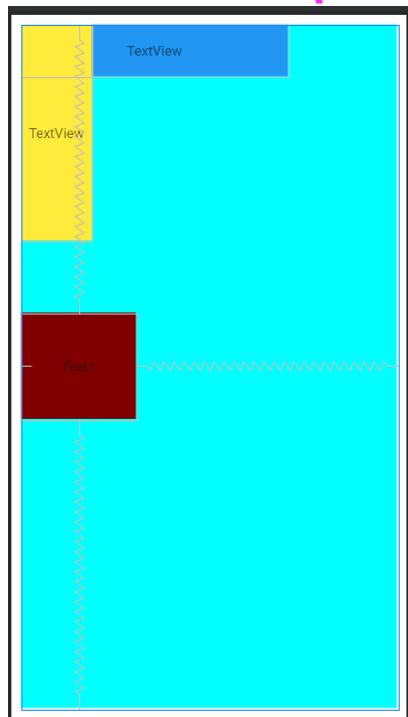


*Layout\_toRightOf= bên phải  
Layout\_toLeftOf= bên phải*

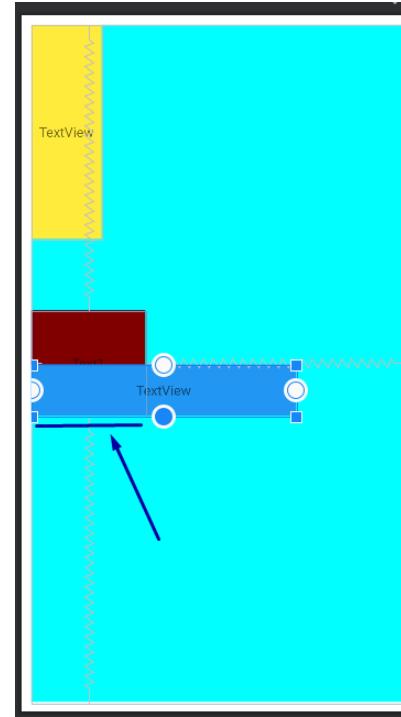


## Relative Layout

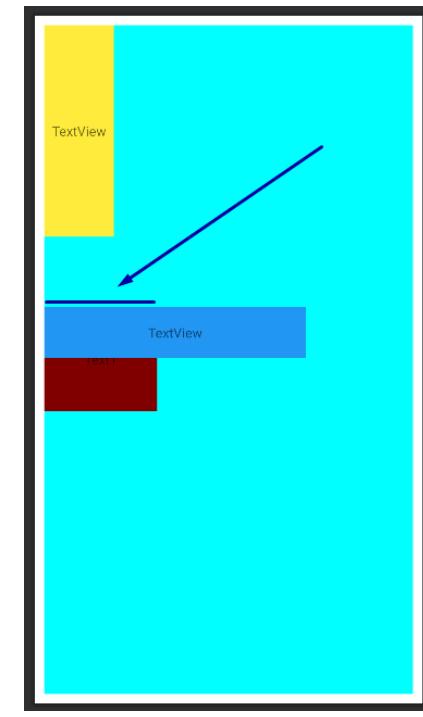
### ❑ 3.2 Thiết lập vị trí giữa view con với view con : Căn các cạnh view con align



```
layout_alignLeft="@+id/txt1"
```



```
layout_alignBottom="@+id/txt1"
```



```
layout_alignTop="@+id/txt1"
```

```
layout_alignLeft="@+id/txt1"
```



## Tổng kết Relative Layout

```
android:layout_alignParentBottom  
android:layout_alignParentEnd  
android:layout_alignParentLeft  
android:layout_alignParentRight  
android:layout_alignParentStart  
android:layout_alignParentTop
```

```
android:layout_centerInParent  
android:layout_centerHorizontal  
android:layout_centerVertical
```

Press Ctrl+Space to view tags from other namespaces [Next Tip](#)

**Căn view con so với view cha**

```
android:layout_alignLeft="@+id/txt2"  
android:layout_alignRight="@+id/txt2"  
android:layout_alignTop="@+id/txt2"  
android:layout_alignBottom="@+id/txt2"
```

```
android:layout_above="@+id/txt2"  
android:layout_below="@+id/txt2"  
android:layout_toLeftOf="@+id/txt2"  
android:layout_toRightOf="@+id/txt2"
```

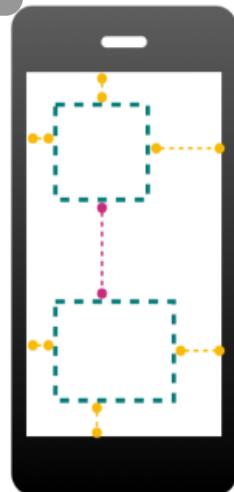
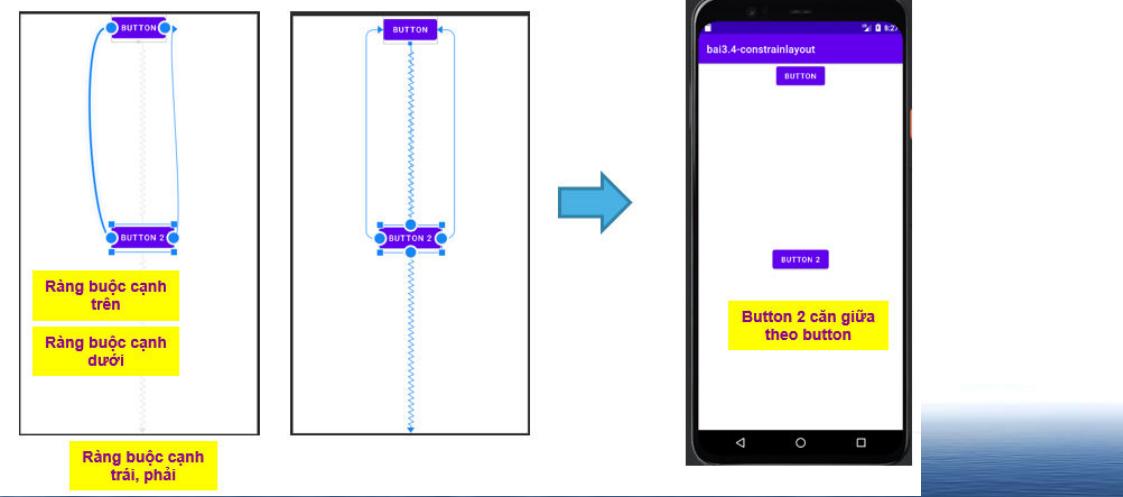
**Căn view con với view con khác**





## Android app with kotlin Bài 3.4

### Constrain Layout



Android  
ConstraintLayout

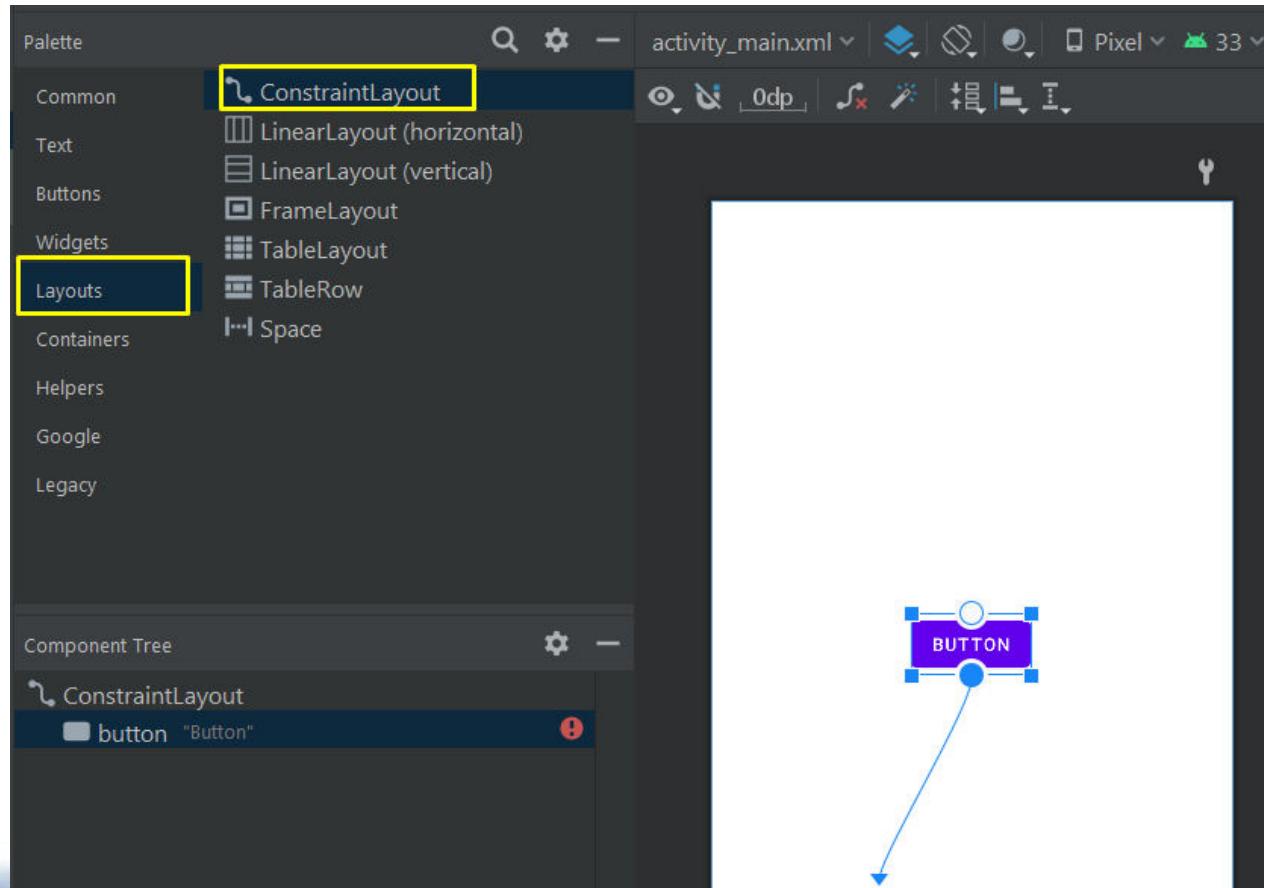


## Constrain Layout

### □ 1. Ràng buộc view với ViewGroup

ConstrainLayout là một *ViewGroup* làm việc dựa vào ràng buộc giữa các *View*:

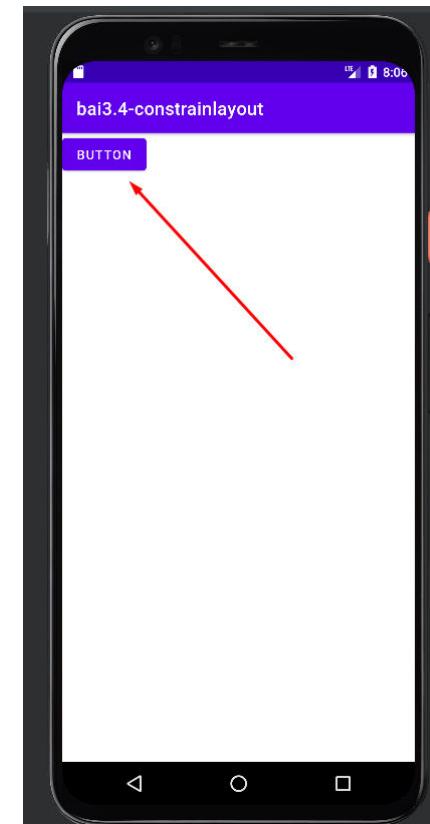
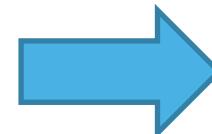
Ví dụ : *Button* phía dưới có 4 vòng tròn tại 4 cạnh, ta có thể click chuột để thiết lập ràng buộc



## Constrain Layout

Mỗi view phải có ít nhất 1 ràng buộc **Horizontal** và 1 ràng buộc **Vertical**.

Nếu ràng buộc này không được thêm vào thì view sẽ được đặt tại điểm gốc zero

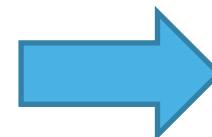
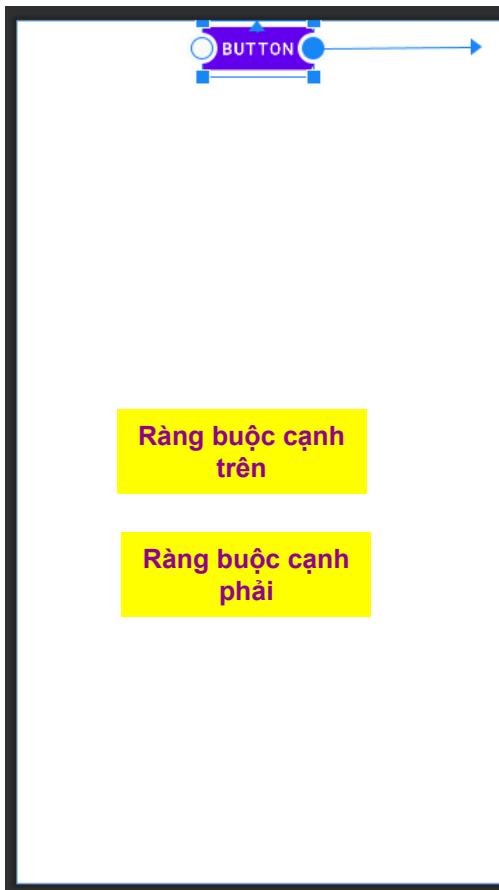


```
app:layout_constraintTop_toTopOf="parent"
```



## Constrain Layout

Sau khi ràng buộc thêm 1 cạnh chiều ngang :

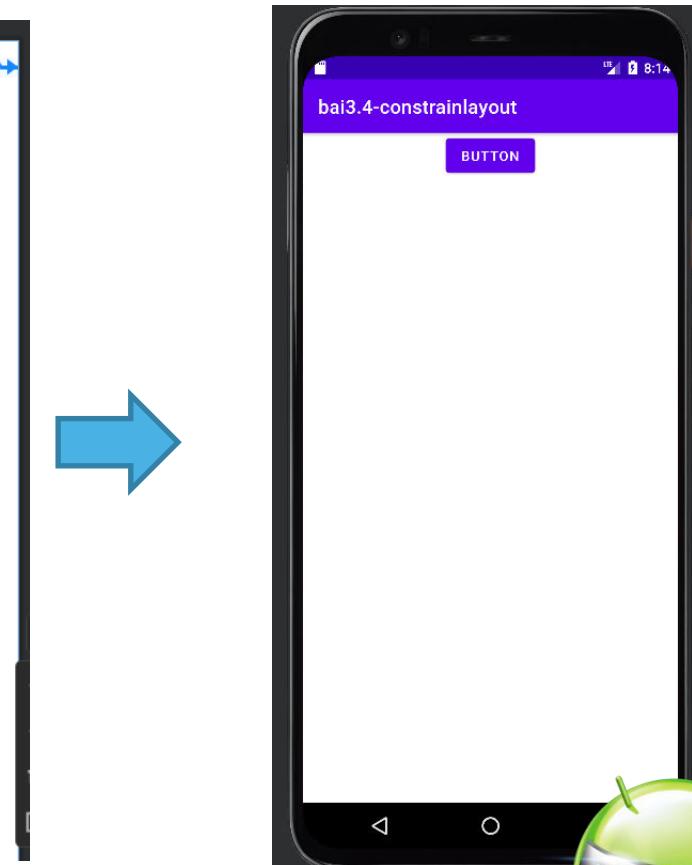
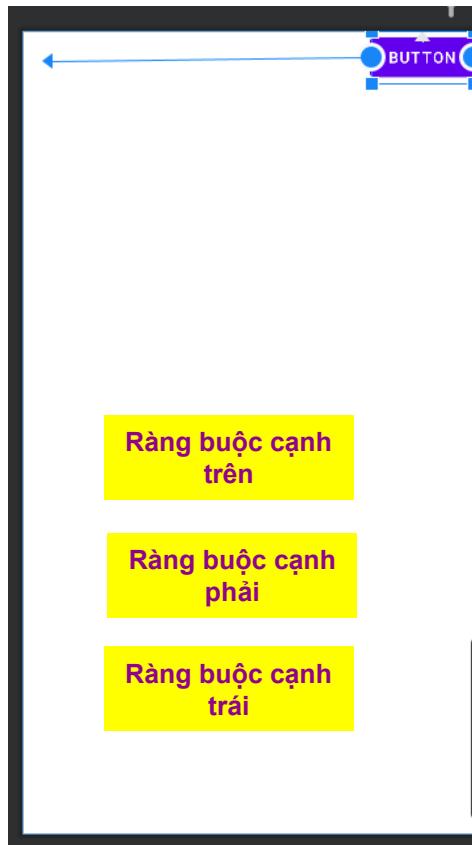


```
app:layout_constraintEnd_toEndOf="parent"
```



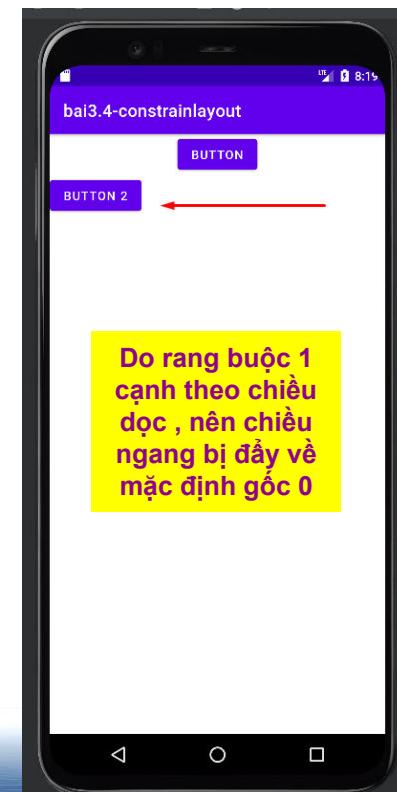
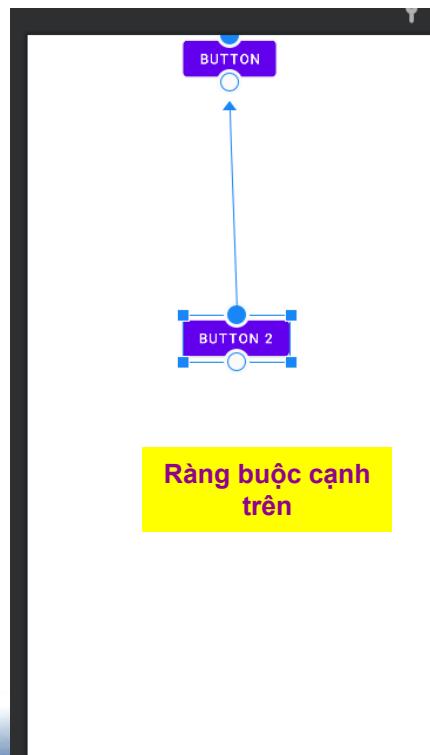
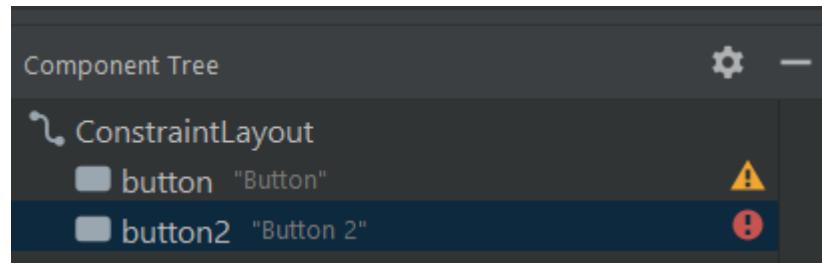
## Constrain Layout

Ràng buộc thêm cạnh trái :



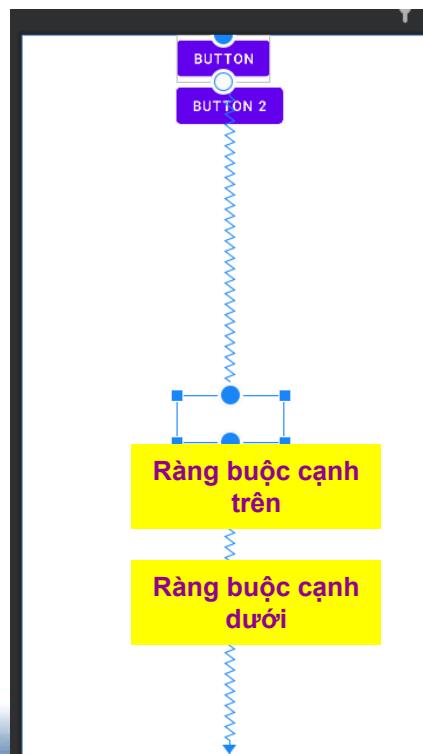
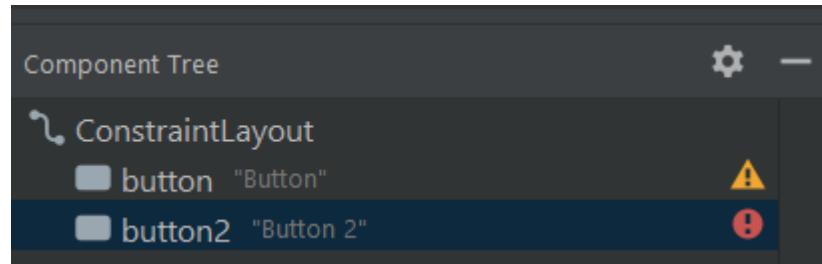
## Constrain Layout

### ❑ 2. Ràng buộc View với View

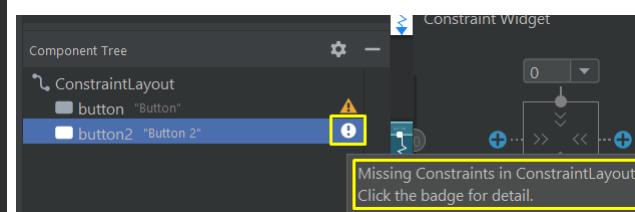
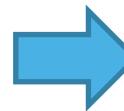


## Constrain Layout

### □ 2. Ràng buộc View với View

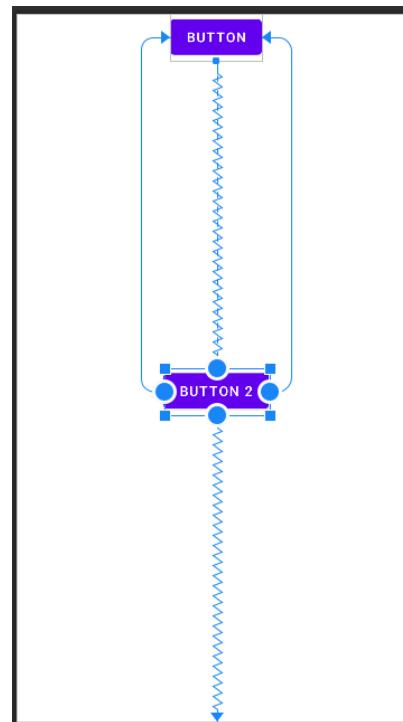
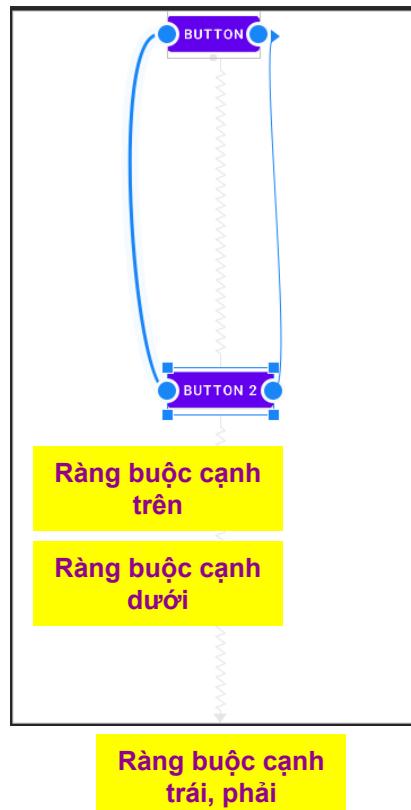


Không có ràng buộc chiều ngang nên vẫn bị đẩy về gốc 0



## Constrain Layout

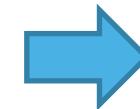
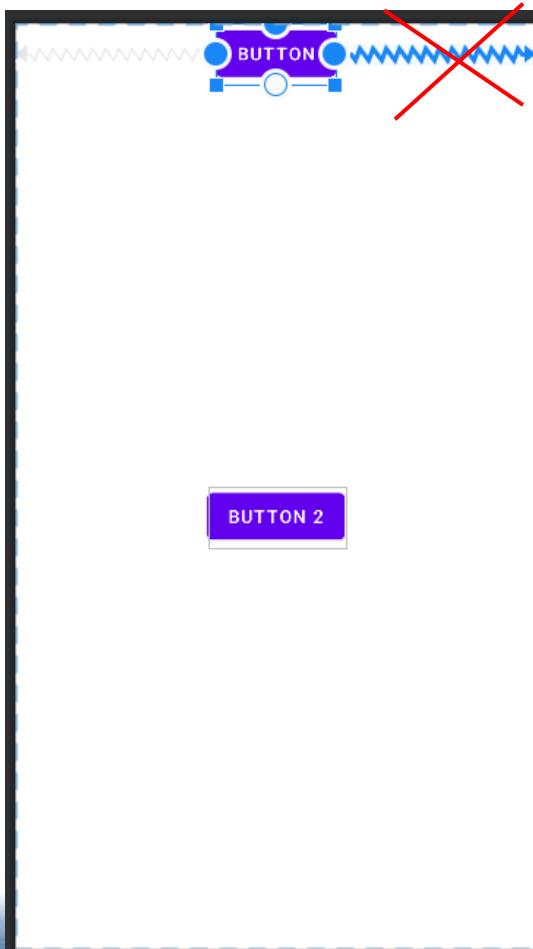
### □ 2. Ràng buộc View với View



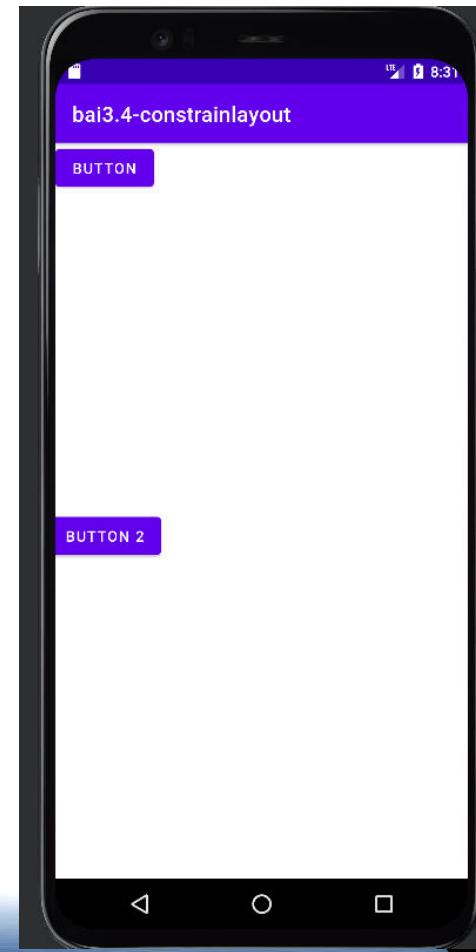
## Constrain Layout

### ❑ 2. Ràng buộc View với View

Nếu Delete ràng buộc phải của  
Button

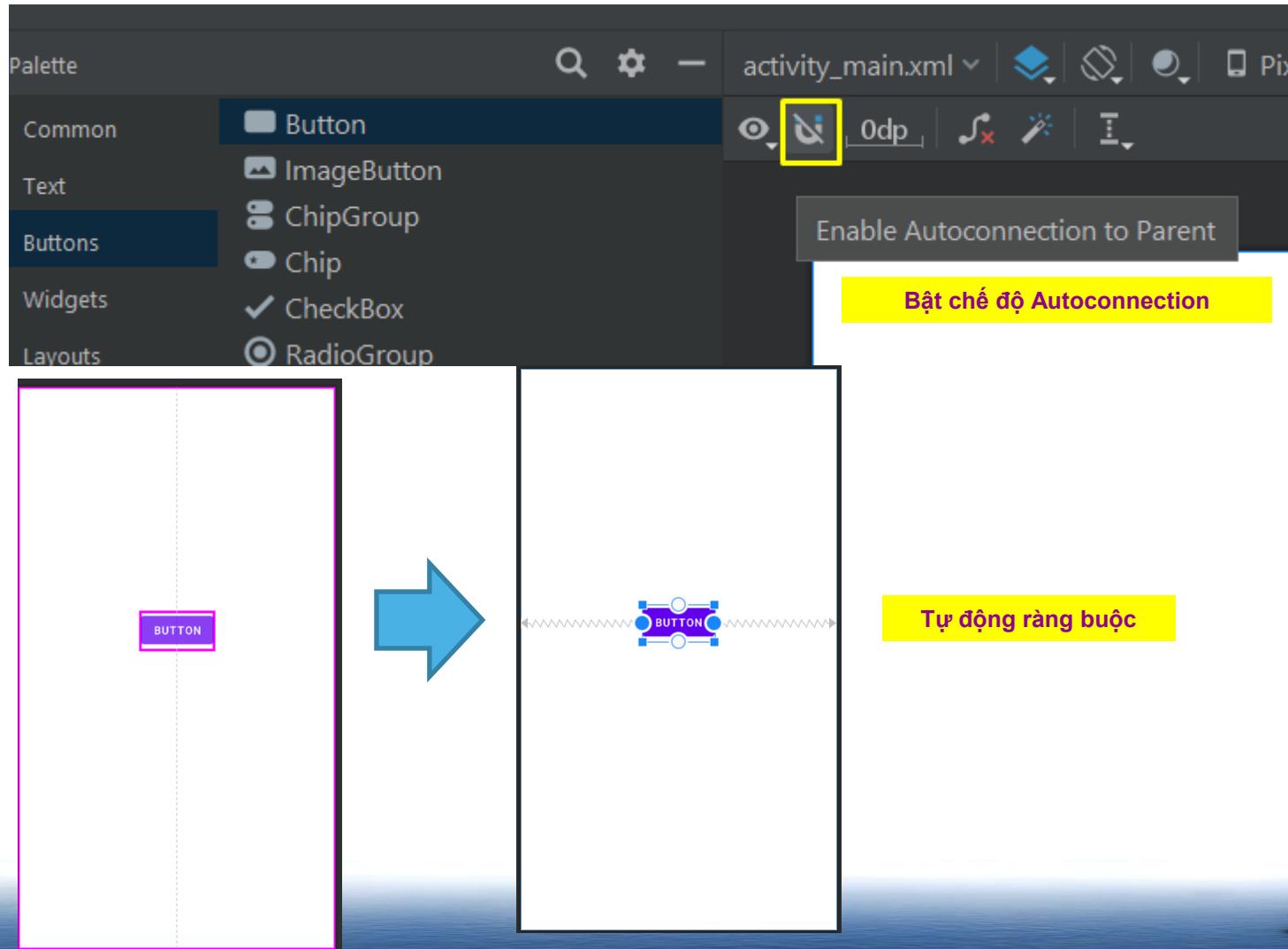


Button 2 tự động vẫn căn theo  
button



## Constrain Layout

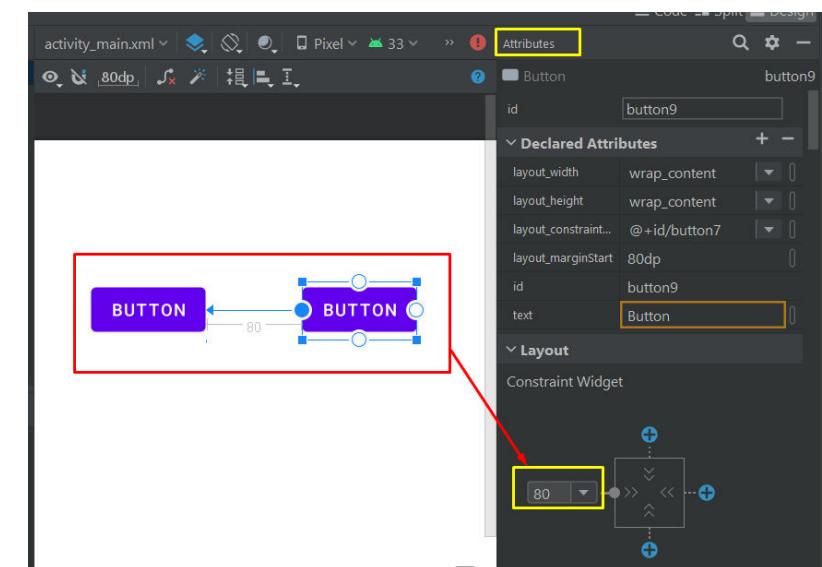
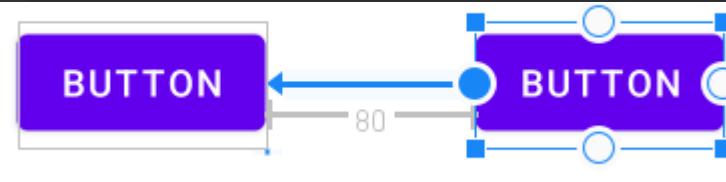
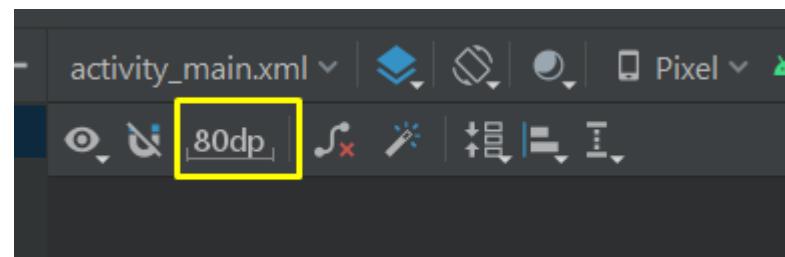
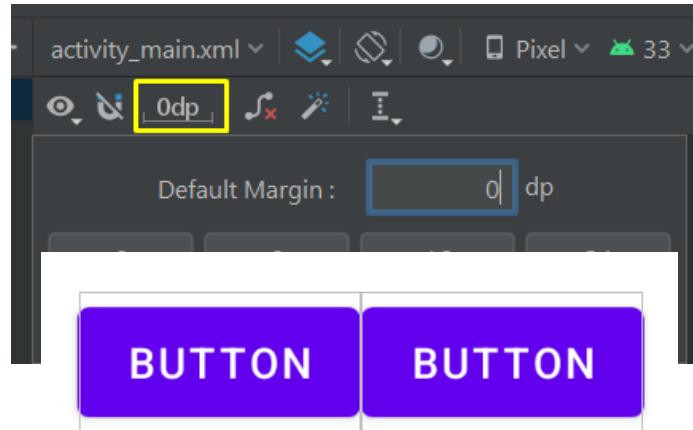
### ❑ 3. Auto ràng buộc khi kéo thả view



## Constrain Layout

### □ 4. Default Margin

*Default Margin : Quy định xem khoảng cách mặc định khi ràng buộc các View là bao nhiêu*

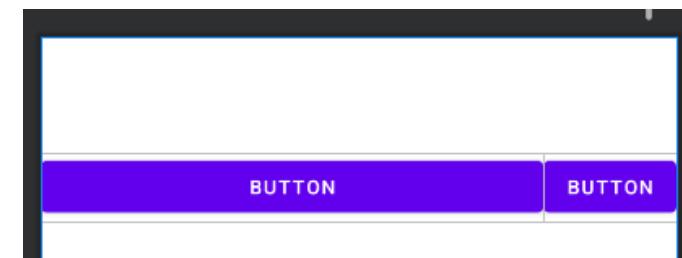
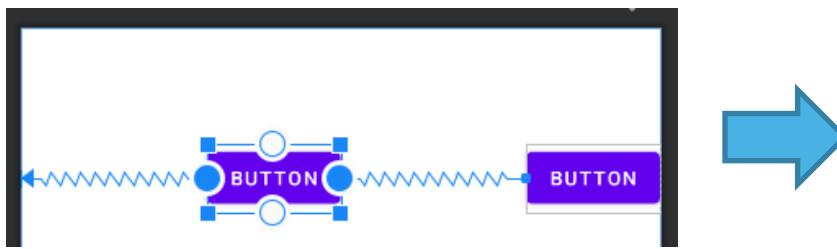
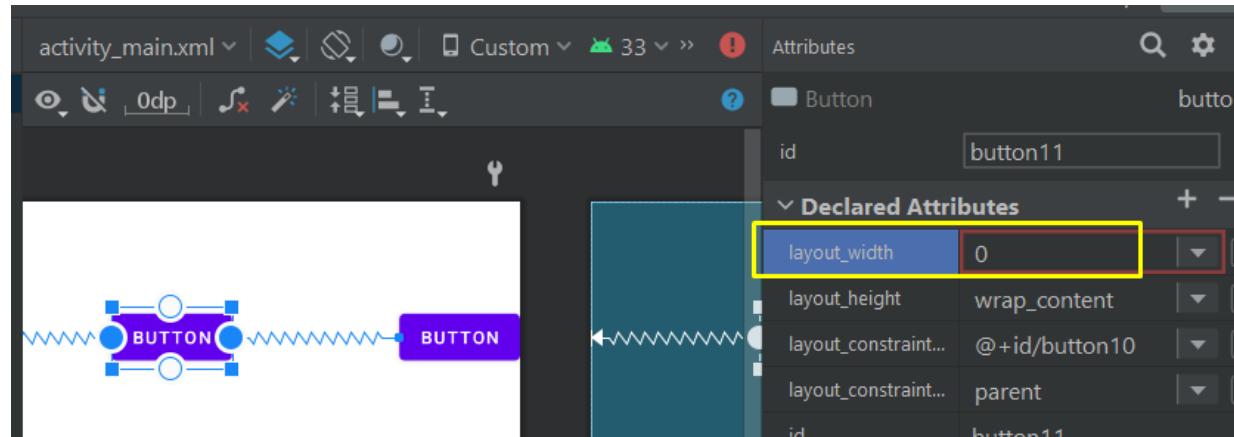


Có thể chỉnh ràng buộc khoảng cách tại Attributes



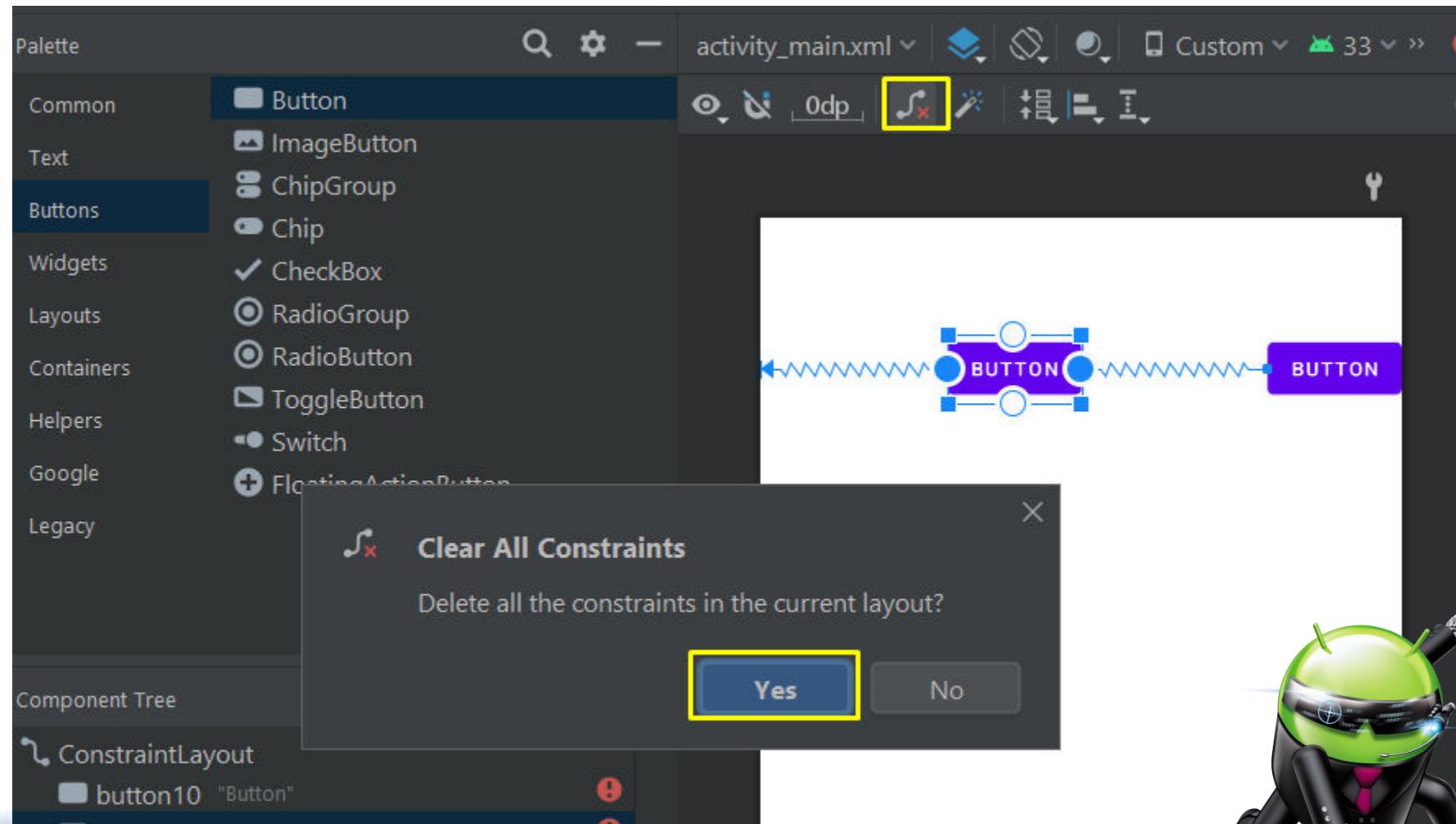
## Constrain Layout

### □ 5. Lắp đầy vùng ràng buộc



## Constrain Layout

### □ 6. Xoá tất cả các ràng buộc



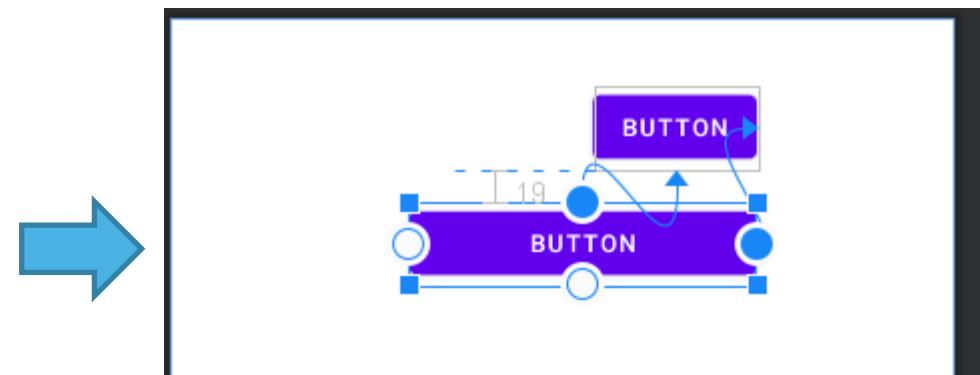
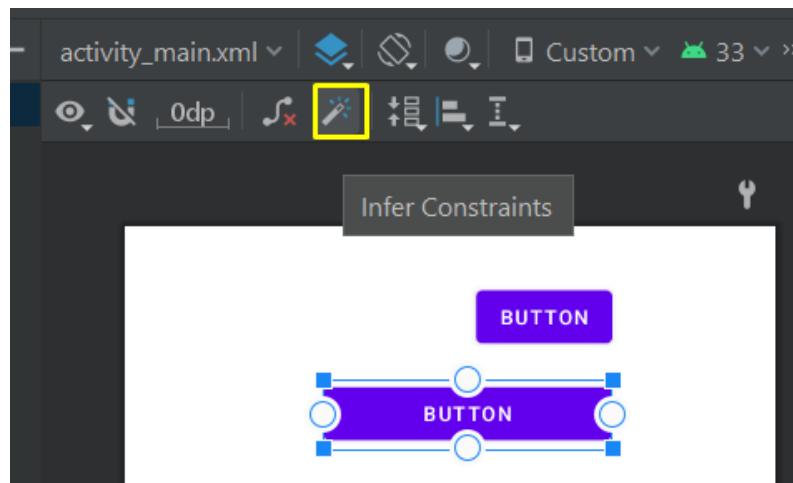
The screenshot shows the Android Studio interface with the following details:

- Palette:** The "Buttons" category is selected in the palette.
- Toolbar:** The "Clear All Constraints" icon (a red circle with a white 'X') is highlighted with a yellow box.
- Preview:** Two purple buttons labeled "BUTTON" are shown with blue constraint lines connecting them horizontally.
- Confirmation Dialog:** A modal dialog titled "Clear All Constraints" asks "Delete all the constraints in the current layout?". The "Yes" button is highlighted with a yellow box.
- Component Tree:** Shows a single node: "button10 "Button"".
- Bottom Bar:** Shows an error icon (red exclamation mark).



## Constrain Layout

### □ 7. Xếp vị trí và tự động ràng buộc





## Android App with Kotlin

### Bài 4

- 1 TextView
- 2 EditText
- 3 Button

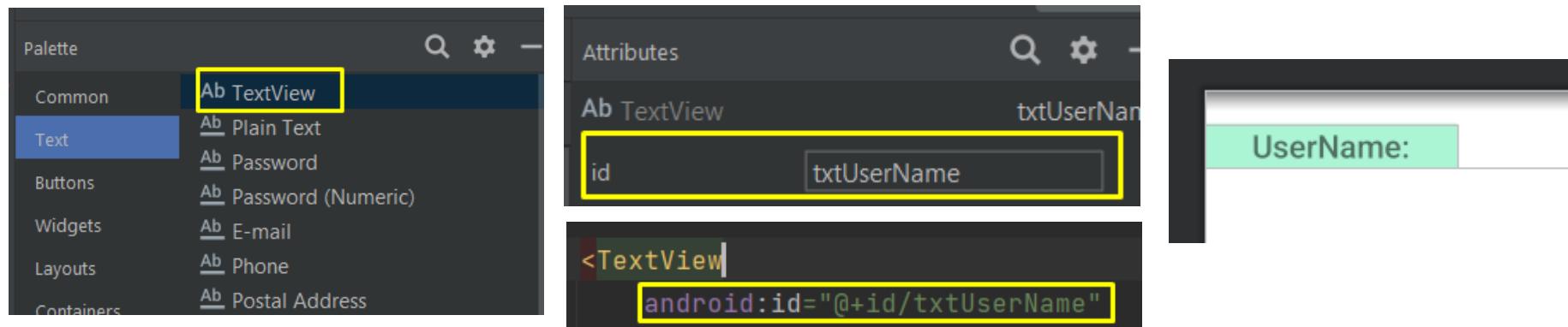


## 1

## TextView

## □ 1. TextView :

- ✓ Dùng để hiển thị văn bản. *Không cho phép ng dùng can thiệp sửa*
- ✓ id : bắt đầu bằng tiền tố **txt** Ví dụ : **txtUserName**



1

## TextView

 2 . Các thuộc tính (Attributes ) cơ bản TextView :

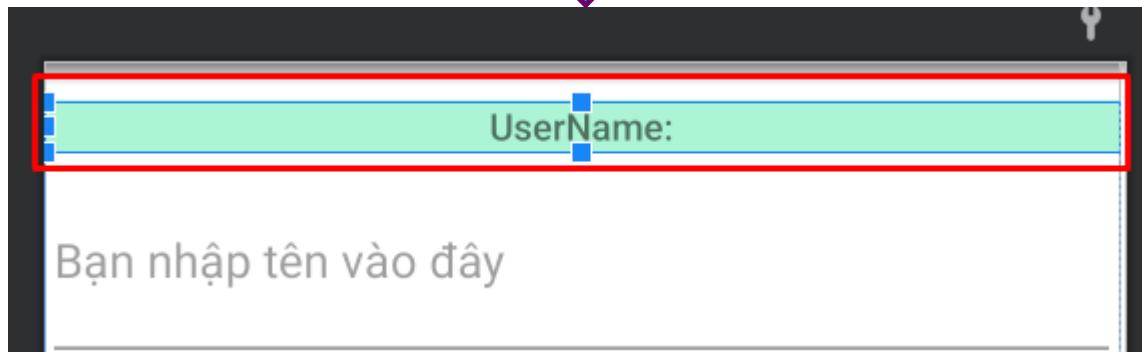
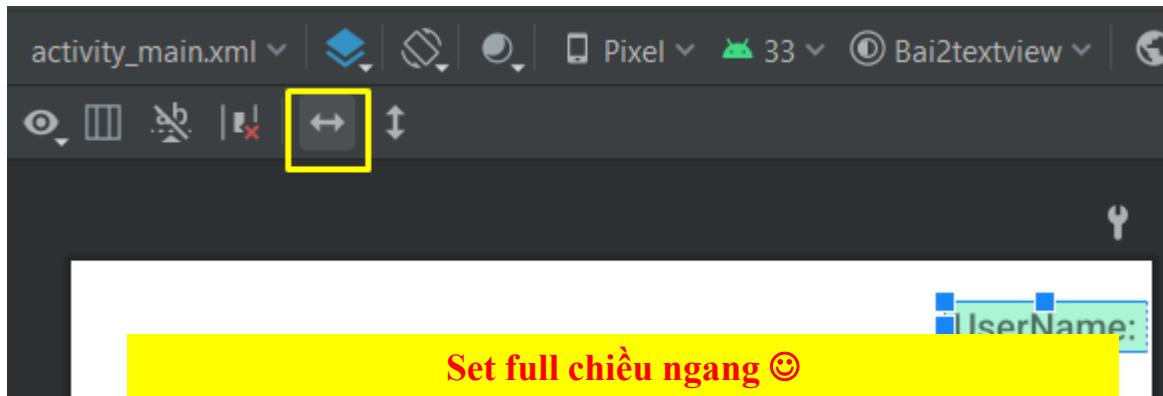
STT	Thuộc tính	Nội dung
1	id	txt_ sử dụng khi coding
2	text	Gán text hiển thị
3	background	Màu nền
4	gravity	Căn chữ trong vùng textview
5	layout_gravity	Căn textview trong view chứa nó
6	txtSize	Cỡ chữ



1

## TextView

## □ 2 . Các thuộc tính (Attributes ) cơ bản TextView :

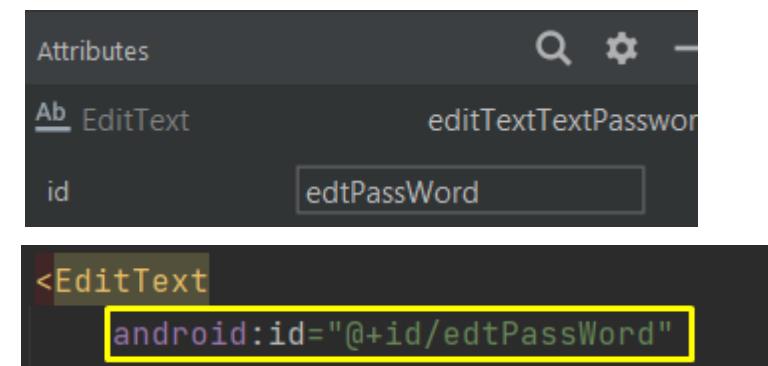
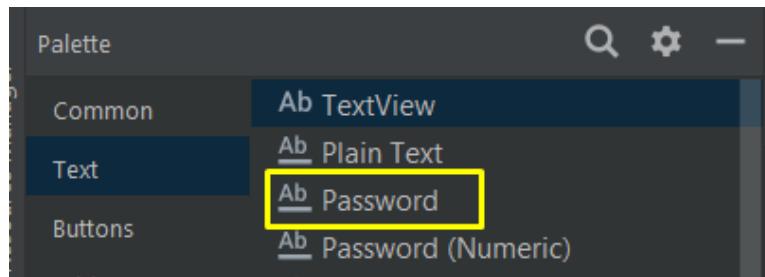


2

## EditText

## □ 1. EditText :

- ✓ Dùng để hiển thị văn bản . Cho phép người dùng nhập liệu
- ✓ id : bắt đầu bằng tiền tố **edt** Ví dụ : **edtPassWord**



Bạn nhập tên vào đây

Nhập Mật khẩu

Pass nhập vào đây nhé thím

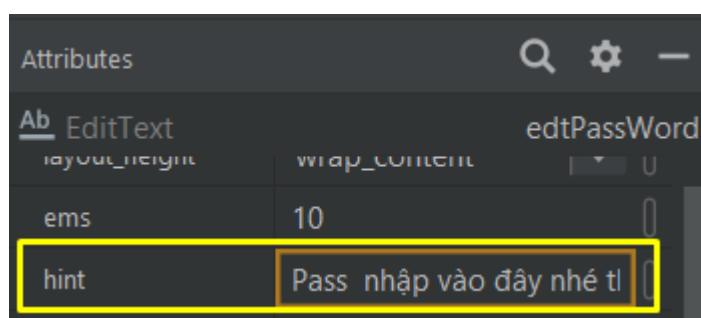


2

## EditText

### □ 2 . Các thuộc tính (Attributes ) cơ bản EditText :

STT	Thuộc tính	Nội dung
1	<b>id</b>	<b>edt_</b> sử dụng khi coding
2	<b>hint</b>	<b>Hiển thị gợi ý</b> ( xuất hiện khi chưa có dữ liệu nhập)



```
android:hint="Pass nhập vào đây nhé thím"
```

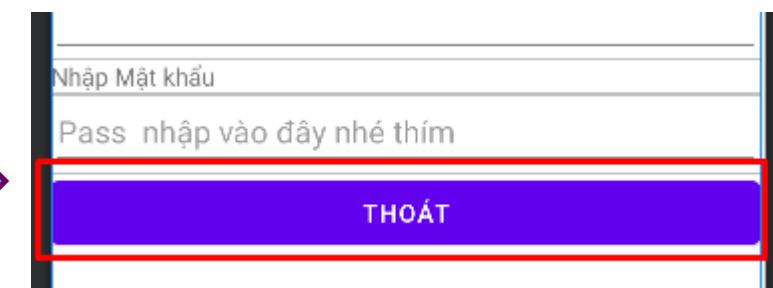
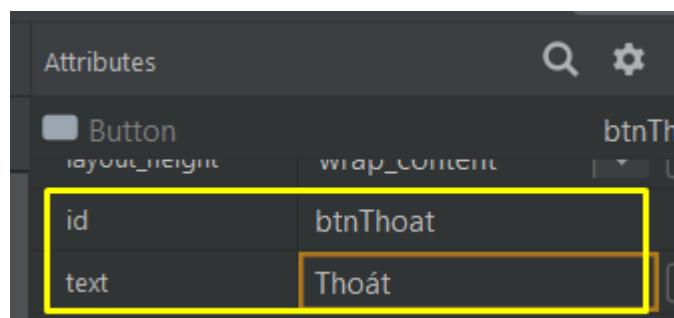


3

## Button

Là nút bấm . Cho phép người dùng click tương tác để thực hiện 1 công việc nào đó

✓ id : bắt đầu bằng tiền tố **btn** Ví dụ : **btnThoat**



```
<Button  
    android:id="@+id/btnThoat"
```





## Bài 5.1



# Xử lý sự kiện trên view

Nhập số a:

Mời nhập a vào đây

Nhập số b:

Mời nhập b vào đây

1

findViewById

2

Ex : code app tính toán

Kết Quả

CỘNG

RESET



1

## findViewById

## □ 1. Thiết kế giao diện sau :

Nhập số a:  
Mời nhập a vào đây

Nhập số b:  
Mời nhập b vào đây

edtNhapA

edtNhapB

edtKQ

Kết Quả

CỘNG

RESET

btnCong

btnReSet

```
android:layout_marginLeft="20dp"  
android:layout_marginRight="20dp"
```

Cách lè phải và trái button



1

## findViewById

## □ 2. Khai báo biến để link đến các view :

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        //mọi code phải đặt sau setContentView  
        val edtNhaphA = findViewById<EditText>(R.id.edtNhaphA)  
        val edtNhaphB = findViewById<EditText>(R.id.edtNhaphB)  
        val edtKQ = findViewById<EditText>(R.id.edtKQ)  
        val btnCong = findViewById<Button>(R.id.btnCong)  
        val btnReset = findViewById<Button>(R.id.btnReset)  
        //set text cho textView  
        edtNhaphA.setText("ô là la")
```

```
//set text cho textView  
edtNhaphA.setText("ô là la")
```



2

## Ex : code app tính toán

## □ 3. Code button cộng :

```
btnCong.setOnClickListener{ it: View!  
    val A = edtNhapA.text.toString().toInt()  
    val B = edtNhapB.text.toString().toInt()  
    val kq = A+B  
  
    //xem log  
    Log.d("tuhoc", "tổng là $kq")  
    //gán kết quả vào ô kết quả edtKQ  
    edtKQ.setText(kq.toString())  
}
```

Xử lý khi click vào button



2

## Ex : code app tính toán

## □ 4. Code button Reset :

```
//code button reset  
btnReset.setOnClickListener { it: View!  
    edtNhaphA.setText("")  
    edtNhaphB.setText("")  
    edtKQ.setText("")  
}
```

Nhập số a:  
666

Nhập số b:  
55

721

**CỘNG**      **RESET**



Nhập số a:  
Mời nhập a vào đây

Nhập số b:  
Mời nhập b vào đây

721

**CỘNG**      **RESET**

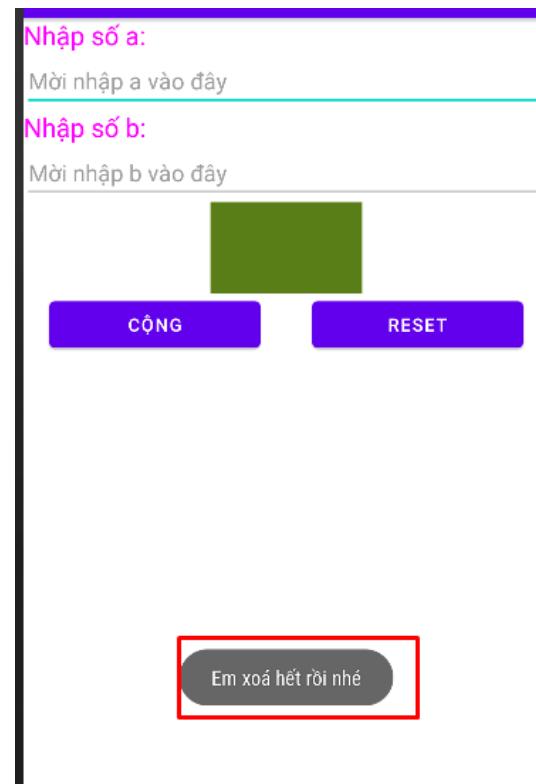


2

## Ex : code app tính toán

## □ 5. Thông báo nhanh trên màn hình :

```
//thông báo nhanh trên màn hình  
Toast.makeText( context: this, text: "Em xoá hết rồi nhé", Toast.LENGTH_SHORT).show()
```





## Bài 5.2

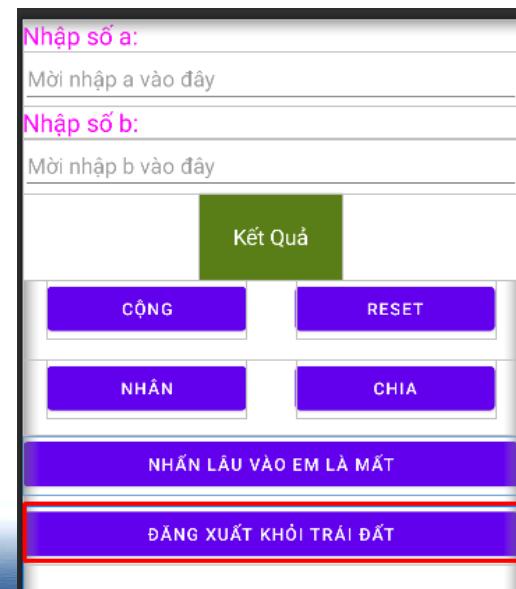
# Xử lý sự kiện trên view

1

Chia sẻ sự kiện

2

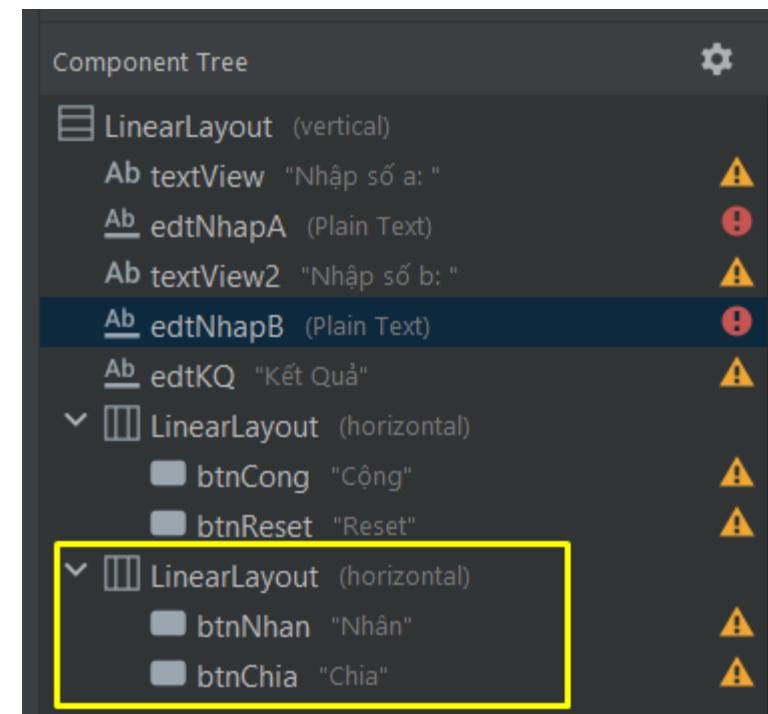
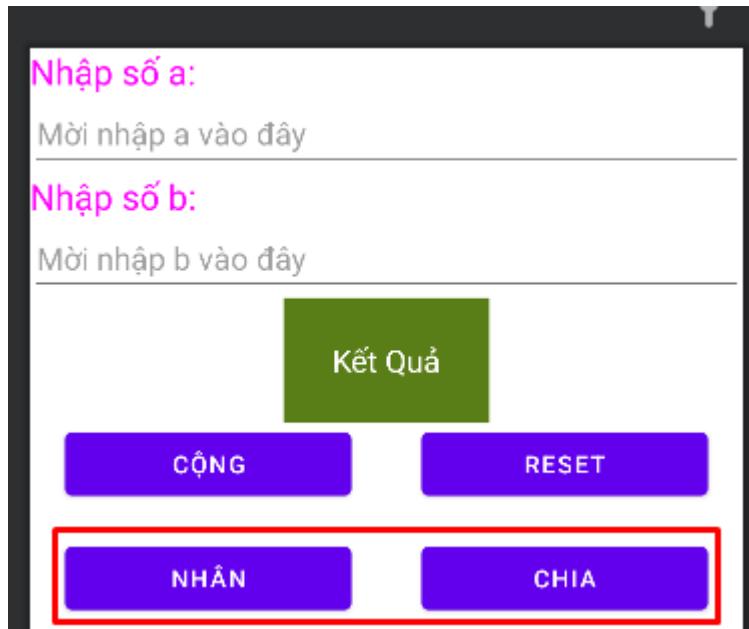
setOnLongClickListener



1

## Chia sẻ sự kiện

✓ Add thêm 2 button nhân và chia:



## 1

## Chia sẻ sự kiện

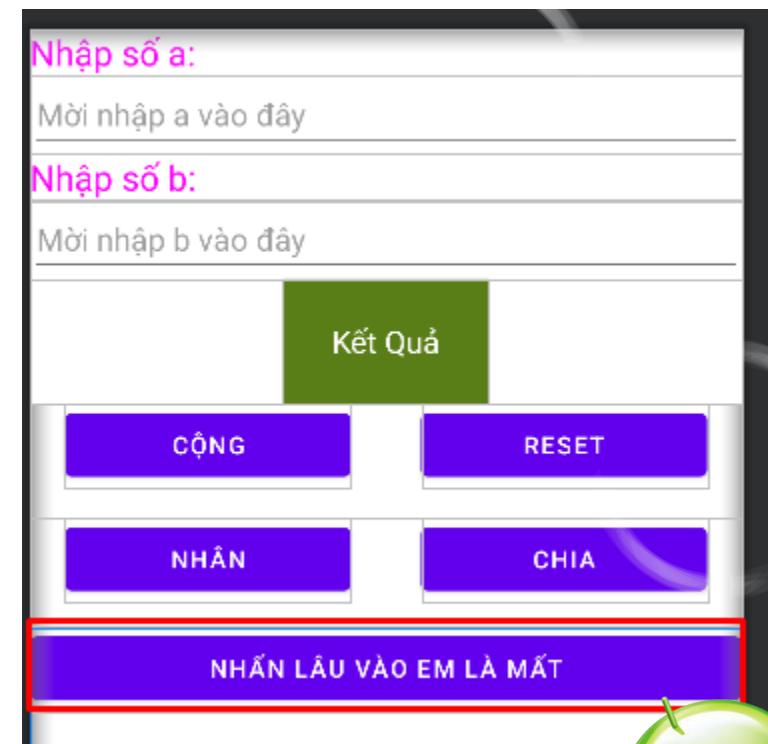
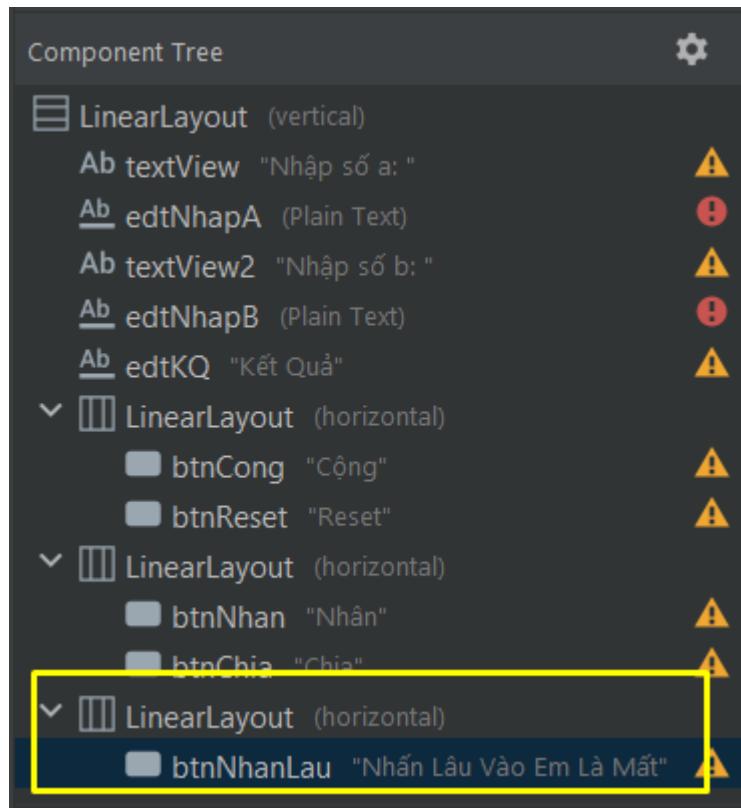
```
//4. Tạo sự kiện chia sẻ: 1 biến có thể chia sẻ sự kiện cho >= 2 control
var suKienChiaSe: View.OnClickListener? = null
suKienChiaSe = View.OnClickListener { it:View!
    if (it==btnNhan)
    {
        var A = edtNhapA.text.toString().toInt()
        var B = edtNhapB.text.toString().toInt()
        var kq = A*B
        edtKQ.setText(kq.toString())
    }
    else if (it==btnChia)
    {
        var A = edtNhapA.text.toString().toDouble()
        var B = edtNhapB.text.toString().toDouble()
        var kq = A/B
        edtKQ.setText(kq.toString())
    }
}
btnNhan.setOnClickListener(suKienChiaSe)
btnChia.setOnClickListener(suKienChiaSe)
```



2

## setOnLongClickListener

- ✓ Là sự kiện dùng khi người dùng thao tác nhấn và giữ



## 2

## setOnLongClickListener

```
//5.setOnLongClickListener : nhấn lâu trên control
var suKienNhanLau: View.OnLongClickListener? = null
suKienNhanLau =View.OnLongClickListener{ it: View!
    if (it==btnNhanLau)
    {
        Toast.makeText( context: this,  text: "Long click detected", Toast.LENGTH_SHORT).show()
        btnNhanLau.visibility = View.GONE //ẩn btn
    }
    else if (it==btnNhan)
    {
        Toast.makeText( context: this,  text: "Long click detected", Toast.LENGTH_SHORT).show()
        btnNhan.visibility = View.GONE //ẩn btn
    }
    true    // chú ý thêm true ở cuối ^OnLongClickListener
}
btnNhanLau.setOnLongClickListener(suKienNhanLau)
btnNhan.setOnLongClickListener(suKienNhanLau)
```

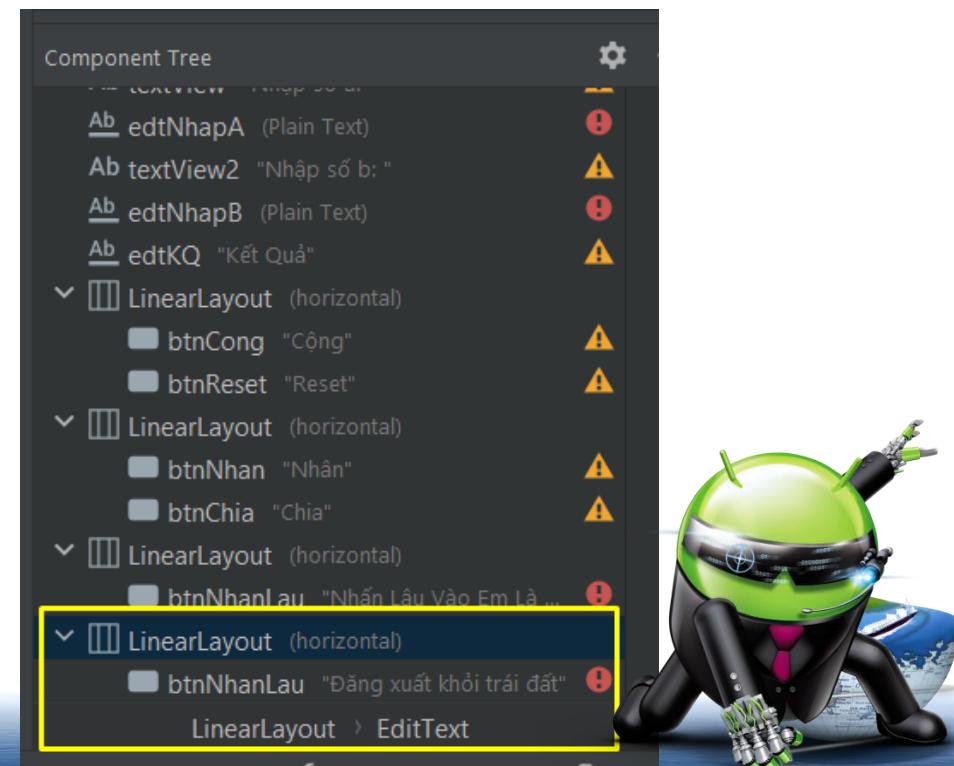
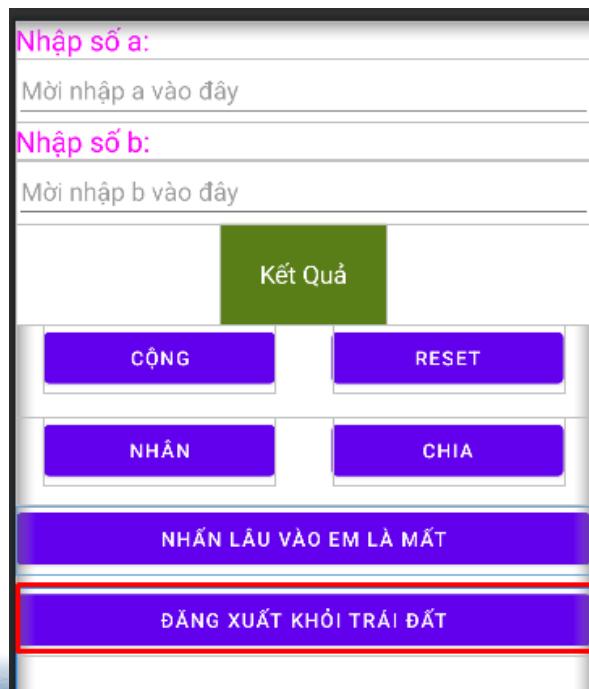


3

## Thoát chương trình

- ✓ dùng `finish()` để thoát

```
btnThoat.setOnClickListener{ it: View!  
    Toast.makeText(context: this, text: "Gà Lại Lập Trình!" +  
        "\nCảm ơn bạn đã sử dụng phần mềm", Toast.LENGTH_LONG).show()  
    finish() // thoát chương trình  
}
```





## Bài 6.1

# Hướng dẫn sử dụng viewbinding kotlin

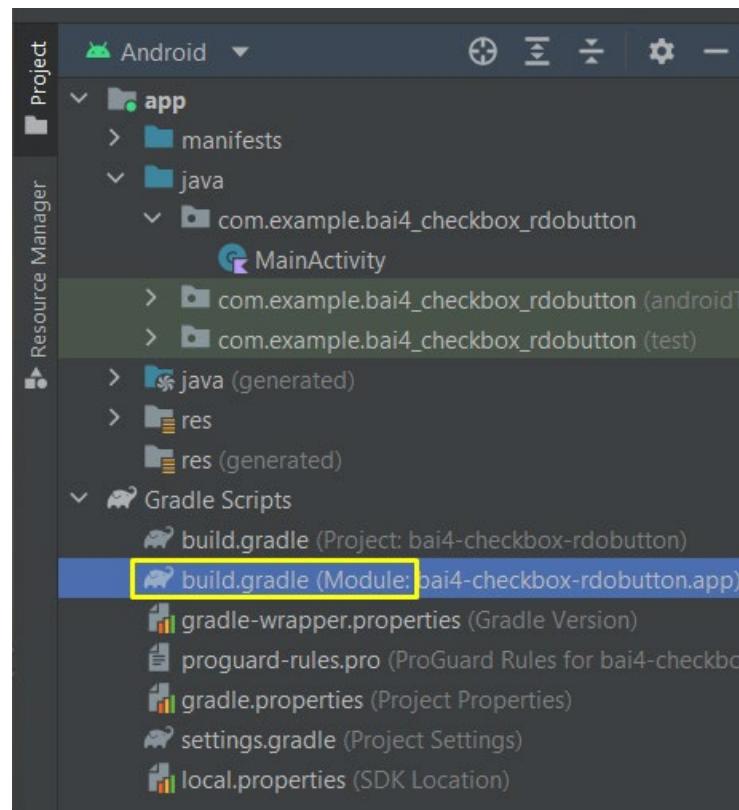
The 'kotlin-android-extensions' Gradle plugin is no longer supported.  
Please use this migration guide (<https://goo.gl/kotlin-android-extensions-deprecation>) to start working with View Binding (<https://developer.android.com/topic/libraries/view-binding>) and the 'kotlin-parcelize' plugin.

## 1

## Tip Bỏ qua khai báo biến

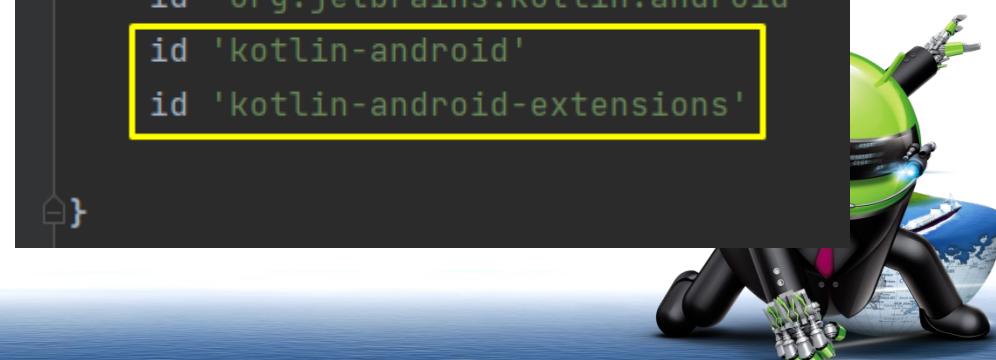
## □ 1. Add plugins :

- ✓ Dùng để bỏ qua bước khai báo biến - kết nối với các view khi coding



```
val edtNhaphA = findViewById<EditText>(R.id.edtNhaphA)
val edtNhaphB = findViewById<EditText>(R.id.edtNhaphB)
val edtKQ = findViewById<EditText>(R.id.edtKQ)
val btnCong = findViewById<Button>(R.id.btnCong)
val btnReset = findViewById<Button>(R.id.btnReSet)
val btnNhan = findViewById<Button>(R.id.btnNhan)
val btnChia = findViewById<Button>(R.id.btnChia)
val btnNhanLau = findViewById<Button>(R.id.btnNhanLau)
val btnThoat = findViewById<Button>(R.id.btnThoat)
```

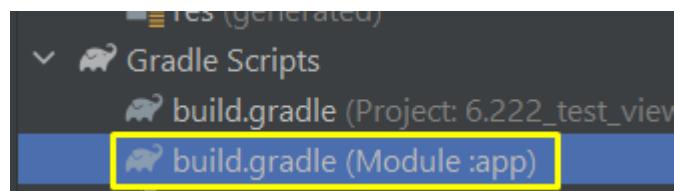
```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
    id 'kotlin-android'
    id 'kotlin-android-extensions'
```



## Viewbinding

### ❑ 2. Viewbinding :

- ✓ *Dùng để bỏ qua bước khai báo biến – phiên bản android mới*



```
kotlinOptions {
    jvmTarget = '1.8' ←
}
// thêm dòng này để sử dụng viewbinding
buildFeatures {
    viewBinding = true
}
```

```
private lateinit var binding: ActivityMainBinding // Tên class sẽ được sinh ra từ file layout
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        //setContentView(R.layout.activity_main)

        // Khởi tạo ViewBinding
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.btnTest.text = "aaaaaaaaaaaaaaaaaa"
```

MainActivity.kt



## Android app with kotlin Bài 6

1

Tip Bỏ qua khai báo biến

2

Checkbox

3

RadioButton

Sở thích của thím:

Nghe Nhạc

Xem Phim

Chơi Thể Thao

Shopping

Du Lịch

Giới tính

Nam

Nữ

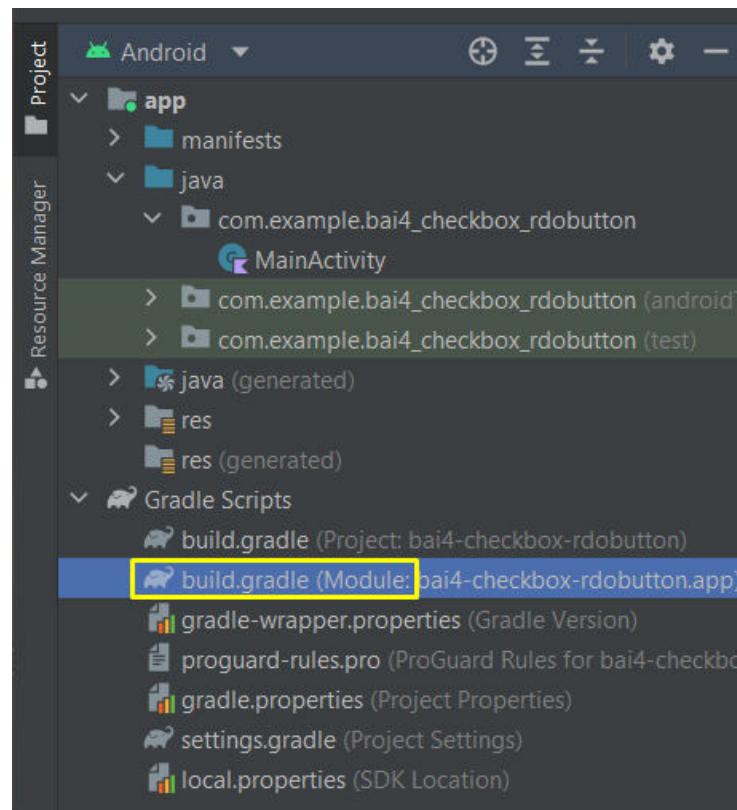


## 1

## Tip Bỏ qua khai báo biến

## □ 1. Add plugins :

- ✓ Dùng để bỏ qua bước khai báo biến - kết nối với các view khi coding



```
val edtNhaphA = findViewById<EditText>(R.id.edtNhaphA)
val edtNhaphB = findViewById<EditText>(R.id.edtNhaphB)
val edtKQ = findViewById<EditText>(R.id.edtKQ)
val btnCong = findViewById<Button>(R.id.btnCong)
val btnReset = findViewById<Button>(R.id.btnReSet)
val btnNhan = findViewById<Button>(R.id.btnNhan)
val btnChia = findViewById<Button>(R.id.btnChia)
val btnNhanLau = findViewById<Button>(R.id.btnNhanLau)
val btnThoat = findViewById<Button>(R.id.btnThoat)
```

```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
    id 'kotlin-android'
    id 'kotlin-android-extensions'
```

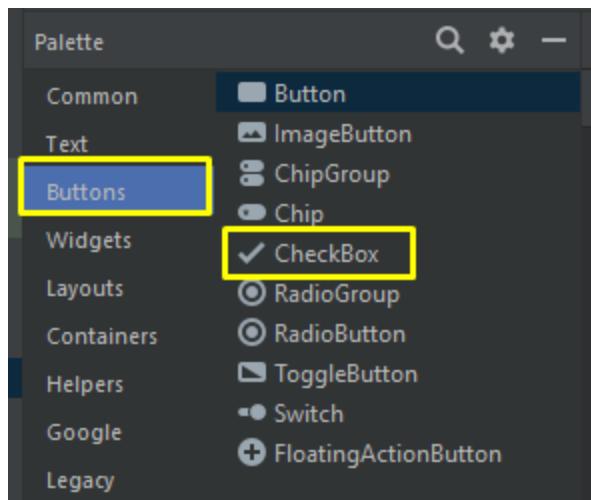


## 2

## Checkbox

 2. Checkbox :

- ✓ Dùng để cho người dùng tích chọn. **Có thể chọn nhiều lựa chọn 1 lúc**
- ✓ id : bắt đầu bằng tiền tố **chk** Ví dụ : **chkSoThich**
- ✓ **chkSoThich.isChecked = true** tức là có lựa chọn được chọn



Sở thích của thím:

- Nghe Nhạc
- Xem Phim
- Chơi Thể Thao
- Shopping
- Du Lịch

**CHỌN SỞ THÍCH**

Sở Thích Của Bạn Là:

Sở thích của bạn sẽ hiển thị ở đây:

```
<CheckBox  
    android:id="@+id/chkShopping"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textColor="#f20"  
    android:textStyle="bold"  
    android:textSize="17dp"  
    android:text="Shopping" />
```



## 2

## Checkbox

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        //add hàm events(), chú ý tất cả viết sau setContentView  
        addEvents()  
    }  
}
```

```
//xử lý sự kiện khi click vào button  
private fun addEvents() {  
    //xử lý sự kiện khi click vào button  
    btnSoThich.setOnClickListener{ it: View!  
        xuLyChonSoThich() //hàm xử lý khi tích chọn  
        //Toast.makeText(this,"Bạn đã click vào tớ",Toast.LENGTH_SHORT).show()  
    }  
}
```



## 2

## Checkbox

```
//hàm xử lý khi tích chọn
private fun xuLyChonSoThich() {
    var s:String =""
    if (chkNgheNhac.isChecked) // <=> chkNgheNhac.isChecked=true
        s+=chkNgheNhac.text.toString()+"\n"
    if (chkXemPhim.isChecked)
        s+=chkXemPhim.text.toString()+"\n"
    if (chkChoiTheThao.isChecked)
        s+=chkChoiTheThao.text.toString()+"\n"
    if (chkShopping.isChecked)
        s+=chkShopping.text.toString()+"\n"
    if (chkDuLich.isChecked)
        s+=chkDuLich.text.toString()+"\n"
    //gán vào ô edtSoThichBanChon
    edtSoThichBanChon.setText(s)
}
```

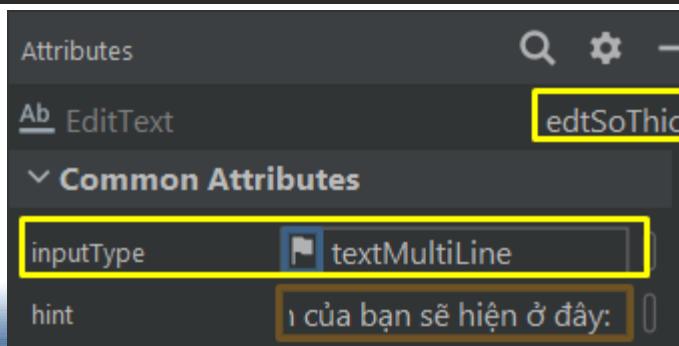
Sở thích của thím:

- Nghe Nhạc
- Xem Phim
- Chơi Thể Thao
- Shopping
- Du Lịch

CHỌN SỞ THÍCH

Sở Thích Của Bạn Là:

Xem Phim  
Chơi Thể Thao



## 3

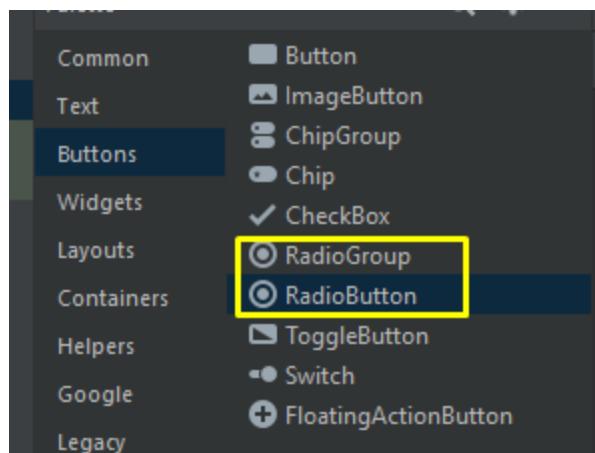
## RadioButton

 2. Radio Group , RadioButton :

Radio Group : Gom nhóm các radio button

RadioButton :

- ✓ Dùng để cho người dùng tích chọn. **Chỉ chọn được 1 lựa chọn**
- ✓ id : bắt đầu bằng tiền tố **rad** Ví dụ : **radGioiTinh**
- ✓ **radGioiTinh.isChecked = true** tức là có lựa chọn được chọn



Giới Tính:

Nam

Nữ

**XÁC NHẬN**

Giới Tính:

Nam

Nữ

**XÁC NHẬN**

Bạn chọn giới tính Nam



3

## RadioButton

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        //add hàm event  
        addEvent()  
    }  
}
```

```
private fun addEvent() {  
    xuLyXacNhan()  
}
```

```
private fun xuLyXacNhan() {  
    var s:String=""  
    if (radNam.isChecked)  
        s=radNam.text.toString()  
    else if (radNu.isChecked)  
        s=radNu.text.toString()  
    //xử lý chuỗi s  
    if (s=="")  
        Toast.makeText(context: this, text: "Bạn chưa chọn giới tính", Toast.LENGTH_SHORT).show()  
    else  
        Toast.makeText(context: this, text: "Bạn chọn giới tính"+s, Toast.LENGTH_SHORT).show()  
}
```





## Android app with kotlin

### Bài 7

1

ImageView

2

ImageButton



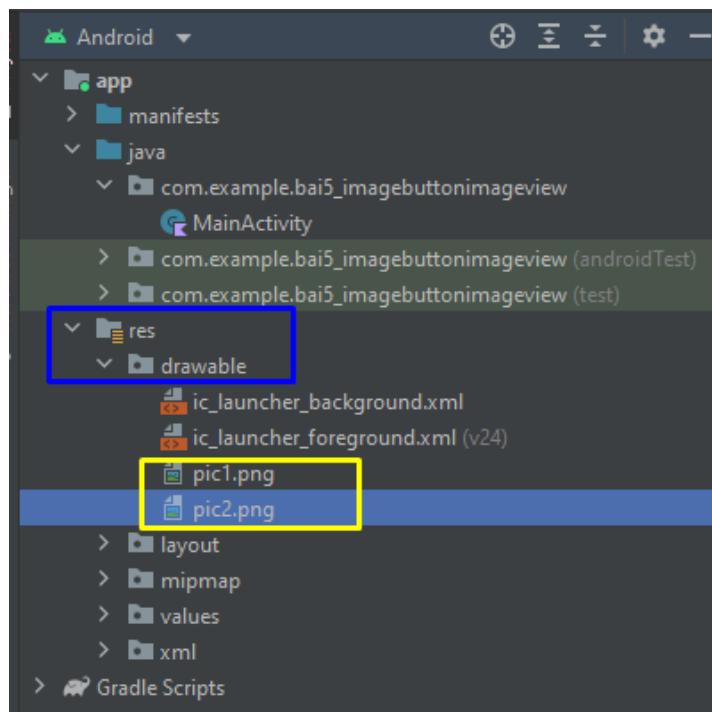
## 1

## ImageView

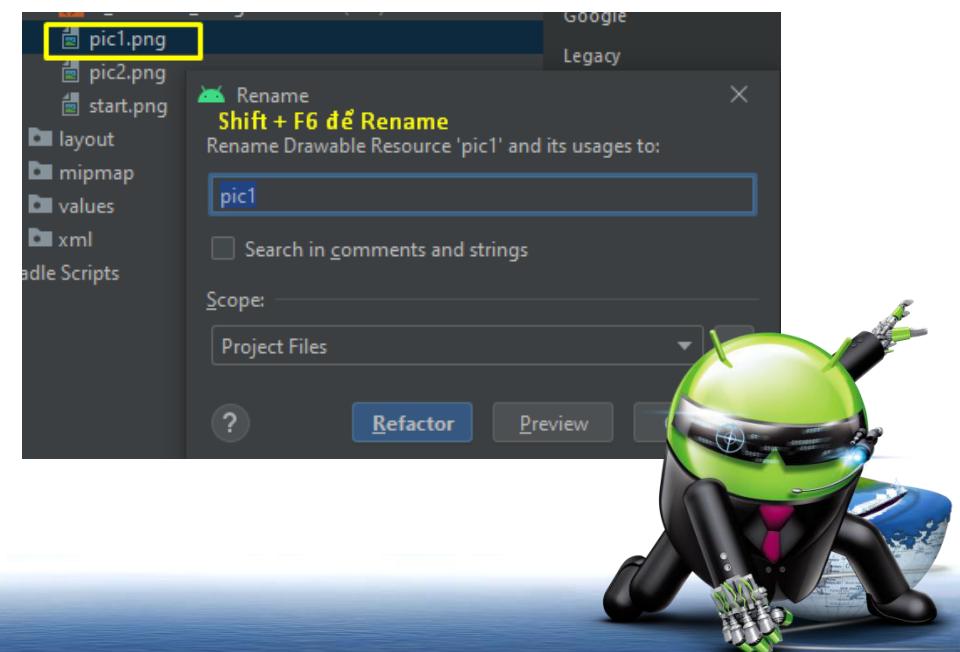
 1. ImageView : Dùng để hiển thị hình ảnh

- ✓ *Quy tắc đặt tên tệp hình ảnh : không bắt đầu bằng số, ký tự đặc biệt, space*
- ✓ *Các tệp hình ảnh đặt trong mục res >> drawable*
- ✓ *Quy tắc đặt id: img Ví dụ : imgHinh1*

*Link file ảnh trong bài :*  
<http://android.tuhoc.cc/>

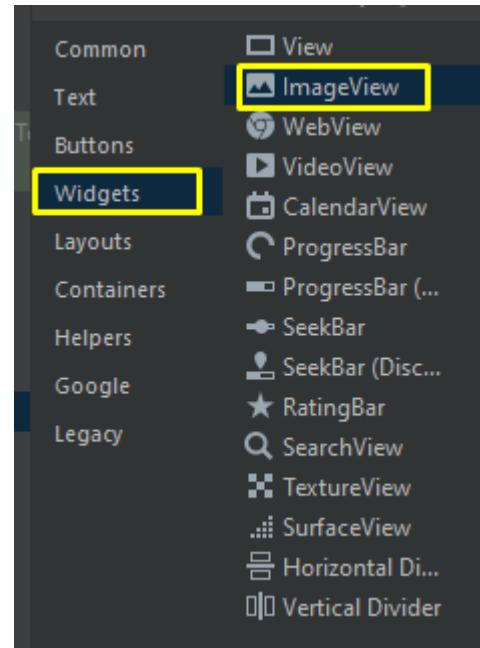
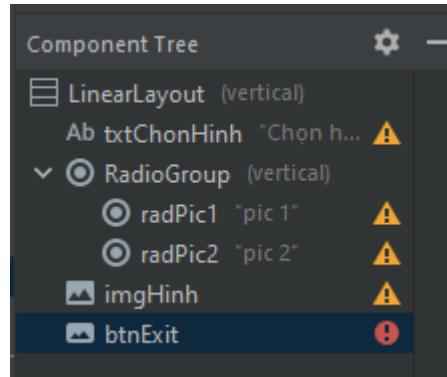


Ctrl + C => Ctrl V vào drawable



1

## Image View

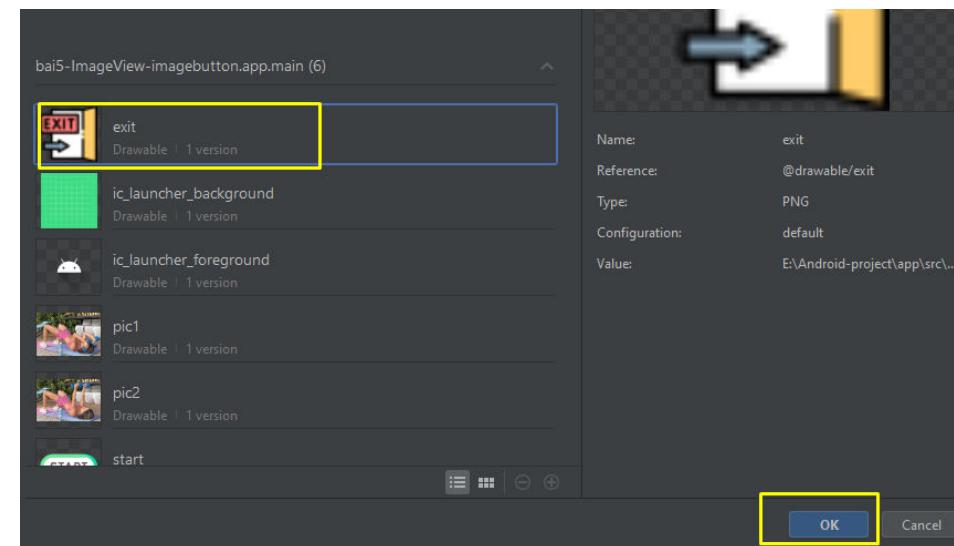
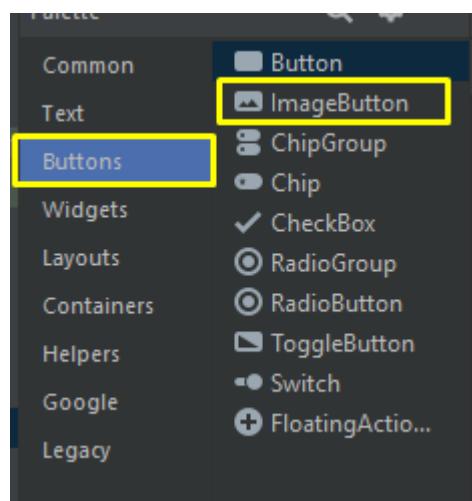


```
<ImageView  
    android:id="@+id/imgHinh"  
    android:layout_width="341dp"  
    android:layout_height="228dp"  
    android:layout_gravity="center"  
    app:srcCompat="@drawable/pic1" />
```



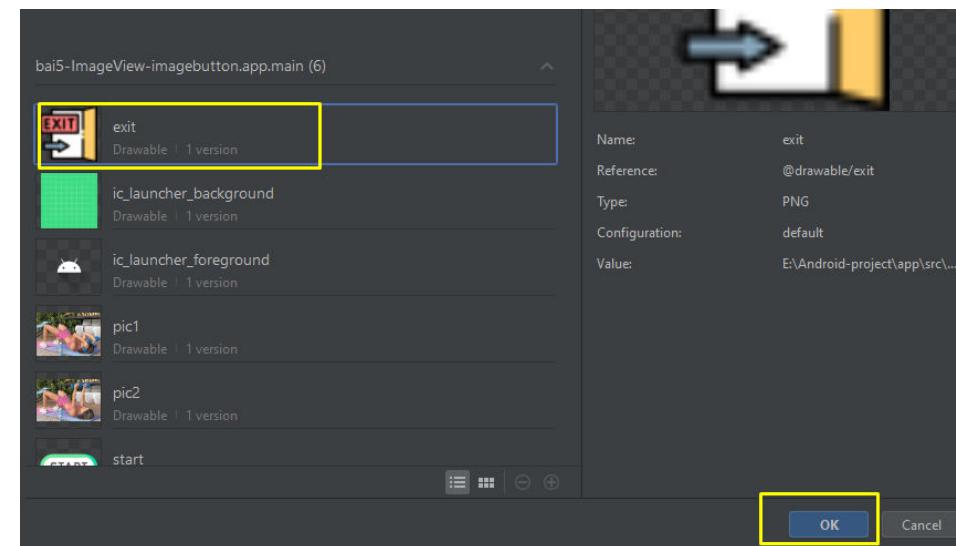
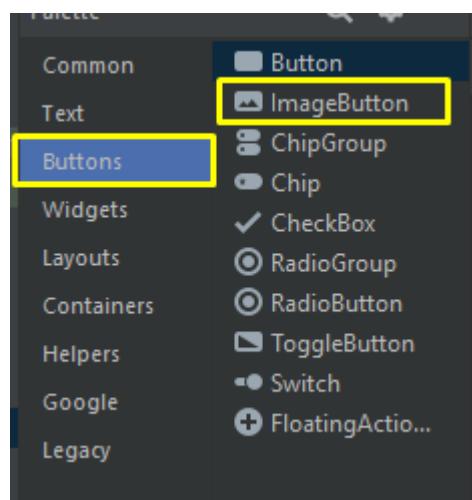
2

## ImageButton



2

## ImageButton



## 2

## ImageButton

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        //ad Events  
        addEvents()  
    }  
}
```

```
private fun addEvents() {  
    //kiểm tra Listener rad button xem có được chọn không?  
    // CompoundButton: Một nút có hai trạng thái, được chọn và bỏ chọn.  
    // Khi nhấn hoặc nhập vào nút, trạng thái sẽ tự động thay đổi.  
    radPic1.setOnCheckedChangeListener{ _, isChecked -> //  
        if (isChecked) {  
            imgHinh.setImageResource(R.drawable.pic1)  
        }  
    }  
    radPic2.setOnCheckedChangeListener { _, isChecked ->  
        if (isChecked) {  
            imgHinh.setImageResource(R.drawable.pic2)  
        }  
    }  
  
    //viết cho nút exit  
    btnExit.setOnClickListener{ v: View!  
        finish()  
    }  
}
```





## Android app with kotlin Bài 8.1

### Listview cơ bản

Nhập số ĐT:

Mời thím nhập số điện thoại vào đây

LƯU DATA

090152111

095241241

Lào
Campuchia
Thái Lan
Ba Lan
Hà Lan
Phần Lan

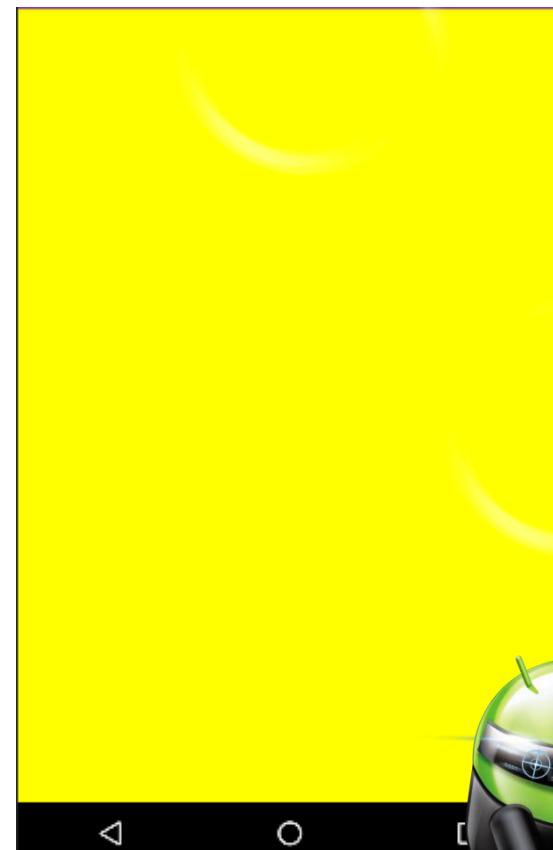
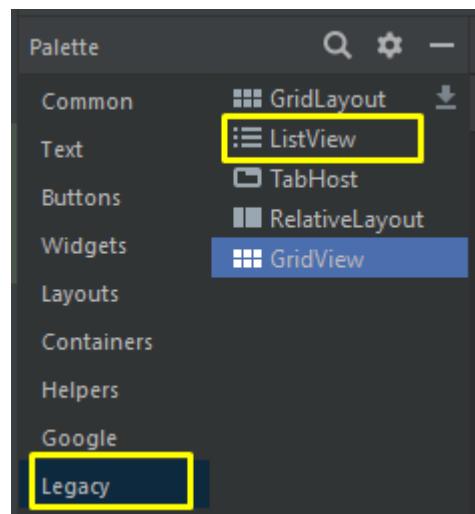


## 1

## Listview

## □ 1. Listview:

✓ *id : bắt đầu bằng tiền tố lv Ví dụ : lvMonAn*



```
<ListView  
    android:id="@+id/lvQuocGia"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="#FF0"  
/>
```



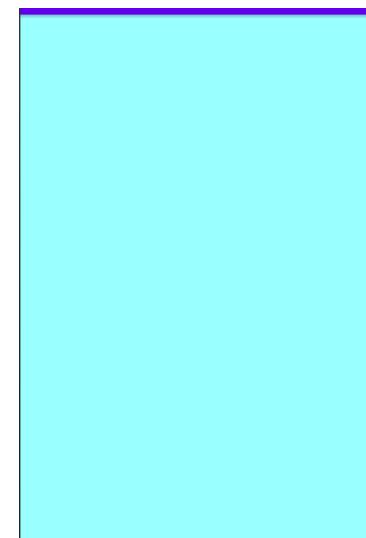
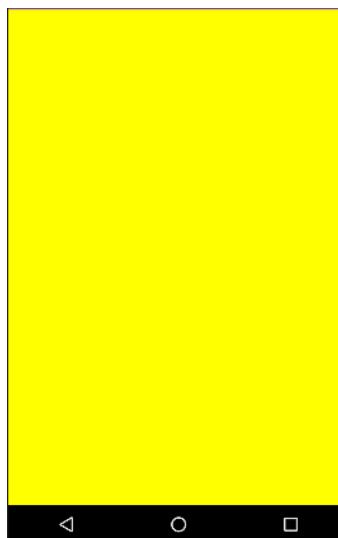
## 1

## Listview

 1. Listview :

✓ *Thử tương tác với listview : Chuyển màu cho listview*

```
//test set màu cho list view
//parseColor=> chuyển từ mã màu sang integer
val mauXanh= Color.parseColor( colorString: "#99FFFF")
lvQuocGia.setBackgroundColor(mauXanh)
//xem Log
Log.d( tag: "tuhoc" , msg: "$mauXanh")
```

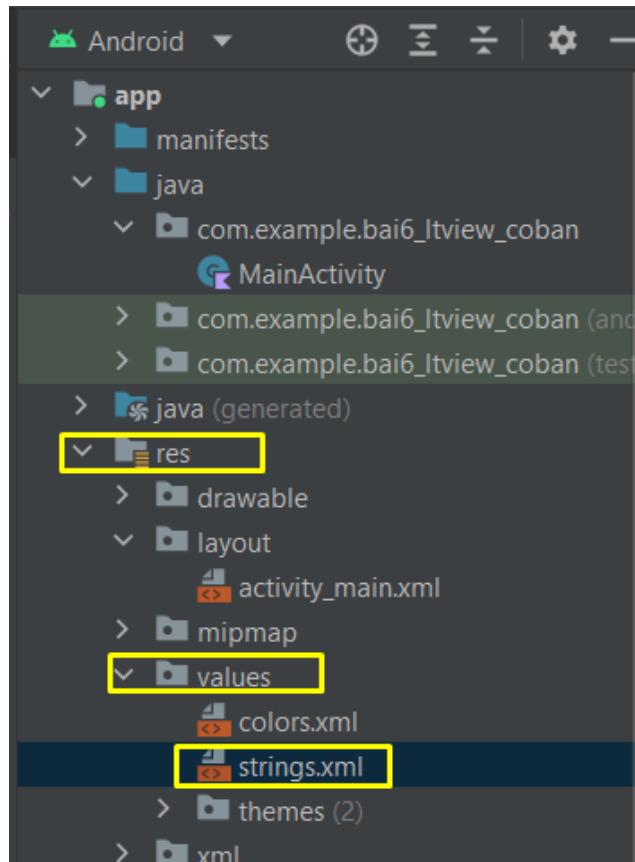


## 1

## Listview

## A . TH1: Nguồn Cố Định Không đổi :

## 2. Khai báo trong string.xml :



```
<string-array name="arrQuocGia">
    <item>Lào</item>
    <item>Campuchia</item>
    <item>Thái Lan</item>
    <item>Bá Lan</item>
    <item>Hà Lan</item>
    <item>Phần Lan</item>
</string-array>
```

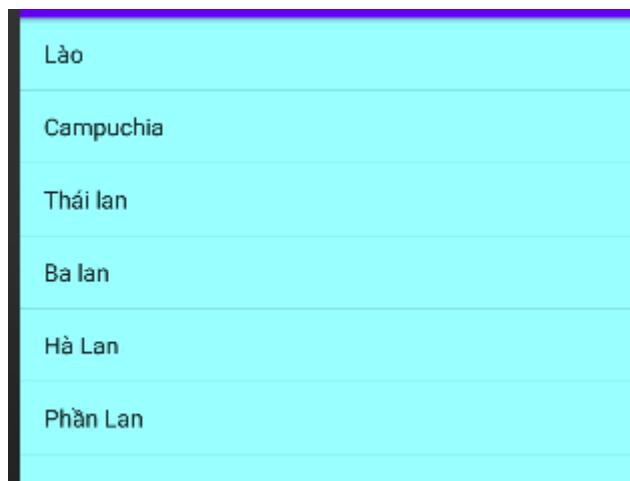


## 1

## Listview

 3 . Hiển thị danh sách lên listview :

```
//khai báo arrQuocGia, sử dụng resources để lấy thông tin từ string.xml
var arrQuocGia= resources.getStringArray(R.array.arrQuocGia)
//gán nguồn cho adapter và hiển thị data lên listview
lvQuocGia.adapter = ArrayAdapter(
    context: this, //màn hình activity hiện tại để hiển thị
    android.R.layout.simple_list_item_1, //vẽ từng dòng cơ bản có sẵn
    arrQuocGia) // nguồn dữ liệu gán cho adapter
```



## 1

## Listview

## □ 3 . Tối ưu hoá code :

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        //test set màu cho list view
        //parseColor=> chuyển từ mã màu sang integer
        val mauXanh= Color.parseColor(colorString: "#99FFFF")
        lvQuocGia.setBackgroundColor(mauXanh)
        //xem Log
        Log.d( tag: "tuhoc" , msg: "$mauXanh")

        addEvents()
    }

    private fun addEvents() {
        hienThiLvQuocGia()
    }

    private fun hienThiLvQuocGia() {
        //khai báo arrQuocGia, sử dụng resources để lấy thông tin từ string.xml
        var arrQuocGia= resources.getStringArray(R.array.arrQuocGia)
        //gán nguồn cho adapter và hiển data lên listview
        lvQuocGia.adapter = ArrayAdapter(
            context: this, //màn hình activity hiện tại để hiển lên
            android.R.layout.simple_list_item_1, //vẽ từng dòng cơ bản có sẵn
            arrQuocGia) // nguồn dữ liệu gán cho adapter
    }
}
```



## 1

## Listview

## ❑ 4 . setOnItemClickListener :

✓ *bắt sự kiện click vào item trên list view*

```
private fun hienThiLvQuocGia() {
    //khai báo arrQuocGia, sử dụng resources để lấy thông tin từ string.xml
    var arrQuocGia= resources.getStringArray(R.array.arrQuocGia)
    //gán nguồn cho adapter và hiển thị data lên listview
    lvQuocGia.adapter = ArrayAdapter(
        context: this, //màn hình activity hiện tại để hiển thị
        android.R.layout.simple_list_item_1, // vẽ từng dòng cơ bản có sẵn
        arrQuocGia) // nguồn dữ liệu gán cho adapter

    //Nghe thông tin click vào list view
    lvQuocGia.setOnItemClickListener { adapterView, view, i, l ->
        Toast.makeText(
            context: this,
            text: "Bạn chọn" + arrQuocGia[i],
            Toast.LENGTH_SHORT
        ).show()
    }
}
```



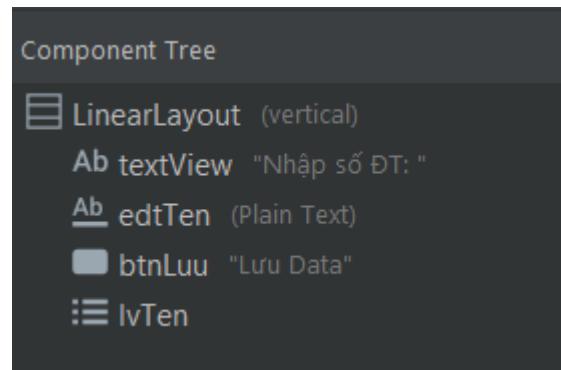
1

## Listview

### B. TH2 : Nguồn dữ liệu thay đổi (có thêm, sửa, xoá)

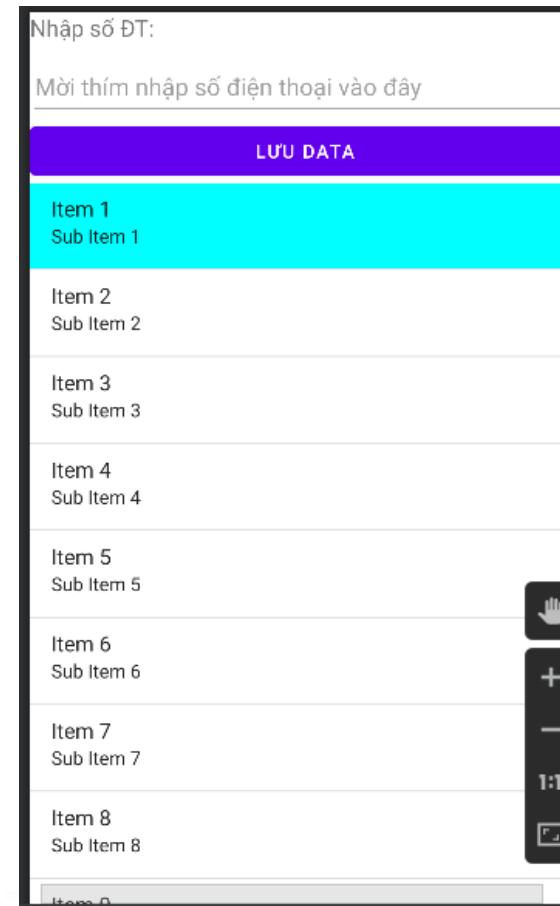
Sử dụng tập collection để quản lý nguồn dữ liệu , cụ thể là sử dụng List:

Kiến thức về list ôn lại bài 23 kotlin cơ bản  
<http://kotlin.tuhoc.cc>



```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
    id 'kotlin-android'
    id 'kotlin-android-extensions'
```

Add plugins

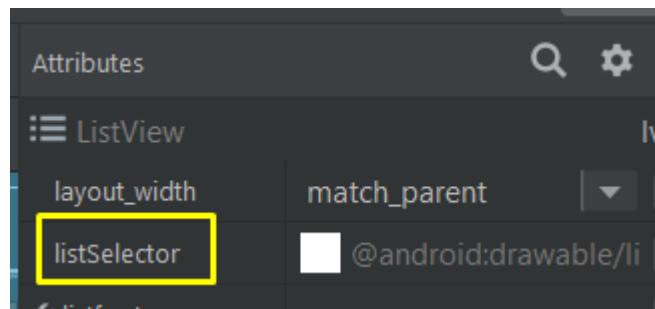


1

## Listview

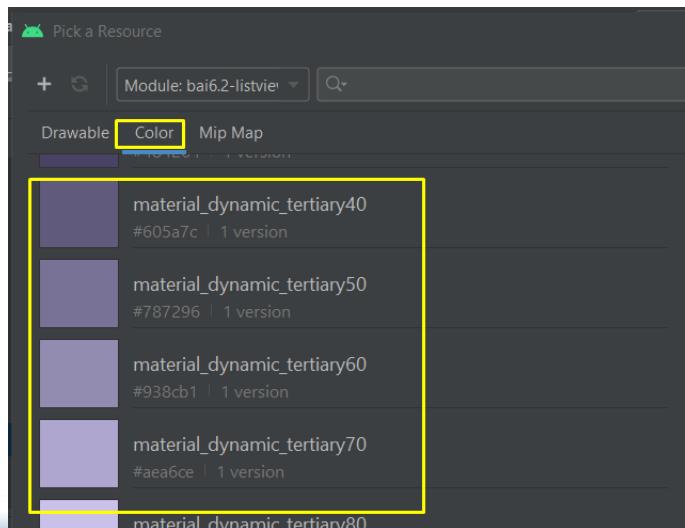
## □ 5 . Set màu cho item trong listview khi được chọn

## Cách 1: Sửa attributes



## Cách 2: Sửa code

```
<ListView  
    android:id="@+id/lvTen"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:listSelector="#00FFFF" />
```



1

## Listview

```
class MainActivity : AppCompatActivity() {  
  
    //khai báo list rỗng  
    var ds:MutableList<String> = mutableListOf("090152111","095241241")  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        //hiển thị danh sách ban đầu  
        lvTen.adapter = ArrayAdapter(  
            context: this,  
            android.R.layout.simple_list_item_1,  
            ds  
        )  
    }  
}
```

Nhập số ĐT:

Mời thím nhập số điện thoại vào đây

LƯU DATA

090152111

095241241

1

## Listview

```
//hàm addevents  
addEvents()
```

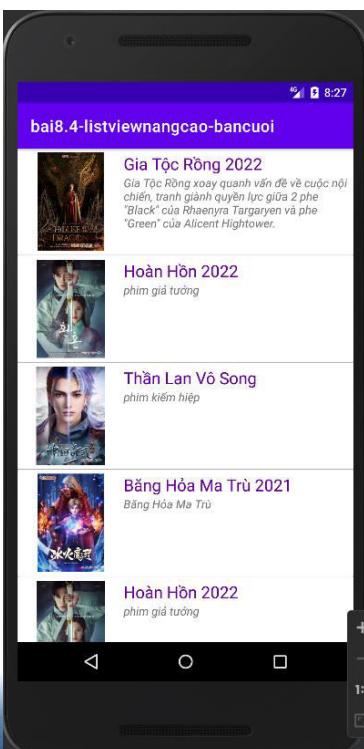
```
private fun addEvents() {  
    //code cho button lưu  
    btnLuu.setOnClickListener { it: View!  
        hamXuLyNutLuu()  
    }  
}  
  
private fun hamXuLyNutLuu() {  
    var s:String = edtTen.text.toString()  
    ds.add(s) // thêm sdt vào list ds  
    edtTen.setText("")  
    edtTen.requestFocus()// đẩy con trỏ văn bản vào ô nhập liệu  
  
    lvTen.adapter = ArrayAdapter(  
        context: this,  
        android.R.layout.simple_list_item_1,  
        ds  
    )  
}
```



Build Android  
App with Kotlin

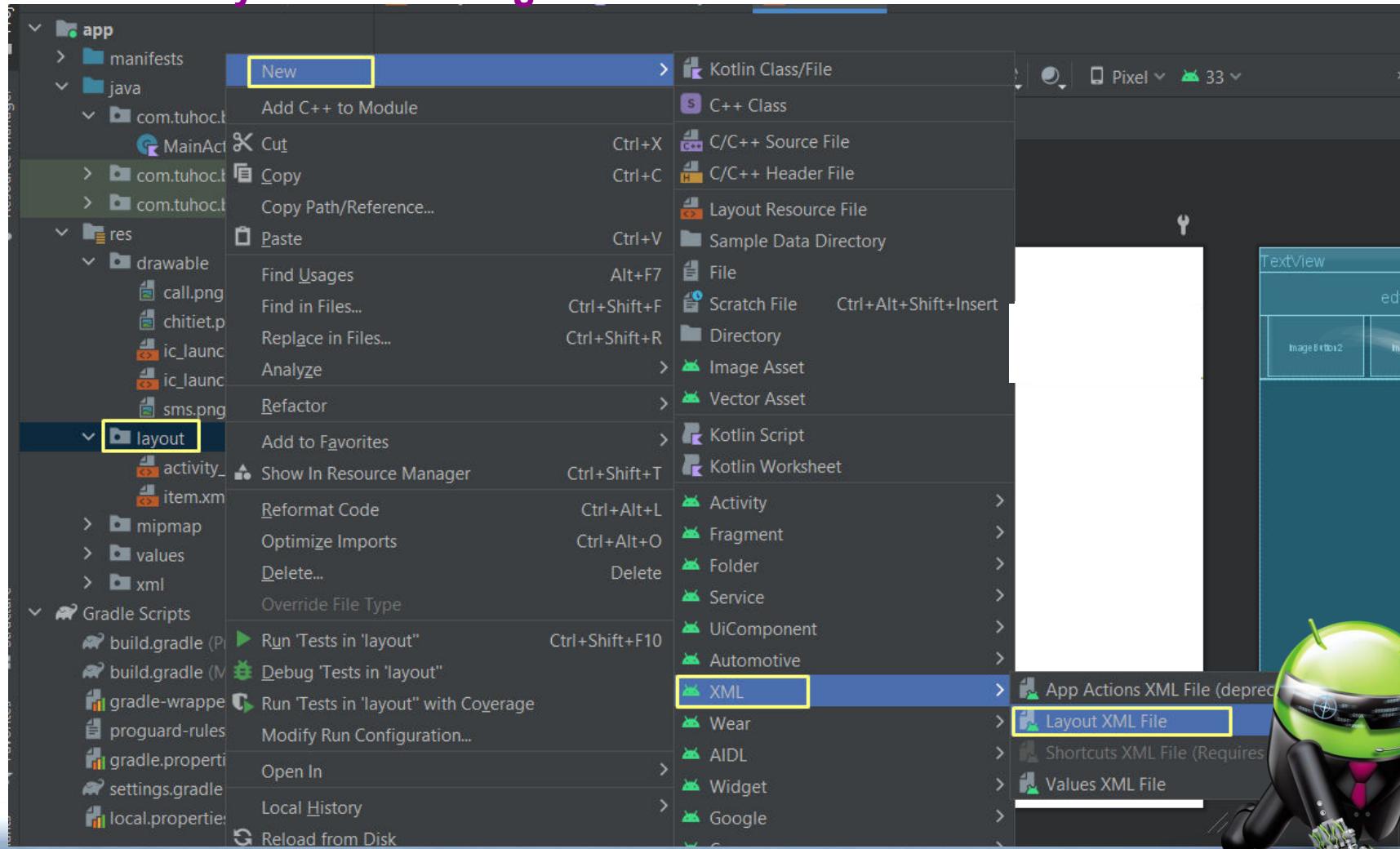
## Android app with kotlin Bài 8.2

Listview nâng cao



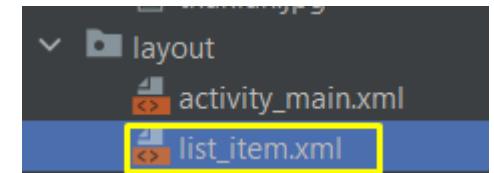
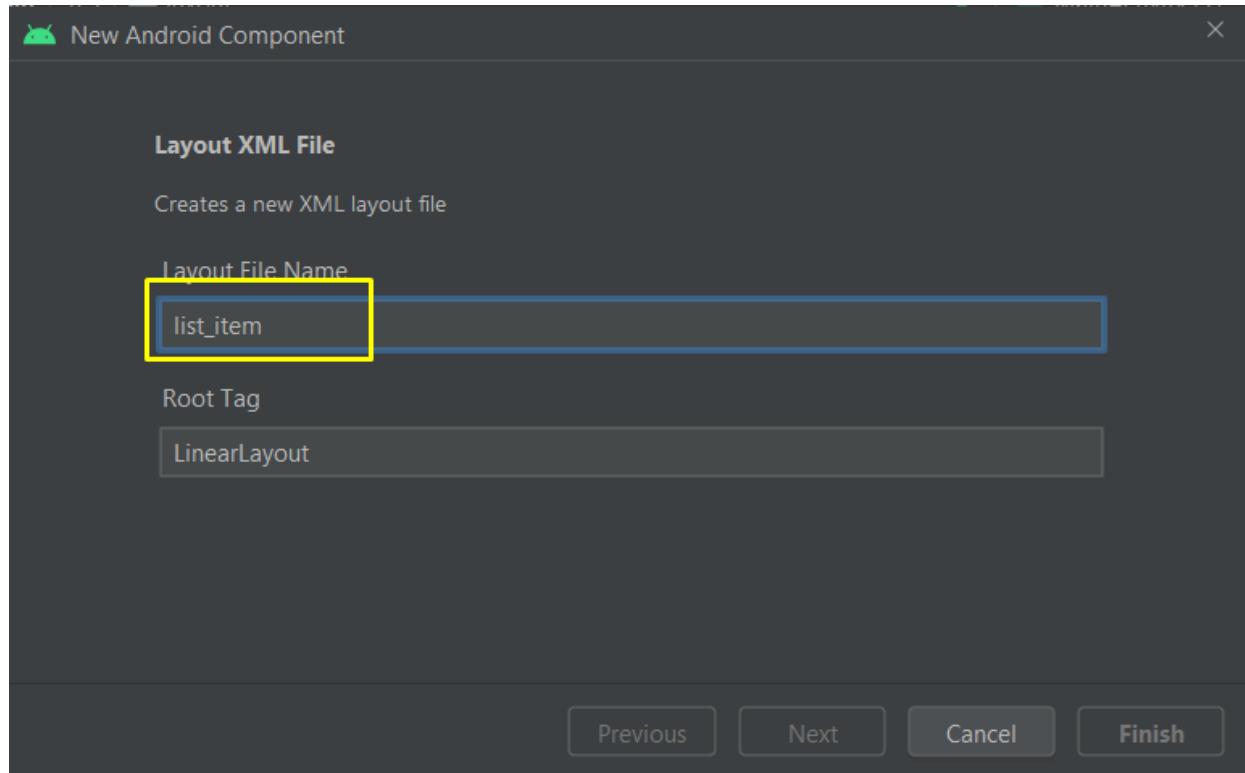
## 1

## Listview nâng cao

 1 . Vẽ layout cho 1 dòng :

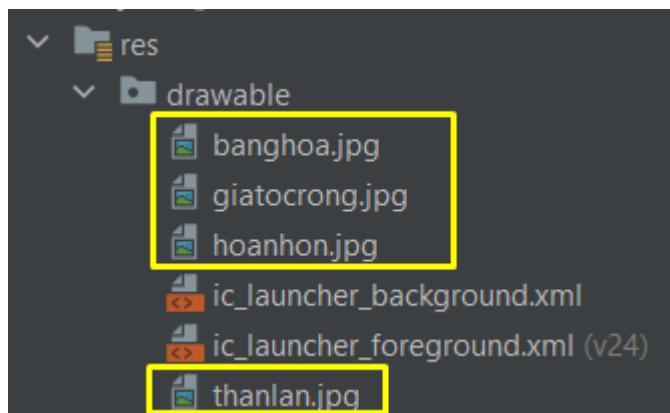
1

## Listview nâng cao

 1 . Vẽ layout cho 1 dòng :

1

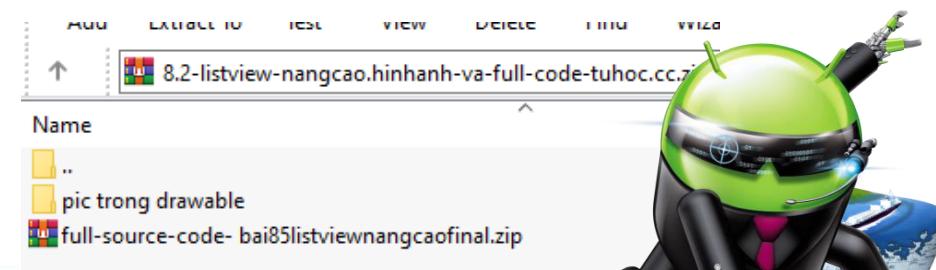
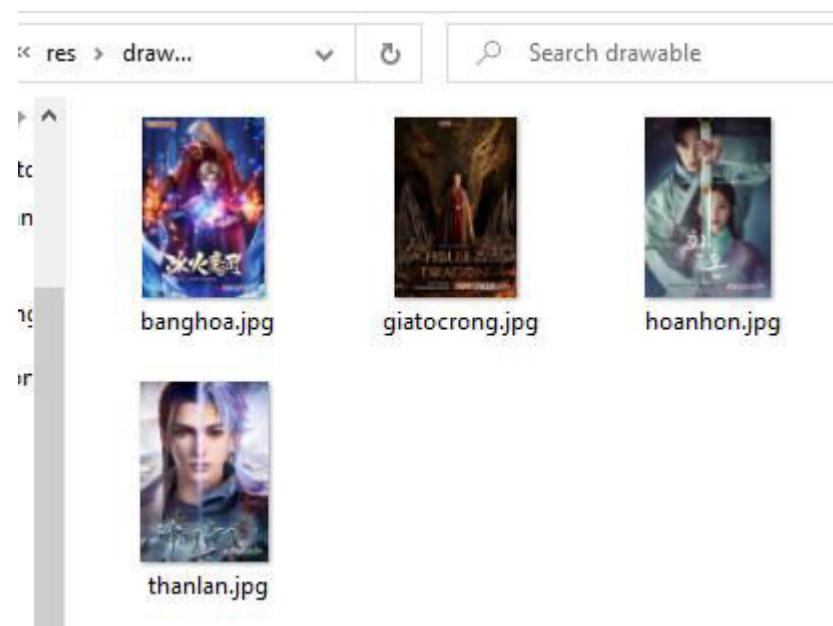
## Listview nâng cao

 1 . Vẽ layout cho 1 dòng : Đưa hình ảnh vào drawable

Ảnh để trong drawable  
và code toàn bộ bài 8,2

<http://android.tuhoc.cc/>

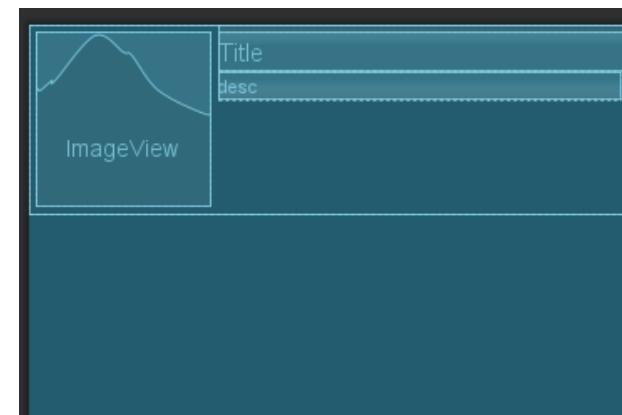
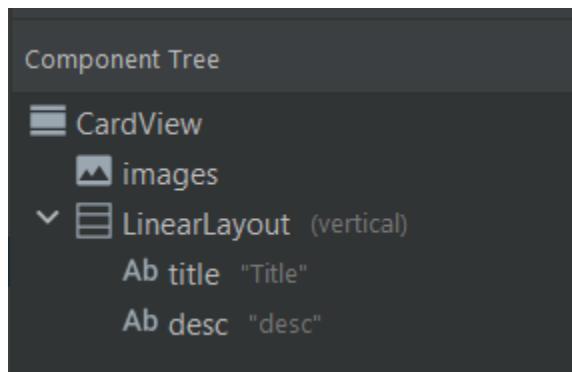
8.2-listview-nangcao.hinhanh-va-full-code-tuhoc...



1

## Listview nâng cao

### 1 . Vẽ layout cho 1 dòng : Thiết kế list\_item.xml



## 1

## Listview nâng cao

 1 . Vẽ layout cho 1 dòng : Thiết kế list\_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_marginTop="0.5dp"
    android:layout_height="130dp">

    <ImageView
        android:id="@+id/images"
        android:layout_width="120dp"
        android:layout_height="120dp"
        android:layout_marginTop="5dp"
        android:layout_marginLeft="5dp"
        app:srcCompat="@drawable/ic_launcher_background" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="130dp"
        android:orientation="vertical">

        <TextView
            android:id="@+id/title"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="20dp"
            android:layout_marginTop="5dp"
            android:textColor="#660099"
            android:text="Title" />

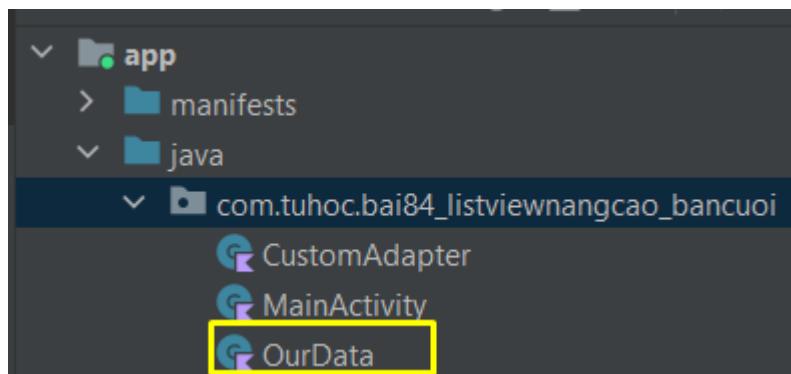
        <TextView
            android:id="@+id/desc"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textStyle="italic"
            android:layout_marginRight="5dp"
            android:text="desc" />
    
</androidx.cardview.widget.CardView>
```

```
<TextView
    android:id="@+id/desc"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textStyle="italic"
    android:layout_marginRight="5dp"
    android:text="desc" />
</LinearLayout>
</androidx.cardview.widget.CardView>
```



1

## Listview nâng cao

 2 . Tạo Class Out Data để định hình các phần tử trong 1 view

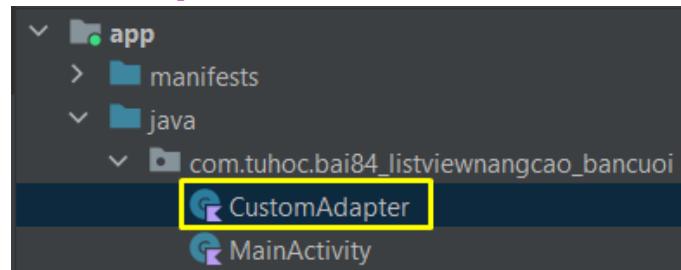
```
class OurData {  
    val image:Int,  
    val title: String,  
    val desc:String}{
```



## 1

## Listview nâng cao

## □ 3 . Tạo CustomAdapter : Class Adapter tùy chỉnh



```
class CustomAdapter(val activity: Activity, val list: List<OurData>): ArrayAdapter<OurData>(activity, R.layout.list_item) {
    override fun getCount(): Int {
        return list.size // số dòng sẽ vẽ lên list view
    }
    override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {
        val contexts = activity.layoutInflater
        //LayoutInflater là 1 component
        // giúp bạn chuyển layout file (xml) thành View (Java code) trong Android
        val rowView = contexts.inflate(R.layout.list_item, parent, attachToRoot: false)

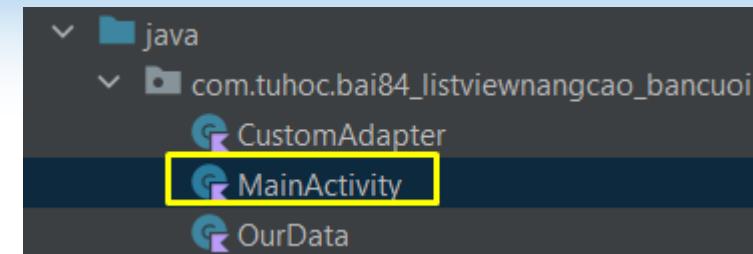
        val images= rowView.findViewById<ImageView>(R.id.images)
        val title = rowView.findViewById<TextView>(R.id.title)
        val desc = rowView.findViewById<TextView>(R.id.desc)

        title.text = list[position].title
        desc.text = list[position].desc
        images.setImageResource(list[position].image)
        return rowView
    }
}
```

1

## Listview nâng cao

## □ 4 . Code MainActivity



```
class MainActivity : AppCompatActivity() {  
    lateinit var customAdapter: CustomAdapter  
    /* lateinit là khởi tạo muộn.  
     * thông thường khai báo biến kotlin sẽ yêu cầu khởi tạo ( gán giá trị cho biến)  
     * Ta có thể dùng lateinit để gán giá trị sau */  
    //ví dụ lateinit var chuoit: String  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        //khai báo danh sách các item list  
        var list = mutableListOf<OurData>()  
        list.add(OurData(R.drawable.giatocrong, title: "Gia Tộc Rồng 2022", desc: "Gia Tộc Rồng xoay qua  
        list.add(OurData(R.drawable.hoanhon, title: "Hoàn Hồn 2022", desc: "phim giả tưởng"))  
        list.add(OurData(R.drawable.thanlan, title: "Thần Lan Võ Song", desc: "phim kiếm hiệp"))  
        list.add(OurData(R.drawable.banghoa, title: "Băng Hòa Ma Trù 2021", desc: "Băng Hòa Ma Trù"))  
  
        customAdapter = CustomAdapter( activity: this, list)  
        //khai báo biến list view Phim link tới view lvPhim  
        val lvPhim = findViewById<ListView>(R.id.lvPhim)  
        lvPhim.adapter = customAdapter  
    }  
}
```

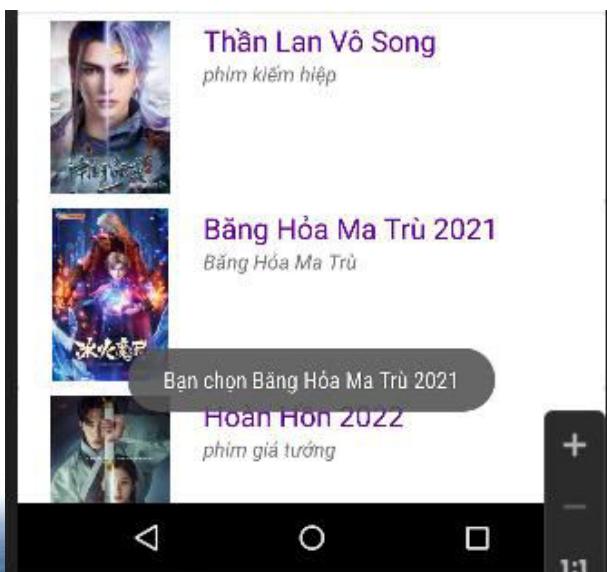


## 1

## Listview nâng cao

## □ 5 . list view item click listener

```
//list view item click listener
lvPhim.onItemClickListener = AdapterView.OnItemClickListener { adapterView, view, i, l ->
    //AdapterView: AdapterView nơi nhấp chuột đã xảy ra.
    //view View: Chế độ xem trong Chế độ xem bộ điều hợp đã được nhấp
    // (đây sẽ là chế độ xem do bộ điều hợp cung cấp)
    //i int: Vị trí của khung nhìn trong bộ điều hợp.
    //l long: Id hàng của mục đã được nhấp.
    Toast.makeText(context: this, text: "Bạn chọn "+list[i].title,Toast.LENGTH_SHORT).show()
}
```





## Android app with kotlin Bài 9,1

### Spinner basic

Mời bạn chọn ngôn ngữ

English  
VietNam  
Hà Lan

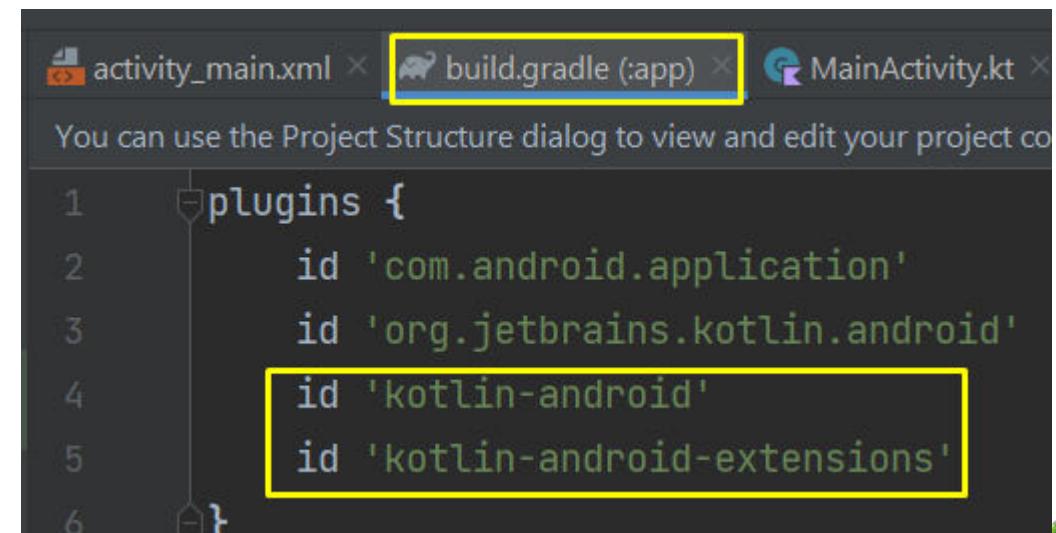
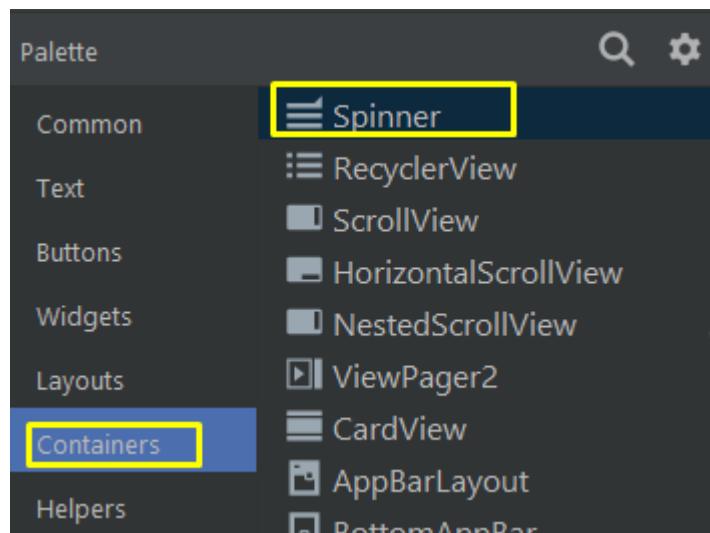


## 1

## Spinner basic

## □ 1. Spinner:

- ✓ *id : bắt đầu bằng tiền tố sp* Ví dụ : spNgonNgu
- ✓ *Spinner gần giống với listview nhưng nó hiển thị dưới dạng drop down*



The screenshot shows the build.gradle file for the app module. A yellow box highlights the 'plugins' section where two new plugins are added: 'kotlin-android' and 'kotlin-android-extensions'. The file content is as follows:

```
1 plugins {  
2     id 'com.android.application'  
3     id 'org.jetbrains.kotlin.android'  
4     id 'kotlin-android'  
5     id 'kotlin-android-extensions'  
6 }
```

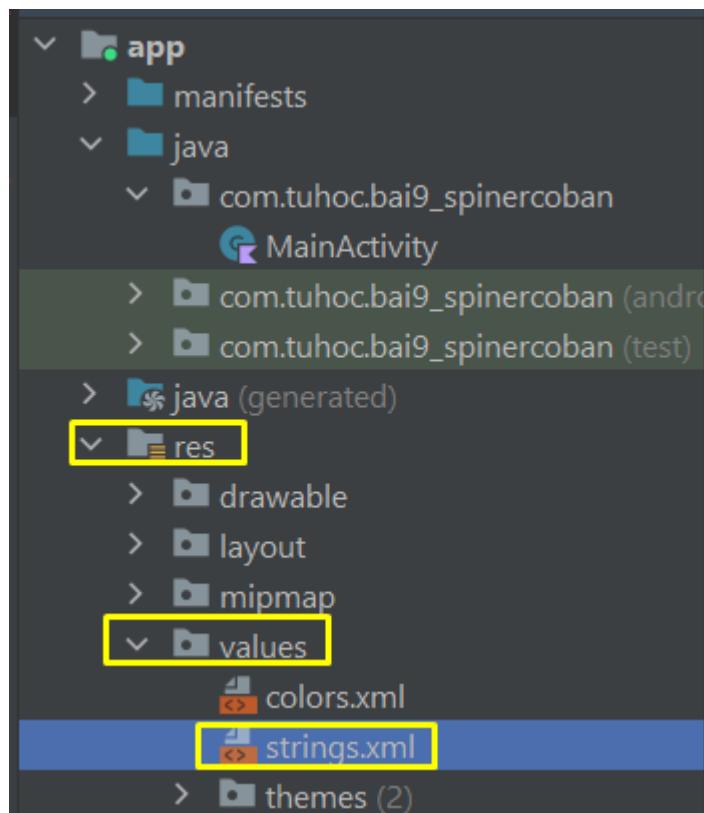
Add plugin để skip khai báo biến



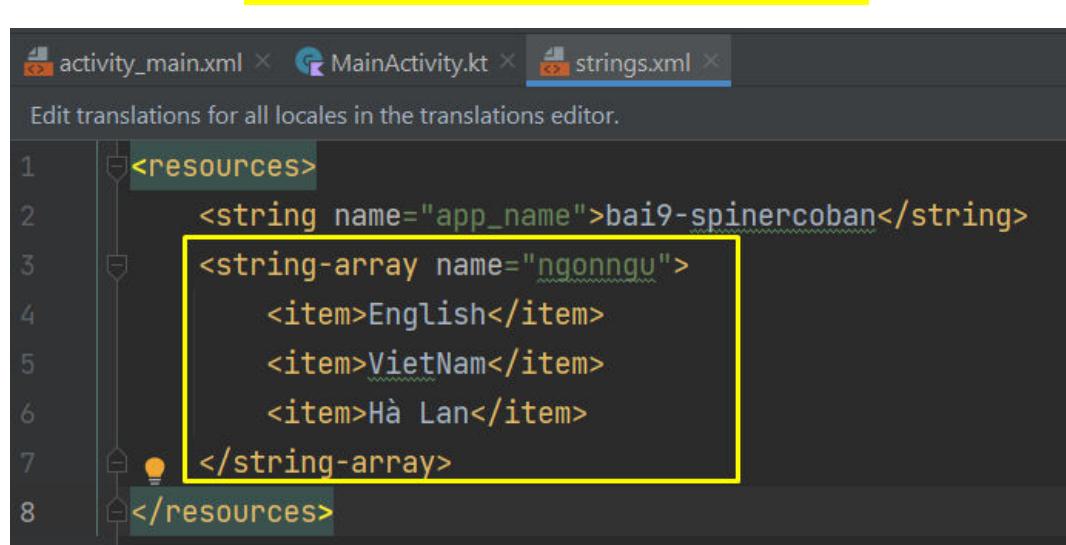
## 1

## Spinner basic

## □ 2. TH1 . Spinner với nguồn cấp cố định :



### Strings.xml



The screenshot shows the Android Studio editor with the 'strings.xml' file open. The code is:

```
<resources>
    <string name="app_name">bai9-spinercoban</string>
    <string-array name="ngonngu">
        <item>English</item>
        <item>VietNam</item>
        <item>Hà Lan</item>
    </string-array>
</resources>
```

A yellow box highlights the entire `<string-array>` block, which contains three items: English, VietNam, and Hà Lan.

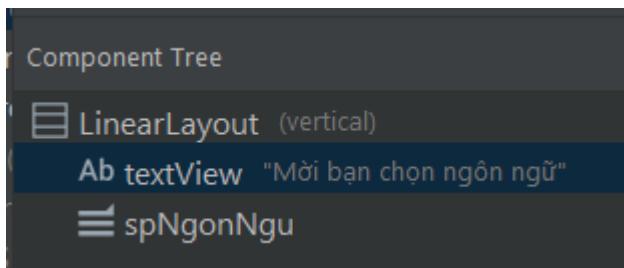


## 1

## Spinner basic

## □ 2. TH1 . Spinner với nguồn cấp cố định :

MainActivity.kt



```
// tạo 1 list ngôn ngữ
val list = resources.getStringArray(R.array.ngonngu)

//khai báo bộ điều hướng
val adt = ArrayAdapter(context: this, // màn hình hiển thị
    android.R.layout.simple_spinner_item, //cách hiển thị
    list) // nguồn cấp cho adapter
//gọi spinner
spNgonNgu.adapter = adt
```

Mời bạn chọn ngôn ngữ

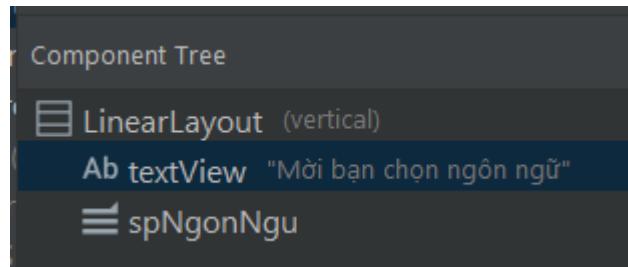
English  
VietNam  
Hà Lan



## 1

## Spinner basic

## □ 3. TH2 . Spinner với nguồn cấp thay đổi:



```
// tạo 1 list ngôn ngữ
val list = mutableListOf<String>()
list.add("English")
list.add("Việt Nam")
list.add("Tây Ban Nha")

// khai báo bộ điều hướng
val adt = ArrayAdapter(context: this, // màn hình hiển thị
    android.R.layout.simple_spinner_item, // cách hiển thị
    list) // nguồn cấp cho adapter

// gọi spinner
spNgonNgu.adapter = adt
```

Mời bạn chọn ngôn ngữ

English  
Việt Nam  
Tây Ban Nha



## 1

## Spinner basic

## ❑ 4. spinner.onItemSelectedListener :

✓ *Lắng nghe xem item nào của spinner đang được chọn*

```
spNgonNgu.onOptionsItemSelected = object : AdapterView.OnItemSelectedListener{
    //Ctrl + O (chữ o) để tái định nghĩa lại phương thức
    override fun onItemSelected(p0: AdapterView<*>?, p1: View?, p2: Int, p3: Long) {
        //alt + enter khi makeText báo lỗi
        //p2:Int là vị trí của phần tử trong list
        Toast.makeText(context: this@MainActivity, text: "Bạn chọn "+list[p2], Toast.LENGTH_SHORT).show()
    }

    override fun onNothingSelected(p0: AdapterView<*>?) {
        TODO(reason: "Not yet implemented")
    }
}
```





## Android app with kotlin Bài 9 ,2

### Custom Spinner

### Custom Spinner:



Xoài lắc nam bộ

### Custom Spinner:



Cam tươi



Dưa hấu sài gòn



Táo china



Sầu riêng không mùi



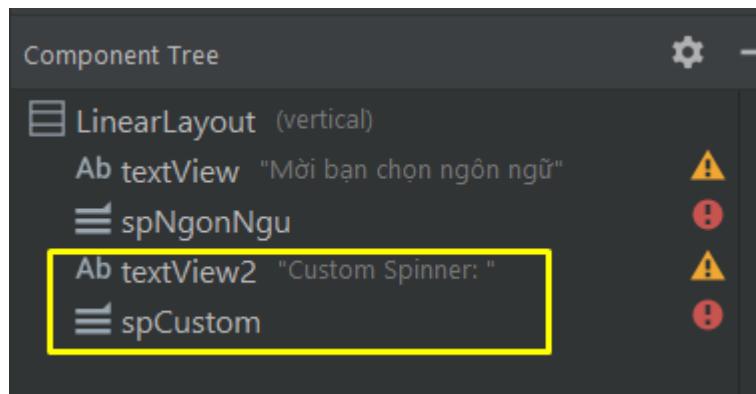
Xoài lắc nam bộ



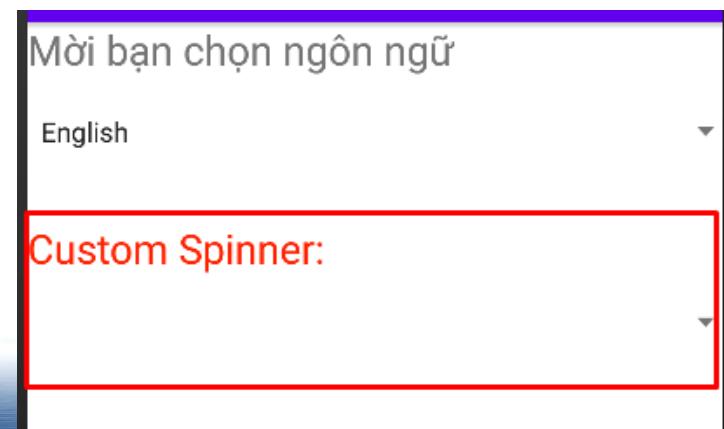
## 2

## Custom Spinner

## □ 1. Thiết kế giao diện design :



```
<TextView  
    android:id="@+id/textView2"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textSize="24dp"  
    android:layout_marginTop="30dp"  
    android:textColor="#f20"  
    android:text="Custom Spinner: " />  
  
<Spinner  
    android:id="@+id/spCustom"  
    android:layout_width="match_parent"  
    android:layout_height="56dp" />
```



2

## Custom Spinner

### □ 2. Các bước tạo custom spinner:

Step 1:  
Make xml layout  
CustomLayout

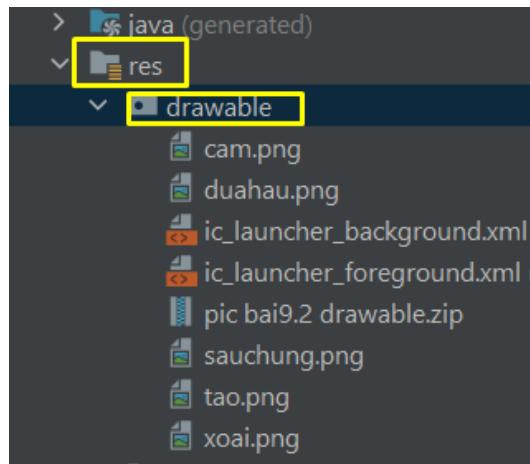
Step 2:  
Make class  
Class OutData

Step 3:  
Make Class  
CustomSpinner



2

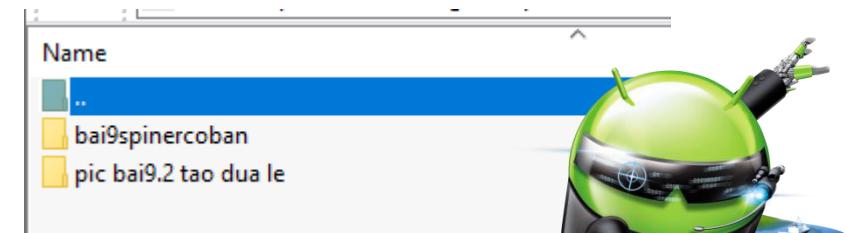
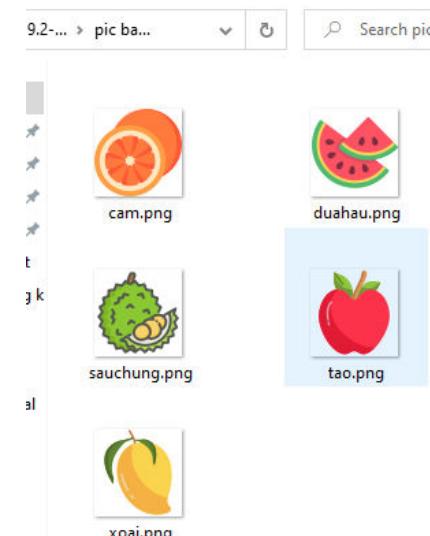
## Custom Spinner

 3. Step 1 - tạo xml layout file tùy chỉnh – Đưa hình ảnh vào drawable:

Ảnh để trong drawable  
và code toàn bộ bài 9,2

<http://android.tuhoc.cc/>

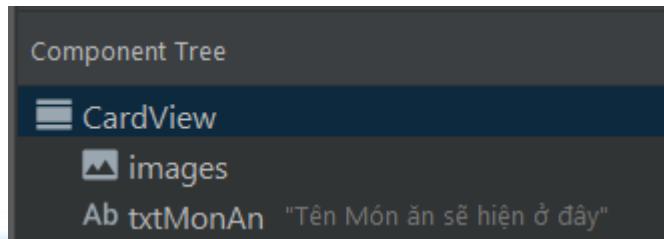
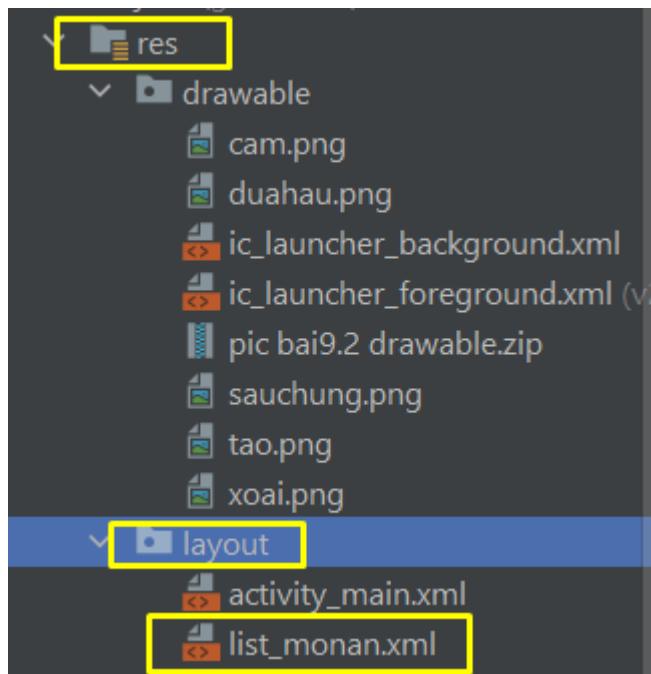
9.2-bai9spinnercoban-nangcao.zip 



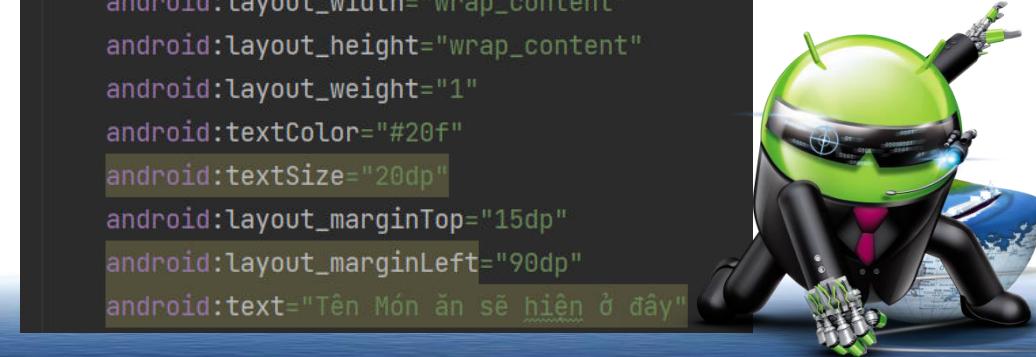
## 2

## Custom Spinner

## □ 3. Step 1: tạo xml layout file tuỳ chỉnh :



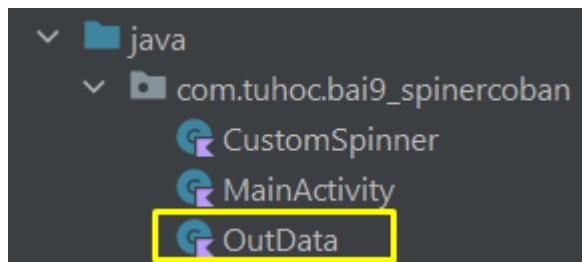
```
<ImageView  
    android:id="@+id/images"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="5dp"  
    android:layout_marginLeft="5dp"  
    app:srcCompat="@drawable/cam" />  
  
<TextView  
    android:id="@+id/txtMonAn"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:textColor="#20f"  
    android:textSize="20dp"  
    android:layout_marginTop="15dp"  
    android:layout_marginLeft="90dp"  
    android:text="Tên Món ăn sẽ hiện ở đây" />
```



## 2

## Custom Spinner

- 4. Step 2: tạo Class OutData lưu trữ các thành phần của 1 dòng :



```
class OutData {  
    val image:Int,  
    val mieuta:String,  
}  
}
```

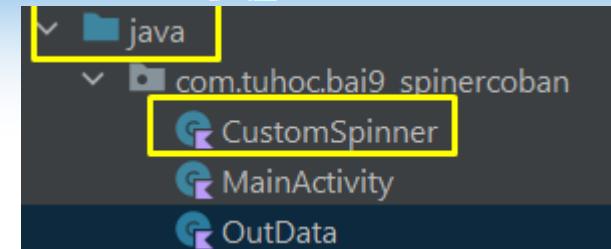


## 2

## Custom Spinner

## □ 5. Step 3: tạo Class CustomSpinner:

```
class CustomSpinner (val activity: Activity, val list2: List<OutData>) : ArrayAdapter<OutData>(activity,R.layout.list_monan){  
    override fun getCount(): Int {  
        return list2.size //vẽ lên spinner bao nhiêu dòng  
    }  
    override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {  
        //position: Int Vị trí của món ăn trong danh sách list  
        //convertView: View? chế độ xem cho từng mục trong spinner  
        // parent: ViewGroup : Viewgroup cha đang chứa spinner  
        return initView(position,convertView,parent)  
    }  
    override fun getDropDownView(position: Int, convertView: View?, parent: ViewGroup): View {  
        return initView(position,convertView,parent)  
    }  
    //hàm xử lý view  
    private fun initView(position: Int, convertView: View?, parent: ViewGroup): View {  
        val contexts = activity.layoutInflater  
        // layoutInflator là 1 component  
        //giúp chuyển đổi layout xml thành View trong android  
        val rowView = contexts.inflate(R.layout.list_monan,parent, attachToRoot: false)  
        //cơ bản ý nghĩa dòng lệnh này là biến 1 xml thành 1 view  
        val images = rowView.findViewById<ImageView>(R.id.images)  
        val txtMonAn = rowView.findViewById<TextView>(R.id.txtMonAn)  
  
        images.setImageResource(list2[position].image)  
        txtMonAn.text= list2[position].mieuta  
        return rowView  
    }  
}
```



## 2

## Custom Spinner

## □ 6. Code Mainactivity:

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
  
    //Tạo hàm custom spinner để dễ quản lý  
    setupCustomSpinner()  
}
```

```
private fun setupCustomSpinner() {  
    //khai báo danh sách các món ăn cho vào list2  
    var list2 = mutableListOf<OutData>()  
    list2.add(OutData(R.drawable.cam, mieuta: "Cam tươi"))  
    list2.add(OutData(R.drawable.duahau, mieuta: "Dưa hấu sài gòn"))  
    list2.add(OutData(R.drawable.tao, mieuta: "Táo china "))  
    list2.add(OutData(R.drawable.sauchung, mieuta: "Sầu riêng không mùi"))  
    list2.add(OutData(R.drawable.xoai, mieuta: "Xoài lắc nam bộ"))  
  
    val customSpinner = CustomSpinner(activity: this, list2)  
    spCustom.adapter = customSpinner  
}
```



## 1

## Spinner

## ❑ 4. spinner.onItemSelectedListener :

✓ *Lắng nghe xem item nào của spinner đang được chọn*

```
//lắng nghe xem item của spinner có được chọn?  
spCustom.onItemSelectedListener = object : AdapterView.OnItemSelectedListener{  
    //Ctrl + O (chữ o) để tái định nghĩa lại phương thức  
    override fun onItemSelected(p0: AdapterView<*>?, p1: View?, p2: Int, p3: Long) {  
        //p2:Int là vị trí của phần tử trong list  
        Toast.makeText(context: this@MainActivity, text: "Bạn chọn "+list2[p2].mieuta,Toast.LENGTH_SHORT).show()  
    }  
    override fun onNothingSelected(p0: AdapterView<*>?) {  
    }  
}
```

Custom Spinner:



Táo china

Bạn chọn Táo china





## Android app with kotlin

### Bài 10

Google

#### AutoCompleteTextView

Select tỉnh thành:

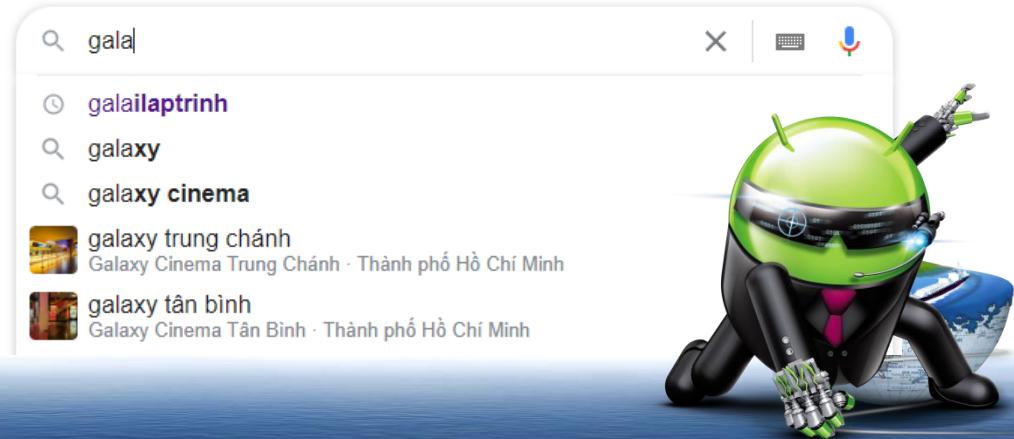
--Select Tỉnh Thành

An Giang

Bà Rịa – Vũng Tàu

Bắc Giang

Bắc Kạn

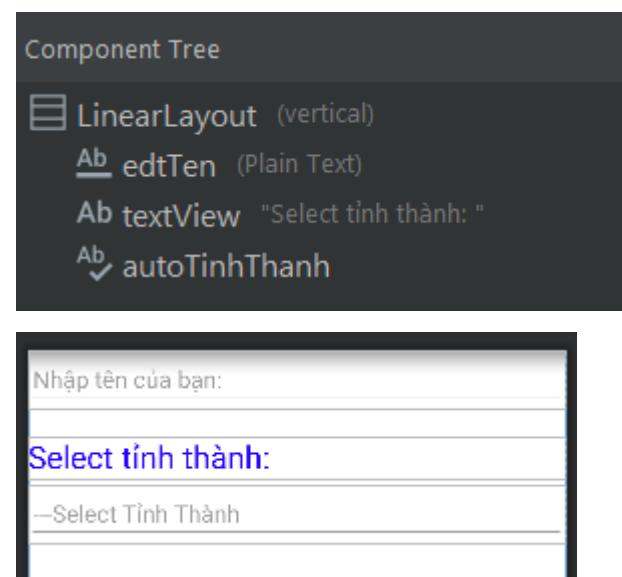
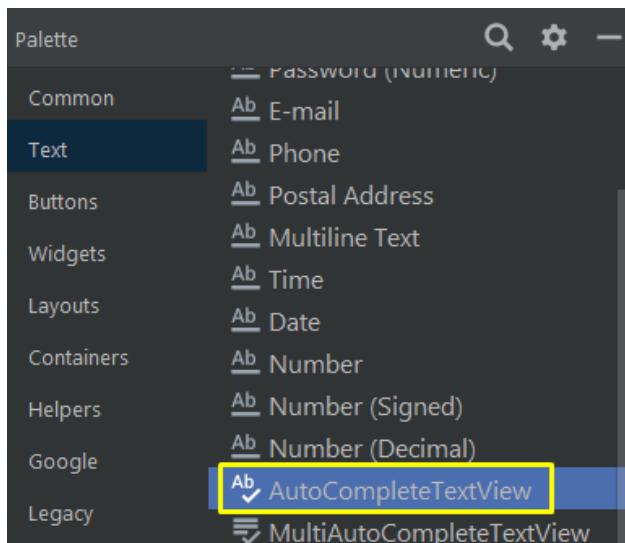


## 1

## AutoCompleteTextView

1. Mục đích sử dụng :

- ✓ *Hiển thị danh sách đề xuất khi người dùng bắt đầu nhập văn bản.*
- ✓ *Khi người dùng bắt đầu nhập, một menu thả xuống sẽ ở đó dựa trên các ký tự đã nhập*
- ✓ *id: bắt đầu bằng tiền tố auto , ví dụ : autoTinhThanh*



```

<EditText
    android:id="@+id/edtTen"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Nhập tên của bạn: "
    android:inputType="textPersonName" />

<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:textSize="24dp"
    android:textColor="#20F"
    android:text="Select tỉnh thành: " />

<AutoCompleteTextView
    android:id="@+id/autoTinhThanh"
    android:layout_width="match_parent"
    android:layout_marginTop="5dp"
    android:layout_height="wrap_content"
    android:completionThreshold="1"
    android:hint="---Select Tỉnh Thành" />
  
```

Thuộc tính đề xuất khi nhập x ký tự

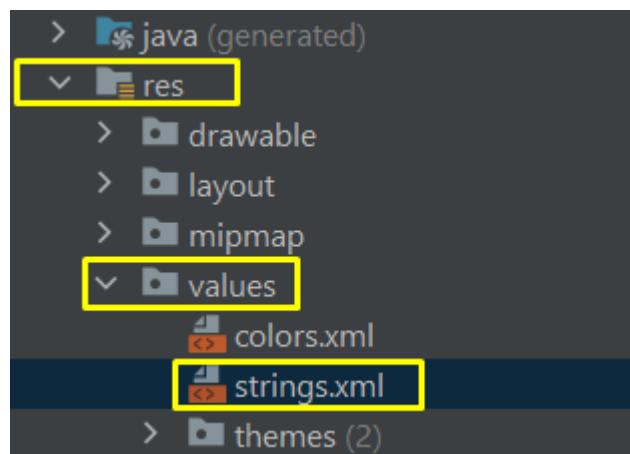
## 1

## AutoCompleteTextView

## □ 2. Cách xây dựng :

✓ *Giống với listview , AutoCompleteTextView cũng cần có nguồn cấp và adapter :*

## □ 2.1 . Step 1 : Tạo nguồn cấp dữ liệu



```
<string-array name="tinhthanh">
    <item>An Giang</item>
    <item>Bà rịa - Vũng tàu</item>
    <item>Bắc Giang</item>
    <item>Bắc Kạn</item>
    <item>Bạc Liêu</item>
    <item>Bắc Ninh</item>
    <item>Bến Tre</item>
    <item>Bình Định</item>
```



## 1

## AutoCompleteTextView

## □ 2.2 . Step 2 : Code trong MainActivity.kt

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        //khai báo list danh sách tinh thành  
        val list = resources.getStringArray(R.array.tinhthanh)  
  
        val adt = ArrayAdapter(context: this,  
            android.R.layout.simple_list_item_1,  
            list)  
        autoTinhThanh.setAdapter(adt)  
    }  
}
```



## 1

## AutoCompleteTextView

## □ 3 . Set đề xuất khi vừa click chuột vào AutoCompleteTextView

```
<AutoCompleteTextView  
    android:id="@+id/autoTinhThanh"  
    android:layout_width="match_parent"  
    android:layout_marginTop="5dp"  
    android:layout_height="wrap_content"  
    android:completionThreshold="1"  
    android:hint="---Select Tỉnh Thành" />
```

Nếu bạn đặt thông số =0 nó vẫn sẽ  
ngầm đẩy lên 1 ký tự mới để xuất

Select tỉnh thành:

---Select Tỉnh Thành

An Giang

Bà rịa – Vũng tàu

Bắc Giang

Bắc Kạn

```
//Gợi ý với 0 ký tự nhập vào, chỉ cần click chuột là xuất hiện gợi ý  
autoTinhThanh.setOnFocusChangeListener(View.OnFocusChangeListener {  
    view, b -> if (b) autoTinhThanh.showDropDown() })  
//View: Chế độ xem có trạng thái đã thay đổi  
///b: boolean true nếu đúng là nó có thay đổi
```



## 1

## AutoCompleteTextView

## □ 4 . setOnItemClickListener : Lắng nghe item nào được click chọn

```
//lắng nghe xem item nào được chọn
autoTinhThanh.setOnItemClickListener(AdapterView.OnItemClickListener { adapterView, view, i, l ->
    //lưu ý i: vị trí của item trong list mà dropview nó xổ ra
    Toast.makeText(context, this,
        text: "click item "+list[i], //không dùng i trong trường hợp này
        Toast.LENGTH_LONG).show()
})

autoTinhThanh.setOnItemClickListener(AdapterView.OnItemClickListener { adapterView, view, i, l ->
    Toast.makeText(context, this,
        text: "click item "+autoTinhThanh.text.toString(),
        Toast.LENGTH_LONG).show()
})
```





## Android app with kotlin

### Bài 11

#### GridView kotlin

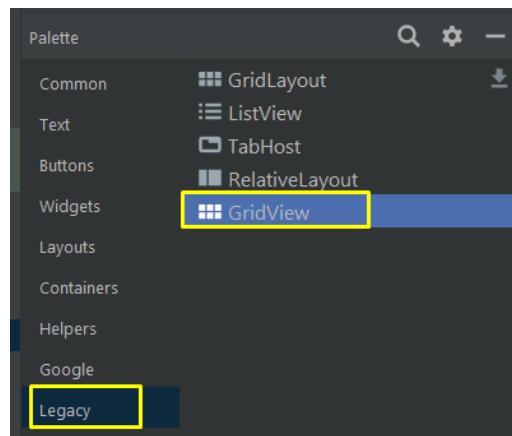


## 1

## GridView kotlin

## □ 1. GridView:

✓ *id : bắt đầu bằng tiền tố gv Ví dụ : gvPhim*



```
<GridView  
    android:id="@+id/gvPhim"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:numColumns="2" />
```

**numColumns : Chia bao nhiêu cột**

Item 1	Item 2
Sub Item 1	Sub Item 2
Item 3	Item 4
Sub Item 3	Sub Item 4
Item 5	Item 6
Sub Item 5	Sub Item 6
Item 7	Item 8
Sub Item 7	Sub Item 8
Item 9	Item 10
Sub Item 9	Sub Item 10
Item 11	Item 12
Sub Item 11	Sub Item 12
Item 13	Item 14
Sub Item 13	Sub Item 14
Item 15	Item 16
Sub Item 15	Sub Item 16
Item 17	Item 18
Sub Item 17	Sub Item 18
Item 19	Item 20
Sub Item 19	Sub Item 20
Item 21	Item 22
Sub Item 21	Sub Item 22



1

## GridView kotlin

### □ 2. Các bước xây dựng GridView:

Step 1:  
Make xml layout  
CustomLayout

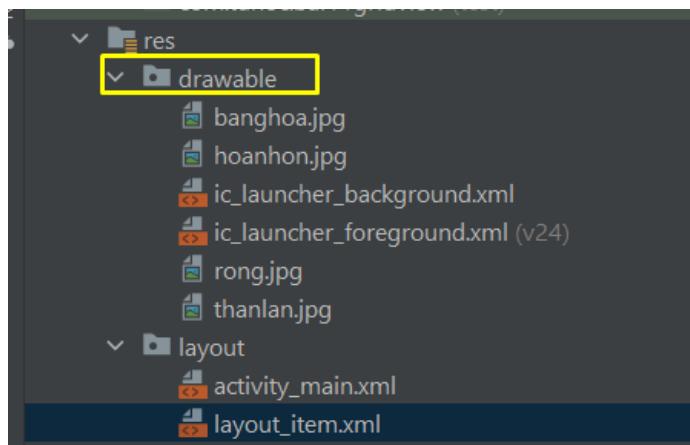
Step 2:  
Make class  
Class OutData

Step 3:  
Make Class  
CustomGridView



1

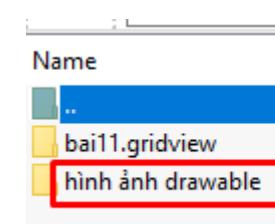
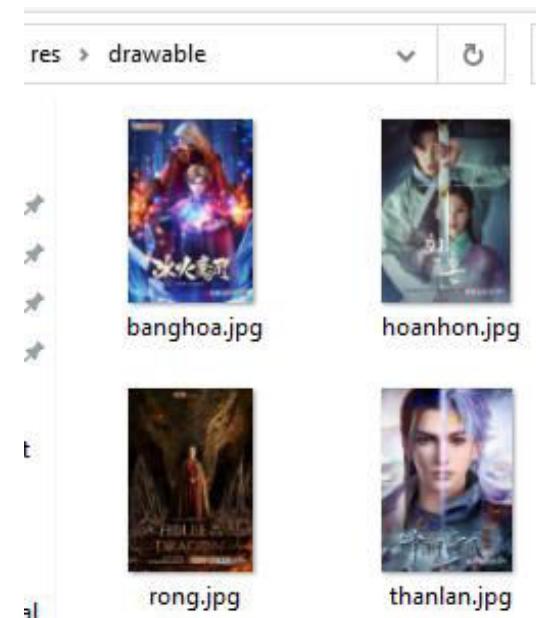
## GridView kotlin

 3. Step 1 - tạo xml layout file tùy chỉnh – Đưa hình ảnh vào drawable:

Ảnh để trong drawable  
và code toàn bộ bài 11

<http://android.tuhoc.cc/>

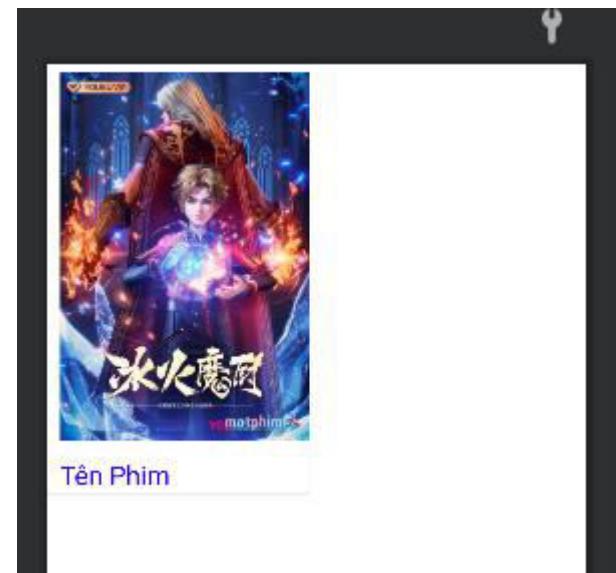
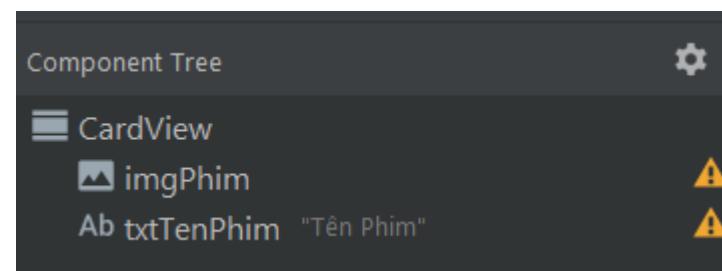
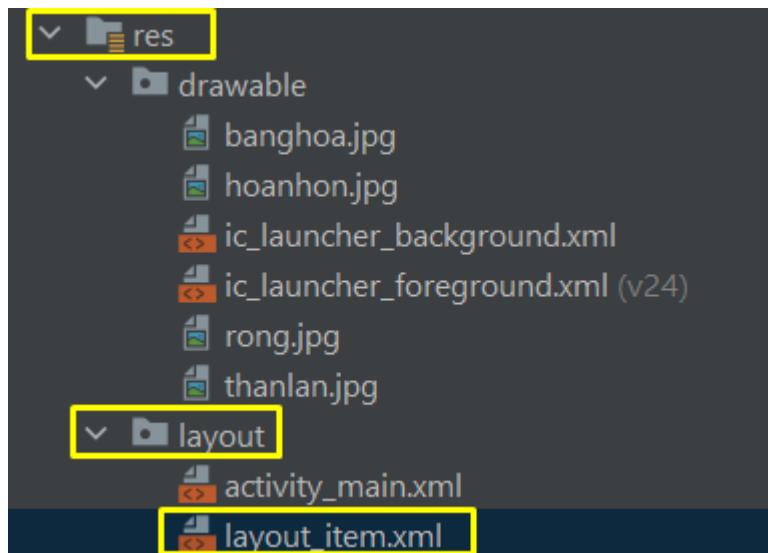
[11.bai11-gridview-tuhoc.cc.zip](#)



1

## GridView kotlin

## □ 3. Step 1: tạo xml layout file tuỳ chỉnh :



## 1

## GridView kotlin

## □ 3. Step 1: tạo xml layout file tuỳ chỉnh (code trong xml):

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="wrap_content"
    android:padding="2dp"
    android:layout_height="wrap_content">

    <ImageView
        android:id="@+id/imgPhim"
        android:layout_width="190dp"
        android:layout_height="283dp"
        android:padding="5dp"
        app:srcCompat="@drawable/banghoa" />

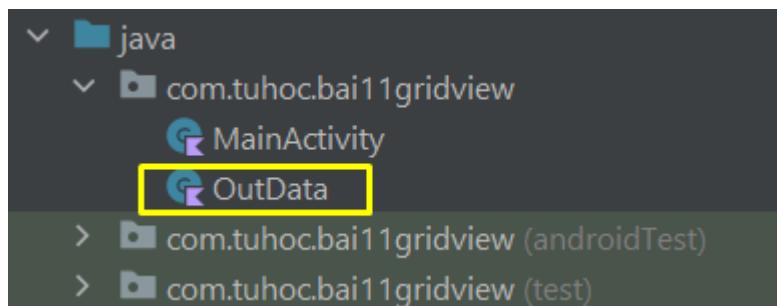
    <TextView
        android:id="@+id/txtTenPhim"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="283dp"
        android:textSize="20sp"
        android:layout_marginStart="10dp"
        android:textColor="#20f"
        android:text="Tên Phim" />
</androidx.cardview.widget.CardView>
```



1

## GridView kotlin

- 4. Step 2: tạo Class OutData lưu trữ các biến cấu trúc của 1 dòng :



```
class OutData(  
    val images:Int,  
    val tenphim:String  
) {  
}
```

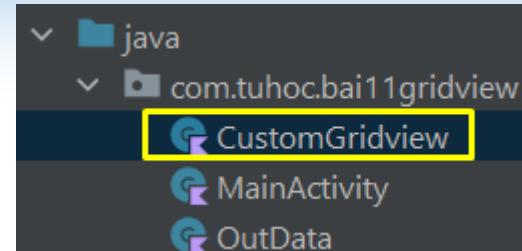


1

## GridView kotlin

## □ 5. Step 3: tạo Class CustomGridView:

```
class CustomGridView (val activity: Activity, val list: List<OutData>) : ArrayAdapter<OutData>(activity, R.layout.layout_item){  
    override fun getCount(): Int {  
        return list.size  
    }  
    @SuppressLint("ViewHolder")  
    override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {  
        val contexts = activity.layoutInflater  
        //layoutInflater giúp chuyển đổi xml layout thành view trong android  
        val rowView = contexts.inflate(R.layout.layout_item, parent, attachToRoot: false)  
        val images = rowView.findViewById<ImageView>(R.id.imgPhim)  
        val txtTenPhim = rowView.findViewById<TextView>(R.id.txtTenPhim)  
  
        images.setImageResource(list[position].images)  
        txtTenPhim.text = list[position].tenphim  
        return rowView  
    }  
}
```



1

## GridView kotlin

## □ 6. Code Mainactivity:

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        //khai báo list danh sách các bộ phim
        var list = mutableListOf<OutData>()
        list.add(OutData(R.drawable.banghoa, tenphim: "Băng Hoả Ma Trù"))
        list.add(OutData(R.drawable.hoanhon, tenphim: "Hoàn Hồn 2022"))
        list.add(OutData(R.drawable.rong, tenphim: "Gia Tộc Rồng"))
        list.add(OutData(R.drawable.thanlan, tenphim: "Thần Lan Chi Mộng"))

        val customGV = CustomGridview( activity: this, list)
        gvPhim.adapter=customGV
    }
}
```



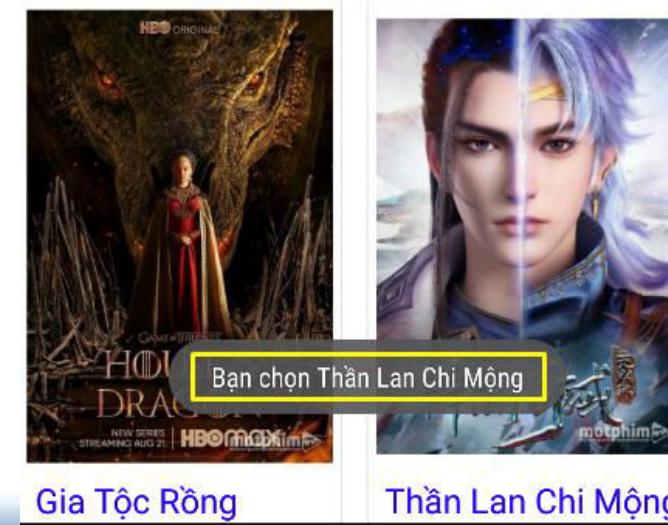
## 1

## GridView kotlin

## ❑ 4. GridView. onItemClickListener :

✓ *Lắng nghe xem item nào của gridView được click chọn*

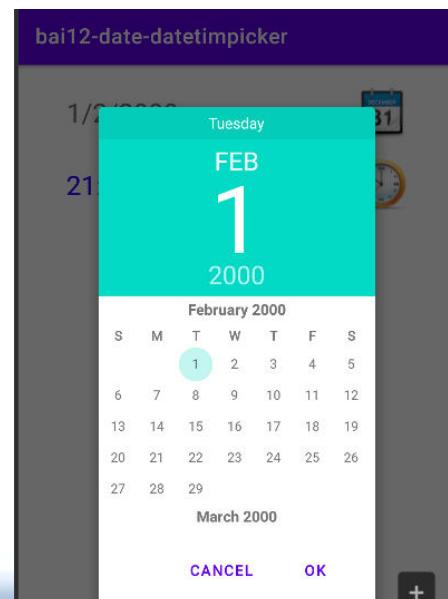
```
//Lắng nghe xem item nào được chọn trên gridView
gvPhim.onItemClickListener = AdapterView.OnItemClickListener { adapterView, view, i, l -
    Toast.makeText(context: this@MainActivity,
        text: "Bạn chọn "+list[i].tenphim,
        Toast.LENGTH_SHORT).show() }
```





## Android app with kotlin

### Bài 12



1

DatePicker

2

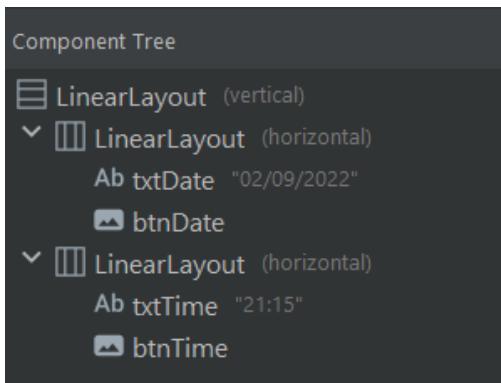
TimePicker



## 1

## DatePicker

## □ 1. Thiết kế giao diện :



Ảnh để trong drawable  
và code toàn bộ bài 12

<http://android.tuhoc.cc/>

12.bai12-datetime-picker kotlin-tuhoc.cc.zip

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="20dp"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/txtDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="25sp"
        android:layout_gravity="center"
        android:paddingStart="25dp"
        android:text="02/09/2022" />

    <ImageButton
        android:id="@+id/btnDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/white"
        android:layout_marginRight="25dp"
        app:srcCompat="@drawable/date" />

</LinearLayout>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/txtTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="25sp"
        android:layout_gravity="center"
        android:textColor="#20F"
        android:paddingStart="45dp"
        android:text="21:15" />

    <ImageButton
        android:id="@+id/btnTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/white"
        android:layout_marginRight="45dp"
        app:srcCompat="@drawable/time" />

```

1

## TimePicker

## □ 2. Code MainActivity.kt :

```
class MainActivity : AppCompatActivity() {
    //khai báo biến cal get thời gian hiện tại
    //xem lại bài 17 phần android cơ bản http://kotlin.tuhoc.cc
    val today = Calendar.getInstance()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        //1. set lại thời gian khi click vào button time
        val startHour = today.get(Calendar.HOUR_OF_DAY)
        val startMinute = today.get(Calendar.MINUTE)
        //lắng nghe click lên button
        btnTime.setOnClickListener { it: View!
            //Timepicker
            TimePickerDialog(context: this, TimePickerDialog.OnTimeSetListener { timePicker, i, i2 -
                //i: Giờ mà người dùng chọn
                //i2: Phút mà người dùng chọn
                txtTime.setText("$i:$i2")
            }, startHour, startMinute, is24HourView: true).show()
        }
    }
}
```

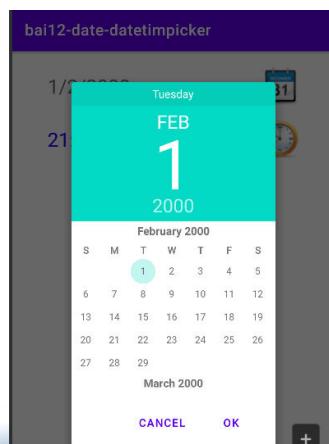


## 2

## DatePicker

## □ 2. Code MainActivity.kt :

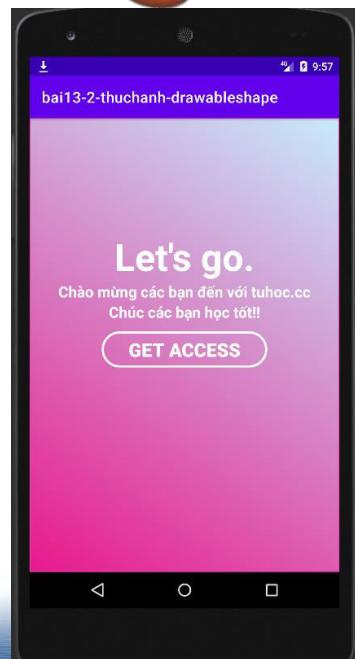
```
//1. set lại thời gian khi click vào button date
//lắng nghe click lên button
btnDate.setOnClickListener { it: View! 
    //Datepicker
    DatePickerDialog(context: this, DatePickerDialog.OnDateSetListener { datePicker, i, i2, i3 ->
        //i: năm
        //i2: Tháng
        //i3: ngày
        txtDate.setText("$i3/${i2+1}/$i")
        year: 2000, month: 1, dayOfMonth: 1).show()
    }
}
```





## Android app with kotlin Bài 13

Make beauty button,  
texview...



0. BUTTON BASIC

1. BUTTON TUHOC.CC

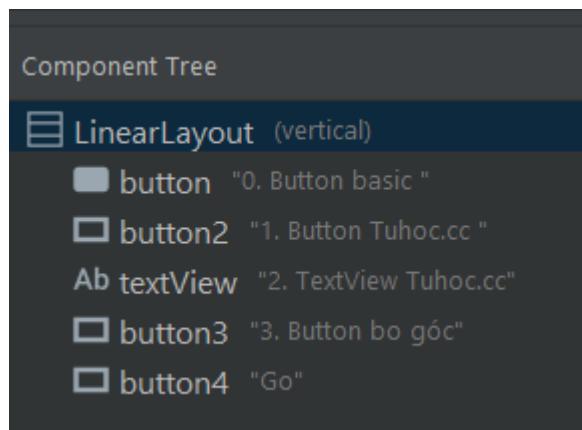
2. TextView Tuhoc.cc

3. BUTTON BO GÓC



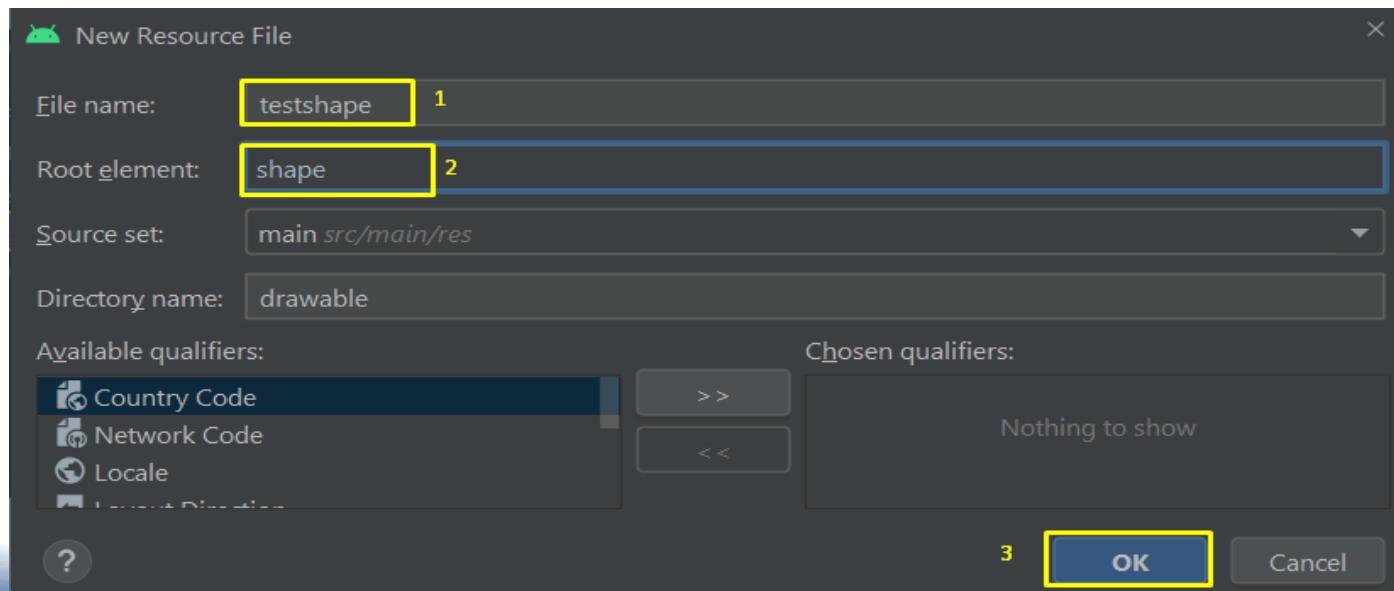
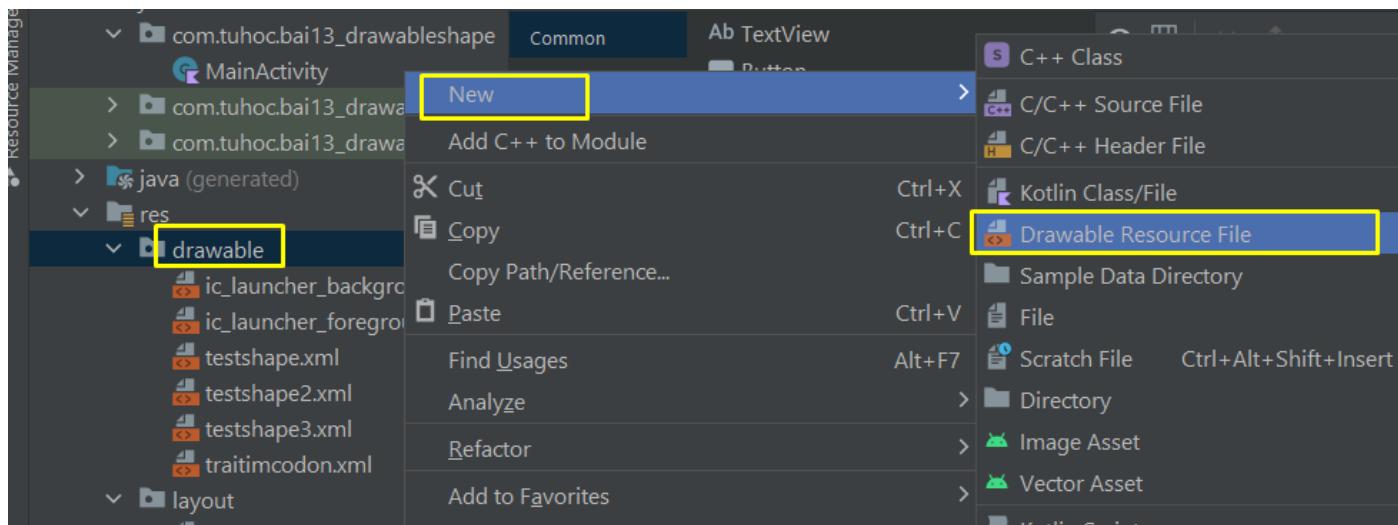
1

## Thiết kế giao diện



2

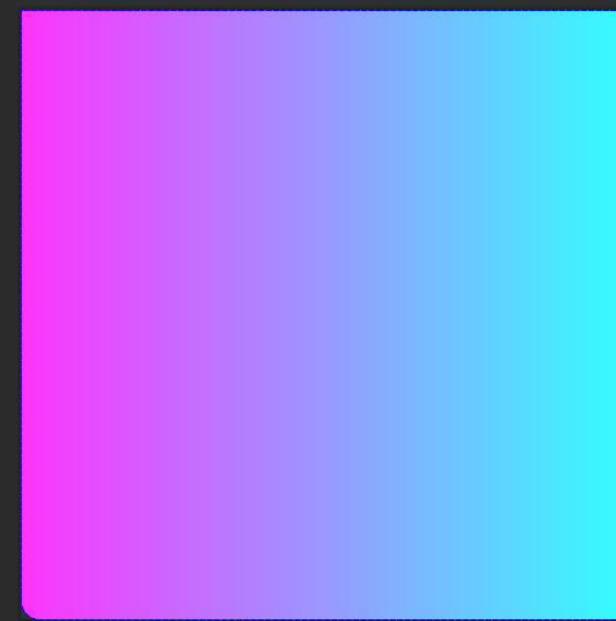
## Tạo shape xml



## 2

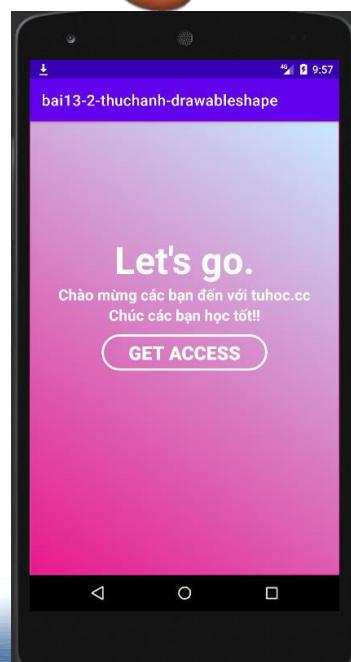
## Tạo shape xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle">
    <!-- ctrl + shift + / để bình luân trong xml --&gt;
    <!-- 1. kích cỡ hình chữ nhật --&gt;
    &lt;!--&lt;size android:height="100dp" android:width="100dp"/&gt;--&gt;
    <!-- 2. solid: màu nền khối shape --&gt;
    &lt;solid android:color="#88FFFFFF"/&gt;
    <!-- 3. strok : tô viền --&gt;
    &lt;stroke android:color="#000000"
        android:width="1dp"
        android:dashGap="1dp"
        android:dashWidth="5dp"/&gt;
    <!-- bo góc
        radius : bo tất cả các góc
        bottomLeftRadius : bo góc dưới, trái
        bottomRightRadius : bo góc dưới, phải
        topRightRadius : bo góc trên, bên phải
        topLeftRadius : bo góc trên, bên trái
    --&gt;
    &lt;corners android:bottomLeftRadius="15dp"
        android:topRightRadius="15dp"/&gt;
    <!-- padding căn lề nội dung nằm trên shape
        ai quên xem lại bài 3.2 =)) --&gt;
    &lt;padding android:bottom="10dp"
        android:left="10dp"
        android:right="10dp"
        android:top="10dp"/&gt;
    <!-- dài mầu chạy gradien <a href="https://webqradient.com/">https://webqradient.com/ -->
    <gradient android:startColor="#FF33FF"
        android:endColor="#33FFFFFF"/>
</shape>
```





## Android app with kotlin Bài 13.2

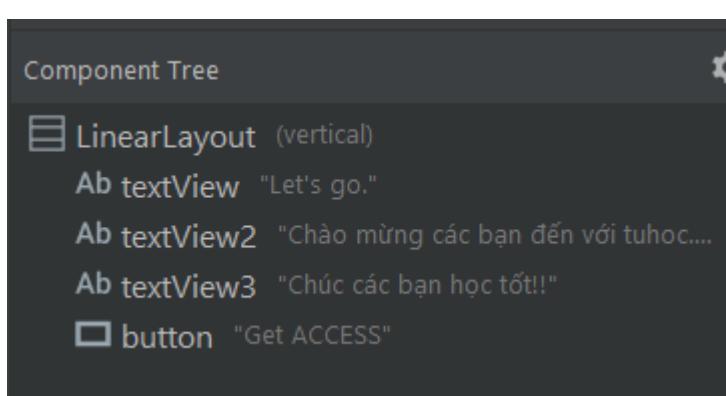


# Thực hành thiết kế giao diện với shape xml



## 1

## Thiết kế giao diện

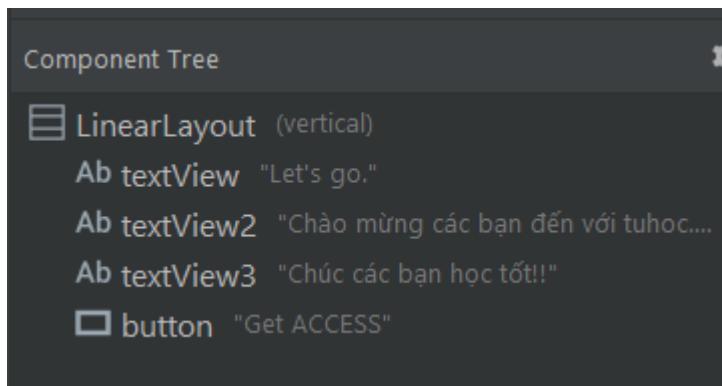


```
<TextView  
    android:id="@+id/textView"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:gravity="center"  
    android:layout_marginTop="150dp"  
    android:textSize="50sp"  
    android:textStyle="bold"  
    android:textColor="@color/white"  
    android:text="Let's go." />
```

```
<TextView  
    android:id="@+id/textView2"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textStyle="bold"  
    android:textColor="@color/white"  
    android:textSize="20sp"  
    android:gravity="center"  
    android:text="Chào mừng các bạn đến với tuhoc.cc" />
```

1

## Thiết kế giao diện

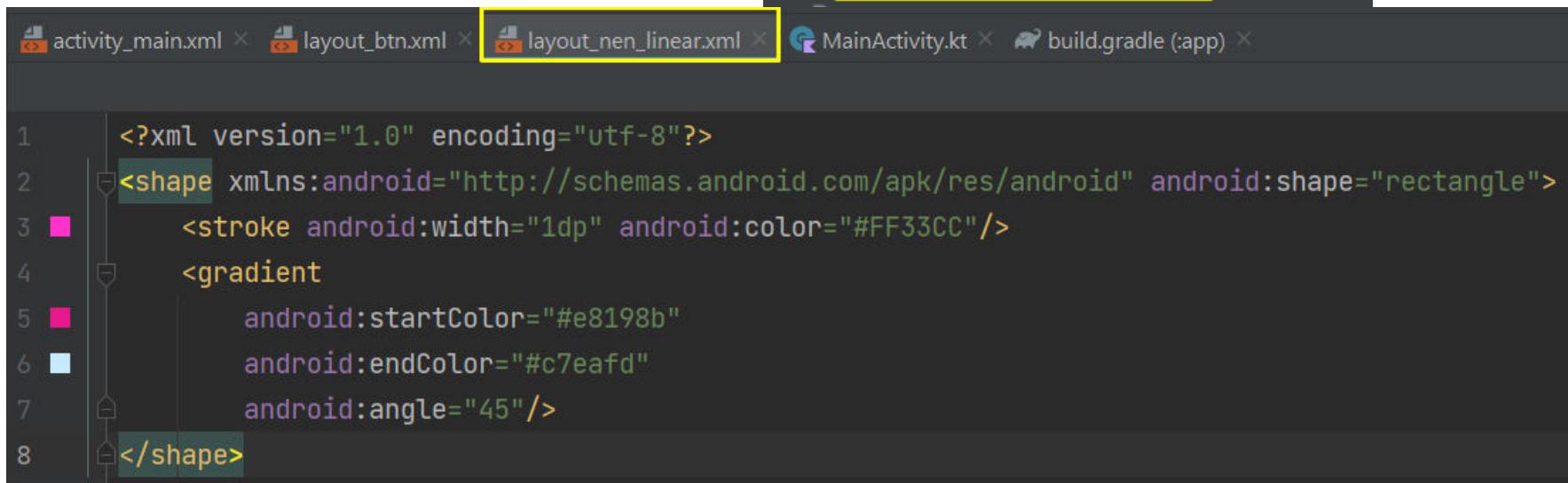
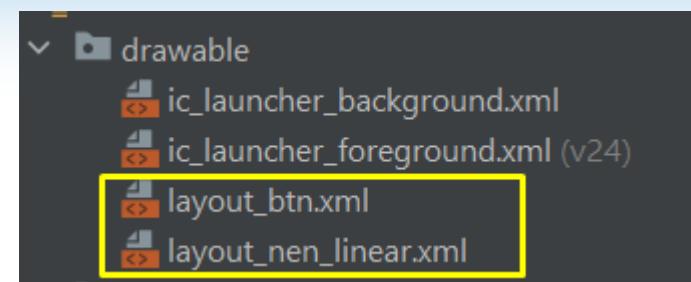


```
<TextView  
    android:id="@+id/textView3"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textStyle="bold"  
    android:textColor="@color/white"  
    android:textSize="20sp"  
    android:gravity="center"  
    android:text="Chúc các bạn học tốt!!" />
```

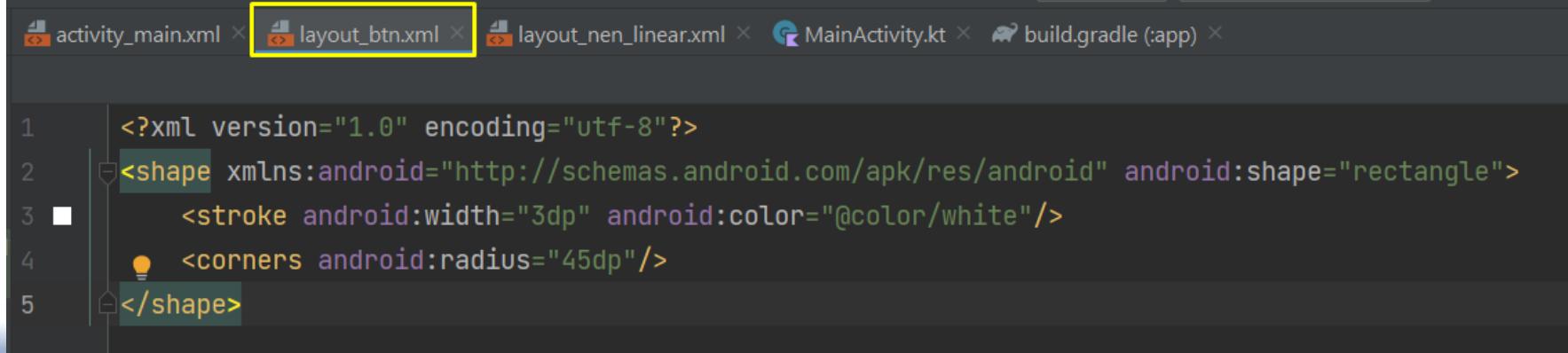
```
<androidx.appcompat.widget.AppCompatButton  
    android:id="@+id/button"  
    android:background="@drawable/layout_btn"  
    android:layout_width="220dp"  
    android:layout_gravity="center"  
    android:layout_marginTop="12dp"  
    android:textColor="@color/white"  
    android:textStyle="bold"  
    android:textSize="25sp"  
    android:layout_height="wrap_content"  
    android:text="Get ACCESS" />
```

## 2

## Tạo shape xml



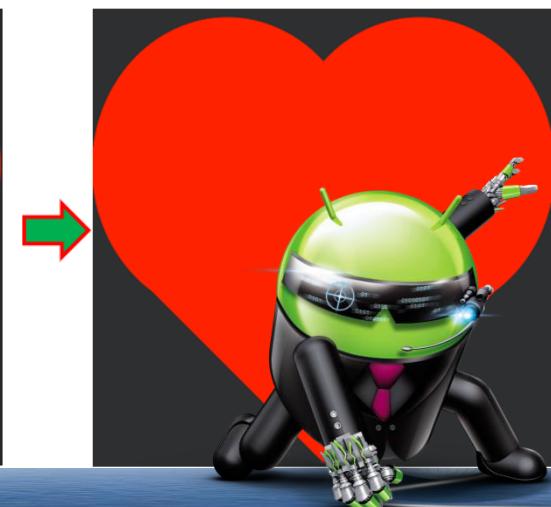
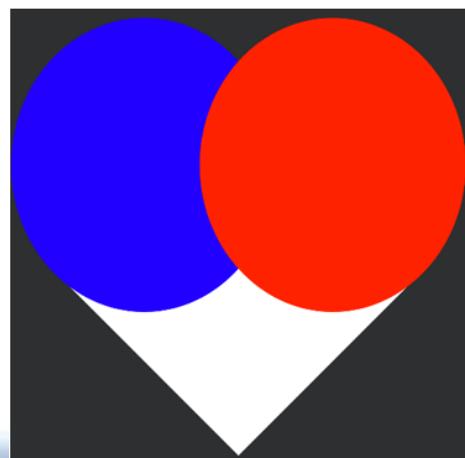
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle">
3     <stroke android:width="1dp" android:color="#FF33CC"/>
4     <gradient
5         android:startColor="#e8198b"
6         android:endColor="#c7eaf7"
7         android:angle="45"/>
8 </shape>
```



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle">
3     <stroke android:width="3dp" android:color="@color/white"/>
4     <corners android:radius="45dp"/>
5 </shape>
```



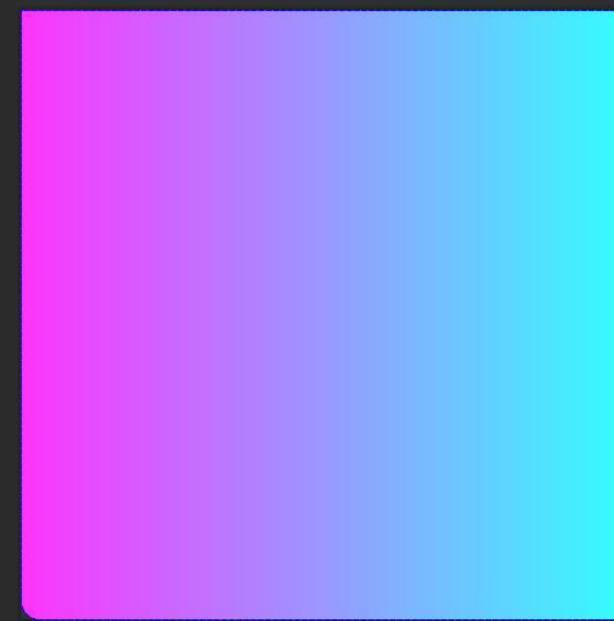
## Layer List kotlin



## 1

## Nhắc lại shape xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle">
    <!-- ctrl + shift + / để bình luân trong xml -->
    <!--1. kích cỡ hình chữ nhật -->
    <!--<size android:height="100dp" android:width="100dp"/>-->
    <!--2. solid: màu nền khối shape -->
    <solid android:color="#88FFFFFF"/>
    <!--3. strok : tô viền -->
    <stroke android:color="#0000FF"
        android:width="1dp"
        android:dashGap="1dp"
        android:dashWidth="5dp"/>
    <!--bo góc
    radius : bo tất cả các góc
    bottomLeftRadius : bo góc dưới, trái
    bottomRightRadius : bo góc dưới, phải
    topRightRadius : bo góc trên, bên phải
    topLeftRadius : bo góc trên, bên trái
    -->
    <corners android:bottomLeftRadius="15dp"
        android:topRightRadius="15dp"/>
    <!--padding căn lề nội dung nằm trên shape
    ai quên xem lại bài 3.2 =)) -->
    <padding android:bottom="10dp"
        android:left="10dp"
        android:right="10dp"
        android:top="10dp"/>
    <!--dài mầu chạy gradien https://webqradient.com/ -->
    <gradient android:startColor="#FF33FF"
        android:endColor="#33FFFFFF"/>
</shape>
```

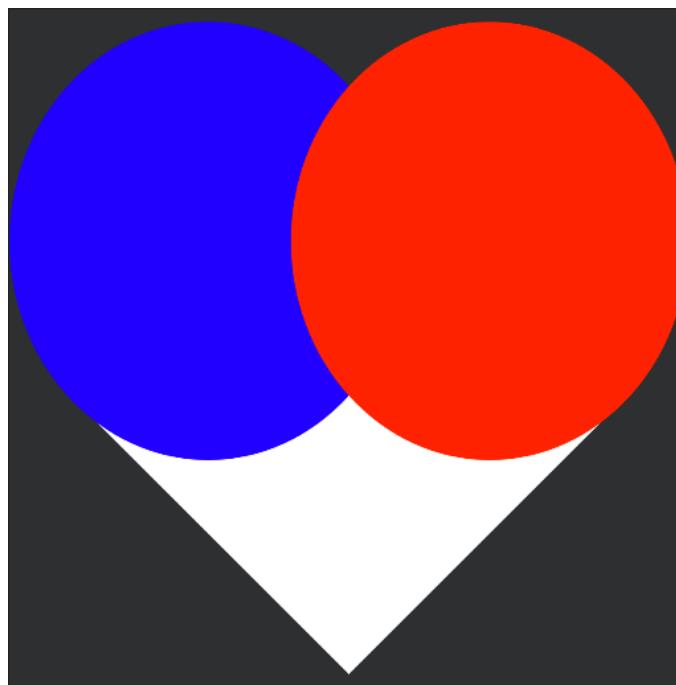


## 2

## Layer List

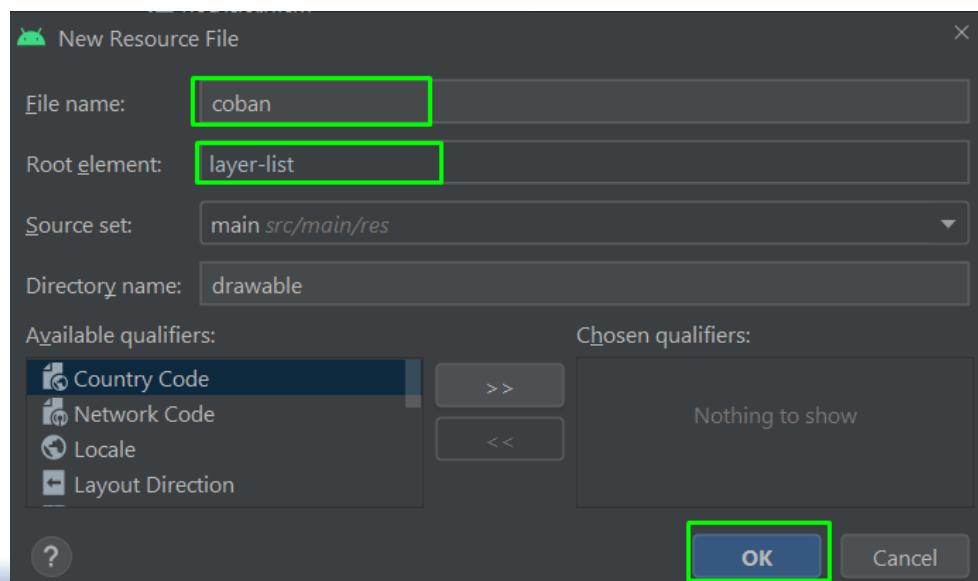
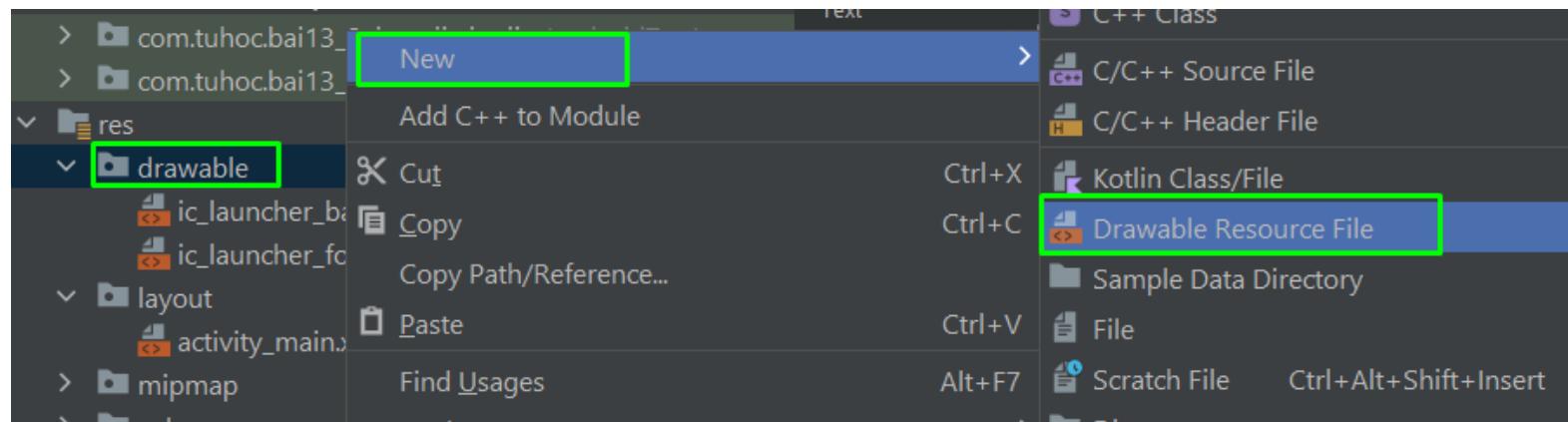
 1. Khái niệm

*Layer List* được xem như là một phần mở rộng hơn của Shape XML, bằng cách kết hợp nhiều hình khối đơn giản đó lại với nhau để tạo thành một khối phức tạp.



2

## Layer List

 1. Create new Layer List

## 2

## Layer List

 2.1 Cấu trúc của Layer List

Mỗi một khối shape được liệt kê trong khoá **<item>** **</item>**

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list
    xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <!--Drawable của item-->
    </item>

    <item>
        <!--Drawable của item-->
    </item>
    <!-- ... -->
</layer-list>
```

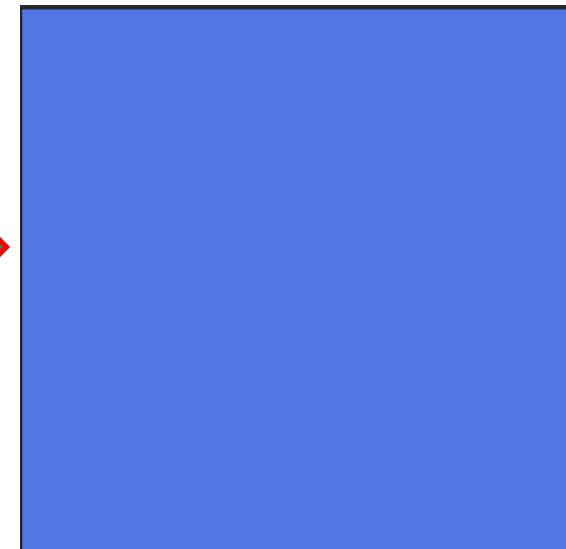
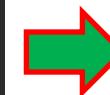


## 2

## Layer List

## □ 2.2 Example Layer List

```
<layer-list xmlns:android="http://schemas.android.  
    <!--khối item 1 -->  
    <item>  
        <shape android:shape="rectangle">  
            <solid android:color="#5275E3"/>  
        </shape>  
    </item>  
</layer-list>
```

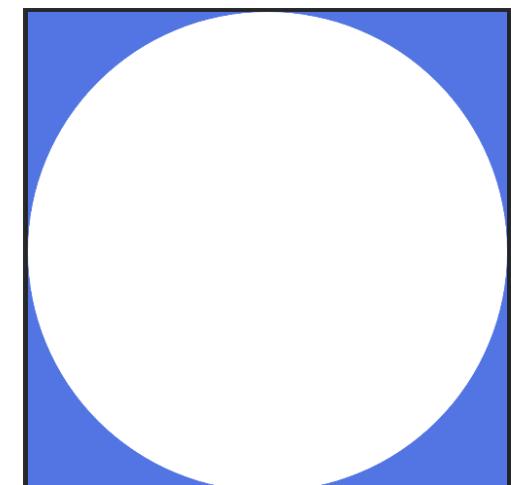
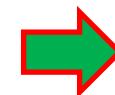


## 2

## Layer List

## □ 2.2 Example Layer List

```
<!--khối item 1 -->
<item>
    <shape android:shape="rectangle">
        <solid android:color="#5275E3"/>
    </shape>
</item>
<!--Khối item 2-->
<item>
    <shape android:shape="oval">
        <solid android:color="@color/white"/>
        <size android:height="100dp"
              android:width="100dp"/>
    </shape>
</item>
```

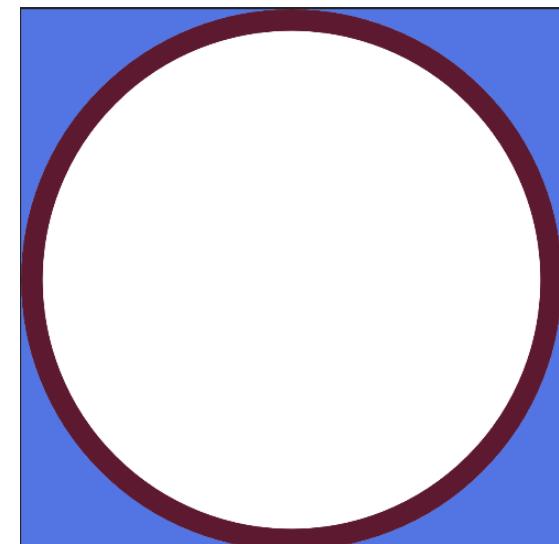
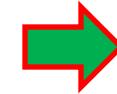


## 2

## Layer List

## □ 2.2 Example Layer List

```
<!--Khối item 3-->
<item>
    <shape android:shape="oval">
        <stroke android:color="#5D192F"
               android:width="4dp"/>
    </shape>
</item>
```



## 2

## Layer List

3. Các thuộc tính cơ bản của <item> trong LayerList :



Nếu bạn quen thuộc với các ngôn ngữ như tiếng Ả Rập hoặc tiếng Urdu, bạn sẽ biết rằng chúng bắt đầu từ phải sang trái không giống như tiếng Anh mà chúng ta đọc từ Trái sang Phải.

Vì vậy, nếu chúng ta đặt lề bằng cách sử dụng Margin phải / trái thì chúng ta không quan tâm đến ngôn ngữ mà chúng ta chỉ trực tiếp thêm lề theo phẩy biến trái và phải.

Nhưng nếu chúng ta sử dụng Margin start / end thì chúng ta quan tâm đến ngôn ngữ. Vì vậy, giả sử nếu ứng dụng có giao diện người dùng được viết bằng ngôn ngữ Ả Rập thì phần bắt đầu sẽ ở bên phải và nếu giao diện người dùng bằng tiếng Anh thì phần bắt đầu sẽ ở bên trái.

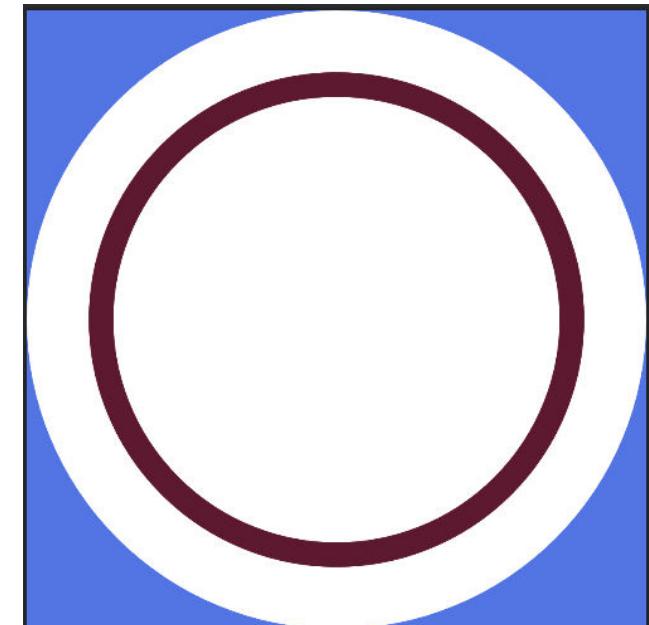
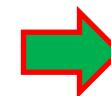


2

## Layer List

## □ 3. Các thuộc tính cơ bản của &lt;item&gt; trong LayerList :

```
<!--khối item 3-->
<item
    android:bottom="10dp"
    android:right="10dp"
    android:left="10dp"
    android:top="10dp">
    <shape android:shape="oval">
        <stroke
            android:color="#5D192F"
            android:width="4dp"/>
    </shape>
</item>
```

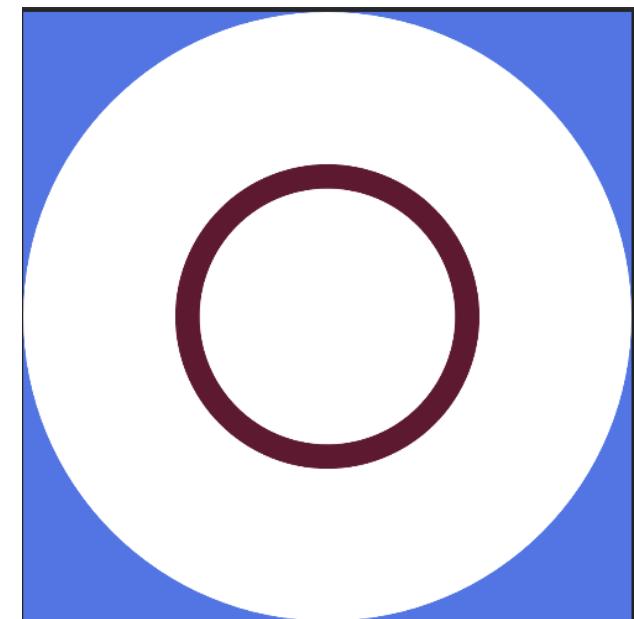
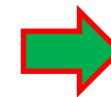


2

## Layer List

## □ 3. Các thuộc tính cơ bản của &lt;item&gt; trong LayerList :

```
<!-- khôi item 3 -->
<item
    android:gravity="center">
    <shape android:shape="oval">
        <size
            android:height="50dp"
            android:width="50dp"/>
        <stroke
            android:color="#5D192F"
            android:width="4dp"/>
    </shape>
</item>
```



2

## Layer List

 4. drawable Resource file :

*Check màu ảnh bất kỳ sử dụng file đính kèm*

Ảnh + check mã màu  
và code toàn bộ bài 13-3  
Pass unrar : **tuhoc1313**

<http://android.tuhoc.cc/>

13-3-layer-kotlin-androidstudio-tuhoc.cc.zip 

INFILE

-  bai13-3layerlistkotlin
-  hinhanh-bai13-3
-  check-mau-tuhoc.cc.exe

```
<item>
    <shape>
        <solid android:color="#FFEA16"/>
    </shape>
</item>
<item android:drawable="@drawable/hinhanh"/>
```



2

## Layer List

 4. drawable Resource file :

*Bo góc viền ngoài, ôn lại thuộc tính của shape xml*

```
<item>
    <shape>
        <solid android:color="#FFEA16"/>
        <corners android:radius="450dp"/>
    </shape>
</item>
<item android:drawable="@drawable/hinhanh"/>
```

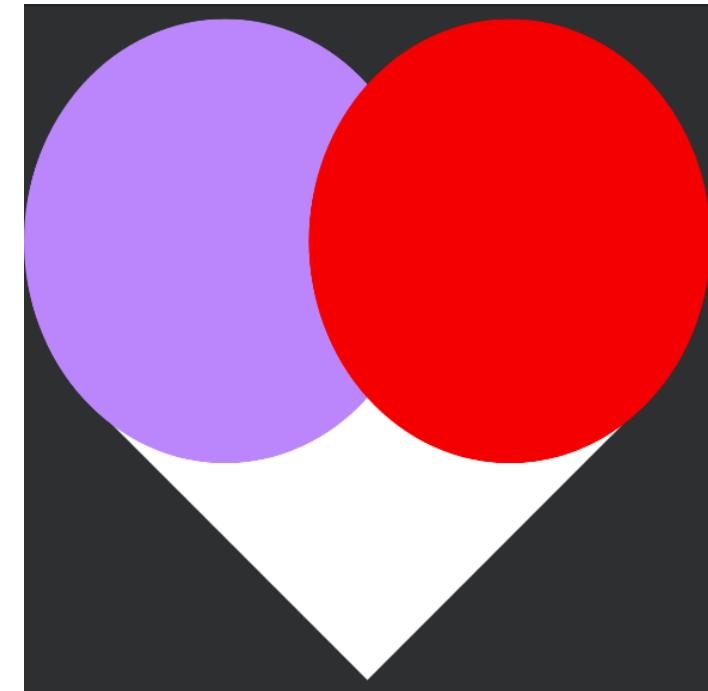


2

## Layer List

## □ 5. Giải thích Code hình trái tim :

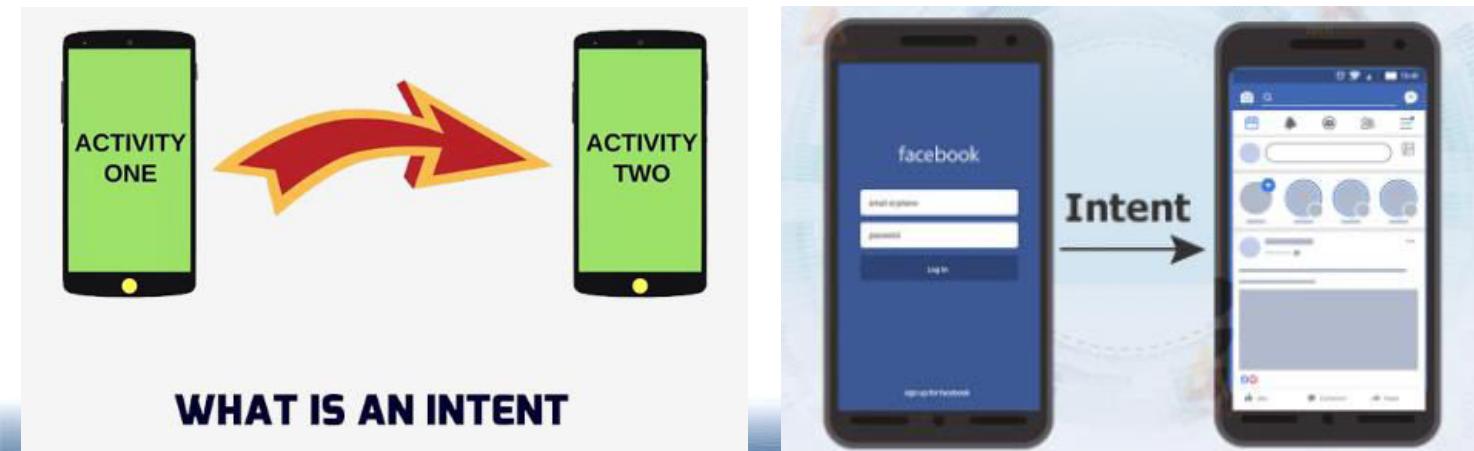
```
<item
    android:bottom="21dp"
    android:left="32dp"
    android:right="32dp"
    android:top="37dp">
    <rotate android:fromDegrees="45">
        <shape>
            <solid android:color="@color/white" />
            <size
                android:width="100dp"
                android:height="100dp" />
        </shape>
    </rotate>
</item>
<item
    android:bottom="52dp"
    android:right="68dp">
    <shape android:shape="oval">
        <solid android:color="@color/purple_200" />
    </shape>
</item>
<item
    android:bottom="52dp"
    android:left="68dp">
    <shape android:shape="oval">
        <solid android:color="#F40000" />
    </shape>
</item>
```





## Android app with kotlin Bài 14

### Intent kotlin



## 1

## Intent

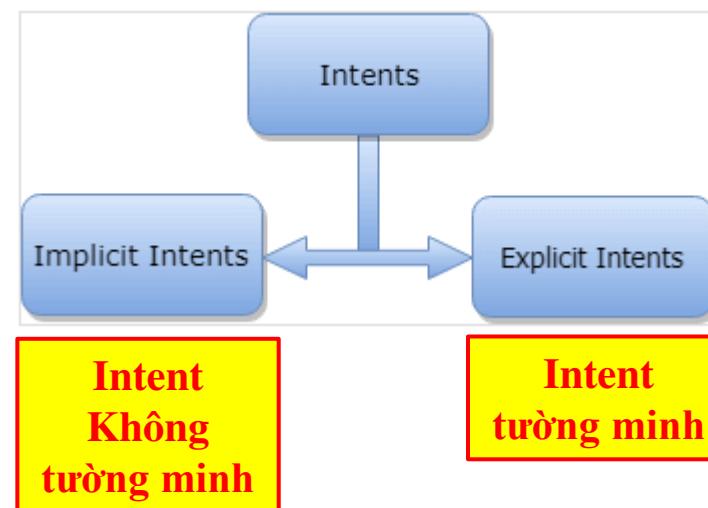
**❑1. Khái niệm Intent là gì?**

*Trong Android, Intent là những tin nhắn không đồng bộ cho phép các component của ứng dụng (như activities, services, broadcast receivers và content providers) yêu cầu chức năng từ một component khác.*

**❑2. Phân loại Intent:**

*Không tường minh vì nó không chỉ định tên của component sẽ bắt đầu*

*Ví dụ: sử dụng để mở 1 url dung trình duyệt của thiết bị, mở camera để chụp ảnh...*



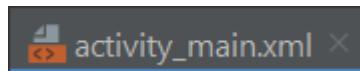
*Thường được sử dụng để khởi chạy các component trong cùng một ứng dụng.*



1

## Intent

## □ 3. Ví dụ Intent:



Component Tree

- LinearLayout (vertical)
  - btnGo "Go Màn Hình 2"

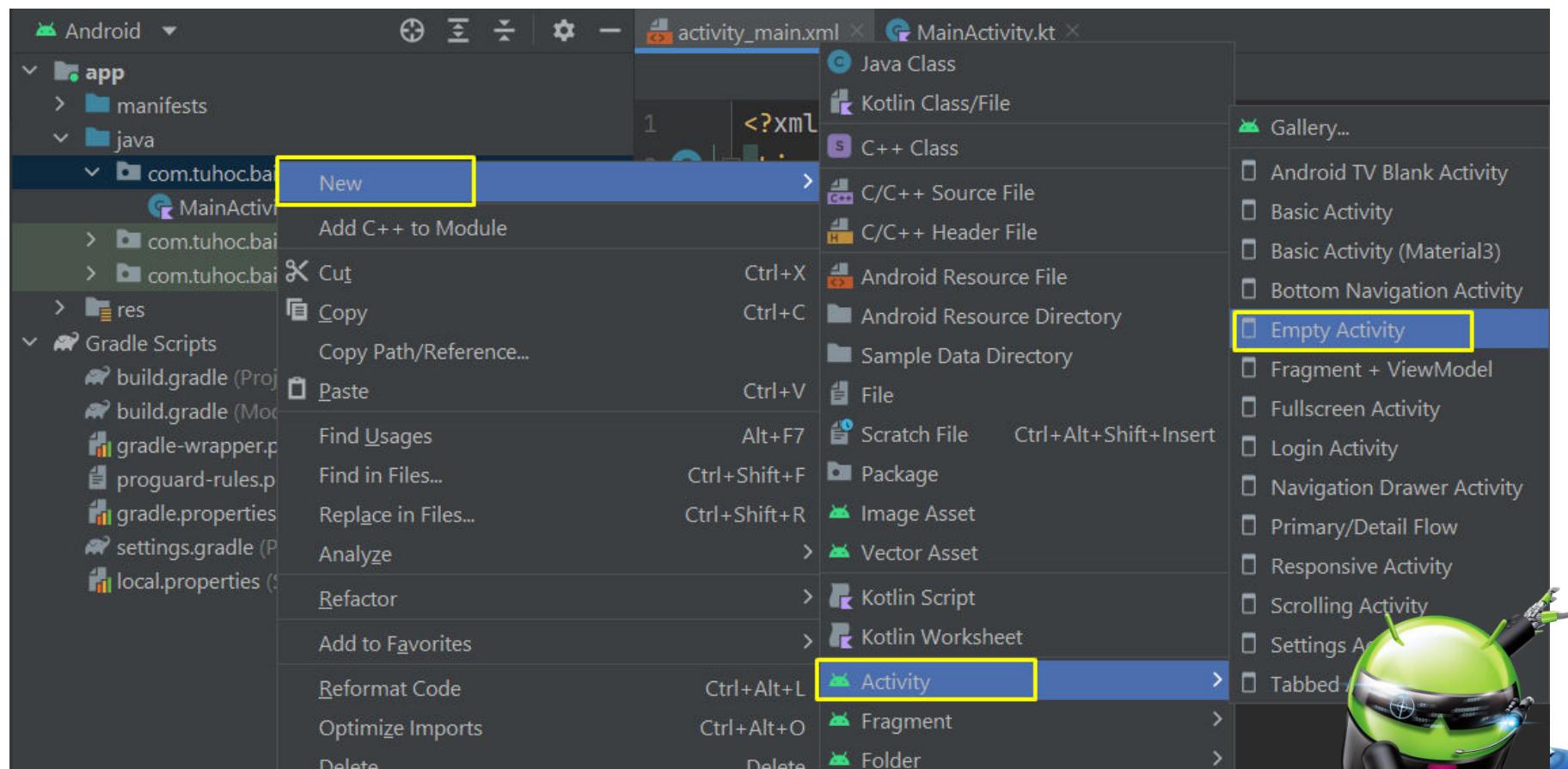
```
<Button  
    android:id="@+id	btnGo"  
    android:layout_width="200dp"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:layout_marginTop="15dp"  
    android:text="Go Màn Hình 2" />
```



1

## Intent

## ❑ 3. Ví dụ Intent:



## 1

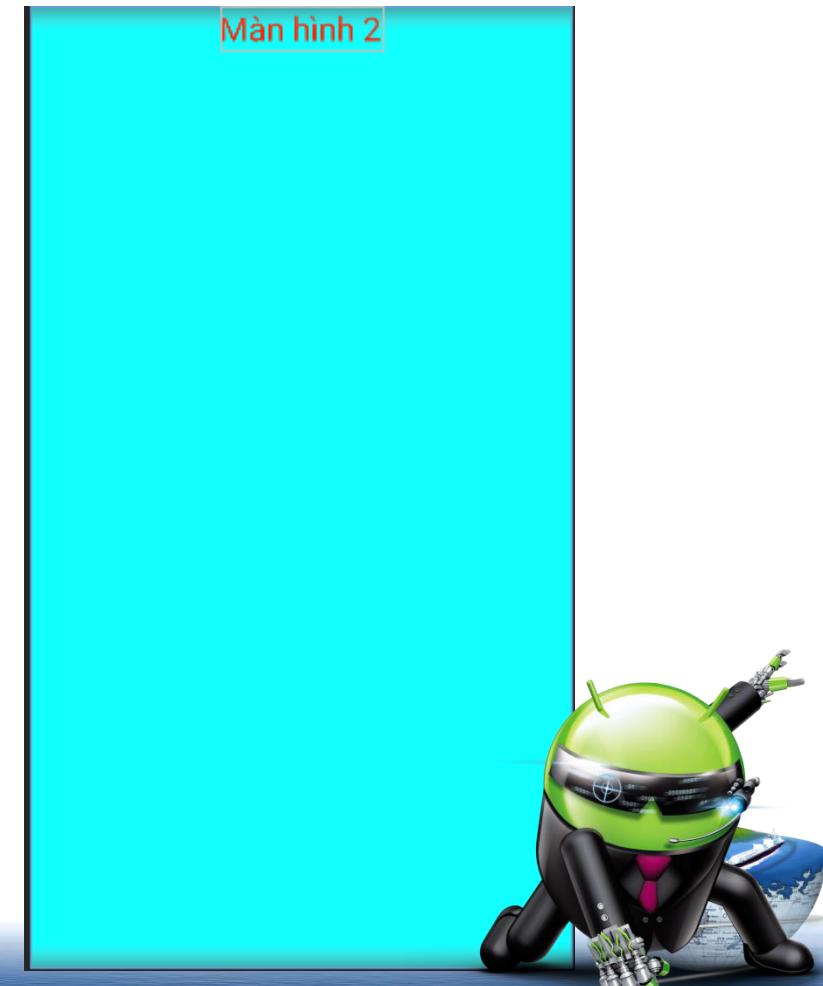
## Intent

 activity\_manhinh2.xml ×

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent"
    android:background="#12ffff"
    tools:context=".manhinh2">

    <TextView
        android:id="@+id/txtManHinh2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25dp"
        android:textColor="#f20"
        android:layout_gravity="center"
        android:text="Màn hình 2" />
</LinearLayout>
```

Màn hình 2



## 1

## Intent

**Implicit Intents : Sử dụng intent gọi màn hình activity khác**

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        //gọi màn hình 2 từ màn hình main
        btnGo.setOnClickListener { it: View!
            val intent1 = Intent(packageContext: this, manhinh2::class.java)
            startActivity(intent1)
        }
    }
}
```



## 1

## Intent

Explicit Intents : *Sử dụng intent gọi đến 1 url*

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        Toast.makeText(context, text: "onCreate", Toast.LENGTH_SHORT).show()
        //gọi màn hình 2 từ màn hình main
        btnGo.setOnClickListener { it: View!
            val intent1 = Intent(packageContext: this, manhinh2::class.java)
            startActivity(intent1)
        }
        //gọi url từ màn hình main
        btnGo4.setOnClickListener { it: View!
            val intent4 = Intent(Intent.ACTION_VIEW,
                Uri.parse(uriString: "http://tuhoc.cc"))
            startActivity(intent4)
        }
    }
}
```





## Android app with kotlin Bài 14.2

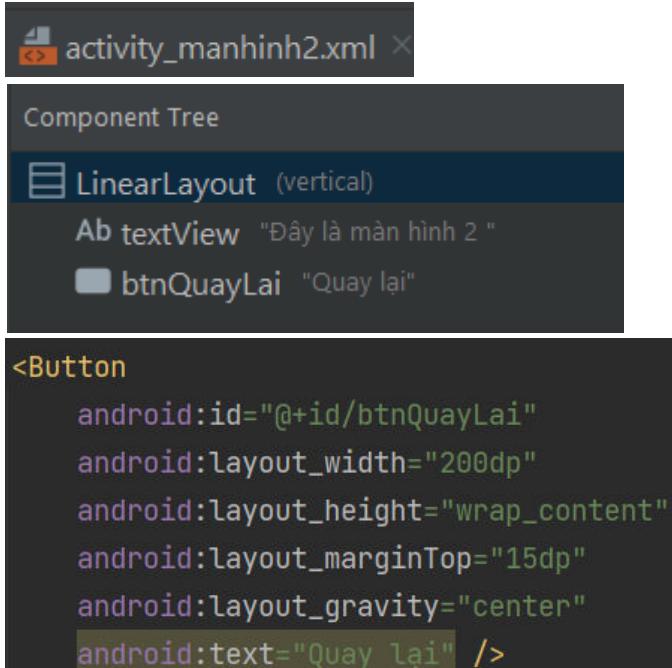
**Intent Bundle  
Truyền, nhận dữ liệu  
giữa các Activity**



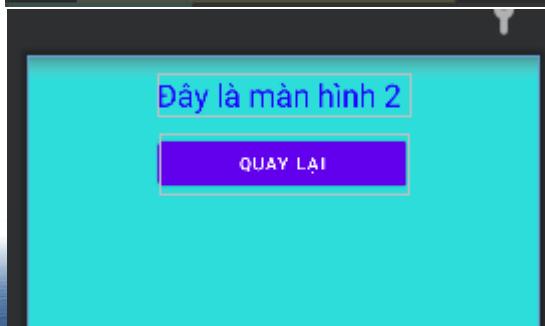
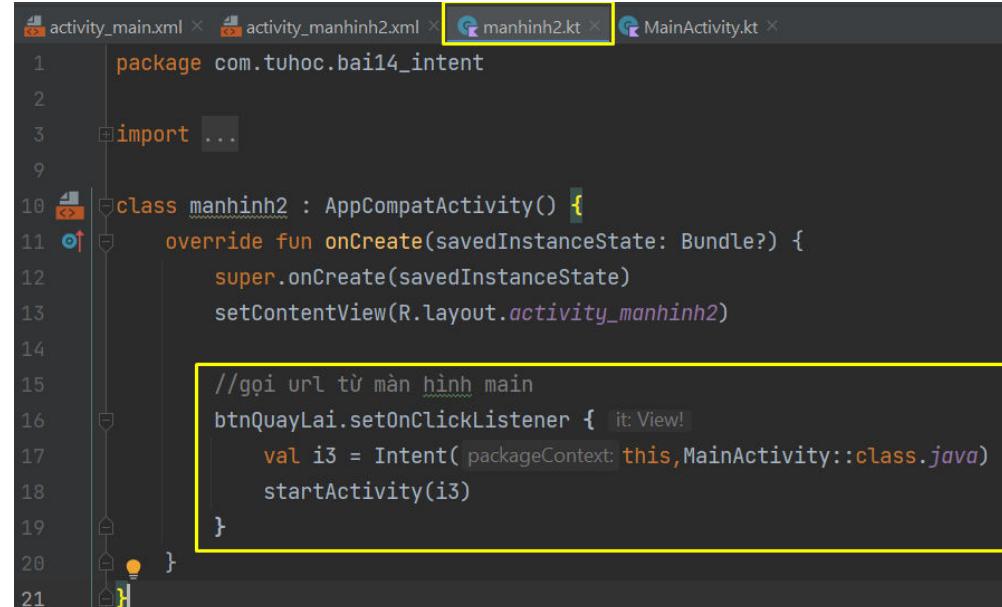
## 1

## Ôn tập Intent

*Ôn tập cùng cỗ lại bài cũ: Giả sử tại màn hình 2 ta muốn quay lại màn hình MainActivity => Thêm 1 button quay lại từ màn hình 2*



```
<Button
    android:id="@+id/btnQuayLai"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    android:layout_gravity="center"
    android:text="Quay lại" />
```

```
1 package com.tuhoc.bai14_intent
2
3 import ...
4
5 class manhinh2 : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         setContentView(R.layout.activity_manhinh2)
9
10        //gọi url từ màn hình main
11        btnQuayLai.setOnClickListener { it: View! ->
12            val i3 = Intent(packageContext, MainActivity::class.java)
13            startActivity(i3)
14        }
15    }
16
17
18
19
20
21 }
```

Code tại manhinh2.kt

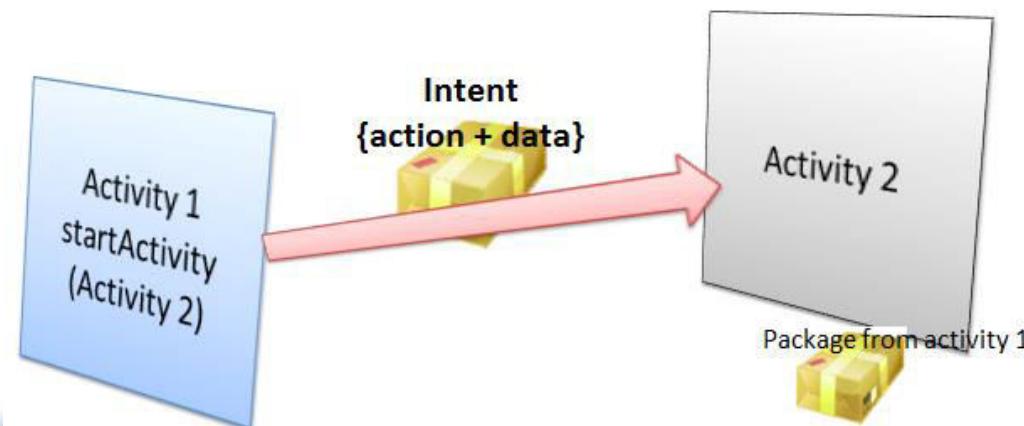


## 2

## Gửi dữ liệu qua các Activity

## □1. Các cách truyền dữ liệu qua Activity:

- *Dữ liệu được “nhét” vào trong Intent và được lấy ra khỏi Intent theo các cặp dữ liệu dạng key/value*
- *Điều này tương tự như khi bạn di chuyển đi chơi xa, thì khi đóng gói hành lý của bạn, nhân viên tiếp nhận hành lý phải dán nhãn tên của bạn hay ID của bạn lên hành lý, để đảm bảo bạn lấy đúng hành lý (chính là value) khi đến nơi dựa vào nhãn tên hay ID đó (chính là key )*
- **Có 2 Cách truyền dữ liệu qua Activity:**
  1. Dùng trực tiếp từ Intent (gửi đơn lẻ từng dữ liệu)
  2. Đóng gói vào Bundle ( đóng gói sau đó gửi 1 lần )

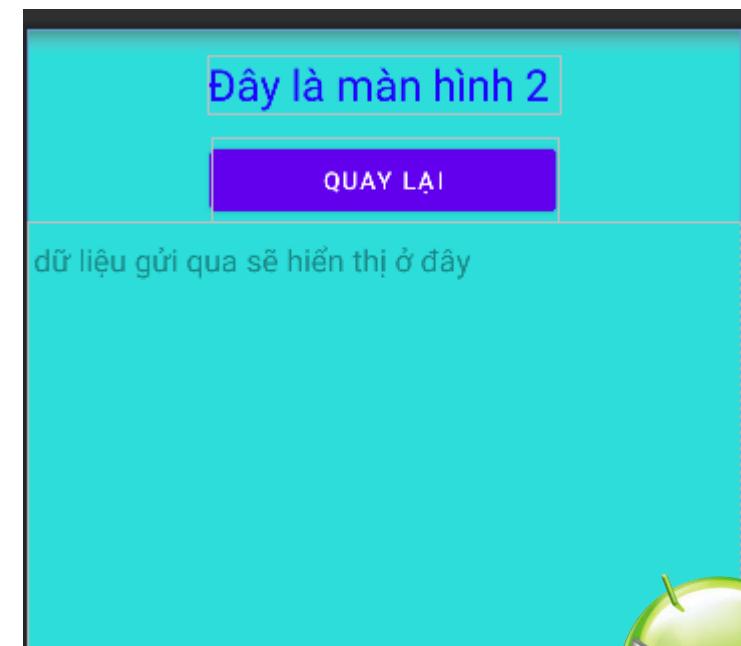
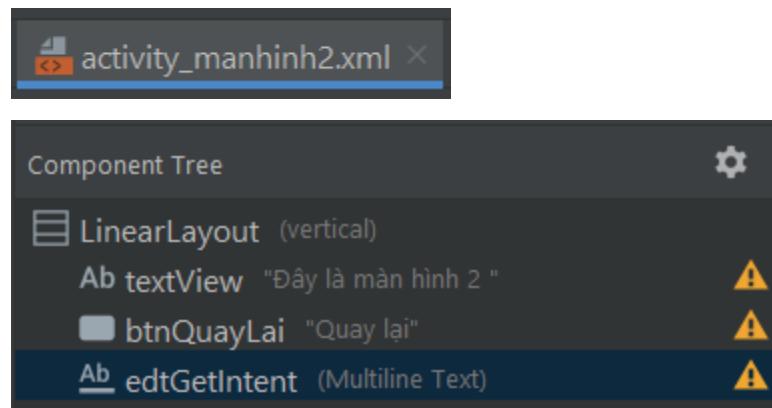


2

## Gửi dữ liệu qua các Activity

## ❑2. Cách 1 : Truyền trực tiếp bang Intent :

Tại `activity_manhinh2.xml` thêm ô `multitext` để hiển thị dữ liệu truyền từ màn hình `MainActivity`



## 2

## Gửi dữ liệu qua các Activity

❑2. Cách 1 : Truyền trực tiếp bằng Intent : **Gửi dữ liệu putExtra()** không có s

```
activity_main.xml × activity_manhinh2.xml × manhinh2.kt × MainActivity.kt ×
package com.tuhoc.bai14_intent

import ...
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        //gọi màn hình 2 từ màn hình main
        btnGo.setOnClickListener { it: View!
            val i = Intent(packageContext: this, manhinh2::class.java)

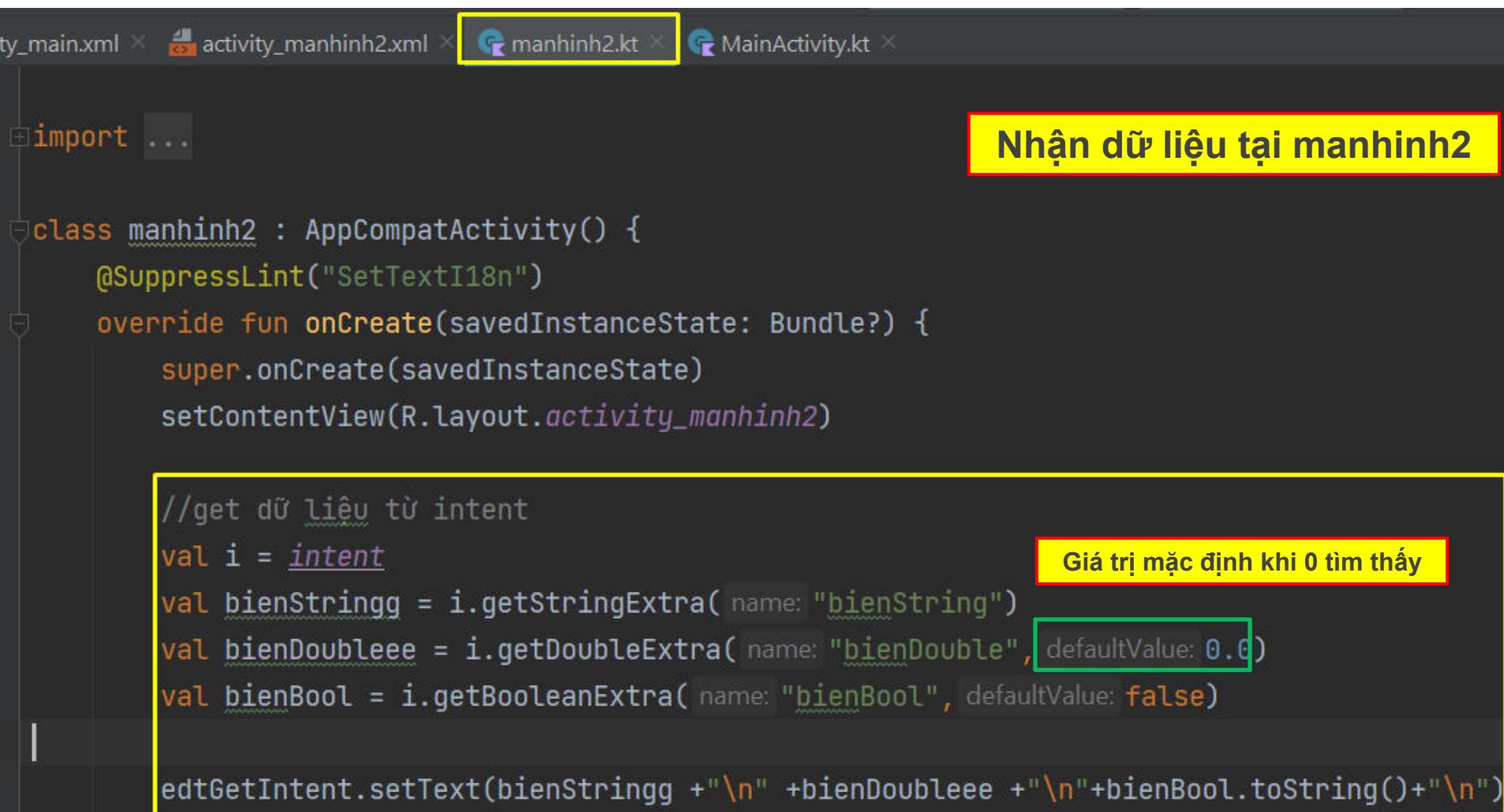
            //truyền dữ liệu trực tiếp intent
            i.putExtra(name: "bienString", value: "chào mừng đến với tuhoc.cc")
            i.putExtra(name: "bienDouble", value: 12345.678)
            i.putExtra(name: "bienBool", value: true)
            startActivity(i)
        }
    }
}
```

**Gửi dữ liệu qua manhinh2**

## 2

## Gửi dữ liệu qua các Activity

## ❑2. Cách 1 : Truyền trực tiếp bằng Intent : Nhận dữ liệu



The screenshot shows the Android Studio interface with the following details:

- Navigation Bar:** Shows tabs for `activity_main.xml`, `activity_manhinh2.xml`, `manhinh2.kt` (highlighted with a yellow box), and `MainActivity.kt`.
- Code Editor:** Displays the `manhinh2.kt` file content.
- Annotations:**
  - A yellow box highlights the text "Nhận dữ liệu tại manhinh2".
  - A green box highlights the parameter `defaultValue: 0.0` in the `getDoubleExtra` call.
  - A red box highlights the parameter `defaultValue: false` in the `getBooleanExtra` call.
- Text in the code editor:**

```
import ...

class manhinh2 : AppCompatActivity() {
    @SuppressLint("SetTextI18n")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_manhinh2)

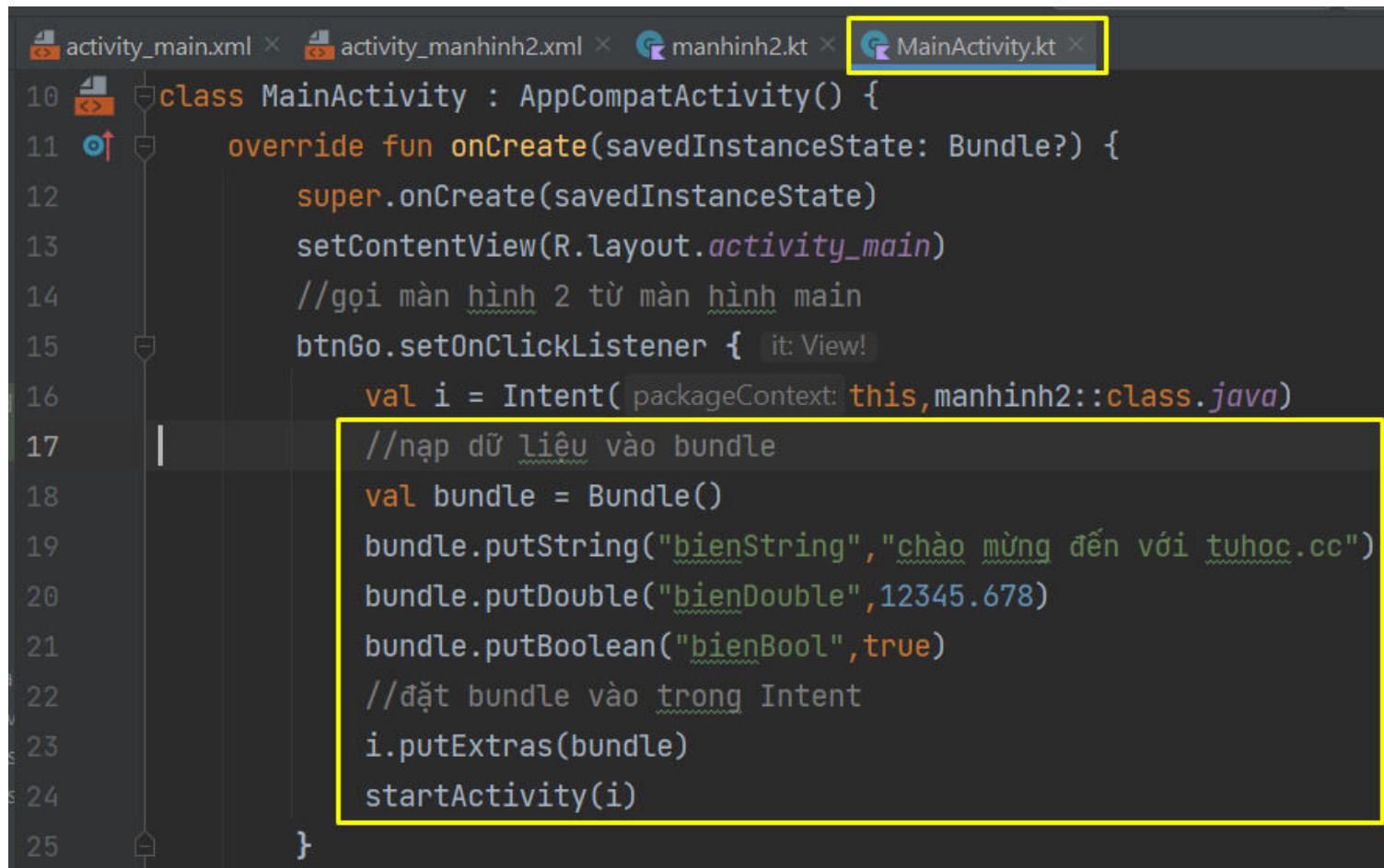
        //get dữ liệu từ intent
        val i = intent
        val bienStringg = i.getStringExtra("bienString")
        val bienDoubleeee = i.getDoubleExtra("bienDouble", defaultValue: 0.0)
        val bienBool = i.getBooleanExtra("bienBool", defaultValue: false)

        edtGetIntent.setText(bienStringg +"\n" +bienDoubleeee +"\n"+bienBool.toString()+"\n")
    }
}
```

## 2

## Gửi dữ liệu qua các Activity

## □3. Cách 2 : Gửi và nhận bằng bundle: Gửi dữ liệu



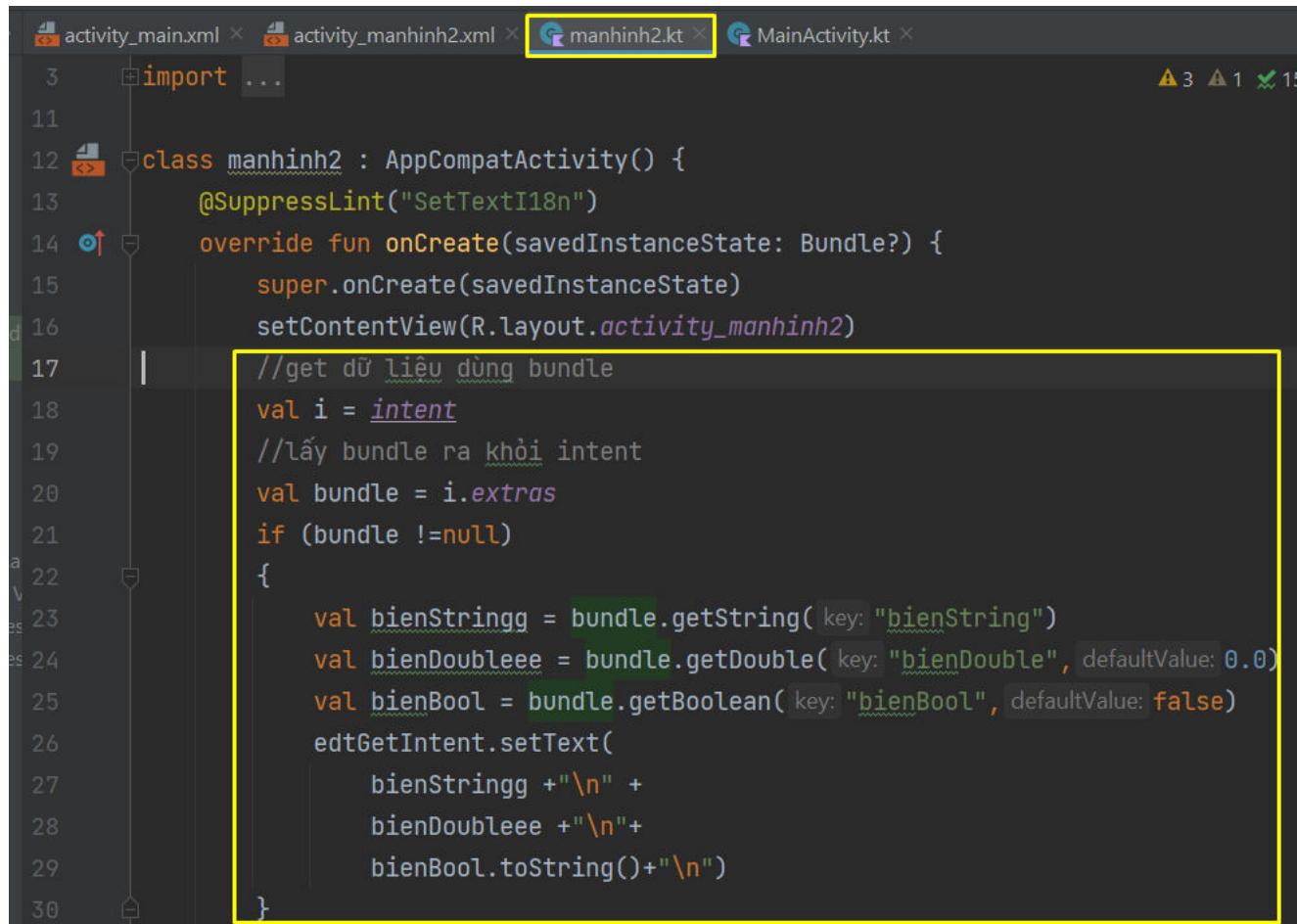
```
activity_main.xml activity_manhinh2.xml manhinh2.kt MainActivity.kt
10 class MainActivity : AppCompatActivity() {
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         setContentView(R.layout.activity_main)
14         //gọi màn hình 2 từ màn hình main
15         btnGo.setOnClickListener { it: View!
16             val i = Intent( packageContext: this,manhinh2::class.java)
17             //nạp dữ liệu vào bundle
18             val bundle = Bundle()
19             bundle.putString("bienString","chào mừng đến với tuhoc.cc")
20             bundle.putDouble("bienDouble",12345.678)
21             bundle.putBoolean("bienBool",true)
22             //đặt bundle vào trong Intent
23             i.putExtra(bundle)
24             startActivity(i)
25 }
```



## 2

## Gửi dữ liệu qua các Activity

## □4. Cách 2 : Gửi và nhận bằng bundle: Nhận dữ liệu



```
activity_main.xml × activity_manhinh2.xml × manhinh2.kt × MainActivity.kt ×
import ...
class manhinh2 : AppCompatActivity() {
    @SuppressLint("SetTextI18n")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_manhinh2)
        //get dữ liệu dùng bundle
        val i = intent
        //lấy bundle ra khỏi intent
        val bundle = i.extras
        if (bundle != null)
        {
            val bienStringg = bundle.getString(key: "bienString")
            val bienDoubleeee = bundle.getDouble(key: "bienDouble", defaultValue: 0.0)
            val bienBool = bundle.getBoolean(key: "bienBool", defaultValue: false)
            edtGetIntent.setText(
                bienStringg +"\n" +
                bienDoubleeee +"\n" +
                bienBool.toString()+"\n")
        }
    }
}
```



## 2

## Gửi dữ liệu qua các Activity

- \* Nếu như **Extra** trên kia sẽ “xé lẻ” dữ liệu ra và gửi theo từng dòng. Thì **Bundle** sẽ giúp bạn “đóng gói” dữ liệu lại và gửi nguyên kiện. Tất nhiên **Bundle** sẽ tiện hơn trong trường hợp bạn muốn gửi cùng một bộ dữ liệu đến nhiều Activity khác nhau.
- \* Ngoài nhiệm vụ đóng gói dữ liệu để truyền qua lại giữa các Activity ở bài học này, thì **Bundle** còn dùng trong một số mục đích khác, đơn cử như truyền dữ liệu qua **Fragment** mà bạn sẽ được biết ở loạt bài sau. Nên tốt hơn hết bạn nên học cách sử dụng **Bundle** ngay từ bài này nhé.

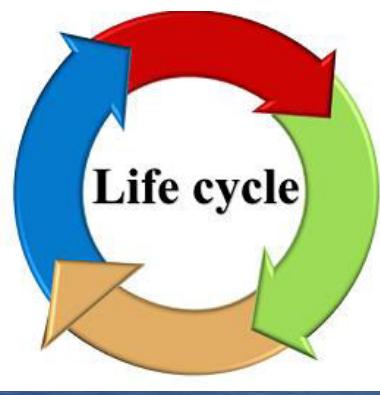
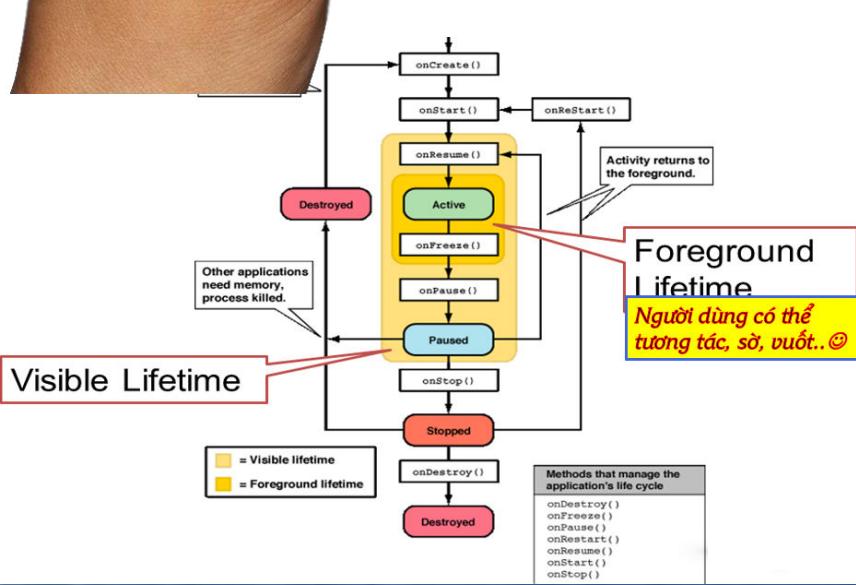




# Android app with kotlin

## Bài 15

# The activity lifecycle Android



1

## Applications?

## □ 1. Khái quát về Applications :

- ✓ Mỗi một Android Project khi bạn biên dịch thành công thì sẽ được đóng gói thành tập tin **.apk**, tập tin **.apk** được gọi là một Application
- ✓ Ví dụ : zalo, facebook, momo....



## 2

## Activity là gì ?

## □ 2. Khái quát về Activity :

✓ Thông thường trong một ứng dụng (Application) sẽ có một hoặc nhiều Activity (Bạn hiểu đại khái là các màn hình tương tác )

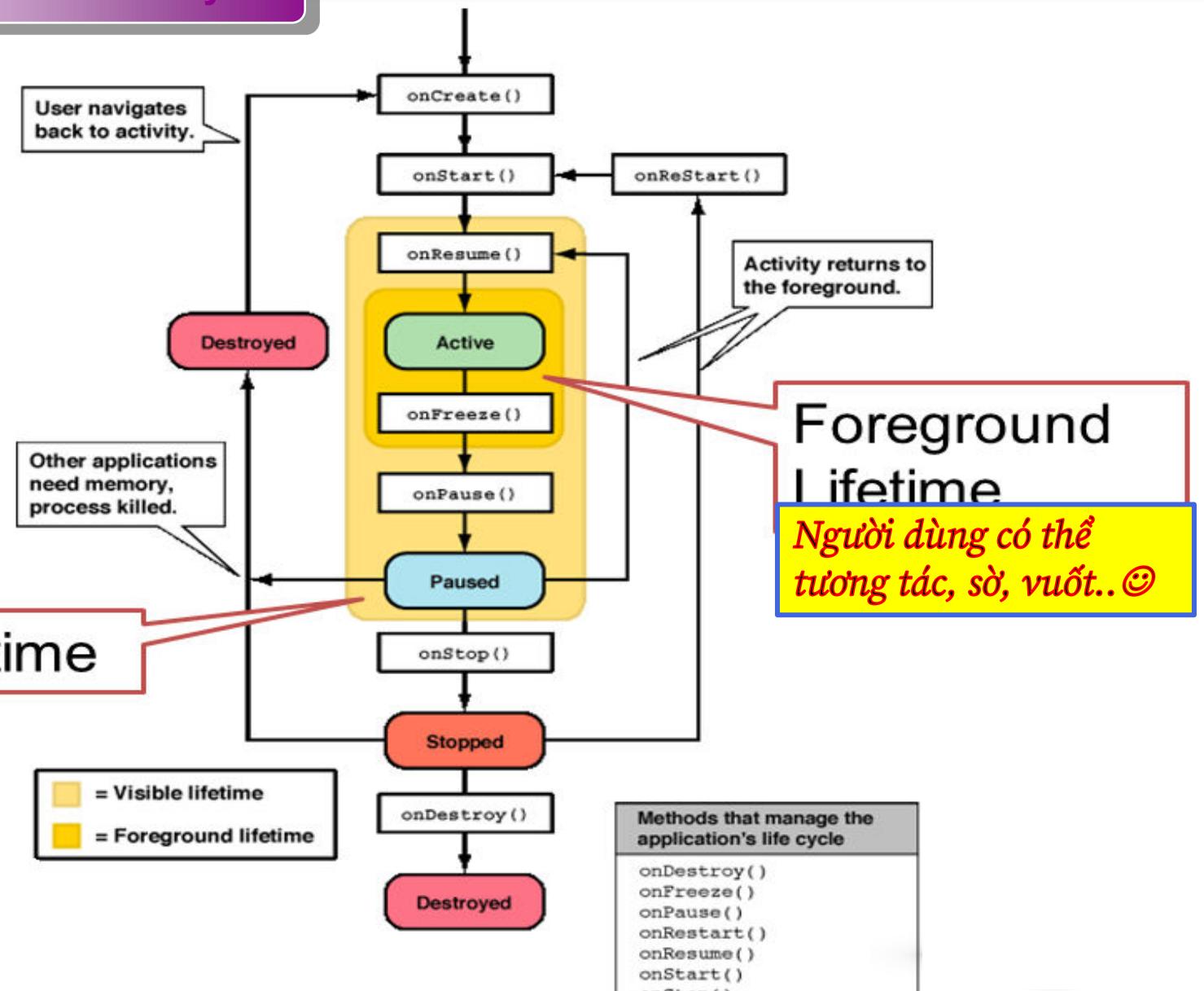


✓ Khi một Activity chỉ định là Main Activity, nó sẽ là màn hình đầu tiên khi khởi chạy ứng dụng. Activity này có thể gọi và kích hoạt một Activity khác



## 3

## Vòng đời của Activity



## 3

## Vòng đời của Activity

 3. Tại sao phải nắm vòng đời của Activity :

- ✓ Đang sử dụng có cuộc gọi đến, chen ngang phần mềm
  - ✓ Đang sử dụng có ứng dụng khác chen 1 phần, hoặc che toàn bộ: Cả hai trường hợp này người dùng đều không tương tác được với phần mềm => activity có thể chuyển sang trạng thái stop
  - ✓ Nếu đang ở trạng thái Stop thì phần mềm có thể sẽ bị tắt ngang,
  - ✓ Ví dụ khi đang nghe điện thoại, ưu tiên số 1 điện thoại sẽ ưu tiên cho cuộc gọi, nên các app chạy nền có thể sẽ bị destroyed
  - ✓ Hoặc chúng ta đang sử dụng ứng dụng khác cần nhiều bộ nhớ hệ thống, nó sẽ ưu tiên cho ứng dụng chúng ta đang sử dụng và nó cũng destroyed ứng dụng chạy nền
- => Chúng ta cần xử lý lưu dữ liệu, hoặc lưu trạng thái... để tránh loss dữ liệu

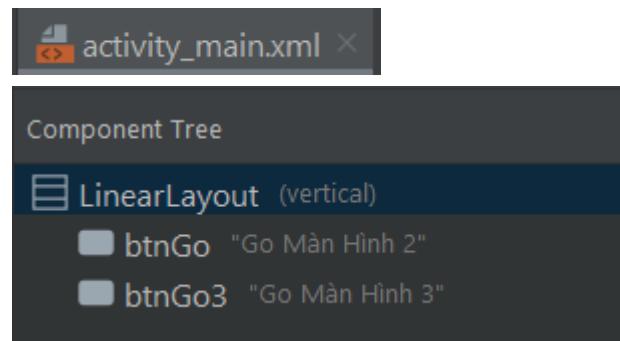


3

## Vòng đời của Activity

## □ 4. Xây dựng mô phỏng :

✓ Step 1: xây dựng layout màn hình chính



```
<Button  
    android:id="@+id	btnGo"  
    android:layout_width="200dp"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:layout_marginTop="15dp"  
    android:text="Go Màn Hình 2" />  
  
<Button  
    android:id="@+id	btnGo3"  
    android:layout_width="200dp"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:layout_marginTop="15dp"  
    android:text="Go Màn Hình 3" />
```



## 3

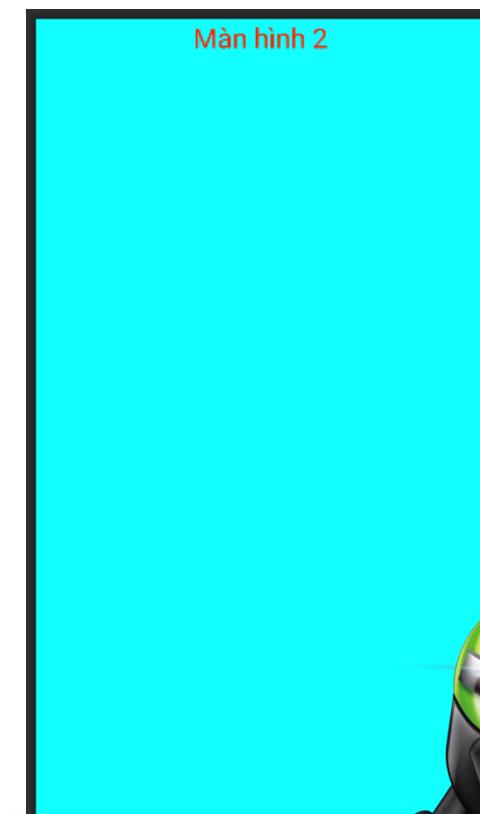
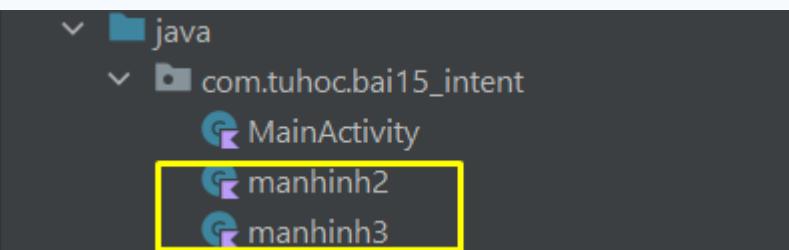
## Vòng đời của Activity

## □ 4. Xây dựng mô phỏng :

✓ Step 2: xây dựng layout màn hình phụ 2, 3

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent"
    android:background="#12ffff"
    tools:context=".manhinh2">

    <TextView
        android:id="@+id/txtManHinh2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25dp"
        android:textColor="#f20"
        android:layout_gravity="center"
        android:text="Màn hình 2" />
</LinearLayout>
```



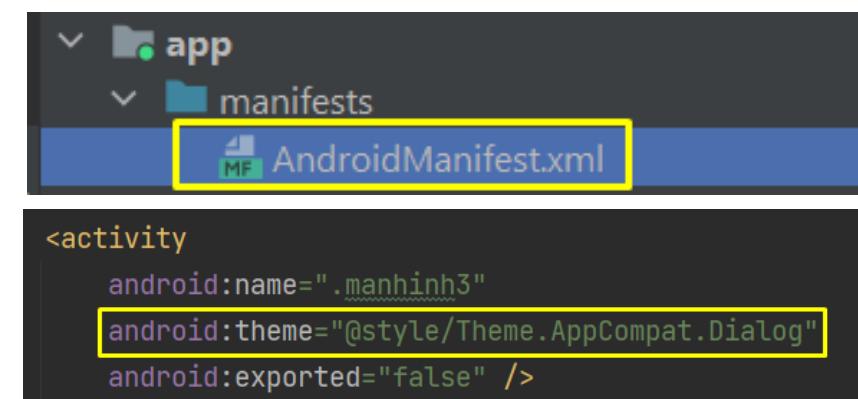
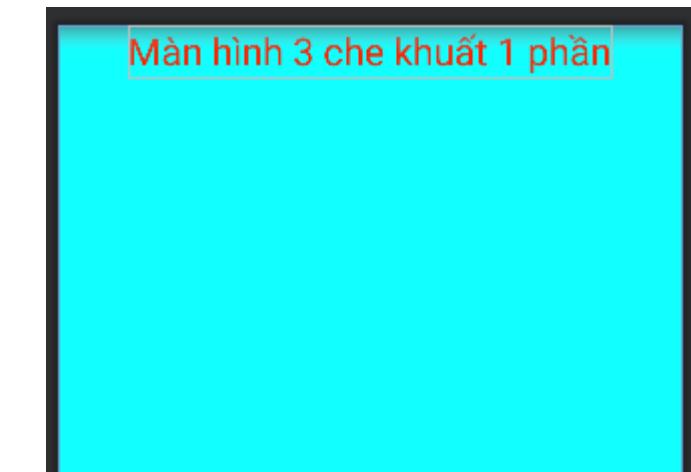
## 3

## Vòng đời của Activity

## □ 4. Xây dựng mô phỏng :

✓ Step 2: xây dựng layout màn hình phụ 2,3

```
<LinearLayout xmlns:android="http://schemas.android.com  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:orientation="vertical"  
    android:layout_height="match_parent"  
    android:background="#12ffff"  
    tools:context=".manhinh2">  
  
<TextView  
    android:id="@+id/txtManHinh3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textSize="25dp"  
    android:textColor="#f20"  
    android:layout_gravity="center"  
    android:text="Màn hình 3 che khuất 1 phần" />  
</LinearLayout>
```



## 3

## Vòng đời của Activity

## □ 4. Xây dựng mô phỏng :

✓ Step 4: *Code mainactivity*

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        Toast.makeText(context, text: "onCreate", Toast.LENGTH_SHORT).show()  
  
        //gọi màn hình 2 từ màn hình main  
        btnGo.setOnClickListener { it: View!  
            val intent1 = Intent(packageContext: this, manhinh2::class.java)  
            startActivity(intent1)  
        }  
  
        //gọi màn hình 3 từ màn hình main  
        btnGo3.setOnClickListener { it: View!  
            val intent2 = Intent(packageContext: this, manhinh3::class.java)  
            startActivity(intent2)  
        }  
    }  
}
```



## 3

## Vòng đời của Activity

## ❑ 4. Xây dựng mô phỏng :

✓ Step 4: *Code mainactivity*

```
//Bấm CTrl + O để override
override fun onStart() {
    super.onStart()
    Toast.makeText(context: this, text: "onStart", Toast.LENGTH_SHORT).show()
}

override fun onRestart() {
    super.onRestart()
    Toast.makeText(context: this, text: "onRestart", Toast.LENGTH_SHORT).show()
}

override fun onResume() {
    super.onResume()
    Toast.makeText(context: this, text: "onResume", Toast.LENGTH_SHORT).show()
}
```



## 3

## Vòng đời của Activity

## □ 4. Xây dựng mô phỏng :

✓ *Step 4: Code mainactivity*

```
override fun onPause() {  
    super.onPause()  
    Toast.makeText(context: this, text: "onPause", Toast.LENGTH_SHORT).show()  
}
```

```
override fun onStop() {  
    super.onStop()  
    Toast.makeText(context: this, text: "onStop", Toast.LENGTH_SHORT).show()  
}
```

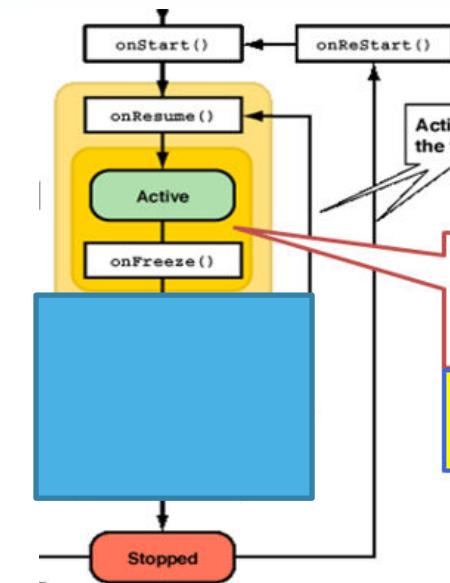
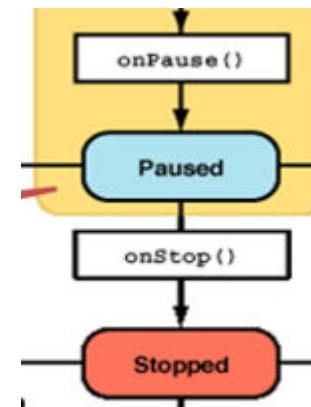
```
override fun onDestroy() {  
    super.onDestroy()  
    Toast.makeText(context: this, text: "onDestroy", Toast.LENGTH_SHORT).show()  
}
```



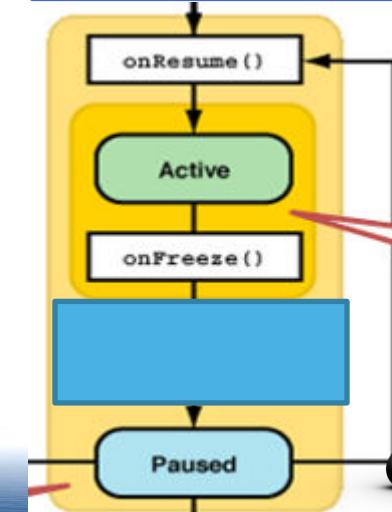
## 3

## Vòng đời của Activity

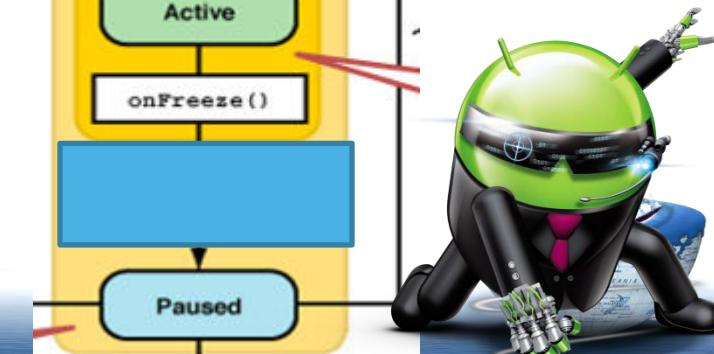
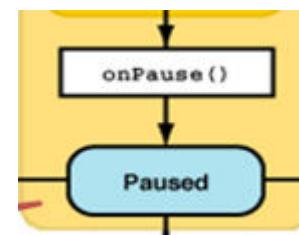
- ✓ Khi bị che khuất toàn bộ  
(thứ bấm phím home,  
hoặc thực tế có cuộc gọi đến,  
ứng dụng khác che toàn bộ)



*Khi activity được gọi lại*



- ✓ Khi bị che khuất 1 phần  
(vẫn nhìn thấy 1 phần  
nhưng người dùng không  
thao tác được)



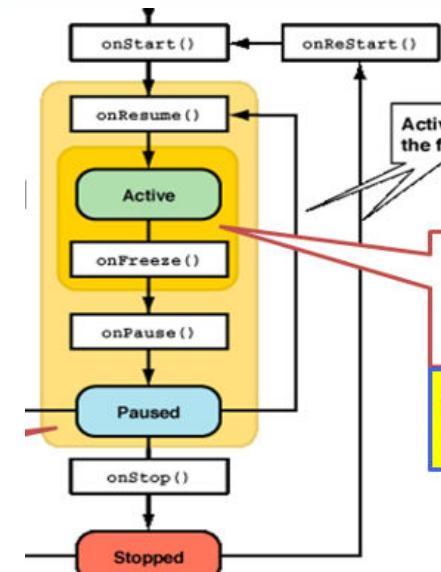
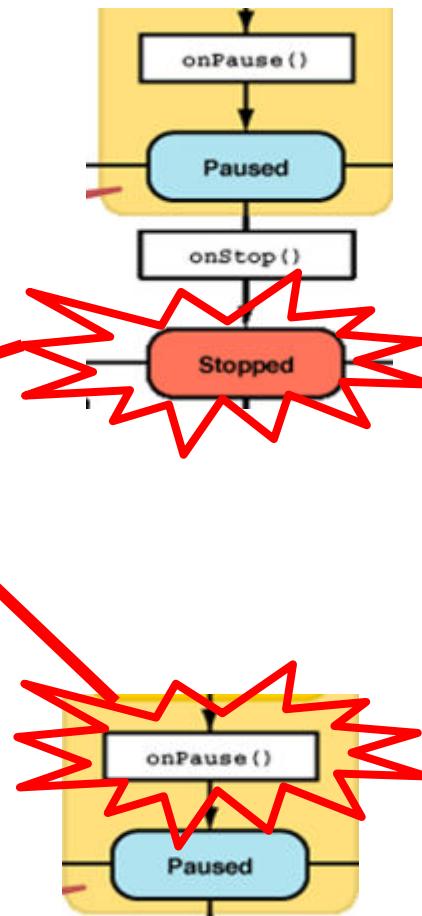
## 3

## Vòng đời của Activity

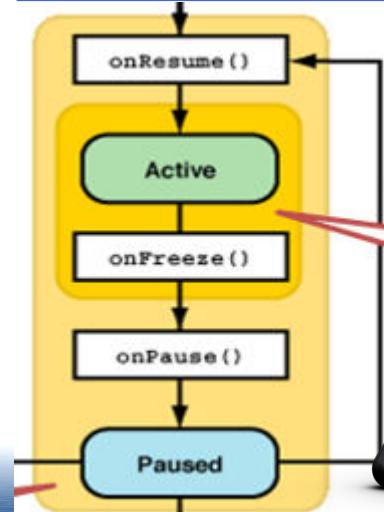
- ✓ Khi bị che khuất toàn bộ  
(thứ bấm phím home,  
hoặc thực tế có cuộc gọi đến,  
ứng dụng khác che toàn bộ)

**Lưu trạng thái  
thì lưu trong sự  
kiện onPause()**

- ✓ Khi bị che khuất 1 phần  
(vẫn nhìn thấy 1 phần  
nhưng người dùng không  
thao tác được)



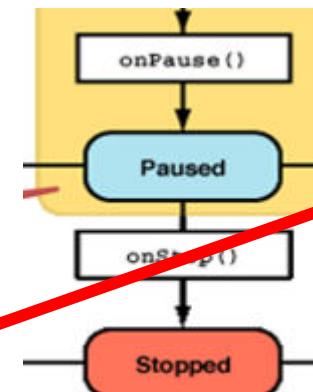
**Khi activity được gọi lại**



## 3

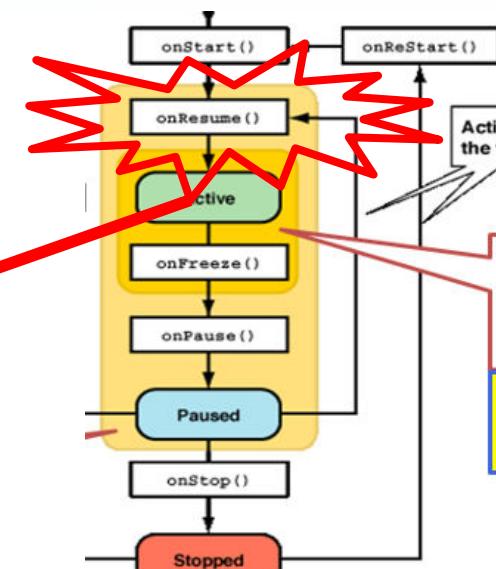
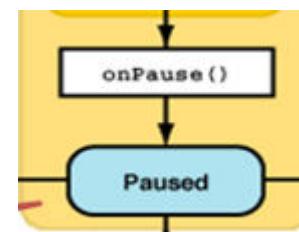
## Vòng đời của Activity

- ✓ Khi bị che khuất toàn bộ  
(thứ bấm phím home,  
hoặc thực tế có cuộc gọi đến,  
ứng dụng khác che toàn bộ)

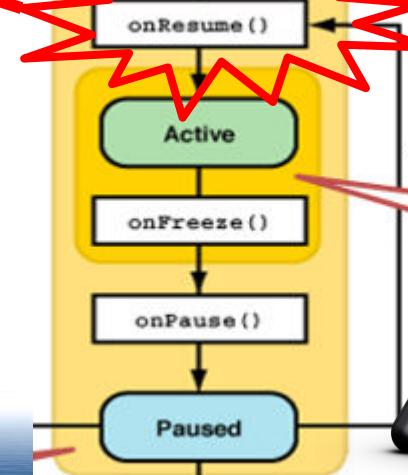


*Phục hồi trạng thái  
thì lưu trong sự kiện  
onResume()*

- ✓ Khi bị che khuất 1 phần  
(vẫn nhìn thấy 1 phần  
nhưng người dùng không  
thao tác được)

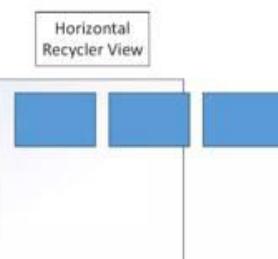
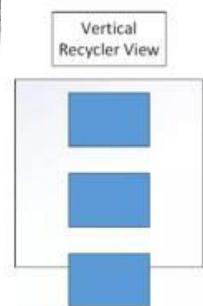


*Khi activity được gọi lại*





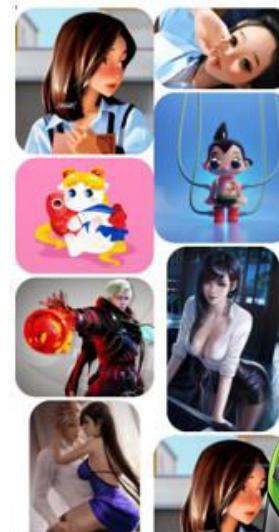
## RecyclerView



1. Hiển thị dạng  
**LinearLayout** có điều  
hướng theo chiều ngang  
hoặc dọc



2. Hiển thị giống  
**GridView** theo hàng và  
cột



3. Hiển thị dạng  
lưới so le

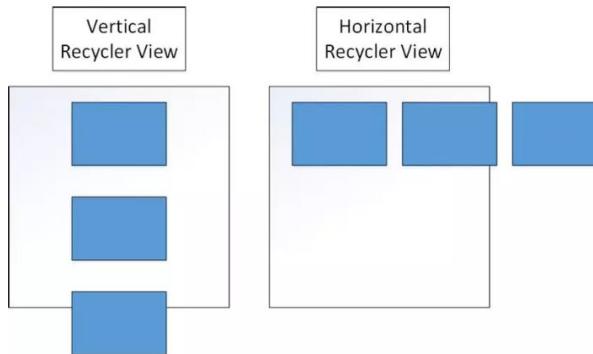


## 1

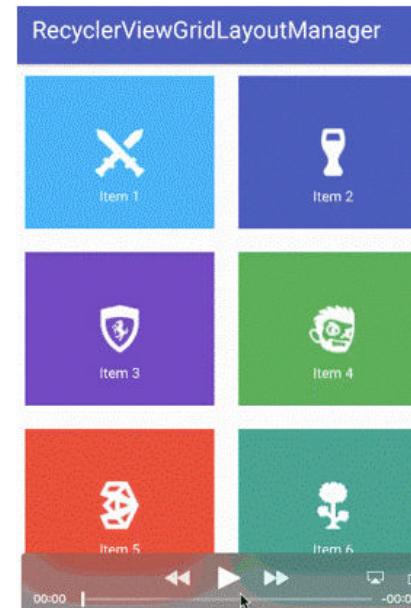
## RecyclerView kotlin

1. RecyclerView:

✓ Là một ViewGroup dùng để chuẩn bị và hiện thị các View tương tự nhau .  
**RecyclerView được cho là sự kế thừa của ListView và GridView, mang lại khả năng sắp xếp view đa dạng hơn**



1. Hiển thị dạng  
**LinearLayout** có điều  
 hướng theo chiều ngang  
 hoặc dọc



2. Hiển thị giống  
**GridView** theo dạng  
 lưới : Hàng và cột



3. Hiển thị dạng  
 lưới so le

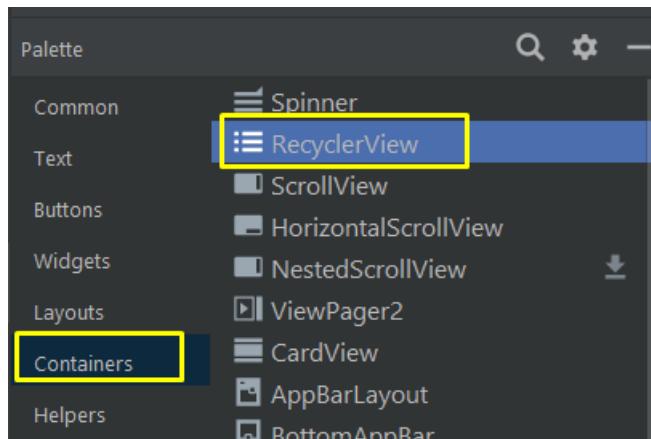


## 1

## RecyclerView kotlin

## □ 1. RecyclerView:

✓ *id : bắt đầu bằng tiền tố rv Ví dụ : rvPhim*



1

## RecyclerView kotlin

### □ 2. Các bước xây dựng RecyclerView:

Step 1:  
Make xml layout  
CustomLayout

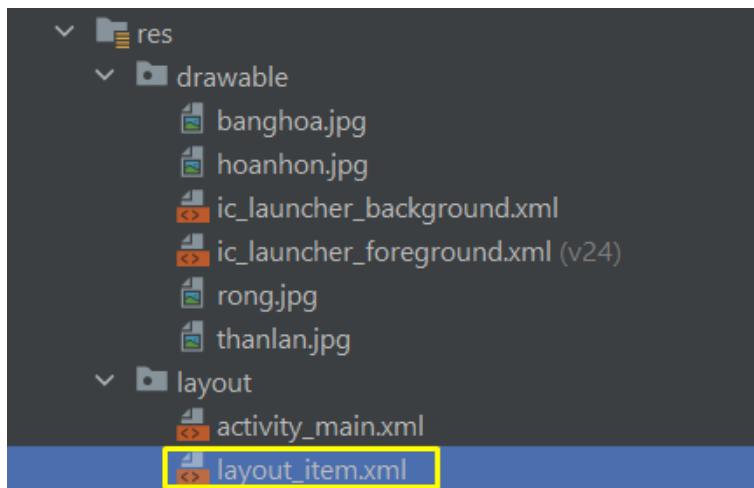
Step 2:  
Make class  
OutData

Step 3:  
Make Class  
RvAdapter  
-ViewHolder



## 1

## RecyclerView kotlin

 3. Step 1 - tạo xml layout file tùy chỉnh – Đưa hình ảnh vào drawable:

Ảnh để trong drawable  
và code toàn bộ bài 16  
Pass unrar : **tuhoc16**

<http://android.tuhoc.cc/>

16.1-bai16-0-recyclerview-tuhoc.cc.rar



bai16-0-recyclerview

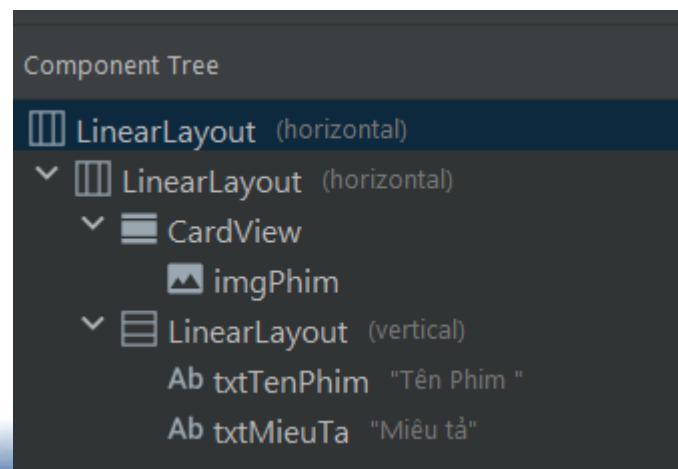
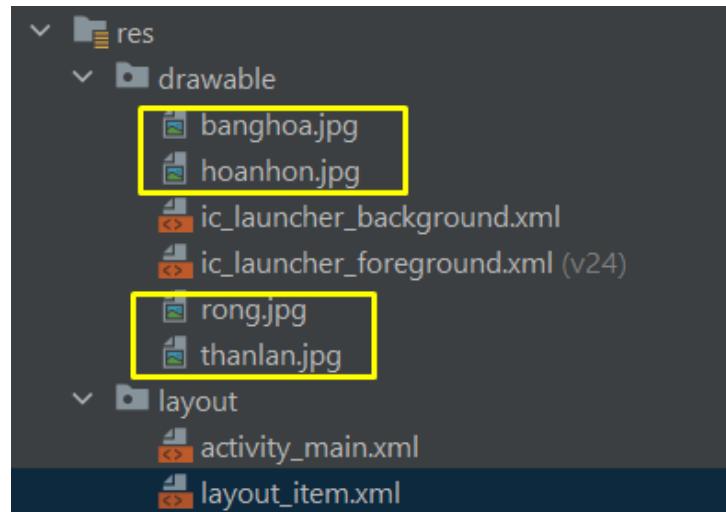
hinhanh-bai16



## 1

## RecyclerView kotlin

## □ 3. Step 1: tạo xml layout file tuỳ chỉnh :



## 1

## RecyclerView kotlin

## □ 3. Step 1: tạo xml layout file tuỳ chỉnh (code trong xml):

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">

        <androidx.cardview.widget.CardView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="5dp"
            app:cardCornerRadius="35dp">

            <ImageView
                android:id="@+id/imgPhim"
                android:layout_width="189dp"
                android:layout_height="291dp"
                android:layout_gravity="center"
                app:srcCompat="@drawable/banghoa" />
        
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="7dp"
    android:layout_marginStart="10dp"
    android:orientation="vertical">

    <TextView
        android:id="@+id/txtTenPhim"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#20f"
        android:textStyle="bold"
        android:textSize="20sp"
        android:text="Tên Phim " />

    <TextView
        android:id="@+id/txtMieuTa"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Miêu tả" />

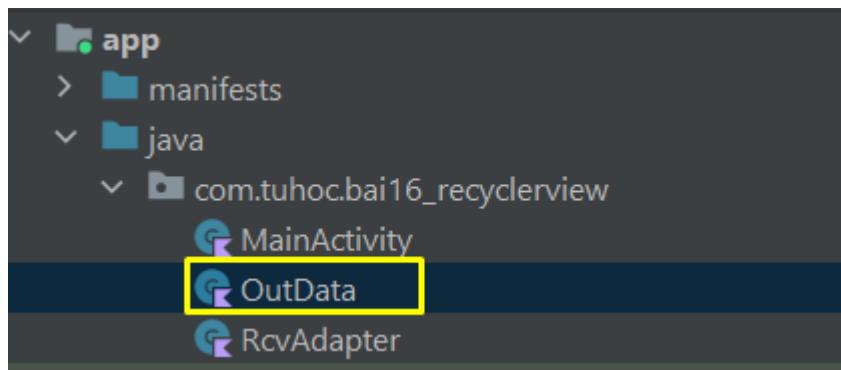
```



## 1

## RecyclerView kotlin

- 4. Step 2: tạo Class OutData lưu trữ các biến cấu trúc của 1 dòng :



```
class OutData(  
    val image:Int,  
    val tenphim:String,  
    val mieuta:String  
) {  
}
```

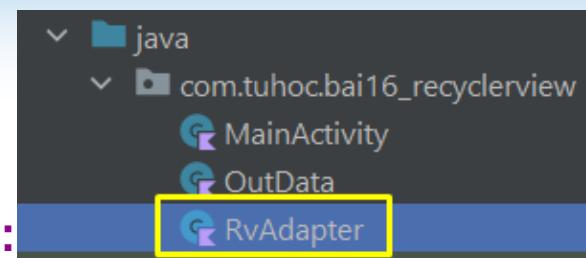


## 1

## RecyclerView kotlin

## □ 5. Step 3: tạo Class RvAdapter – ViewHolder :

```
class RvAdapter (var ds: List<OutData>): RecyclerView.Adapter<RvAdapter.PhimViewHolder>(){  
    //make class viewHolder  
    inner class PhimViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView)  
  
    //ctrl + i  
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PhimViewHolder {  
        val view = LayoutInflater.from(parent.context).inflate(R.layout.layout_item, parent, attachToRoot: false)  
        return PhimViewHolder(view)  
    }  
  
    override fun onBindViewHolder(holder: PhimViewHolder, position: Int) {  
        holder.itemView.apply { this: View  
            txtMieuTa.text = ds[position].mieuTa  
            txtTenPhim.text = ds[position].tenphim  
            imgPhim.setImageResource(ds[position].image)  
        }  
    }  
    override fun getItemCount(): Int {  
        return ds.size  
    }  
}
```



## 1

## RecyclerView kotlin

## □ 6. Code Mainactivity:

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        //tạo ds
        val ds = mutableListOf<OutData>()
        ds.add(OutData(R.drawable.banghoa, tenphim: "Băng Hoả Ma Trù", mieuta: "Phim Trung Quốc"))
        ds.add(OutData(R.drawable.hoanhon, tenphim: "Hoàn Hồn", mieuta: "Phim Trung Quốc"))
        ds.add(OutData(R.drawable.rong, tenphim: "Gia Tộc Rồng", mieuta: "Phim Trung Quốc"))
        ds.add(OutData(R.drawable.thanlan, tenphim: "Thần Lan Chi Mông", mieuta: "Phim Trung Quốc"))

        val adaptervv = RvAdapter(ds)
        rvDemo.adapter= adaptervv
        rvDemo.layoutManager = LinearLayoutManager(
            context: this,
            LinearLayoutManager.HORIZONTAL,
            reverseLayout: false
        )
    }
}
```



## 1

## RecyclerView kotlin

## □ 6. Code Mainactivity: Hiển thị dạng gridLayout

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        //tạo ds
        val ds = mutableListOf<OutData }()
        ds.add(OutData(R.drawable.banghoa, tenphim: "Băng Hoả Ma Trù", mieuta: "Phim Trung Quốc"))
        ds.add(OutData(R.drawable.hoanhon, tenphim: "Hoàn Hồn", mieuta: "Phim Trung Quốc"))
        ds.add(OutData(R.drawable.rong, tenphim: "Gia Tộc Rồng", mieuta: "Phim Trung Quốc"))
        ds.add(OutData(R.drawable.thanlan, tenphim: "Thần Lan Chi Mông", mieuta: "Phim Trung Quốc"))

        val adaptervv = RvAdapter(ds)
        rvDemo.adapter= adaptervv
        rvDemo.layoutManager = GridLayoutManager(
            context: this,
            spanCount: 2,
            GridLayoutManager.HORIZONTAL,
            reverseLayout: false
        )
    }
}
```

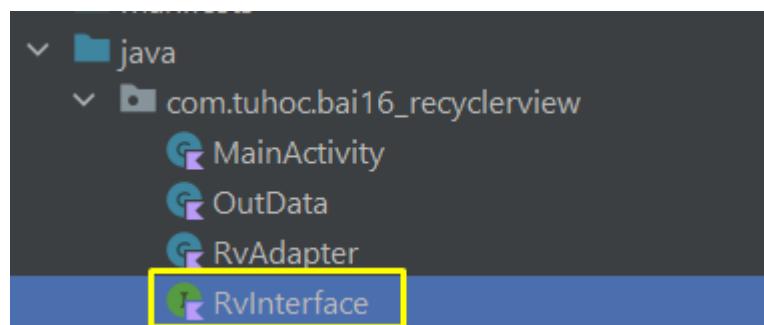


## 1

## RecyclerView kotlin

 4. holder.itemView.setOnClickListener :

✓ *Lắng nghe xem item nào của RecyclerView được click chọn*

 4.1 Step 1: tạo Class Interface

```
interface RvInterface {  
    fun OnClickPhim(pos : Int)  
}
```



Thần Lan Chi Mộng  
Phim trung quốc



## 1

## RecyclerView kotlin

## □ 4.2 Step 2: Code RvAdapter

```
class RvAdapter (var ds: List<OutData>, val onPhimClick : RvInterface): RecyclerView.Adapter<RvAdapter.PhimViewHolder>(){

    inner class PhimViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView)

    //ctrl + i
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PhimViewHolder {
        val view = LayoutInflater.from(parent.context).inflate(R.layout.layout_item,parent, attachToRoot: false)
        return PhimViewHolder(view)
    }

    override fun onBindViewHolder(holder: PhimViewHolder, position: Int) {
        holder.itemView.apply { this: View
            txtMieuTa.text = ds[position].mieuTa
            txtTenPhim.text =ds[position].tenphim
            imgPhim.setImageResource(ds[position].image)

            //lắng nghe item click chọn
            holder.itemView.setOnClickListener { it: View!
                onPhimClick.OnClickPhim(position)
            }
        }
    }
}
```



## 1

## RecyclerView kotlin

## □ 4.3 Step 3: Code MainActivity

```
class MainActivity : AppCompatActivity(){
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val ds= mutableListOf<OutData }()
        ds.add(OutData(R.drawable.banghoa, tenphim: "Băng hoả ma trù", mieuta: "Phim trung quốc"))
        ds.add(OutData(R.drawable.hoanhon, tenphim: "Hoàn Hồn", mieuta: "Phim trung quốc"))
        ds.add(OutData(R.drawable.rong, tenphim: "Gia Tộc Rồng", mieuta: "Phim trung quốc"))
        ds.add(OutData(R.drawable.thanlan, tenphim: "Thần Lan Chi Mông", mieuta: "Phim trung quốc"))

        val adapter = RvAdapter(ds)
        object :RvInterface{
            //Alt + enter để override
            override fun OnClickPhim(pos: Int) {
                Toast.makeText(context: this@MainActivity,
                    text: "Bạn đã click vào ${ds[pos].tenphim}",
                    Toast.LENGTH_SHORT).show()
            }
        })
        rvDemo.adapter = adapter
        rvDemo.layoutManager = LinearLayoutManager(
            context: this,
            GridLayoutManager.VERTICAL,
            reverseLayout: false )
    }
}
```





## RecyclerView itemView.setOnClickListener



→ Click!



Thần Lan Chi Mộng  
Phim trung quốc

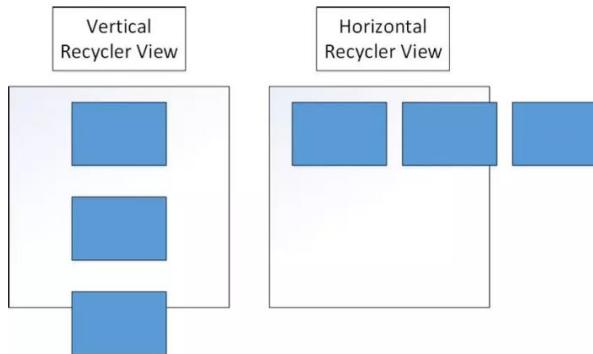


## 1

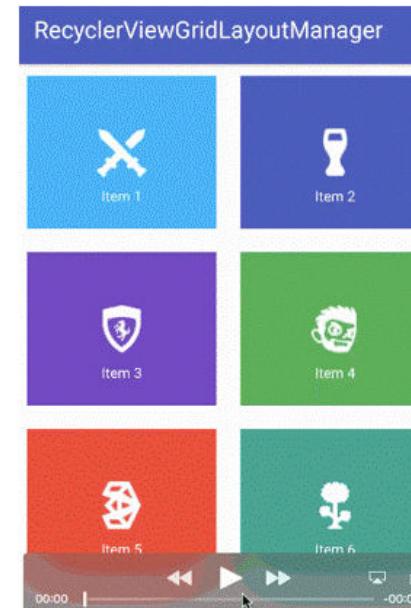
## RecyclerView kotlin

1. RecyclerView:

✓ Là một ViewGroup dùng để chuẩn bị và hiện thị các View tương tự nhau .  
**RecyclerView được cho là sự kế thừa của ListView và GridView, mang lại khả năng sắp xếp view đa dạng hơn**



1. Hiển thị dạng  
**LinearLayout** có điều  
 hướng theo chiều ngang  
 hoặc dọc



2. Hiển thị giống  
**GridView** theo dạng  
 lưới : Hàng và cột



3. Hiển thị dạng  
 lưới so le

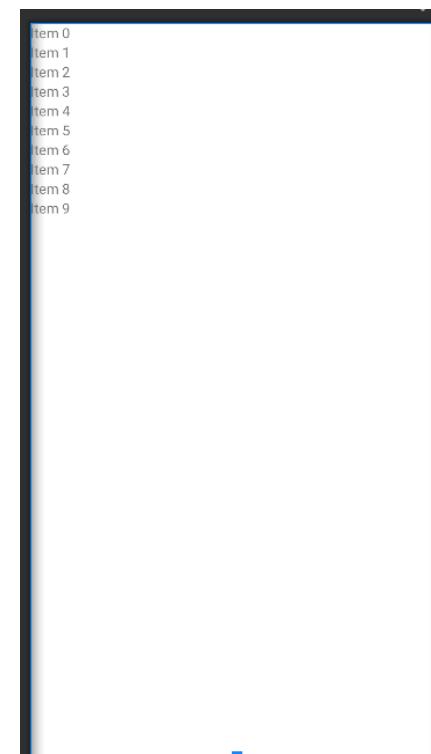
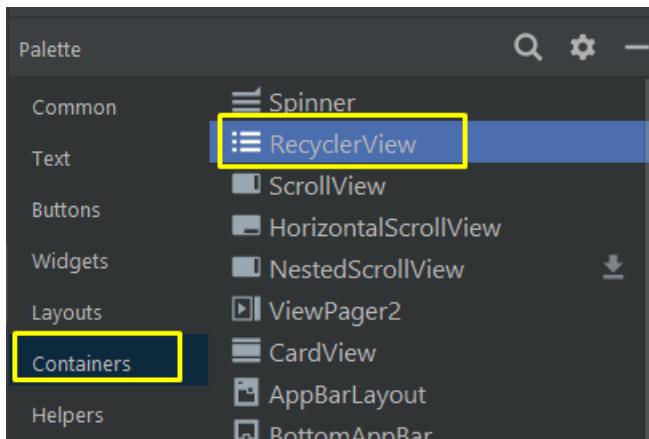


## 1

## RecyclerView kotlin

## □ 1. RecyclerView:

✓ *id : bắt đầu bằng tiền tố rv Ví dụ : rvPhim*



1

## RecyclerView kotlin

### □ 2. Các bước xây dựng RecyclerView:

Step 1:  
Make xml layout  
CustomLayout

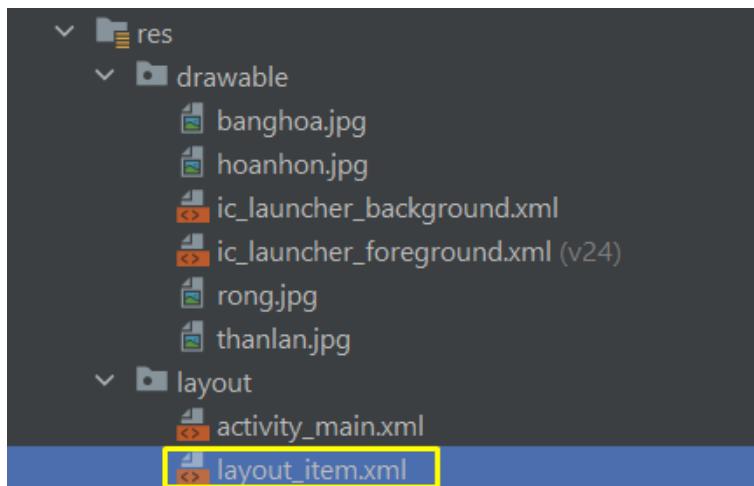
Step 2:  
Make class  
OutData

Step 3:  
Make Class  
RvAdapter  
-ViewHolder



## 1

## RecyclerView kotlin

 3. Step 1 - tạo xml layout file tùy chỉnh – Đưa hình ảnh vào drawable:

Ảnh để trong drawable  
và code toàn bộ bài 16  
Pass unrar : **tuhoc16**

<http://android.tuhoc.cc/>

16.1-bai16-0-recyclerview-tuhoc.cc.rar



banghoa.jpg



hoanhon.jpg



rong.jpg



thanlan.jpg

bai16-0-recyclerview

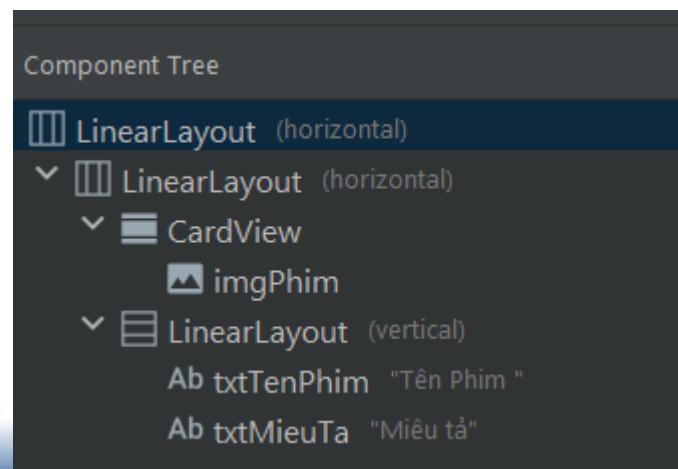
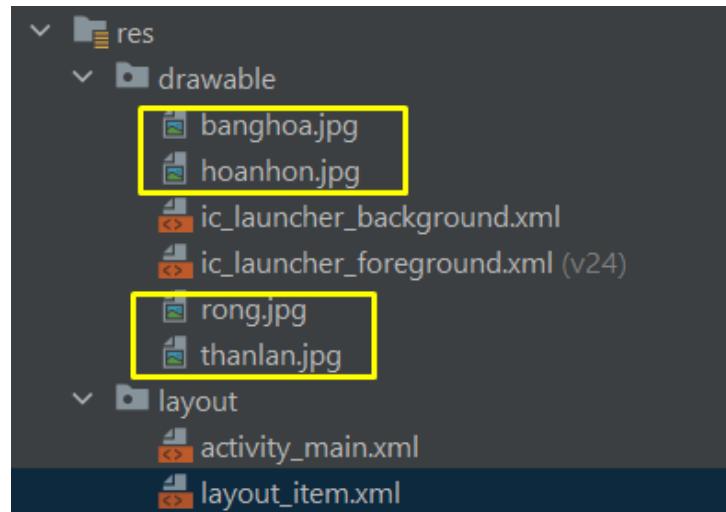
hinhanh-bai16



## 1

## RecyclerView kotlin

## □ 3. Step 1: tạo xml layout file tuỳ chỉnh :



## 1

## RecyclerView kotlin

## □ 3. Step 1: tạo xml layout file tuỳ chỉnh (code trong xml):

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">

        <androidx.cardview.widget.CardView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="5dp"
            app:cardCornerRadius="35dp">

            <ImageView
                android:id="@+id/imgPhim"
                android:layout_width="189dp"
                android:layout_height="291dp"
                android:layout_gravity="center"
                app:srcCompat="@drawable/banghoa" />
        
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="7dp"
    android:layout_marginStart="10dp"
    android:orientation="vertical">

    <TextView
        android:id="@+id/txtTenPhim"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#20f"
        android:textStyle="bold"
        android:textSize="20sp"
        android:text="Tên Phim " />

    <TextView
        android:id="@+id/txtMieuTa"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Miêu tả" />

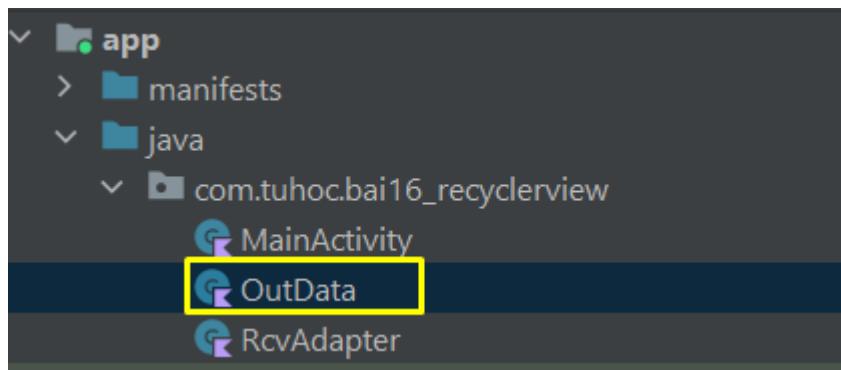
```



## 1

## RecyclerView kotlin

- 4. Step 2: tạo Class OutData lưu trữ các biến cấu trúc của 1 dòng :



```
class OutData(  
    val image:Int,  
    val tenphim:String,  
    val mieuta:String  
) {  
}
```

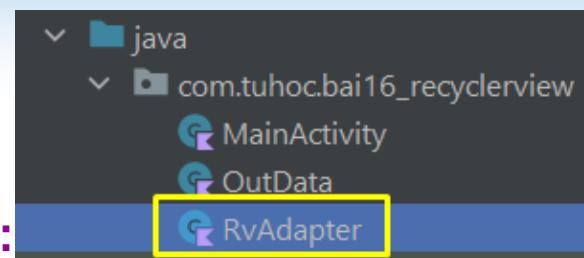


## 1

## RecyclerView kotlin

## □ 5. Step 3: tạo Class RvAdapter – ViewHolder :

```
class RvAdapter (var ds: List<OutData>): RecyclerView.Adapter<RvAdapter.PhimViewHolder>(){  
    //make class viewHolder  
    inner class PhimViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView)  
  
    //ctrl + i  
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PhimViewHolder {  
        val view = LayoutInflater.from(parent.context).inflate(R.layout.layout_item, parent, attachToRoot: false)  
        return PhimViewHolder(view)  
    }  
  
    override fun onBindViewHolder(holder: PhimViewHolder, position: Int) {  
        holder.itemView.apply { this: View  
            txtMieuTa.text = ds[position].mieuuta  
            txtTenPhim.text = ds[position].tenphim  
            imgPhim.setImageResource(ds[position].image)  
        }  
    }  
    override fun getItemCount(): Int {  
        return ds.size  
    }  
}
```



## 1

## RecyclerView kotlin

## □ 6. Code Mainactivity:

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        //tạo ds
        val ds = mutableListOf<OutData>()
        ds.add(OutData(R.drawable.banghoa, tenphim: "Băng Hoả Ma Trù", mieuta: "Phim Trung Quốc"))
        ds.add(OutData(R.drawable.hoanhon, tenphim: "Hoàn Hồn", mieuta: "Phim Trung Quốc"))
        ds.add(OutData(R.drawable.rong, tenphim: "Gia Tộc Rồng", mieuta: "Phim Trung Quốc"))
        ds.add(OutData(R.drawable.thanlan, tenphim: "Thần Lan Chi Mông", mieuta: "Phim Trung Quốc"))

        val adaptervv = RvAdapter(ds)
        rvDemo.adapter= adaptervv
        rvDemo.layoutManager = LinearLayoutManager(
            context: this,
            LinearLayoutManager.HORIZONTAL,
            reverseLayout: false
        )
    }
}
```



## 1

## RecyclerView kotlin

## □ 6. Code Mainactivity: Hiển thị dạng gridLayout

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        //tạo ds
        val ds = mutableListOf<OutData }()
        ds.add(OutData(R.drawable.banghoa, tenphim: "Băng Hoả Ma Trù", mieuta: "Phim Trung Quốc"))
        ds.add(OutData(R.drawable.hoanhon, tenphim: "Hoàn Hồn", mieuta: "Phim Trung Quốc"))
        ds.add(OutData(R.drawable.rong, tenphim: "Gia Tộc Rồng", mieuta: "Phim Trung Quốc"))
        ds.add(OutData(R.drawable.thanlan, tenphim: "Thần Lan Chi Mông", mieuta: "Phim Trung Quốc"))

        val adaptervv = RvAdapter(ds)
        rvDemo.adapter= adaptervv
        rvDemo.layoutManager = GridLayoutManager(
            context: this,
            spanCount: 2,
            GridLayoutManager.HORIZONTAL,
            reverseLayout: false
        )
    }
}
```



## 1

## RecyclerView kotlin

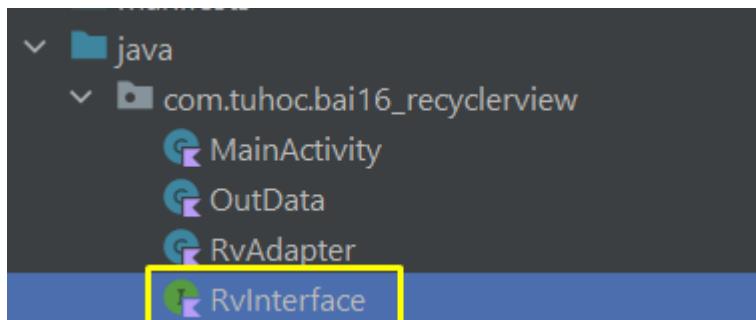
## □ 4. holder.itemView.setOnClickListener :

✓ *Lắng nghe xem item nào của RecyclerView được click chọn*



Thần Lan Chi Mộng  
Phim trung quốc

## □ 4.1 Step 1: tạo Class Interface



```
interface RvInterface {  
    fun OnClickPhim(pos : Int)  
}
```



## 1

## RecyclerView kotlin

## □ 4.2 Step 2: Code RvAdapter

```
class RvAdapter (var ds: List<OutData>, val onPhimClick : RvInterface): RecyclerView.Adapter<RvAdapter.PhimViewHolder>(){

    inner class PhimViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView)

    //ctrl + i
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PhimViewHolder {
        val view = LayoutInflater.from(parent.context).inflate(R.layout.layout_item,parent, attachToRoot: false)
        return PhimViewHolder(view)
    }

    override fun onBindViewHolder(holder: PhimViewHolder, position: Int) {
        holder.itemView.apply { this: View
            txtMieuTa.text = ds[position].mieuTa
            txtTenPhim.text =ds[position].tenphim
            imgPhim.setImageResource(ds[position].image)

            //lắng nghe item click chọn
            holder.itemView.setOnClickListener { it: View!
                onPhimClick.OnClickPhim(position)
            }
        }
    }
}
```



## 1

## RecyclerView kotlin

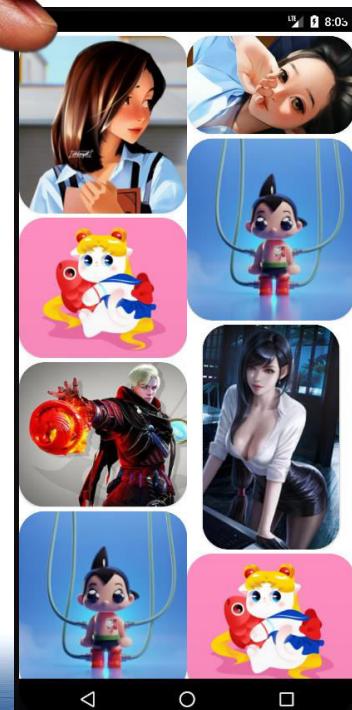
## □ 4.3 Step 3: Code MainActivity

```
class MainActivity : AppCompatActivity(){
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

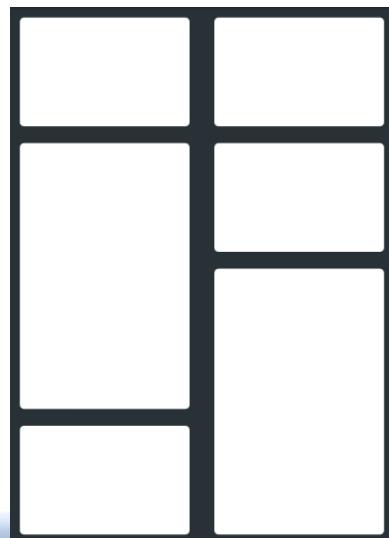
        val ds= mutableListOf<OutData }()
        ds.add(OutData(R.drawable.banghoa, tenphim: "Băng hoả ma trù", mieuta: "Phim trung quốc"))
        ds.add(OutData(R.drawable.hoanhon, tenphim: "Hoàn Hồn", mieuta: "Phim trung quốc"))
        ds.add(OutData(R.drawable.rong, tenphim: "Gia Tộc Rồng", mieuta: "Phim trung quốc"))
        ds.add(OutData(R.drawable.thanlan, tenphim: "Thần Lan Chi Mông", mieuta: "Phim trung quốc"))

        val adapter = RvAdapter(ds)
        object :RvInterface{
            //Alt + enter để override
            override fun OnClickPhim(pos: Int) {
                Toast.makeText(context: this@MainActivity,
                    text: "Bạn đã click vào ${ds[pos].tenphim}",
                    Toast.LENGTH_SHORT).show()
            }
        })
        rvDemo.adapter = adapter
        rvDemo.layoutManager = LinearLayoutManager(
            context: this,
            GridLayoutManager.VERTICAL,
            reverseLayout: false )
    }
}
```





## RecyclerView StaggeredGridLayout



1

## RecyclerView kotlin

### □ 1. Các bước xây dựng RecyclerView:

Step 1:  
Make xml layout  
CustomLayout

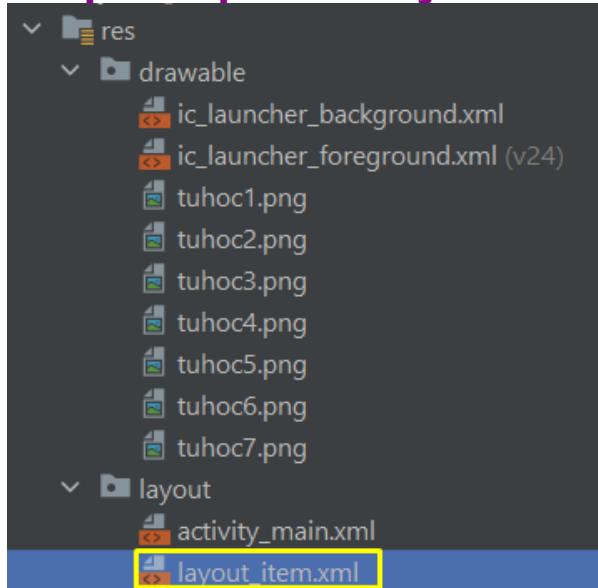
Step 2:  
Make class  
OutData

Step 3:  
Make Class  
RvAdapter  
-ViewHolder



## 1

## RecyclerView kotlin

 2. Step 1 - tạo xml layout file tùy chỉnh – Đưa hình ảnh vào drawable:

Ảnh để trong drawable  
và code toàn bộ bài 16  
Pass unrar : **tuhoc163**

<http://android.tuhoc.cc/>

 16.3-recyclerview-gridlayout.zip 



Name

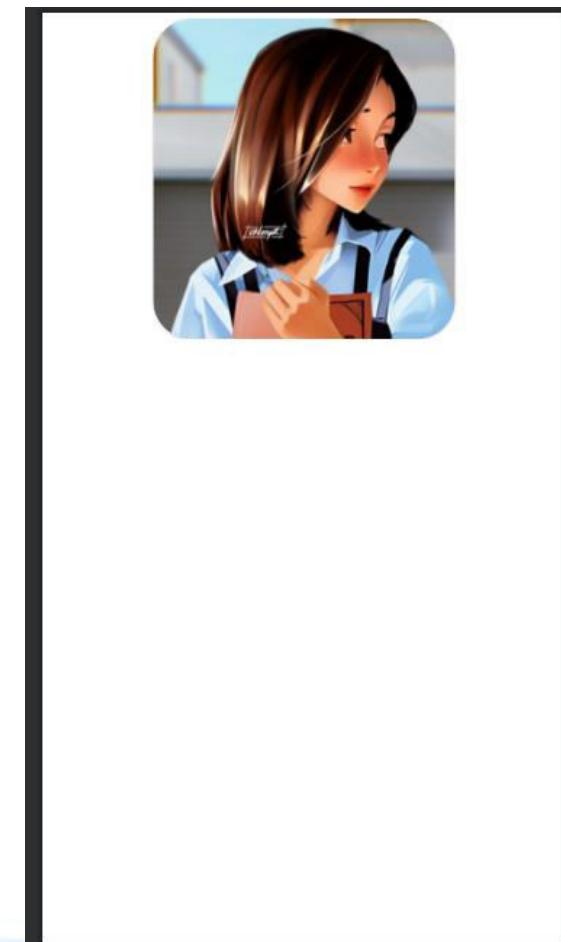
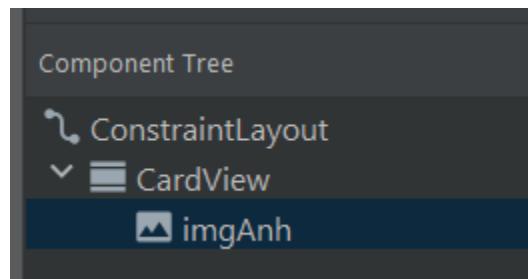
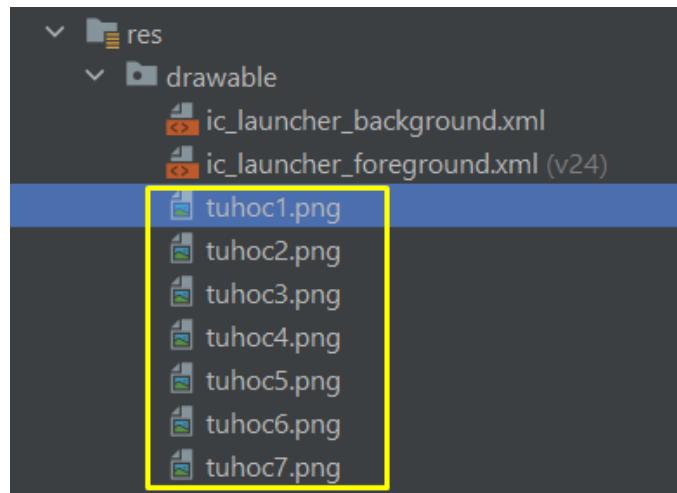
 16-2recyclerviewcustomgridlayout  
 **hinhanh-bai16.3**



1

## RecyclerView kotlin

## □ 3. Step 1: tạo xml layout file tuỳ chỉnh :



## 1

## RecyclerView kotlin

## □ 4. Step 1: tạo xml layout file tuỳ chỉnh (code trong xml):

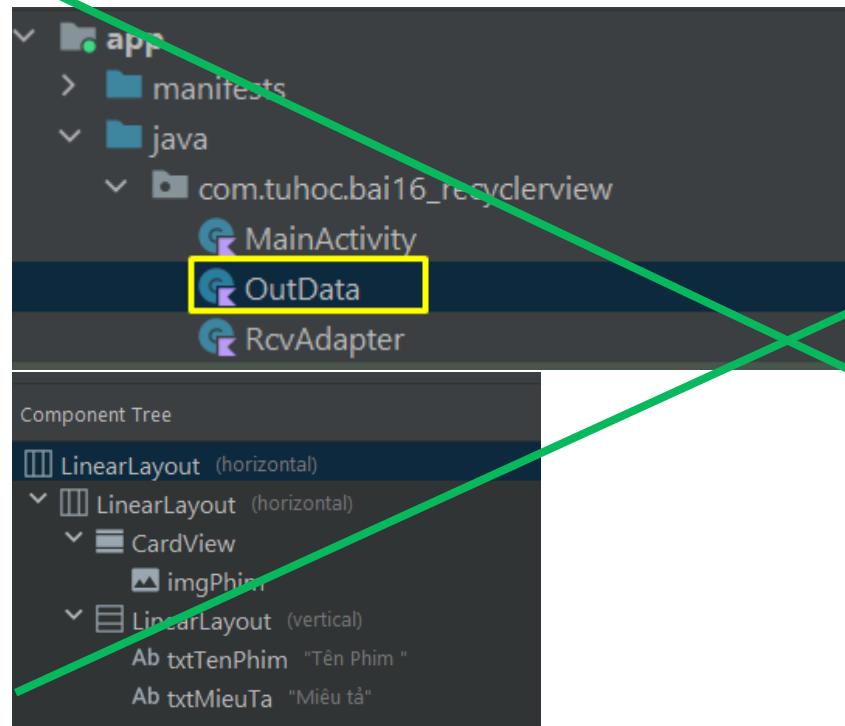
```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:andro  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">  
  
    <androidx.cardview.widget.CardView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_margin="5dp"  
        app:cardCornerRadius="30dp"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent">  
  
        <ImageView  
            android:id="@+id/imgAnh"  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:adjustViewBounds="true"  
            app:srcCompat="@drawable/tuhoc1" />  
    </androidx.cardview.widget.CardView>  
</androidx.constraintlayout.widget.ConstraintLayout>
```



## 1

## RecyclerView kotlin

- 5. Step 2: tạo Class OutData lưu trữ các biến cấu trúc của 1 dòng :



```
class OutData(  
    val image:Int,  
    val tenphim:String,  
    val mieuta:String  
) {  
}
```

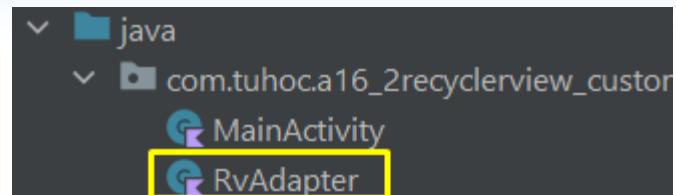
**Do layout\_item chỉ có 1 phần tử là imageView nên có thể bỏ qua bước này**



## 1

## RecyclerView kotlin

## □ 6. Step 3: tạo Class RvAdapter – ViewHolder :



```
class RvAdapter (private val ds:List<Int>) :  
    RecyclerView.Adapter<RvAdapter.ItemViewHolder>() {  
    class ItemViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView)  
  
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ItemViewHolder {  
        val view = LayoutInflater.from(parent.context).inflate(R.layout.layout_item, parent, attachToRoot: false)  
        return ItemViewHolder(view)  
    }  
    override fun onBindViewHolder(holder: ItemViewHolder, position: Int) {  
        holder.itemView.apply { this: View  
            imgAnh.setImageResource(ds[position])  
        }  
    }  
    override fun getItemCount(): Int {  
        return ds.size  
    }  
}
```



## 1

## RecyclerView kotlin

## □ 7. Code Mainactivity:

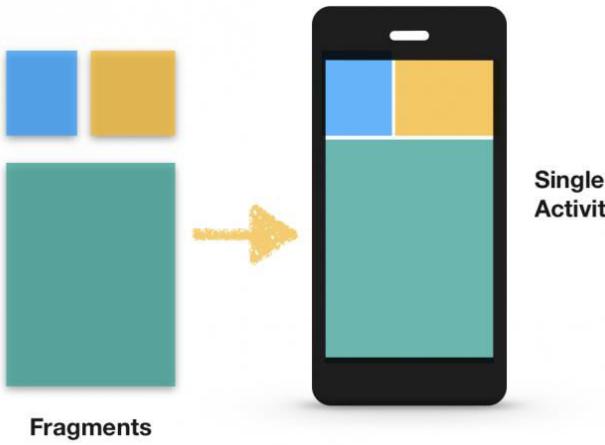
```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        XuLy()
    }
    private fun XuLy() {
        var ds = mutableListOf<Int>()
        ds.add(R.drawable.tuhoc1)
        ds.add(R.drawable.tuhoc2)
        ds.add(R.drawable.tuhoc3)
        ds.add(R.drawable.tuhoc4)
        ds.add(R.drawable.tuhoc5)
        ds.add(R.drawable.tuhoc6)
        ds.add(R.drawable.tuhoc7)

        rvView.layoutManager = StaggeredGridLayoutManager( spanCount: 2, StaggeredGridLayoutManager.VERTICAL)
        val itemAdapter = RvAdapter(ds)
        rvView.adapter = itemAdapter
    }
}
```





## Fragments

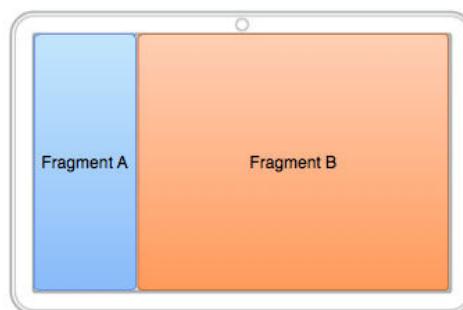


## 1

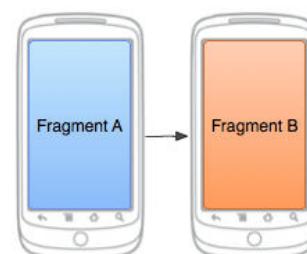
## Fragments kotlin

 1. Tại sao cần dùng Fragment:

- ✓ *Tại một thời điểm thì chỉ có 1 Activity được hiển thị => Khi muốn sử dụng nhiều màn hình con trên cùng 1 activity, ta sử dụng Fragment.*
- ✓ *Fragment có thể coi là sub-activity (activity con).*
- ✓ *Dùng trong trường hợp hiển thị tương thích trên các thiết bị có độ lớn màn hình khác nhau*



Giao diện linh động trên  
Tablet & Mobile



## 1

## Fragments kotlin

 2. Một số đặc điểm tiêu biểu :

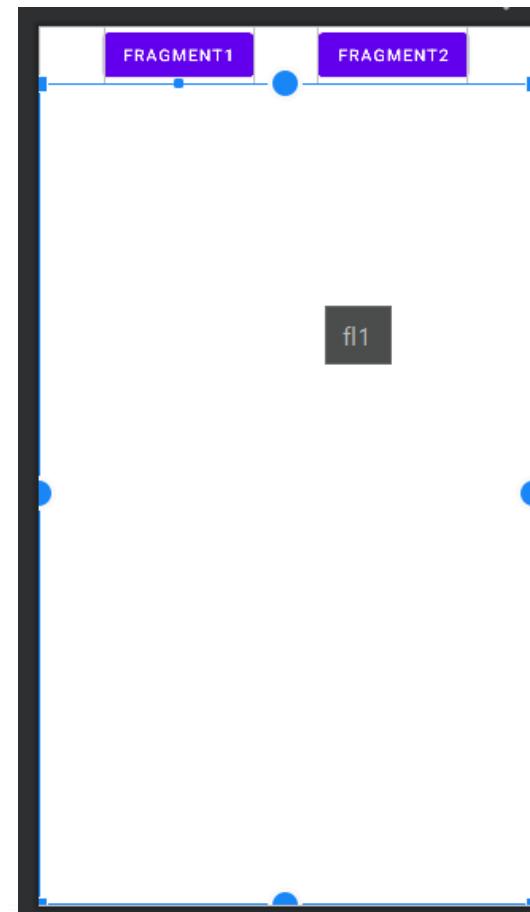
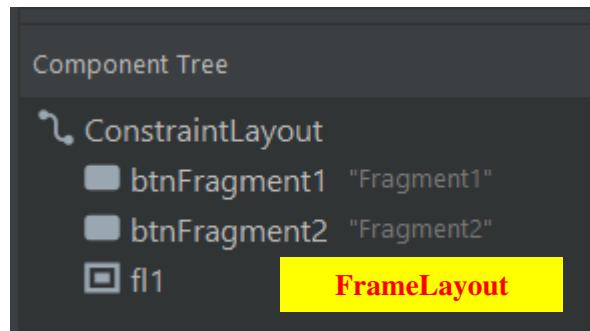
- ✓ *Fragment cũng có vòng đời riêng của nó.*
- ✓ *Có thể thêm, xóa, thay đổi fragment trong activity khi activity vẫn đang chạy.*
- ✓ *Một Fragment có thể được sử dụng trong nhiều Activities.*
- ✓ *Vòng đời của Fragment có quan hệ chặt chẽ với vòng đời của Activity đang dùng, khi Activity bị tạm dừng thì các Fragment sẽ dừng lại.*



1

## Fragments kotlin

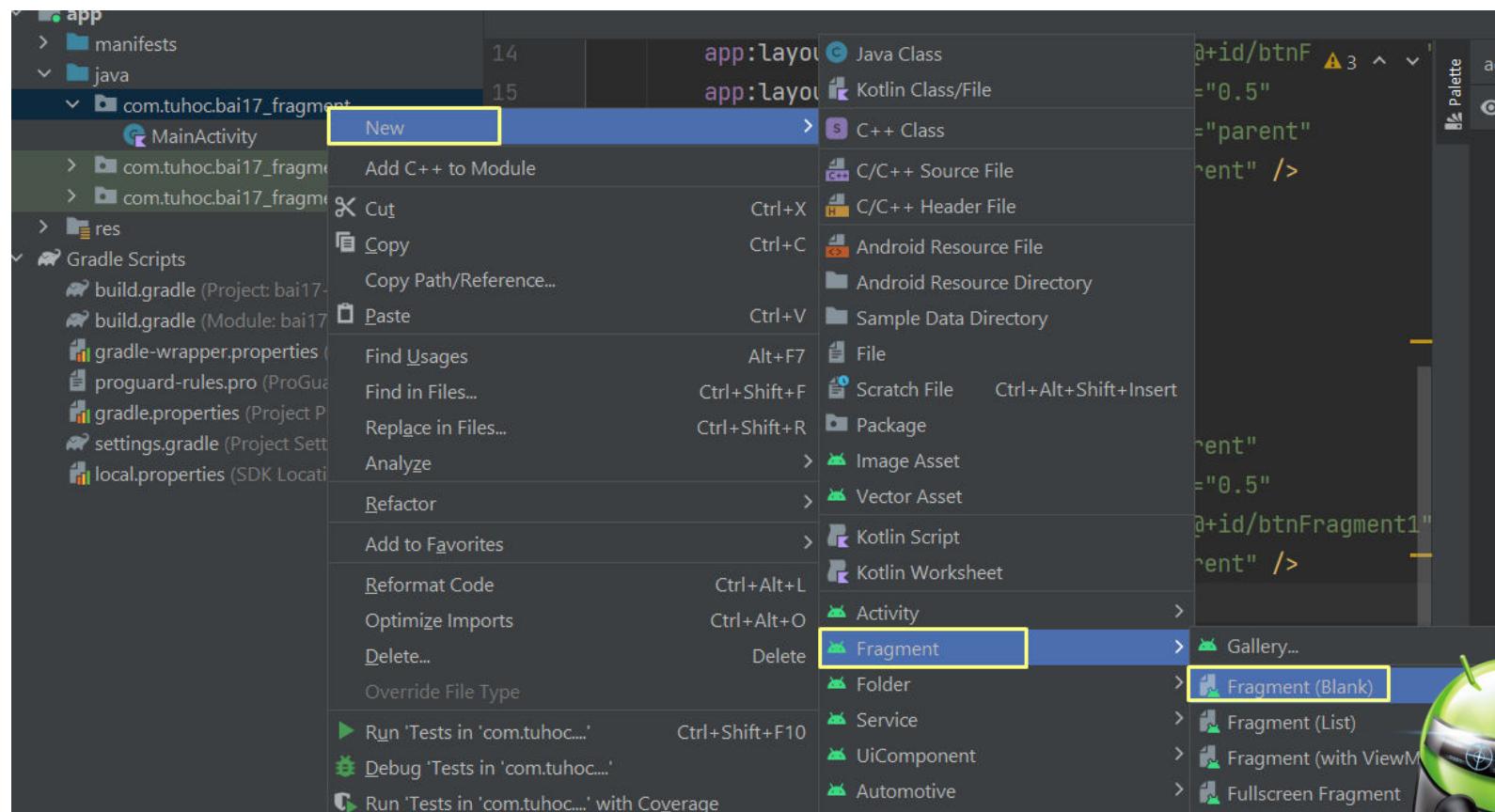
## □ 3. Thiết kế ứng dụng demo Fragment :



## 1

## Fragments kotlin

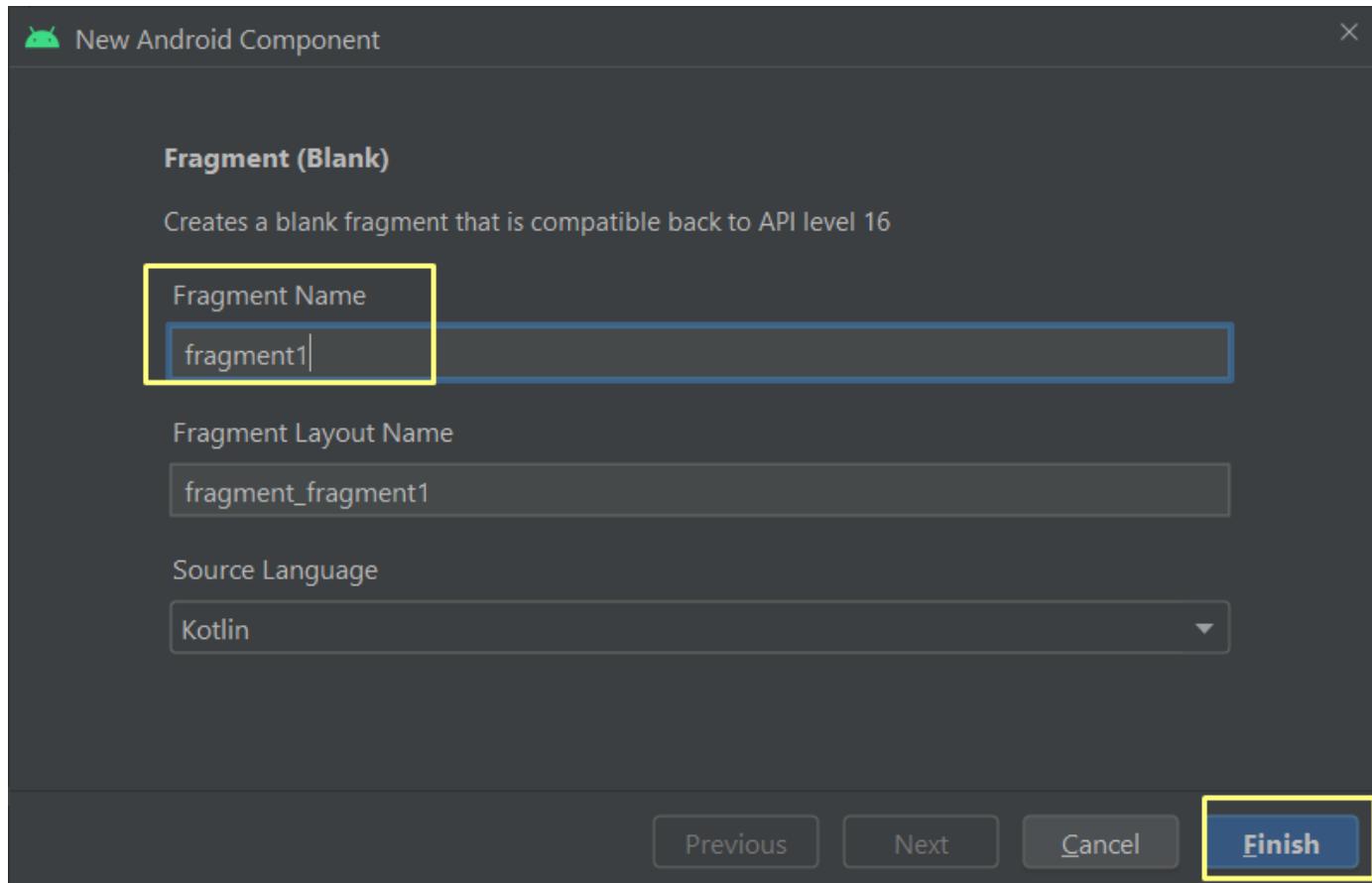
## □ 4. Tạo mới fragment:



1

## Fragments kotlin

## □ 4. Tạo mới fragment:



## 1

## Fragments kotlin

## □ 6. Code fragment1.kt :

```
package com.tuhoc.bai17_fragment

import ...

// TODO: Rename parameter arguments, choose names that match
// the fragment initialization parameters, e.g. ARG_ITEM_NUM
private const val ARG_PARAM1 = "param1"
private const val ARG_PARAM2 = "param2"

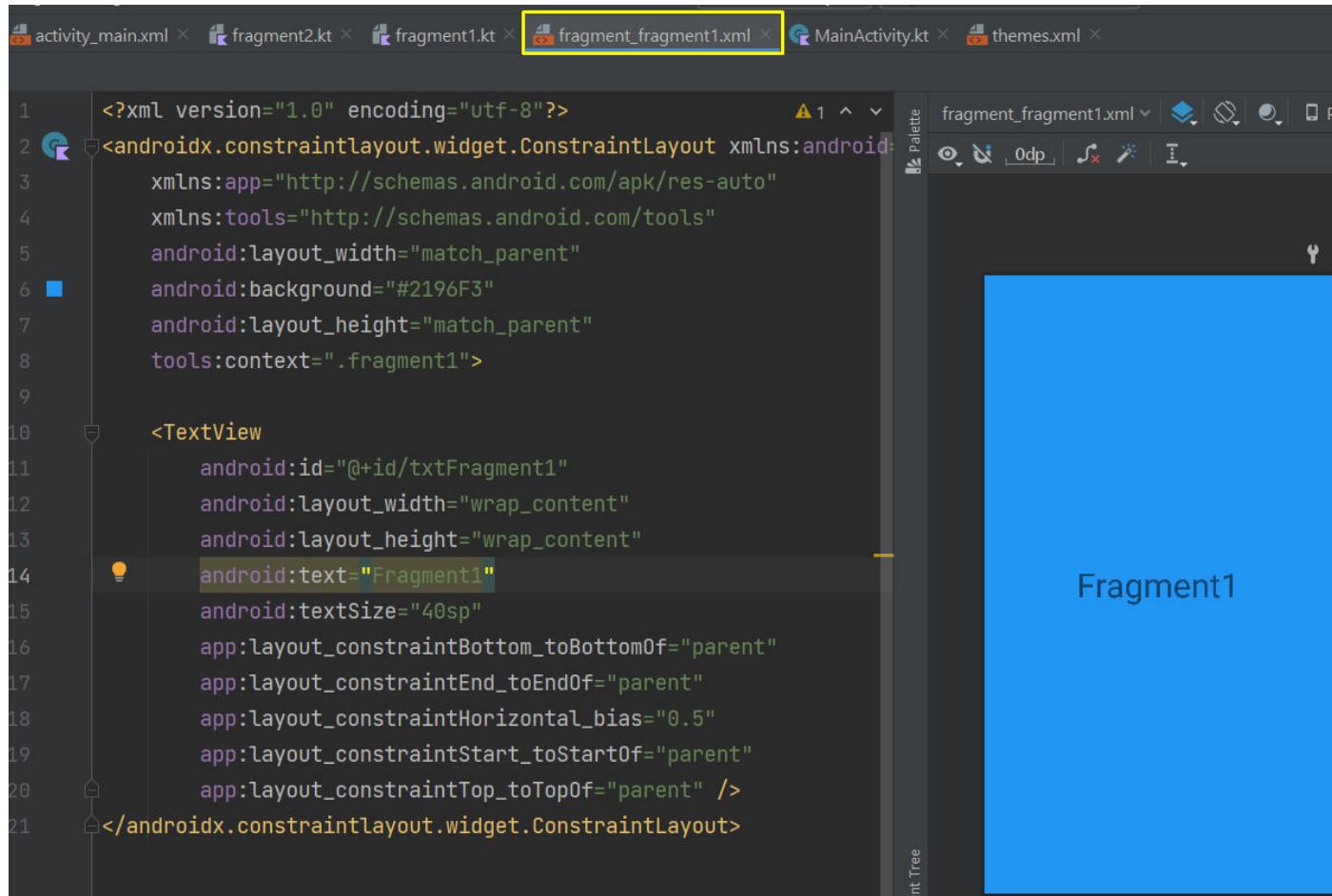
/**
 * A simple [Fragment] subclass.
 * Use the [fragment1.newInstance] factory method to
 * create an instance of this fragment.
 */
class fragment1 : Fragment(R.layout.fragment_fragment1) {
```



## 1

## Fragments kotlin

## □ 5. Thiết kế giao diện cho fragment 1 :



The screenshot shows the Android Studio interface with the following details:

- Top Bar:** activity\_main.xml, fragment2.kt, fragment1.kt, fragment\_fragment1.xml (highlighted with a yellow box), MainActivity.kt, themes.xml.
- Left Panel:** Shows the XML code for fragment\_fragment1.xml.
- Right Panel:** Shows the preview of the fragment layout, which contains a single TextView with the text "Fragment1".

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:background="#2196F3"
    android:layout_height="match_parent"
    tools:context=".fragment1">

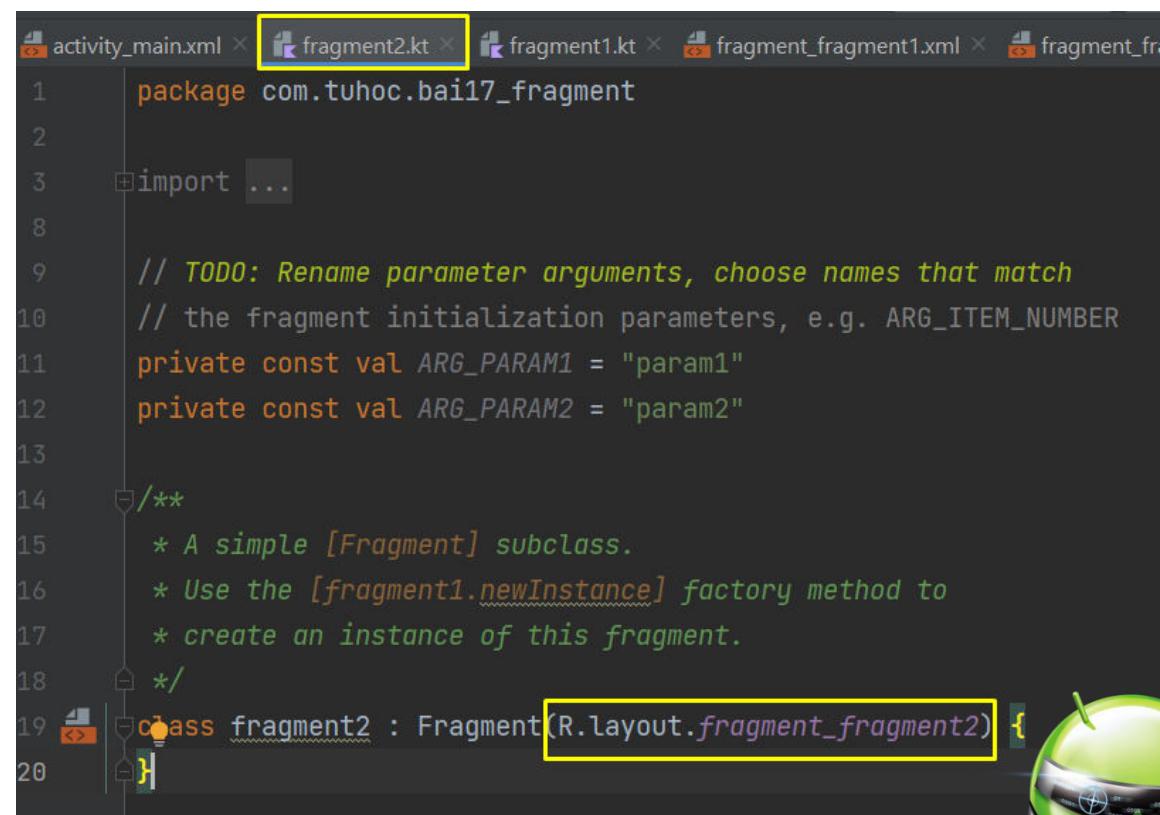
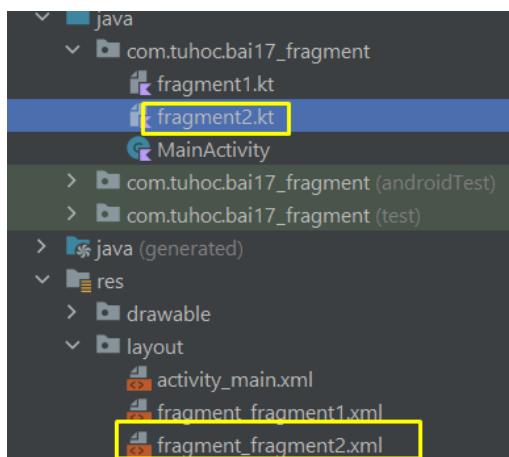
    <TextView
        android:id="@+id/txtFragment1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fragment1"
        android:textSize="40sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



## 1

## Fragments kotlin

## □ 6. Tạo fragment 2



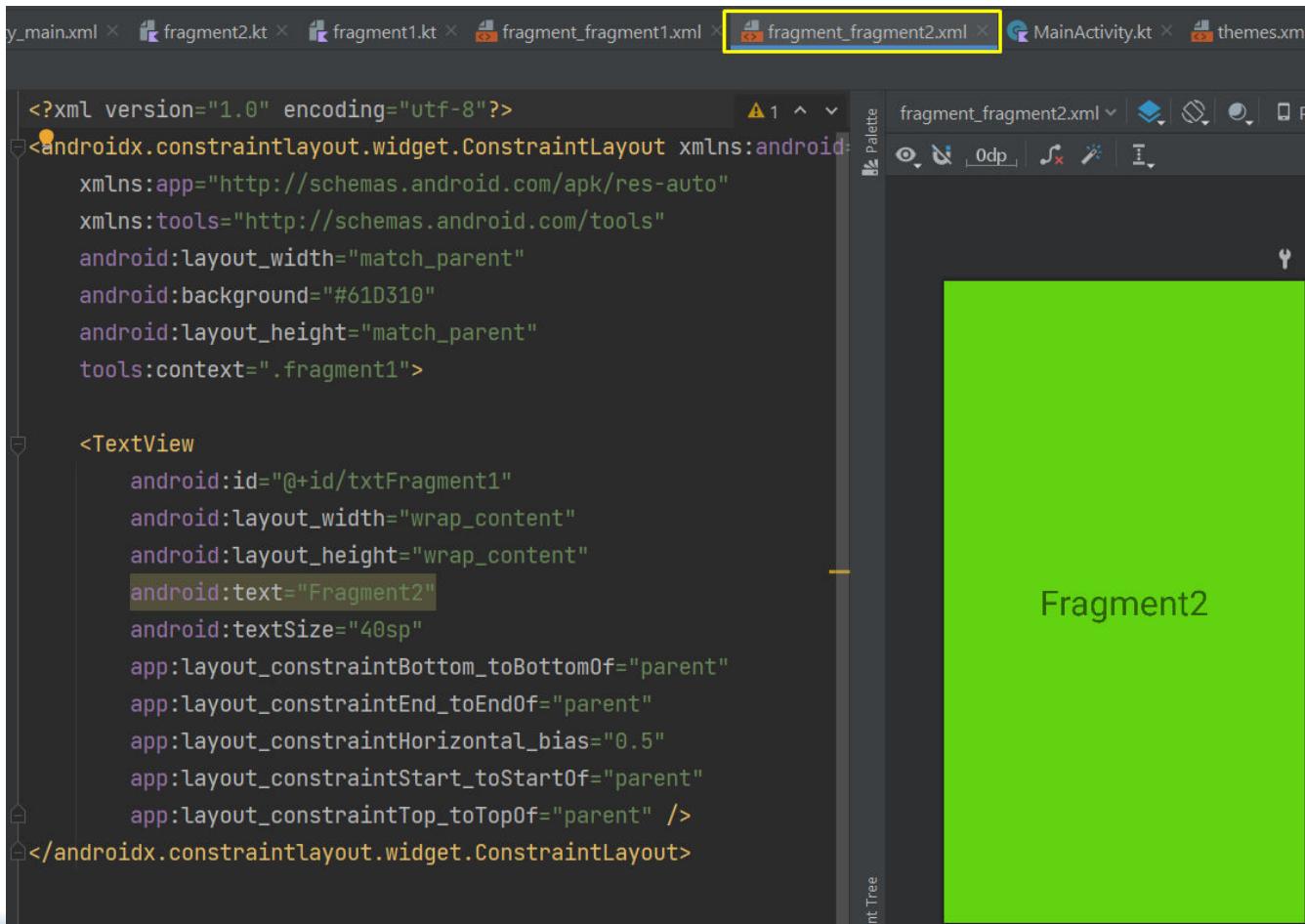
```
1 package com.tuhoc.bai17_fragment
2
3 import ...
4
5 // TODO: Rename parameter arguments, choose names that match
6 // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
7 private const val ARG_PARAM1 = "param1"
8 private const val ARG_PARAM2 = "param2"
9
10 /**
11  * A simple [Fragment] subclass.
12  * Use the [fragment1.newInstance] factory method to
13  * create an instance of this fragment.
14  */
15 class fragment2 : Fragment(R.layout.fragment_fragment2) {
16
17 }
```



## 1

## Fragments kotlin

## □ 6. Thiết kế giao diện fragment 2:



The screenshot shows the Android Studio interface with the XML layout file for Fragment 2 selected in the tab bar. The code in the editor defines a ConstraintLayout containing a single TextView with the text "Fragment2".

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:background="#61D310"
    android:layout_height="match_parent"
    tools:context=".fragment1">

    <TextView
        android:id="@+id/txtFragment1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fragment2"
        android:textSize="40sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

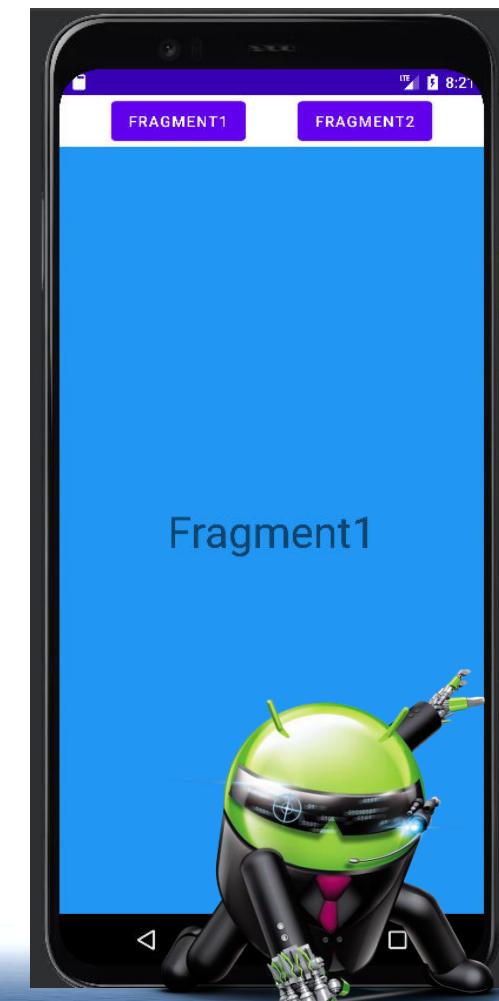


## 1

## Fragments kotlin

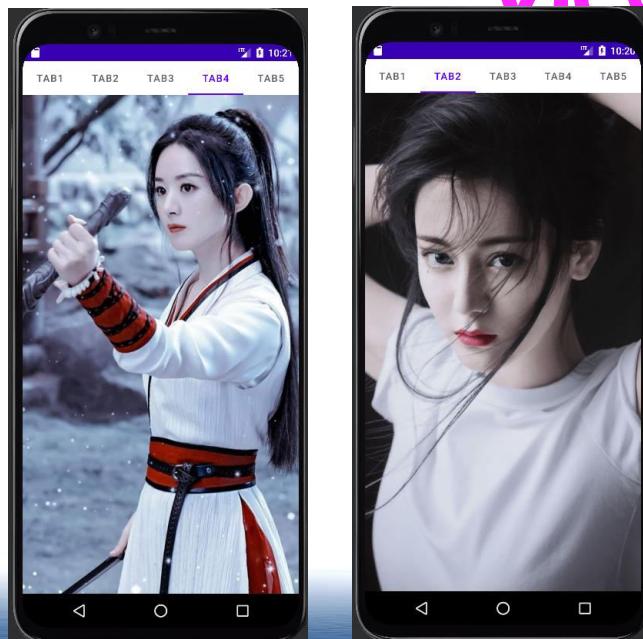
## □ 7. Code mainActivity gọi fragment :

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        val sub1 = fragment1()  
        val sub2 = fragment2()  
        supportFragmentManager.beginTransaction().apply { this: FragmentTransaction  
            replace(R.id.fl1, sub1)  
            commit()  
        }  
        btnFragment1.setOnClickListener { it: View!  
            supportFragmentManager.beginTransaction().apply { this: FragmentTransaction  
                replace(R.id.fl1, sub1)  
                commit()  
            }  
        }  
        btnFragment2.setOnClickListener { it: View!  
            supportFragmentManager.beginTransaction().apply { this: FragmentTransaction  
                replace(R.id.fl1, sub2)  
                commit()  
            }  
        }  
    }  
}
```





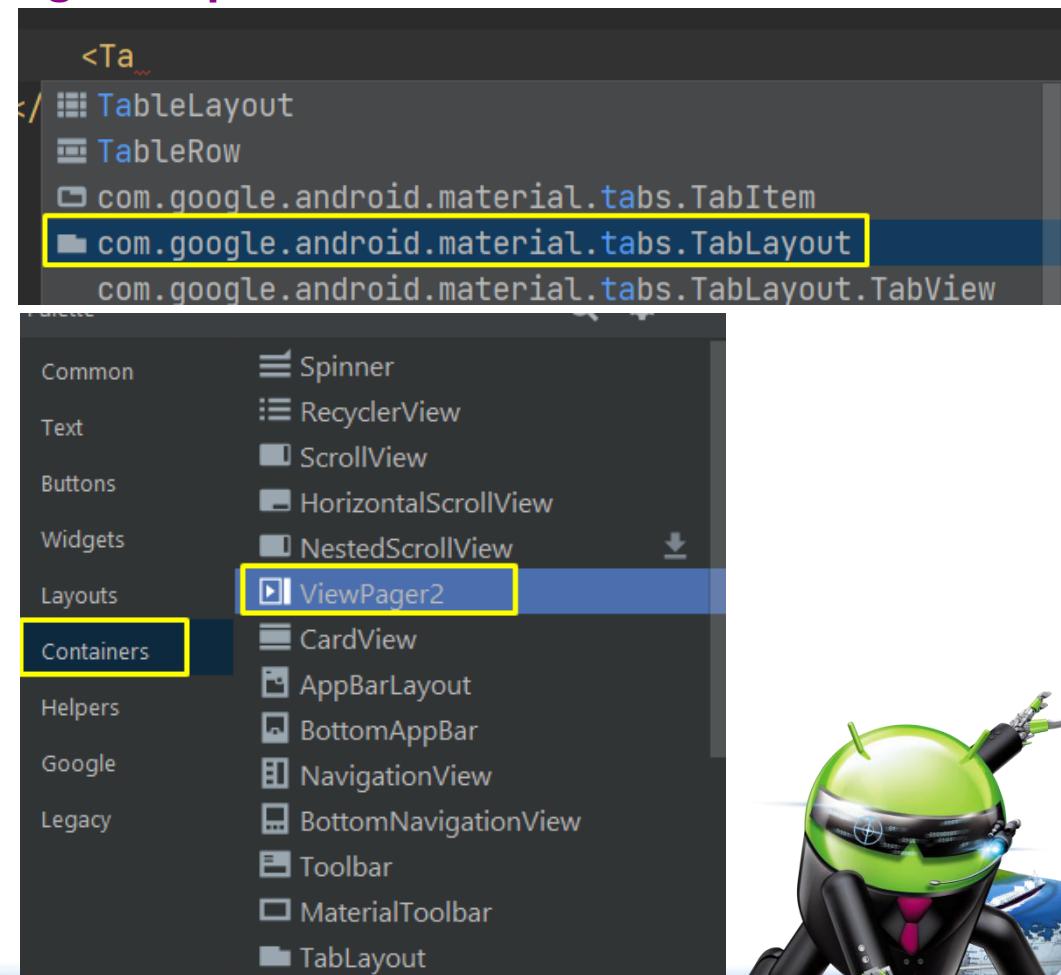
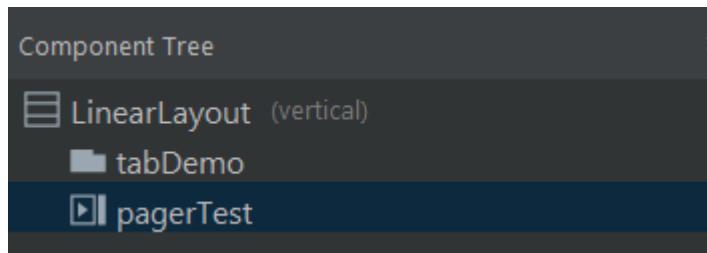
## TabLayout & ViewPager2 Kotlin



## 1

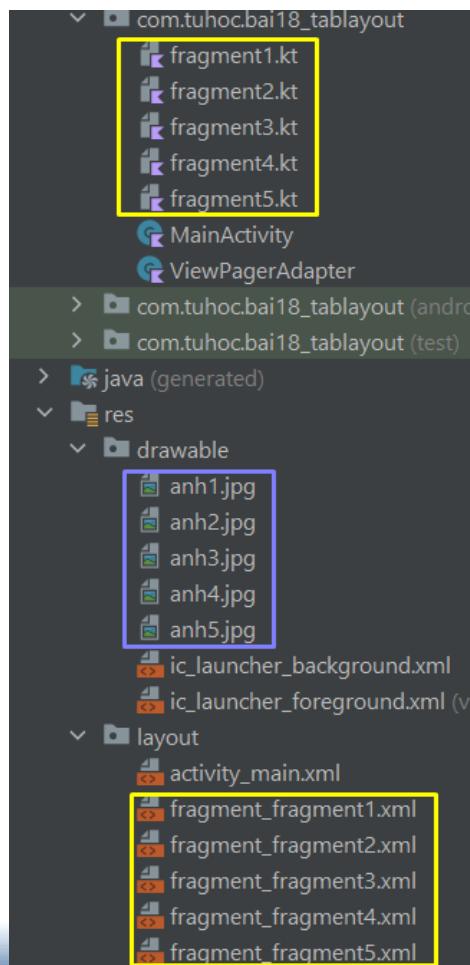
## TabLayout &amp; ViewPager2

## □ Add tablayout, viewpager2 vào giao diện :



## 1

## TabLayout &amp; ViewPager2

 2. Step 1- tạo các fragments– Đưa hình ảnh vào drawable:

Ảnh để trong drawable  
và code toàn bộ bài 18  
Pass unrar : **tuhoc180**

<http://android.tuhoc.cc/>

18.bai18-androikotlin.zip 

bai18tablayout  
hinhanh-bai18

hinhanh-bai18    Search hinhanh



anh1.jpg



anh2.jpg



anh3.jpg



anh4.jpg



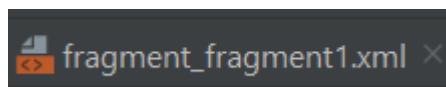
anh5.jpg



## 1

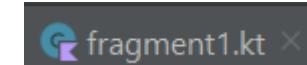
## TabLayout &amp; ViewPager2

## □ step 1. Code trong fragments :

 fragment\_fragment1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".fragment1">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:adjustViewBounds="true"
        android:scaleType="centerCrop"
        android:src="@drawable/anh1" />
</FrameLayout>
```

 fragment1.kt

```
class fragment1 : Fragment(R.layout.fragment_fragment1) {
```

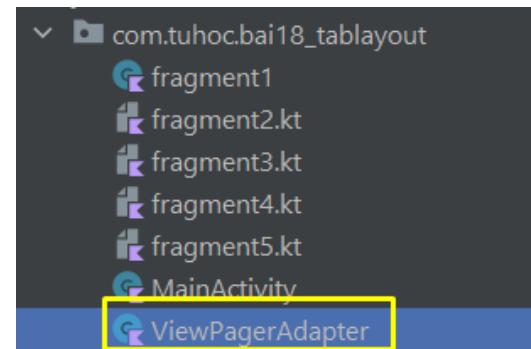


## 1

## TabLayout &amp; ViewPager2

## □ 3. Step 2- tạo class Adapter:

```
class ViewPagerAdapter(fragmentManager: FragmentManager, lifecycle: Lifecycle):  
    FragmentStateAdapter(fragmentManager, lifecycle) {  
    //ctrl + i  
    override fun getItemCount(): Int {  
        return 5  
    }  
    override fun createFragment(position: Int): Fragment {  
        return when(position){  
            0->{  
                fragment1()  
            }  
            1->{  
                fragment2()  
            }  
            2->{  
                fragment3()  
            }  
            3->{  
                fragment4()  
            }  
            else->{  
                fragment5()  
            }  
        }  
    }  
}
```

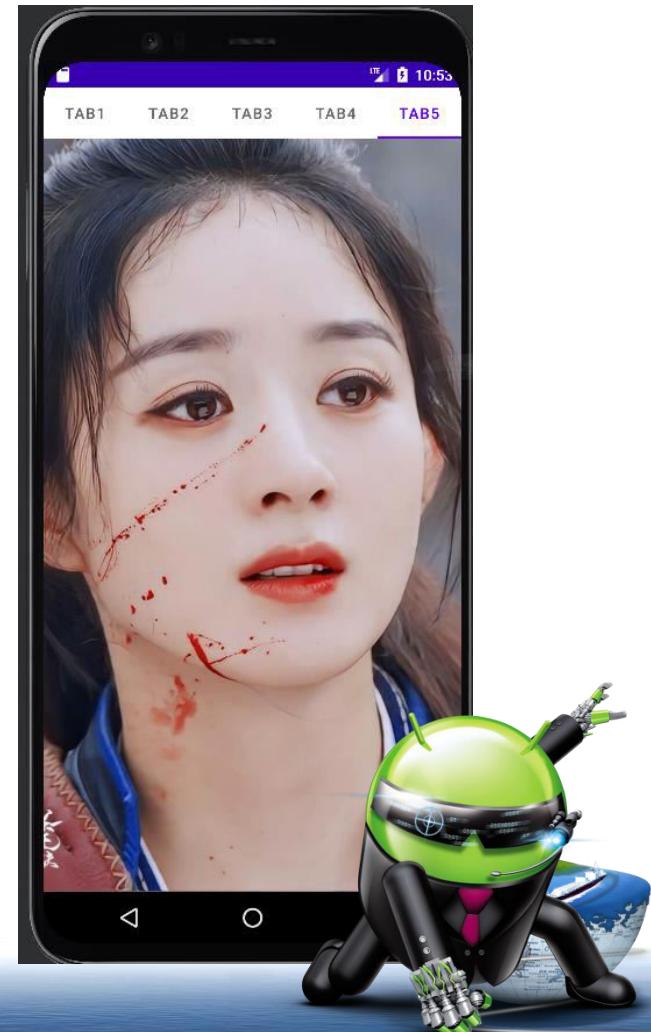


## 1

## TabLayout &amp; ViewPager2

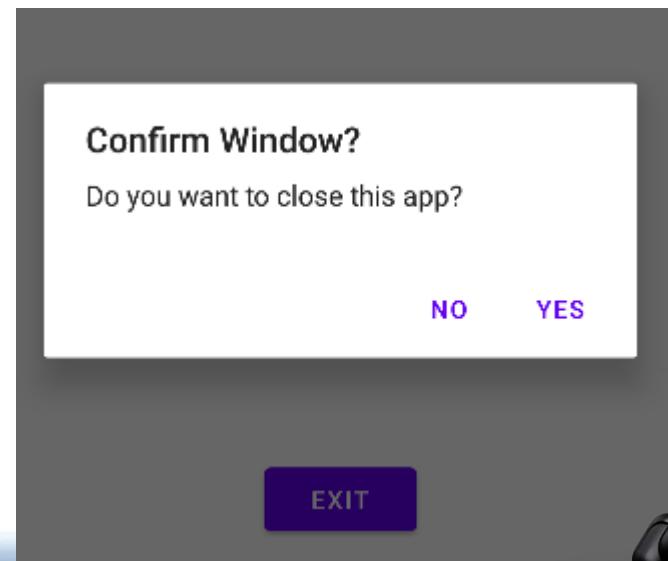
## □ 3. Step 3- Code MainActivity:

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val adapter = ViewPagerAdapter(supportFragmentManager, lifecycle)
        pagerTest.adapter = adapter
        TabLayoutMediator(tabDemo, pagerTest){tab, position ->
            when(position){
                0->{
                    tab.text="Tab1"
                }
                1->{
                    tab.text="Tab2"
                }
                2->{
                    tab.text="Tab3"
                }
                3->{
                    tab.text="Tab4"
                }
                4->{
                    tab.text="Tab5"
                }
            }
        }.attach()
    }
}
```





## AlertDialog

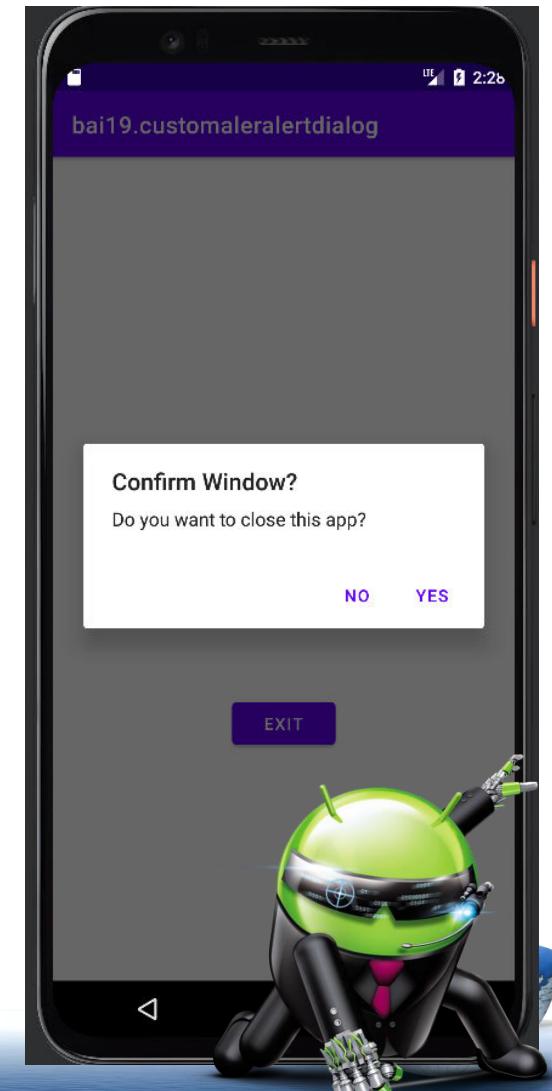


## 1

## AlertDialog Basic

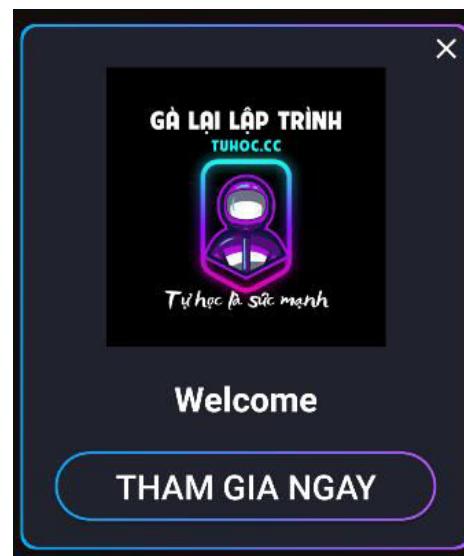
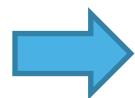
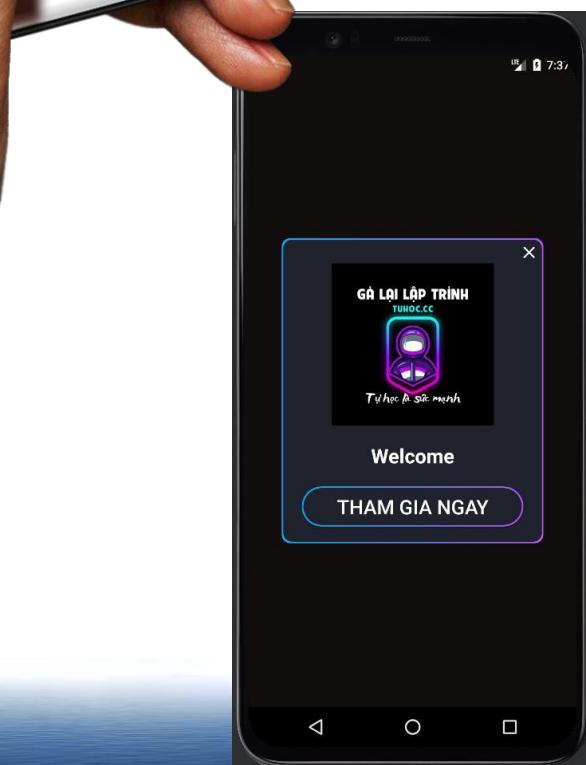
```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        btnExit.setOnClickListener { it: View!
            val dialog = AlertDialog.Builder(context: this)
            dialog.apply { this: AlertDialog.Builder
                //Tiêu đề thông báo
                setTitle("Confirm Window?")
                //Nội dung thông báo
                setMessage("Do you want to close this app?")
                //nút phủ định (không đồng ý)
                setNegativeButton(text: "No"){ dialogInterface: DialogInterface, i: Int ->
                    dialogInterface.dismiss()
                }
                //nút khẳng định, xác nhận có
                setPositiveButton(text: "Yes"){ dialogInterface: DialogInterface, i: Int ->
                    finish()
                }
                //ngăn đóng hộp thoại khi click ra ngoài hộp thoại
                setCancelable(false)
            }
            dialog.show()
        }
    }
}
```





## Custom AlertDialog

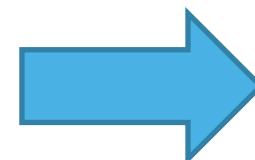


1

## Custom AlertDialog

### □ 1. Các bước xây dựng:

Step 1:  
Make xml layout  
CustomLayout



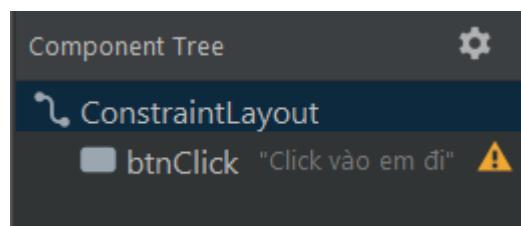
Step 2:  
Chuyển xml thành  
view



## 1

## Custom AlertDialog

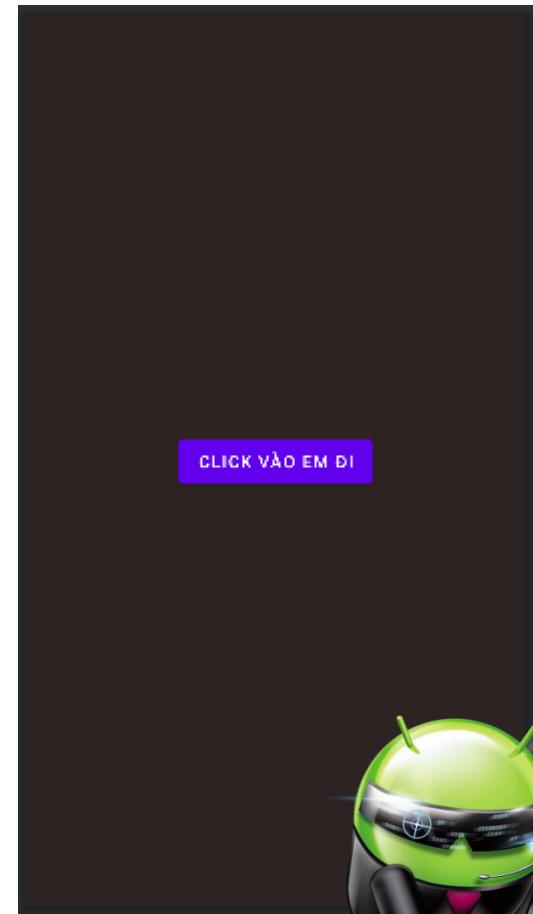
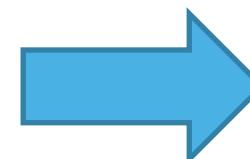
## □ 2. Xây dựng layout :



```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:background="#2E2323"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id	btnClick"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click vào em đi"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

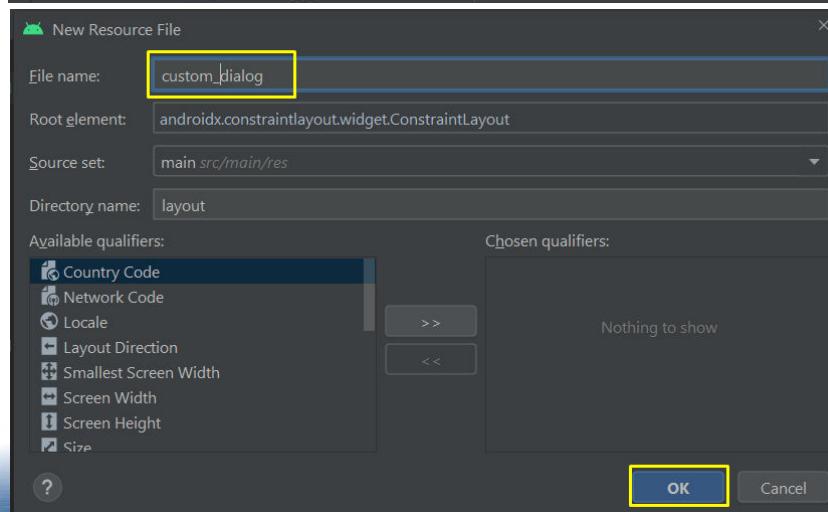
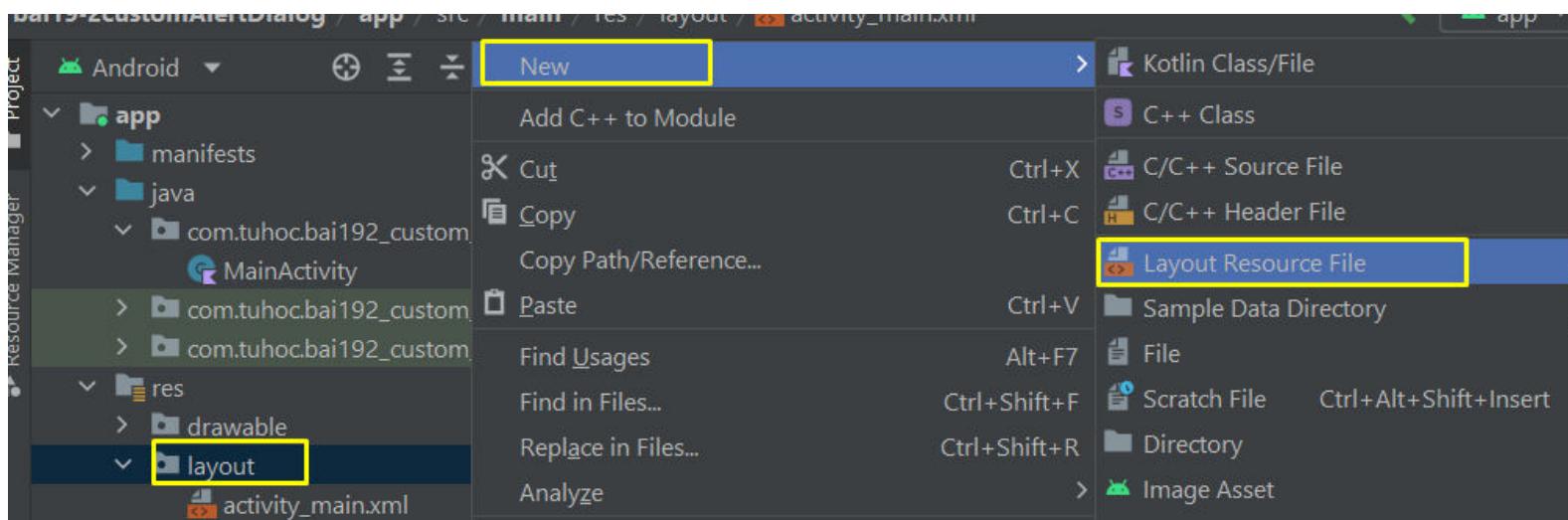
</androidx.constraintlayout.widget.ConstraintLayout>
```



1

## Custom AlertDialog

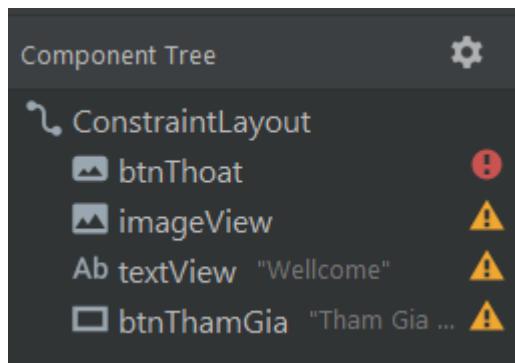
## ☐ Step 1: Xây dựng Custom layout xml :



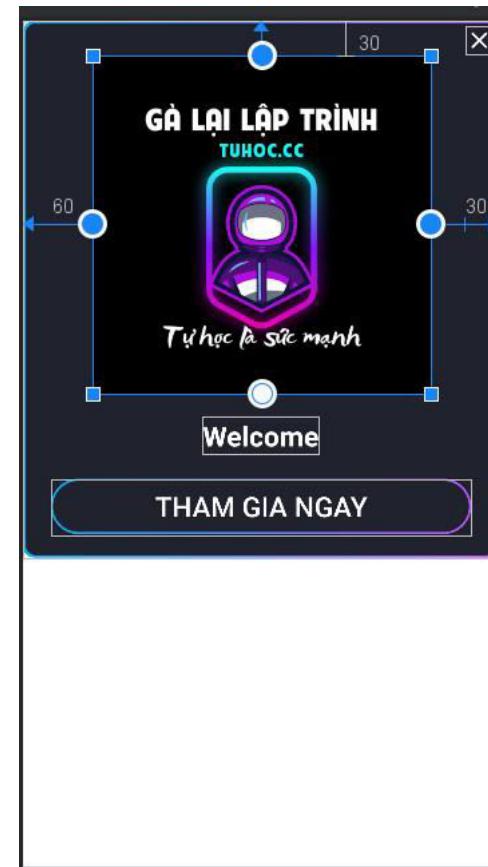
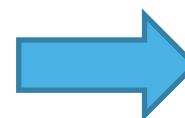
## 1

## Custom AlertDialog

## □ Step 1: Xây dựng Custom layout xml :

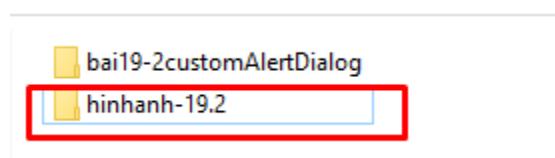


Ảnh đẻ trong drawable  
và code toàn bộ bài 19.2  
(API 24 android 7.0 )  
Pass unrar : **tuhoc192**



<http://android.tuhoc.cc/>

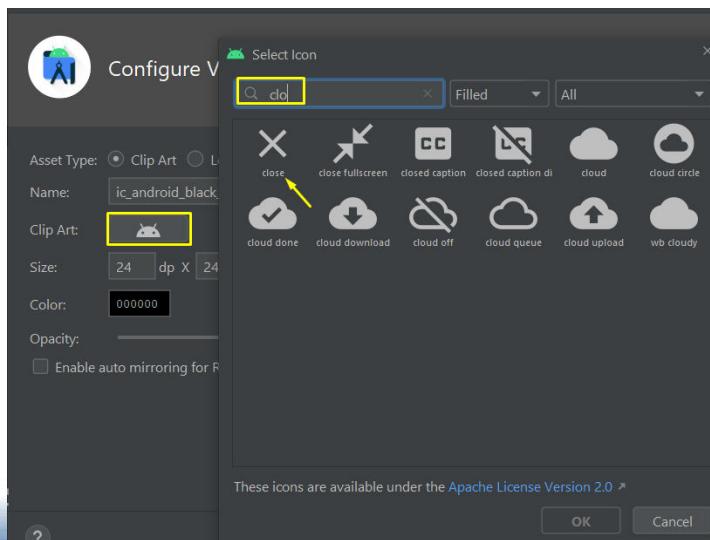
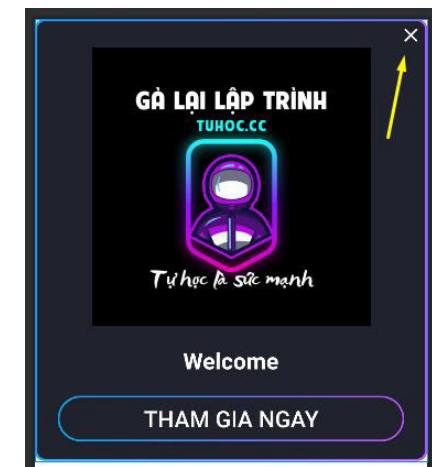
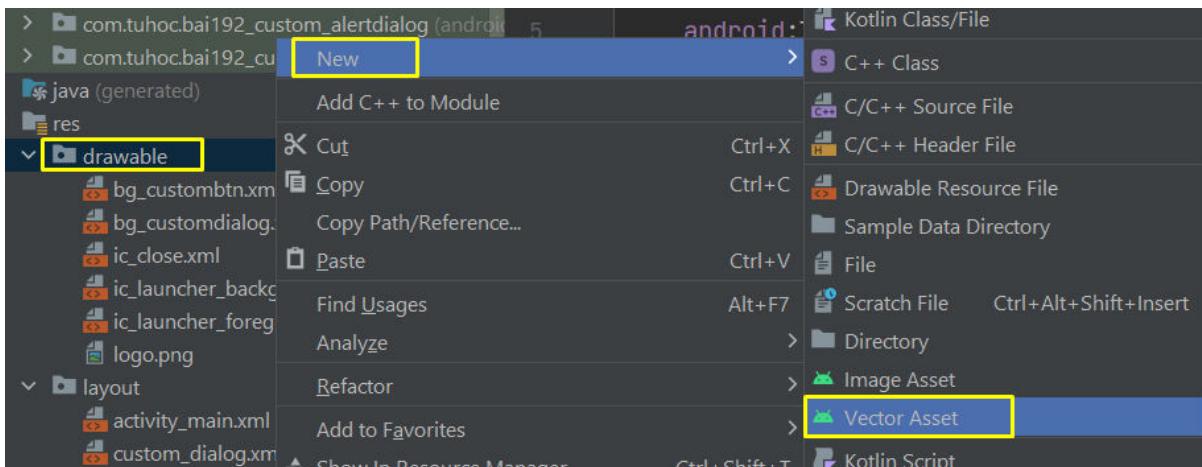
19.2-fullcode+hinhanh-dialog.zip



## 1

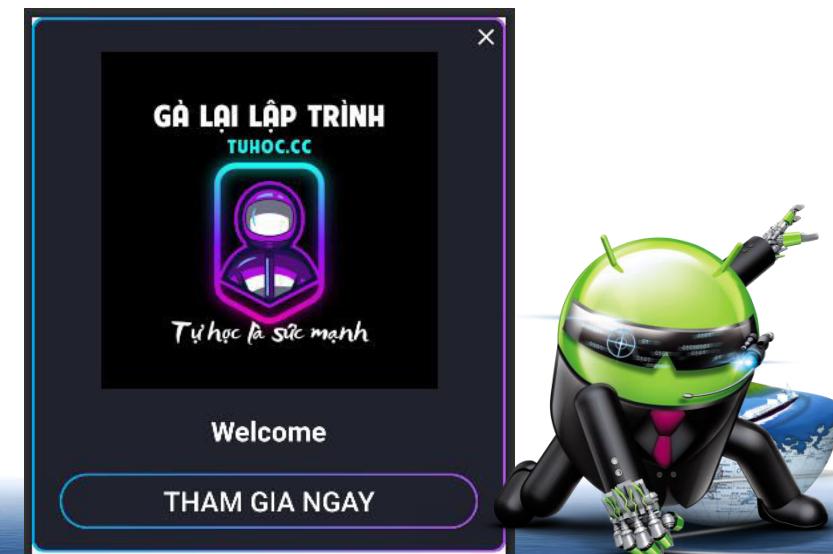
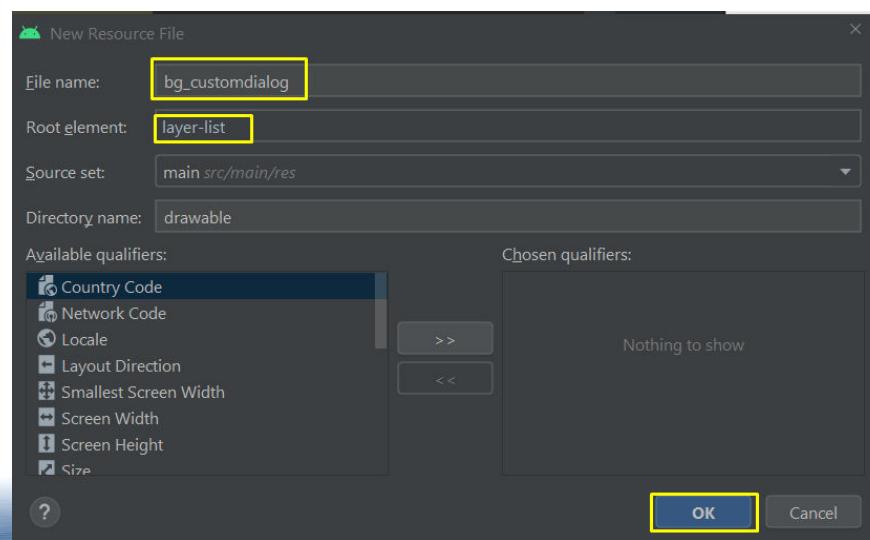
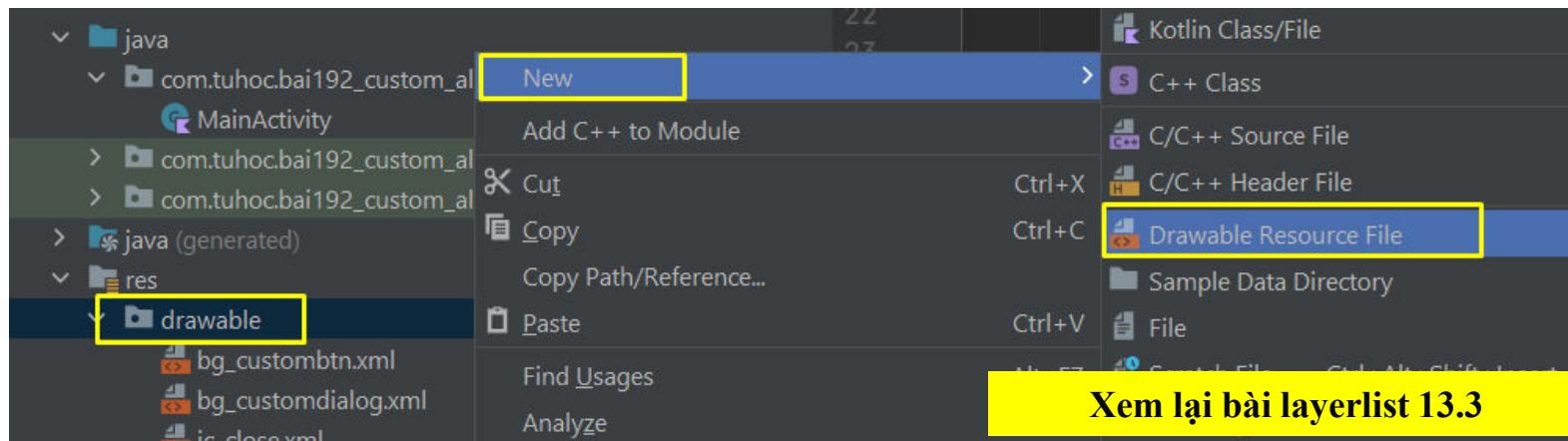
## Custom AlertDialog

## ☐ Step 1: Xây dựng Custom layout xml - Thêm nút x bằng vector



## 1

## Custom AlertDialog

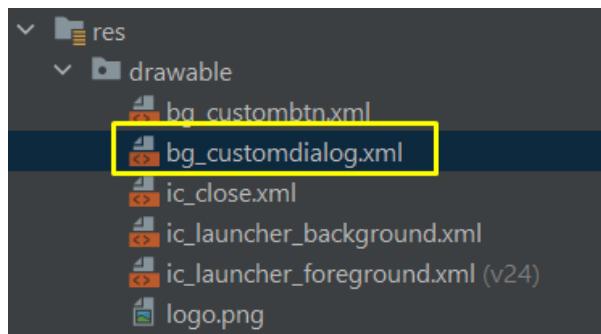
 Step 1: Xây dựng Custom layout xml - Gradient color bằng layerlist

## 1

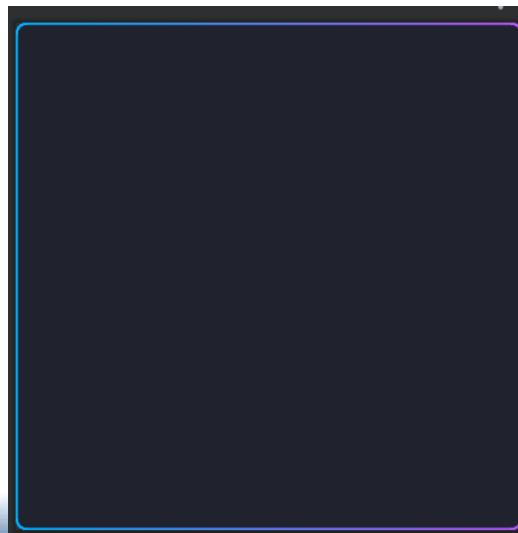
## Custom AlertDialog

 Step 1: Xây dựng Custom layout xml - Gradient color bằng layerlist

Code Gradien (Xem lại bài layerlist 13.3)



```
<layer-list xmlns:android="http://schemas.android.com  
    <item>  
        <shape android:shape="rectangle">  
            <gradient  
                android:angle="0"  
                android:startColor="#00AFFA"  
                android:endColor="#BA53FA"/>  
            <corners android:radius="10dp"/>  
        </shape>  
    </item>  
  
    <item  
        android:bottom="2dp"  
        android:top="2dp"  
        android:left="2dp"  
        android:right="2dp">  
  
        <shape android:shape="rectangle">  
            <solid android:color="#20222D"/>  
            <corners android:radius="10dp"/>  
        </shape>  
    </item>  
</layer-list>
```



## 1

## Custom AlertDialog

## □ Step 2: Code Main Activity- cách 1: Normal

```
class MainActivity : AppCompatActivity() {
    lateinit var dialog: AlertDialog
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

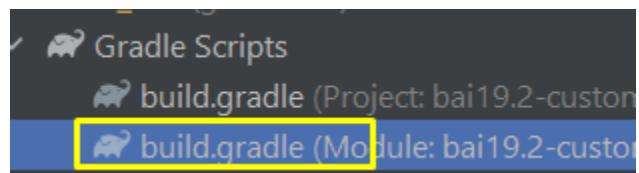
        btnClick.setOnClickListener { it: View!
            showDialogNormal()
        }
    }
    private fun showDialogNormal() {
        val build = AlertDialog.Builder(context: this)
        val view = layoutInflater.inflate(R.layout.custom_dialog, root: null)
        build.setView(view)
        dialog = build.create()
        dialog.show()
    }
}
```



## 1

## Custom AlertDialog

## □ Step 2: Code Main Activity- cách 2: ViewBinding



```
    kotlinOptions {
        jvmTarget = '1.8'
    }
    viewBinding {
        enabled = true
    }
}
dependencies {
    implementation 'androidx.core:core-ktx:1.7.0'
```

```
class MainActivity : AppCompatActivity() {
    lateinit var dialog: AlertDialog
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        btnClick.setOnClickListener { it: View!
            showDialogBinding()
        }
    }
    private fun showDialogBinding() {
        val build = AlertDialog.Builder(context: this)
        val dialogBinding = CustomDialogBinding
            .inflate(LayoutInflater.from(context: this))
        build.setView(dialogBinding.root)
        dialog = build.create()
        dialog.show()
    }
}
```



## 1

## Custom AlertDialog

## □ Step 2.2: Code Main Activity - : Code bntThoat, btnThamGia

```
private fun showDialogBinding() {
    val build = AlertDialog.Builder(context: this)
    val dialogBinding = CustomDialogBinding
        .inflate(LayoutInflater.from(context: this))
    dialogBinding.btnThoat.setOnClickListener { dialog.dismiss() }
    dialogBinding.btnThamGia.setOnClickListener { it: View!
        Toast.makeText(context: this,
            text: "Bạn đã tham gia thành công",
            Toast.LENGTH_SHORT).show() }
    build.setView(dialogBinding.root)
    dialog = build.create()
    dialog.show()
}
```

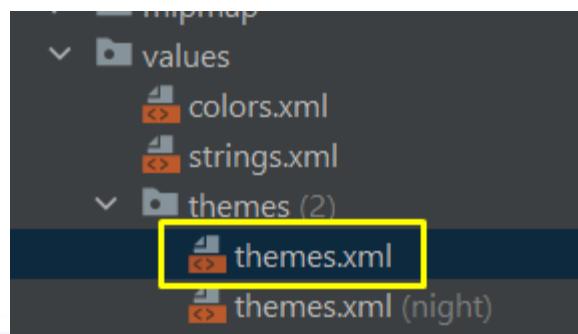
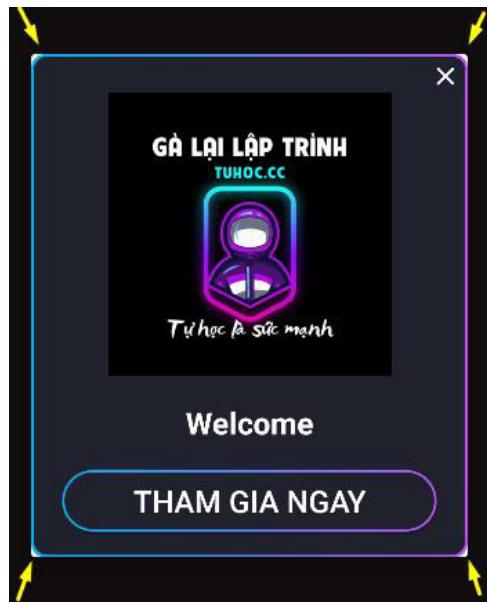
```
private fun showDialogNormal() {
    val build = AlertDialog.Builder(context: this,R.style.Themecustom)
    val view = LayoutInflater.inflate(R.layout.custom_dialog, root: null)
    build.setView(view)
    //code nút close
    val btnClose = view.findViewById<ImageButton>(R.id.btnClose)
    btnClose.setOnClickListener { dialog.dismiss() }
    //code nút tham gia
    val btnThamGia = view.findViewById<Button>(R.id.btnThamGia)
    btnThamGia.setOnClickListener { it: View!
        Toast.makeText(context: this,
            text: "Bạn đã tham gia thành công",
            Toast.LENGTH_SHORT).show() }
    dialog = build.create()
    dialog.show()
}
```



## 1

## Custom AlertDialog

Step 2.3: Code Main Activity- Thiết lập nền trong suốt



```
<style name="Themecustom" parent="Theme.AppCompat.Light.Dialog.Alert">
    <item name="android:windowBackground">@android:color/transparent</item>
    <item name="android:windowMinWidthMinor">80%</item>
</style>
```

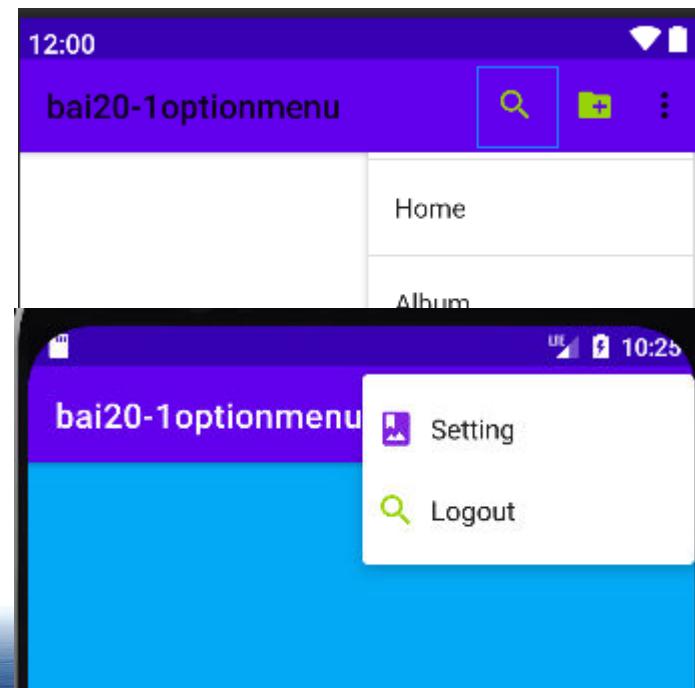
```
private fun showDialogNormal() {
    val build = AlertDialog.Builder(context: this, R.style.Themecustom)
    val view = LayoutInflater.inflate(R.layout.custom_dialog, root: null)
    build.setView(view)
    //code nút close
    val btnClose = view.findViewById<ImageButton>(R.id.btnClose)
    btnClose.setOnClickListener { dialog.dismiss() }
    //code nút tham gia
    val btnThamGia = view.findViewById<Button>(R.id.btnThamGia)
    btnThamGia.setOnClickListener { it: View!
        Toast.makeText(context: this,
            text: "Bạn đã tham gia thành công",
            Toast.LENGTH_SHORT).show()
    }
    dialog = build.create()
    dialog.show()
}
```

```
private fun showDialogBinding() {
    val build = AlertDialog.Builder(context: this, R.style.Themecustom)
    val dialogBinding = CustomDialogBinding
        .inflate(LayoutInflater.from(context: this))
    dialogBinding.btnClose.setOnClickListener { dialog.dismiss() }
    dialogBinding.btnThamGia.setOnClickListener { it: View!
        Toast.makeText(context: this,
            text: "Bạn đã tham gia thành công",
            Toast.LENGTH_SHORT).show()
    }
    build.setView(dialogBinding.root)
    dialog = build.create()
    dialog.show()
}
```





## OptineMenu Kotlin



1

Option Menu basic

2

Add icon in Options Menu



1

## Option Menu basic

### Các bước tạo menu cơ bản :

Step 1:  
**Make xml layout  
Menu**

Step 2:  
**Khởi tạo menu**

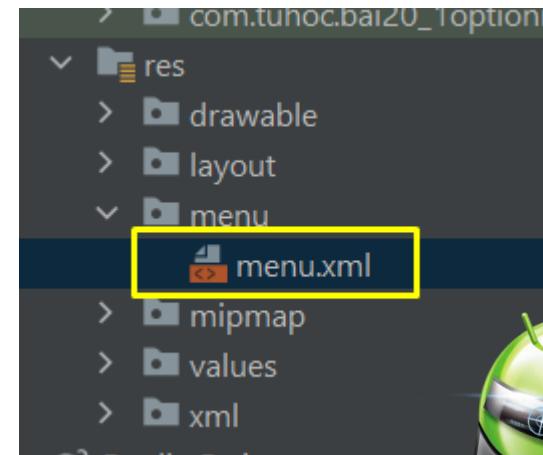
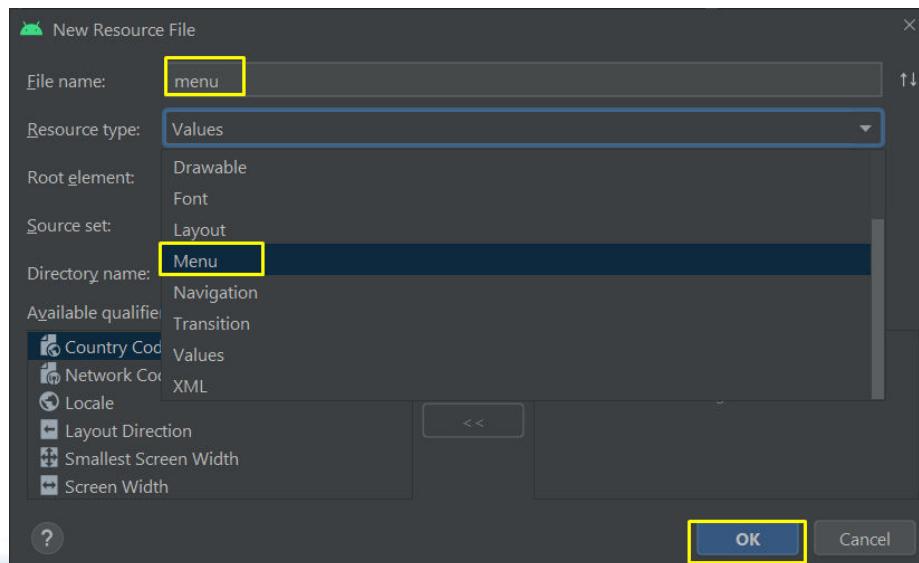
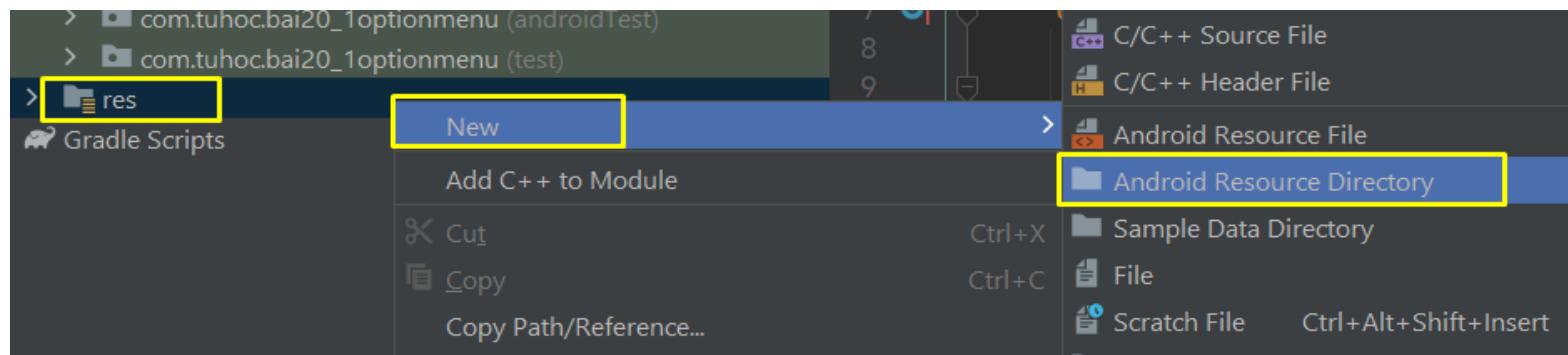
Step 3:  
**Xử lý sự kiện trên  
menu**



1

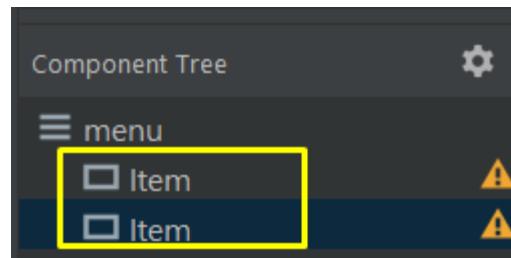
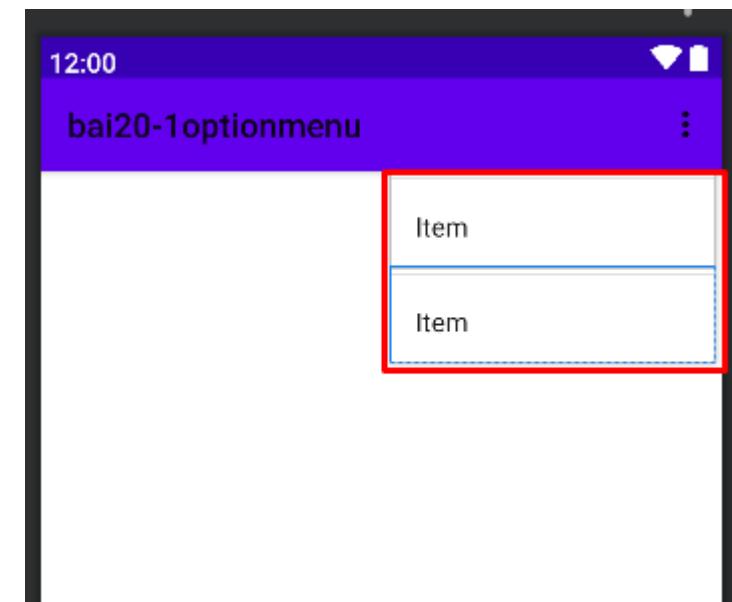
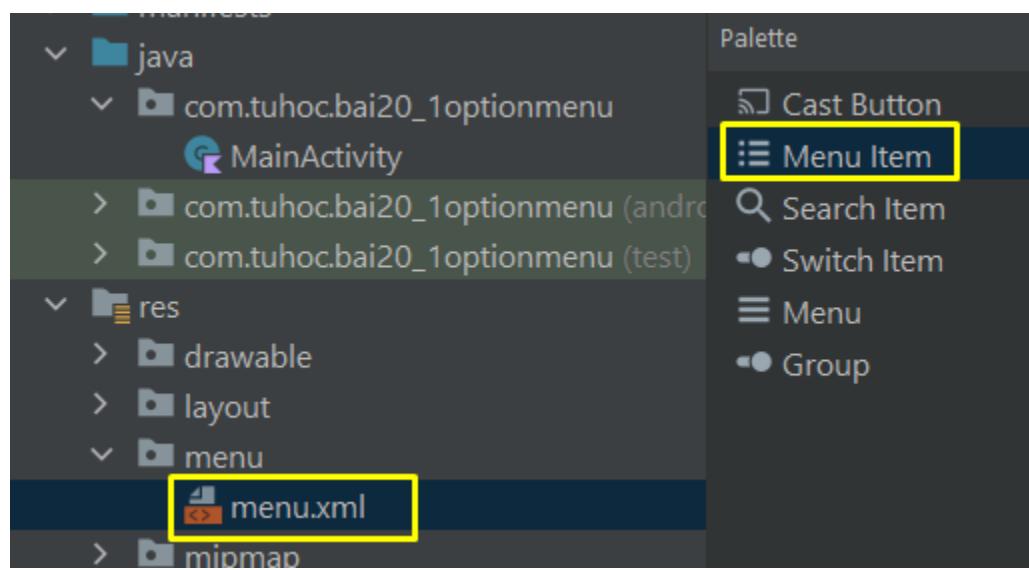
## Option Menu basic

## □ Step 1. Tạo menu.xml:



## 1

## Option Menu basic

 Step 1. Tạo menu.xml: Tạo Menu Item trong menu.xml

```
<item android:title="Item" />
<item android:title="Item" />
```



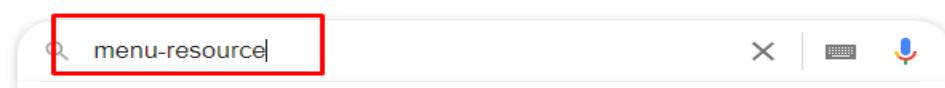
## 1

## Option Menu basic

Các thuộc tính quan trọng thường gặp :

STT	Thuộc tính	Nội dung	Chi tiết
1	<b>id</b>	Id của menu Item	<b>mnuXXX</b>
2	<b>title</b>	Tex hiển thị	Home, Exit...
3	<b>icon</b>	Biểu tượng của menu	
5	<b>showAsAction</b>	Đặt trong thanh ứng dụng	<b>ifRoom</b>

<https://developer.android.com/guide/topics/resources/menu-resource>

<https://developer.android.com/guide/topics/resources/menu-resource>

Menu resource | Android Developers

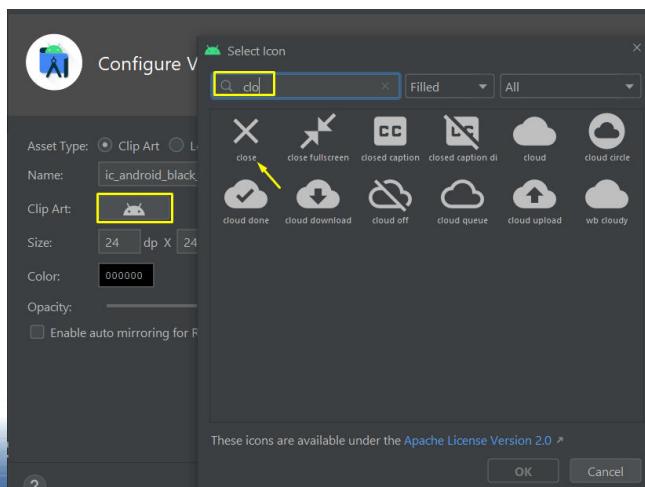
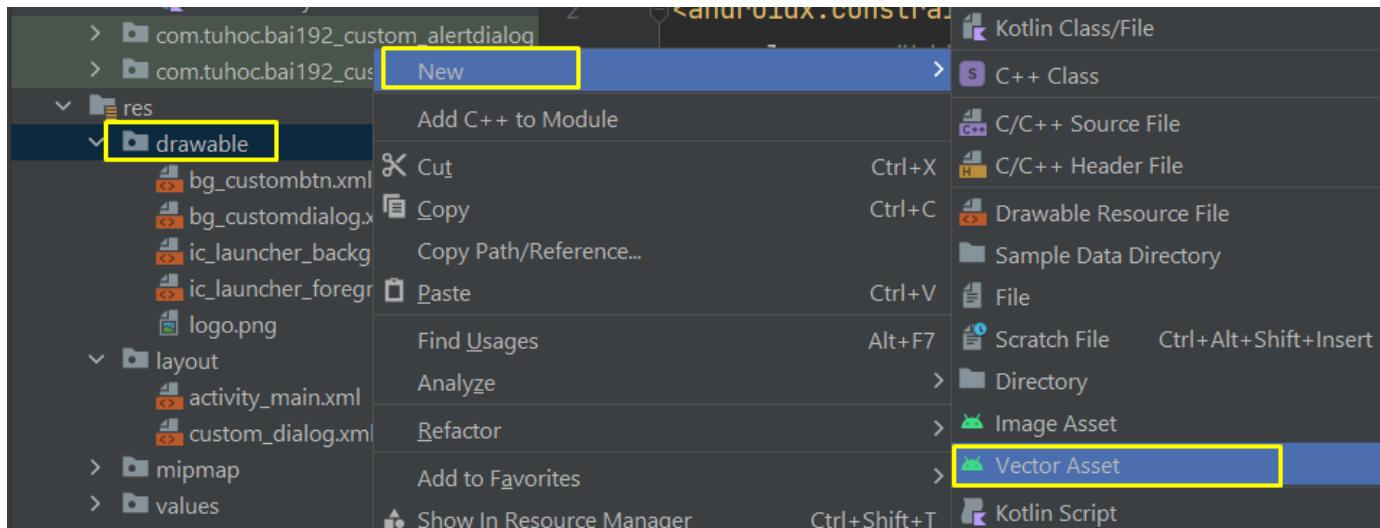
27 thg 10, 2021 — A **menu resource** defines an application menu (Options Menu, Context Menu, or submenu) that can be inflated with `MenuInflater`. For a guide to ...



## 1

## Option Menu basic

☐ Sử dụng icon trong thư viện có sẵn kotlin :

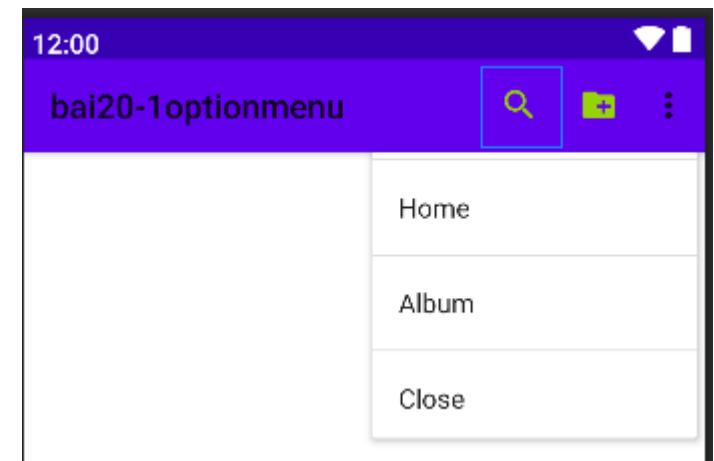


## 1

## Option Menu basic

## □ Code menu.xml :

```
5      <item
6          android:id="@+id/mnuHome"
7          android:icon="@drawable/ic_home"
8          android:title="Home" />
9
10     <item
11         android:id="@+id/mnuAlbum"
12         android:icon="@drawable/ic_album"
13         android:title="Album" />
14
15     <item
16         android:id="@+id/mnuClose"
17         android:icon="@drawable/ic_close"
18         android:title="Close" />
19
20     <item
21         android:id="@+id/mnuSearch"
22         android:icon="@drawable/ic_search"
23         android:title="Tìm kiếm"
24         app:showAsAction="ifRoom" />
25
26     <item
27         android:id="@+id/mnuNewFolder"
28         android:icon="@drawable/ic_new_folder"
29         android:title="Tìm kiếm"
30         app:showAsAction="ifRoom" />
```



## 1

## Option Menu basic

## □ Step 2. Khởi tạo menu từ file xml



```
activity_main.xml × themes.xml × MainActivity.kt × menu.xml ×
package com.tuhoc.bai20_1optionmenu

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
    //khởi tạo menu
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.menu, menu)
        return true
    }
}
```

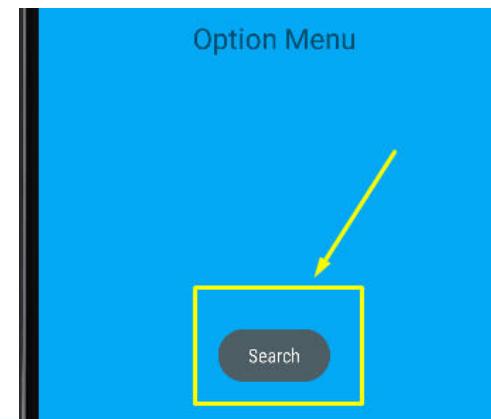
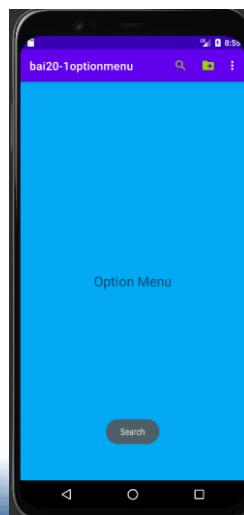
## 1

## Option Menu basic

## □ Step 3. Xử lý sự kiện trên menu Item



```
// lắng nghe xem item nào được chọn
override fun onOptionsItemSelected(item: MenuItem): Boolean {
    when(item.itemId){
        R.id.mnuHome -> Toast.makeText( context: this, text: "Home",Toast.LENGTH_SHORT).show()
        R.id.mnuAlbum -> Toast.makeText( context: this, text: "Album",Toast.LENGTH_SHORT).show()
        R.id.mnuNewFolder -> Toast.makeText( context: this, text: "NewFolder",Toast.LENGTH_SHORT).show()
        R.id.mnuSearch -> Toast.makeText( context: this, text: "Search",Toast.LENGTH_SHORT).show()
        R.id.mnuClose -> finish()
    }
    return super.onOptionsItemSelected(item)
}
```



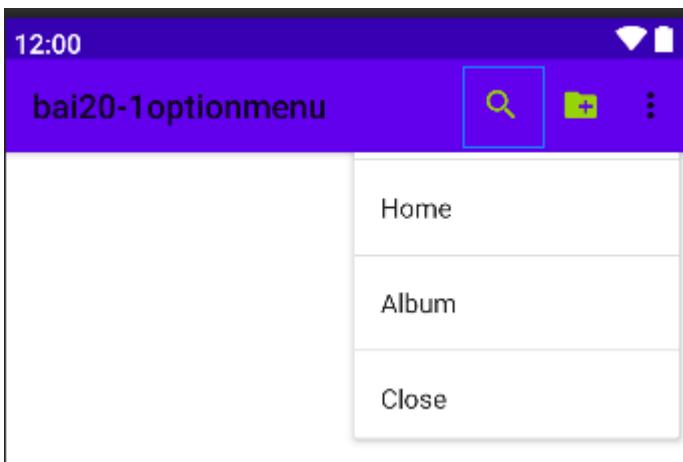
## 2

## Add icon in Options Menu

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/mnuHome"
        android:icon="@drawable/ic_home"
        android:title="Home" />

</menu>
```



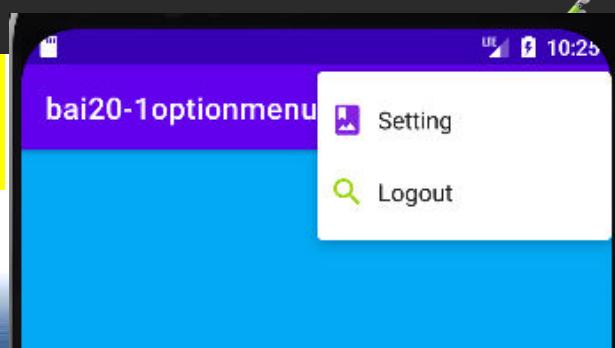
Menu basic

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/overflowMenu"
        android:icon="@drawable/ic_baseline_more_vert_24"
        android:title=""
        app:showAsAction="always" />

    <menu>
        <item
            android:icon="@drawable/ic_album"
            android:title="Setting" />
        <item
            android:icon="@drawable/ic_search"
            android:title="Logout" />
    </menu>
</item>
</menu>
```

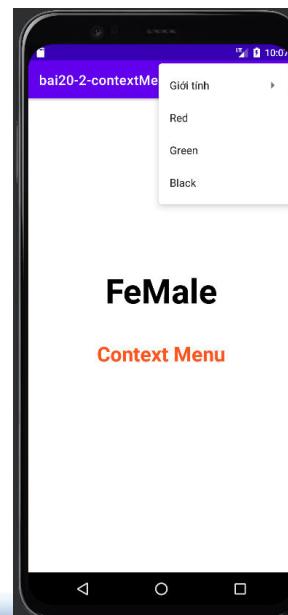
Sub menu  
with icon



Setting  
Logout



## ContextMenu Kotlin



### ContextMenu

Mời bạn chọn màu

Red

Green

Black



1

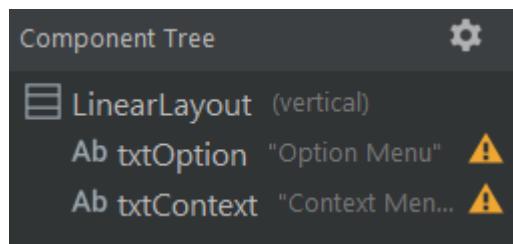
Option Menu part 2

2

ContextMenu

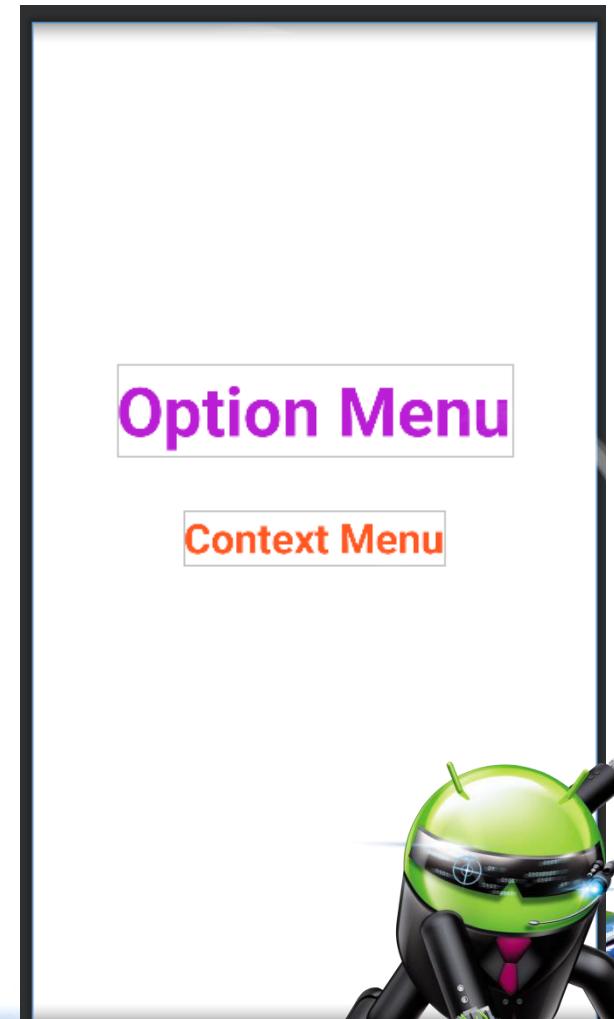
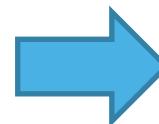
## 1

## Option Menu part 2

 Thiết kế giao diện cơ bản demo :

```
<TextView  
    android:id="@+id/txtOption"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:layout_marginTop="250dp"  
    android:text="Option Menu"  
    android:textStyle="bold"  
    android:textColor="#BA1ED4"  
    android:textSize="50sp" />
```

```
<TextView  
    android:id="@+id/txtContext"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:layout_marginTop="40dp"  
    android:text="Context Menu"  
    android:textColor="#FF5722"  
    android:textSize="30sp"  
    android:textStyle="bold" />
```



## 1

## Option Menu part 2

## □ 1.1 Khởi tạo menu bằng code trong MainActivity :

```
_main.xml x MainActivity.kt x
//khởi tạo menu (add bằng lệnh)
override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    //menu chính
    //add(int groupId, int itemId, int order, CharSequence title)
    menu?.add(1,1,1,"Red")
    menu?.add(1,2,2,"Green")
    menu?.add(1,3,3,"Black")
    //add sub menu
    var subMenu = menu?.addSubMenu("Giới tính")
    subMenu?.add(2,21,4,"Male")?.setChecked(false)
    subMenu?.add(2,22,5,"Female")?.setChecked(false)
    /* setGroupCheckable (int group,
        boolean checkable,
        boolean exclusive)
     1. group: id của group muốn có nút check
     2. boolean: true để cho phép dấu kiểm,
        false thành không cho phép. Mặc định sẽ là false
     3. Chon true để chỉ được chọn 1 item trong group
        Chon false tích chọn được nhiều lựa chọn
    */
    subMenu?.setGroupCheckable(2,true,true)
    return super.onCreateOptionsMenu(menu)
}
```



## 1

## Option Menu part 2

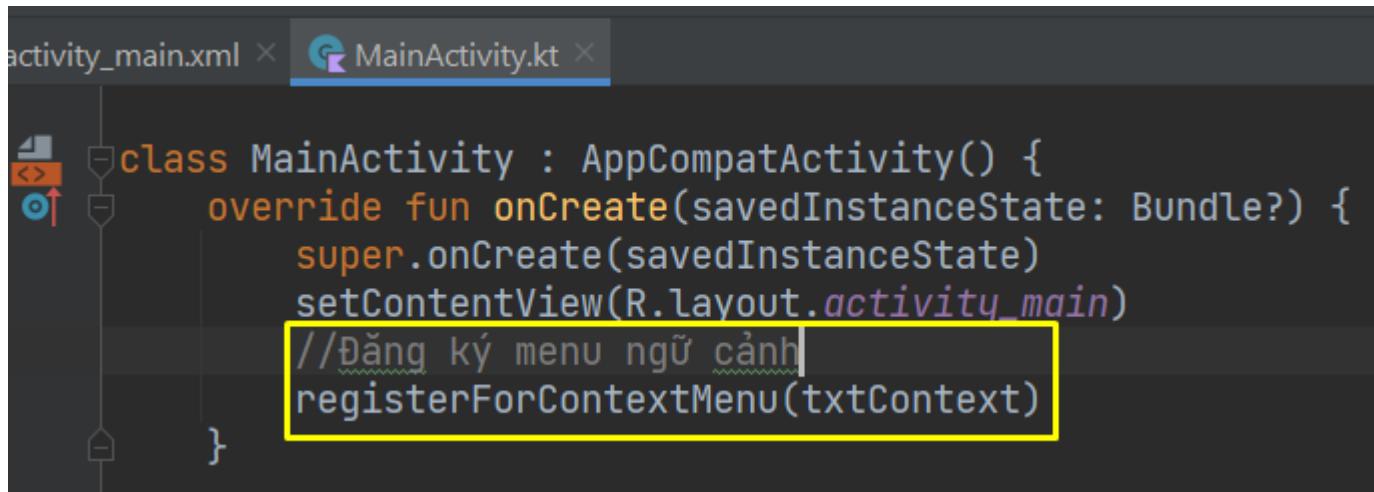
 1.2. Xử lý sự kiện trên menu Item:

```
//xử lý sự kiện trên item menu
override fun onOptionsItemSelected(item: MenuItem): Boolean {
    when(item.itemId){
        1->{txtOption.setTextColor(Color.RED)}
        2->{txtOption.setTextColor(Color.GREEN)}
        3->{txtOption.setTextColor(Color.BLACK)}
        //tương tác với submenu
        21->{txtOption.setText("Male")}
        22->{txtOption.setText("FeMale")}
    }
    return super.onOptionsItemSelected(item)
}
```



## 2

## ContextMenu

 2.1 Đăng ký menu ngữ cảnh cho view:

```
activity_main.xml × MainActivity.kt ×

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        //Đăng ký menu ngữ cảnh
        registerForContextMenu(txtContext)
    }
}
```



2

## ContextMenu

### □ 2.2 Khởi tạo context menu :

```
//khởi tạo context menu
override fun onCreateContextMenu(
    menu: ContextMenu?,
    v: View?,
    menuInfo: ContextMenu.ContextMenuItemInfo?
) {
    super.onCreateContextMenu(menu, v, menuInfo)
    menu?.add(3,31,1,"Red")
    menu?.add(3,32,1,"Green")
    menu?.add(3,33,1,"Black")
    menu?.setHeaderTitle("Mời bạn chọn màu")
}
```



2

## ContextMenu

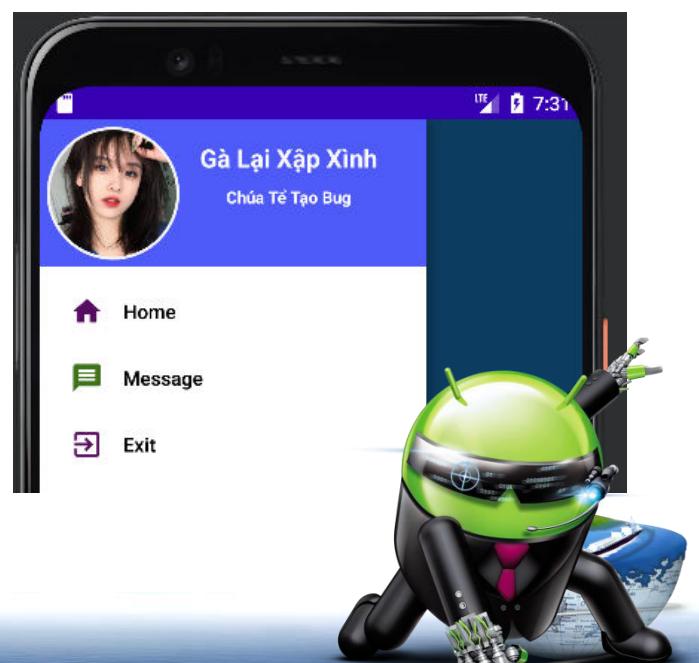
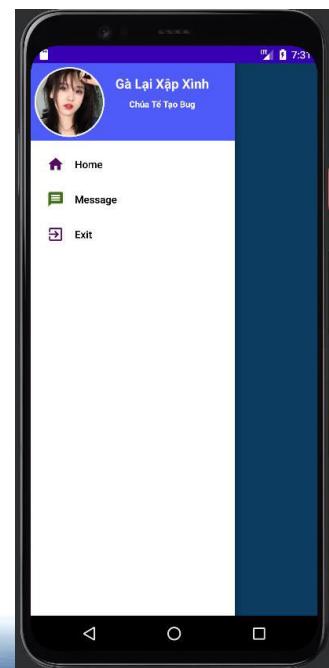
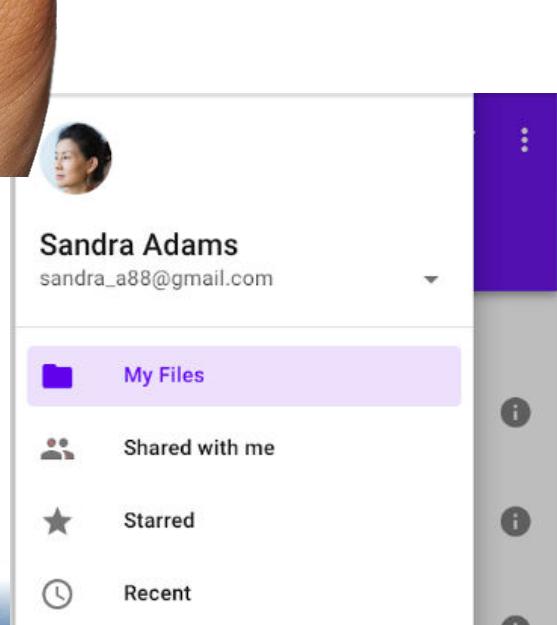
## □ 2.3 Xử lý sự kiện trên item context Menu:

```
//xử lý sự kiện trên item Context Menu
override fun onContextItemSelected(item: MenuItem): Boolean {
    when(item.itemId) {
        31->{txtContext.setTextColor(Color.RED)}
        32->{txtContext.setTextColor(Color.GREEN)}
        33->{txtContext.setTextColor(Color.BLACK)}
    }
    return super.onContextItemSelected(item)
}
```





## Navigation Kotlin



1

## Navigation View kotlin

### □ 1. Các bước xây dựng :

Step 1:  
Make xml layout  
Header

Step 2:  
Make menu xml

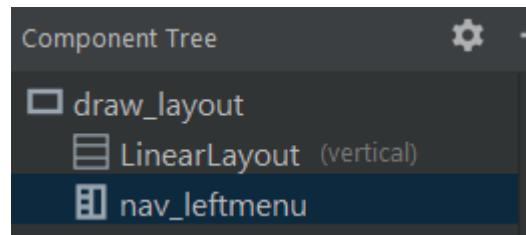
Step 3:  
Khai báo header +  
menu vào  
NavigationView



## 1

## NavigationView kotlin

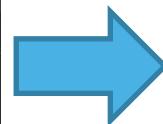
## □ Thiết kế giao diện :



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/draw_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#2196F3"
        android:orientation="vertical" />

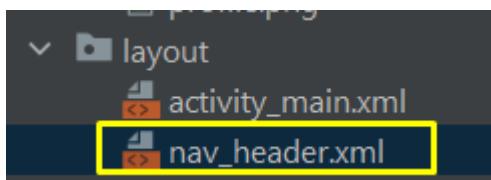
    <com.google.android.material.navigation.NavigationView
        android:id="@+id/nav_leftmenu"
        android:layout_width="wrap_content"
        android:layout_height="match_parent" />
</androidx.drawerlayout.widget.DrawerLayout>
```



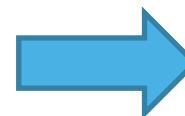
1

## Navigation View kotlin

### Step 1: Tạo header xml :



Ảnh đế trong drawable  
và code toàn bộ bài 21  
(API 24 android 7.0 )  
Pass unrar : **tuhoc211**



<http://android.tuhoc.cc/>

21-navimenu.zip

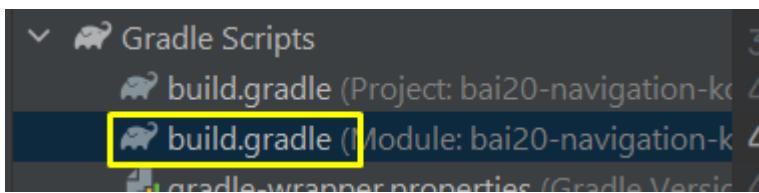
bai21navigationkotlin  
hinhanh-21



1

## Navigation View kotlin

### Step 1: Tạo header xml :



```
dependencies {
    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.appcompat:appcompat:1.5.1'
    implementation 'com.google.android.material:material'
    implementation 'androidx.constraintlayout:constraintlayout'
    implementation 'de.hdodenhof:circleimageview:3.1.0'
    implementation 'androidx.mediarouter:mediarouter:1.3.0'
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:background="#4E5BF8"
    android:padding="5dp"
    android:layout_height="wrap_content">

    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/profile_image"
        android:layout_width="96dp"
        android:layout_height="96dp"
        android:src="@drawable/profile"
        app:civ_border_color="#F6F5F5"
        app:civ_border_width="2dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="15dp"
        android:layout_marginTop="10dp"
        android:text="Gà Lại Xấp Xinh"
        android:textColor="#FBF9F9"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintStart_toEndOf="@+id/profile_image"
        app:layout_constraintTop_toTopOf="parent" />

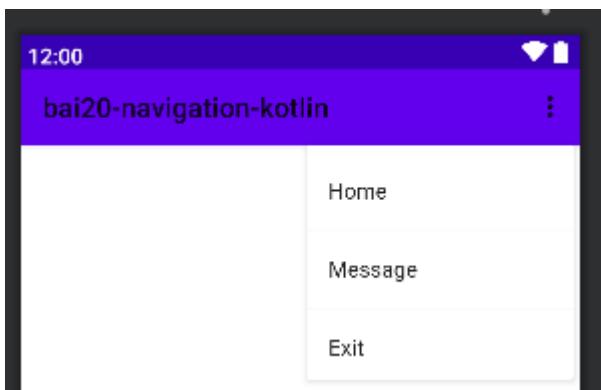
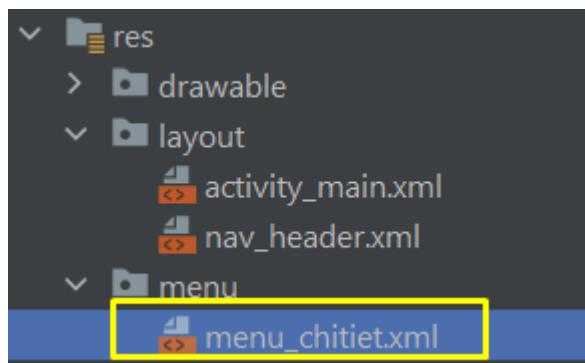
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="Chúa Tể Tạo Bug"
        android:textColor="#FFFFFF"
        android:textSize="12sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="@+id/textView"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

## 1

## Navigation View kotlin

## □ Step 2: Tạo menu xml :



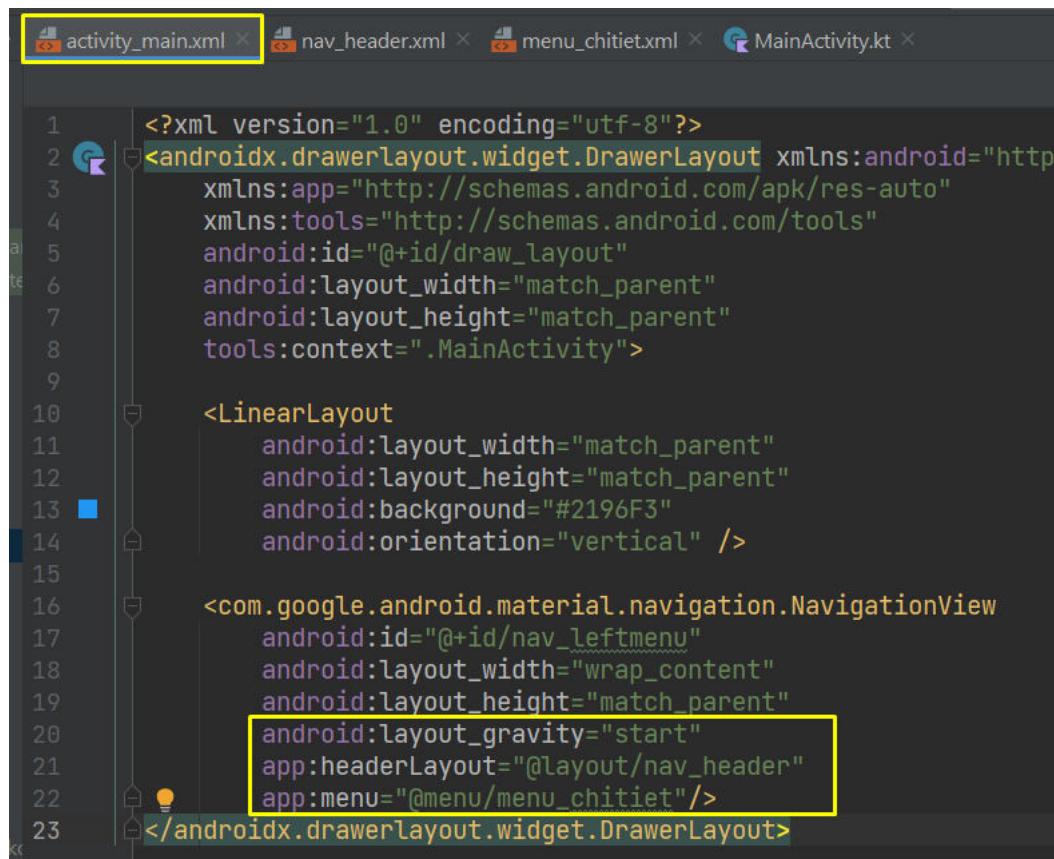
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/
    xmlns:app="http://schemas.android.com/apk/res-
        <item
            android:id="@+id/nav_home"
            android:icon="@drawable/ic_home"
            android:title="Home" />
        <item
            android:id="@+id/nav_message"
            android:icon="@drawable/ic_message"
            android:title="Message" />
        <item
            android:id="@+id/nav_exit"
            android:icon="@drawable/ic_exit"
            android:title="Exit" />
    </menu>
```



## 1

## NavigationView kotlin

## □ Step 2: Add header và menu vào NavigationView :

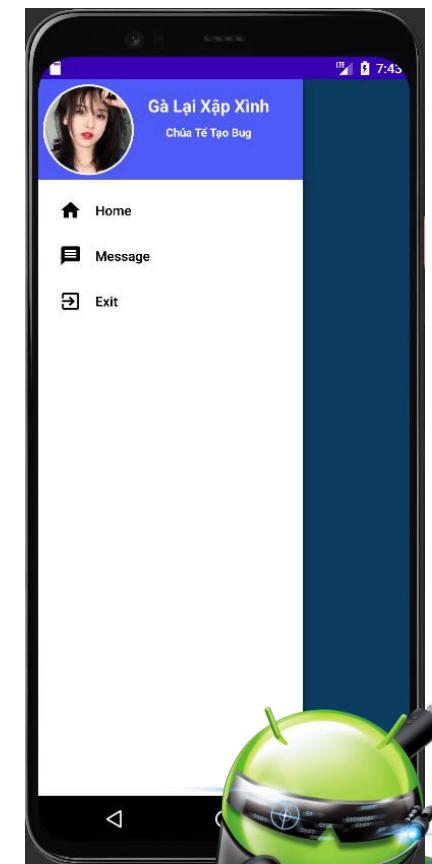


```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/draw_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#2196F3"
        android:orientation="vertical" />

    <com.google.android.material.navigation.NavigationView
        android:id="@+id/nav_leftmenu"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        app:headerLayout="@layout/nav_header"
        app:menu="@menu/menu_chitiet"/>

</androidx.drawerlayout.widget.DrawerLayout>
```



## 1

## NavigationView kotlin

## □ Xử lý sự kiện trên item menu :



```
activity_main.xml × nav_header.xml × menu_chitiet.xml × MainActivity.kt ×
package com.tuhoc.bai20_navigation_kotlin
import ...
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        //sử dụng màu cho icon
        nav_leftmenu.itemIconTintList = null
        //lắng nghe sự kiện click lên item menu
        nav_leftmenu.setNavigationItemSelectedListener { it: MenuItem
            when(it.itemId){
                R.id.nav_home->Toast.makeText( context: this, text: "Home", Toast.LENGTH_SHORT).show()
                R.id.nav_message->Toast.makeText( context: this, text: "Message", Toast.LENGTH_SHORT).show()
                R.id.nav_exit->finish()
            }
            true ^setNavigationItemSelectedListener
        }
    }
}
```



## SQLite How to use SQLite Studio

	USERID	UNAME	PWD
1	1	teo@gmail.com	123456
2	2	ti@gmail.com	654321
3	3	ba@gmail.com	111111
4	4	bon@gmail.com	666666
5	5	oi@gmail.com	333333
6	6	bona@gmail.com	12355555



## 1

## Khái quát SQLite

- ❑ là 1 trong những hệ cơ sở dữ liệu quan trọng trong các hệ điều hành mobile:  
Dùng để lưu trữ dữ liệu => **Giải quyết bài toán, khi tắt ứng dụng, dữ liệu không bị mất mà lưu trong máy khách hàng**
- ❑ khi học SQLlite có thể áp dụng từ android, sang các hệ điều hành khác như Ios, ...
- ❑ **Bản chất SQLite là 1 cơ sở dữ liệu mà ở đó chúng ta có thể :**

- ✓ **Creat**
- ✓ **Reward**
- ✓ **Update**
- ✓ **Delete**

USERID	UNAME	PWD
1	teo@gmail.com	123456
2	ti@gmail.com	654321
3	ba@gmail.com	111111
4	bon@gmail.com	666666
5	oi@gmail.com	333333
6	bona@gmail.com	12355555



## 2

## SQLite Studio

## □ 2. Download công cụ SQLite Studio :

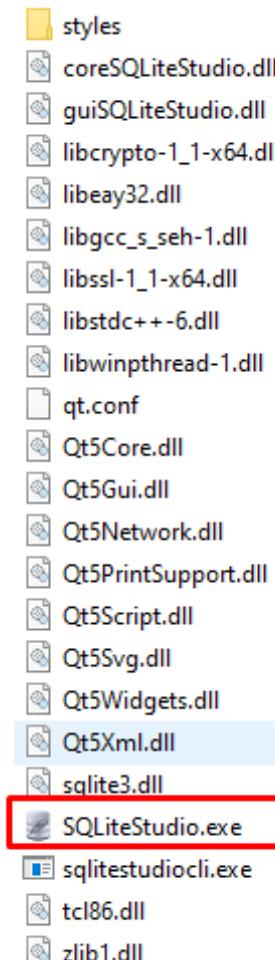
Cách 1: Download link bản cùng  
version với khoá học này tại tuhoc.cc  
( sử dụng cho hệ điều hành Window)

<http://android.tuhoc.cc/>

▼ 22.sqlitestudio-3.3.3-tuhoc.cc.zip 

Cách 2: Download từ trang chủ

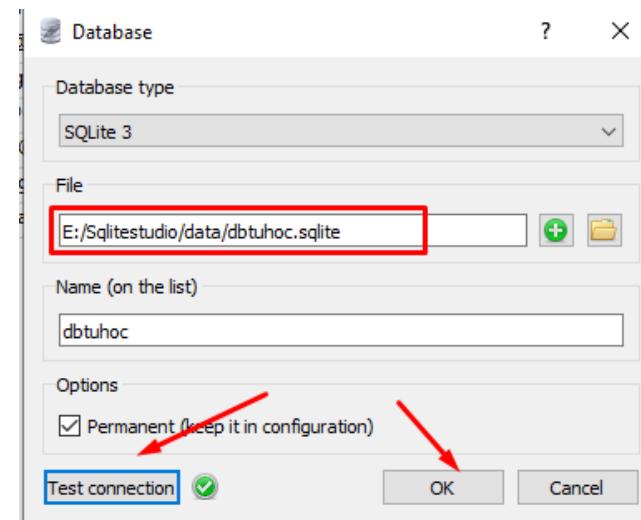
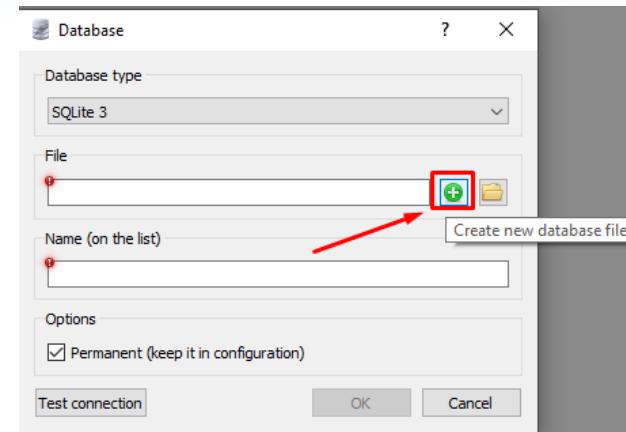
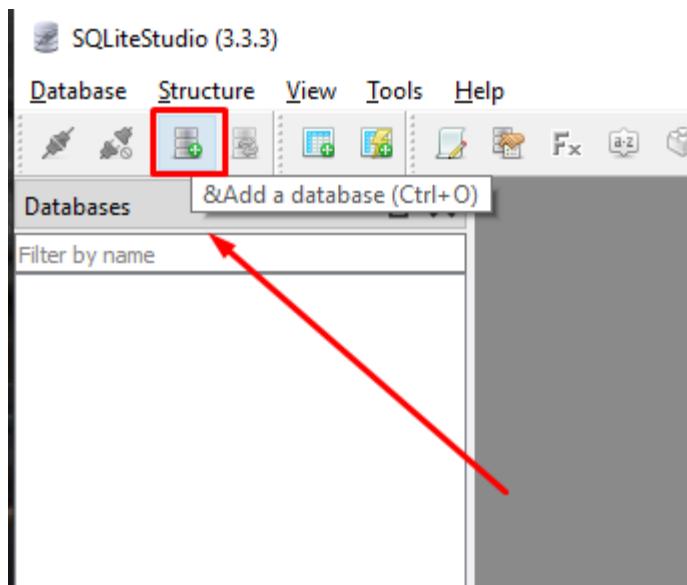
<https://sqlitestudio.pl/>



## 2

## SQLite Studio

## □ 3. Tạo mới database :



Database sẽ được tạo ngay sau khi  
test connection

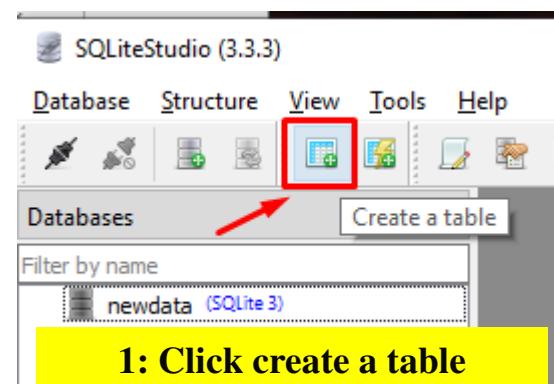


## 2

## SQLite Studio

## □ 4. Tạo table :

□ *Chú ý 1 table luôn có khoá chính : là số không trùng nhau, ví dụ căn cước công dân, mã sinh viên.... Để lập trình viên có thể truy xuất đến các nội dung 1 cách chính xác*



Structure Data Constraints Indexes Triggers DDL

dbtuhoc Table name: USERS

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate
------	-----------	-------------	-------------	--------	-------	----------	---------

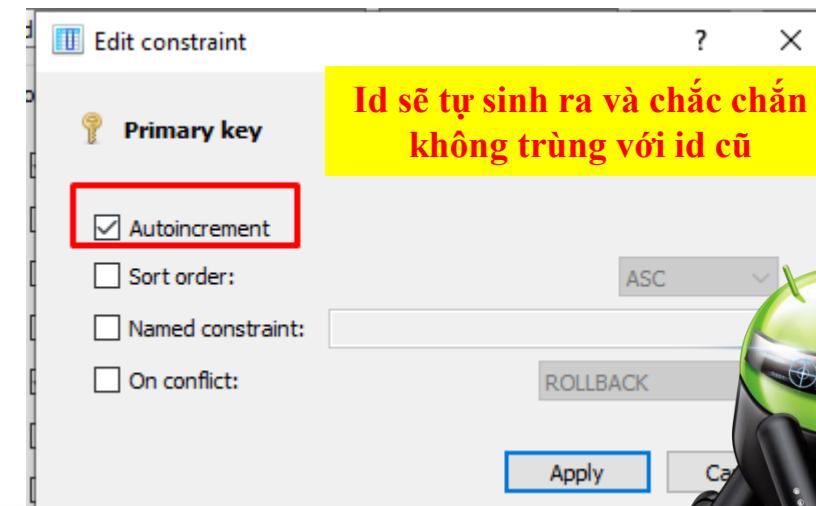
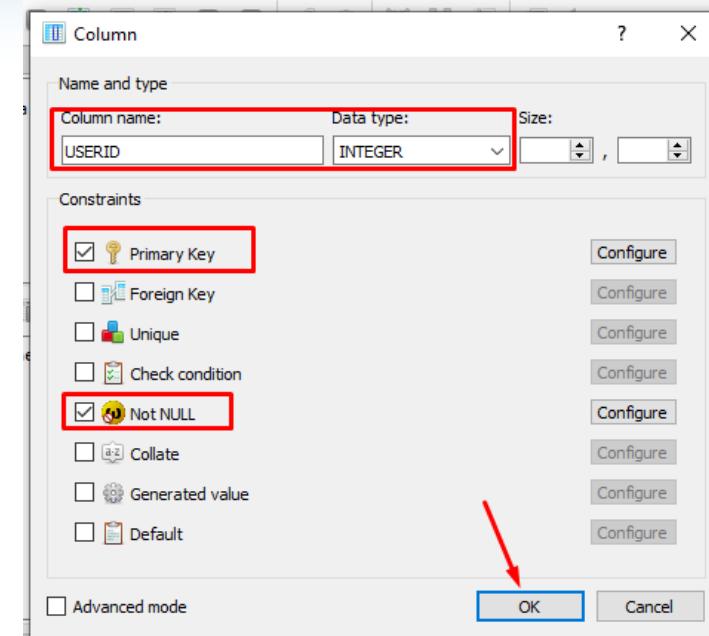
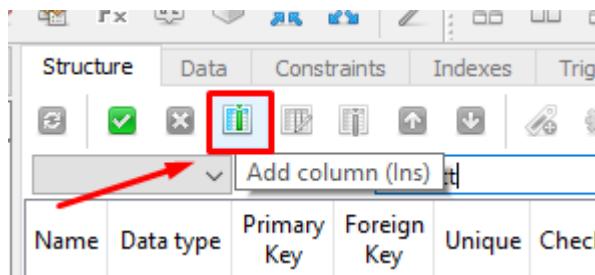
2: Nhập tên bảng



## 2

## SQLite Studio

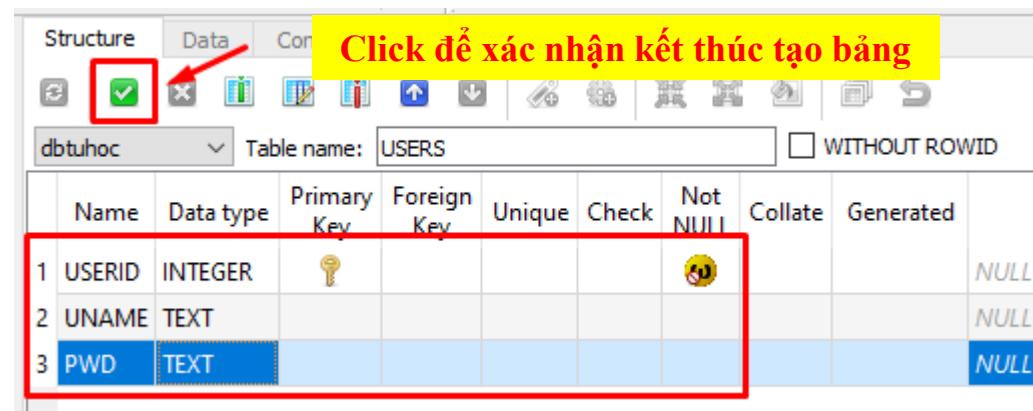
## □ 5. Tạo cột :



2

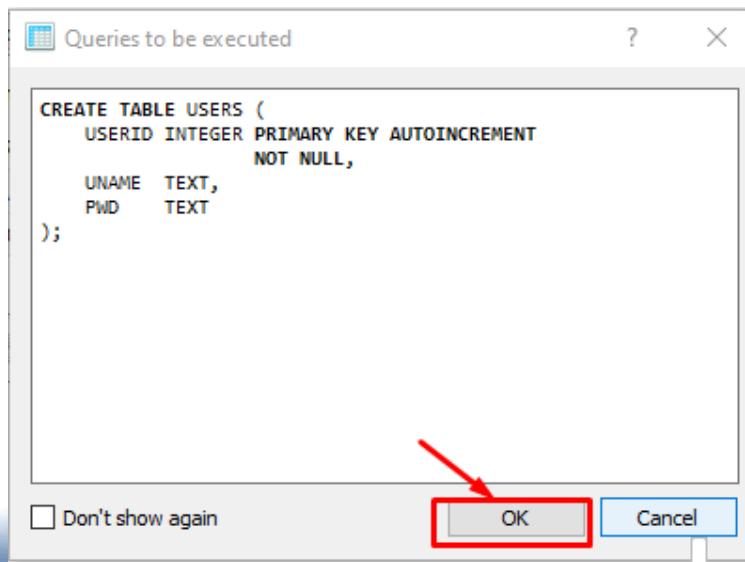
## SQLite Studio

## □ 5. Tạo các cột tiếp theo :



The screenshot shows the SQLite Studio interface. A yellow callout box with the text "Click để xác nhận kết thúc tạo bảng" (Click to confirm the end of table creation) points to the green checkmark icon in the toolbar. The table structure is defined as follows:

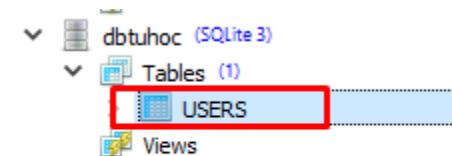
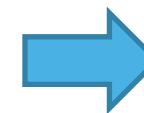
Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated
1 USERID	INTEGER	KEY				NULL		
2 UNAME	TEXT					NULL		
3 PWD	TEXT					NULL		



The dialog box contains the following SQL code:

```
CREATE TABLE USERS (
    USERID INTEGER PRIMARY KEY AUTOINCREMENT
        NOT NULL,
    UNAME TEXT,
    PWD TEXT
);
```

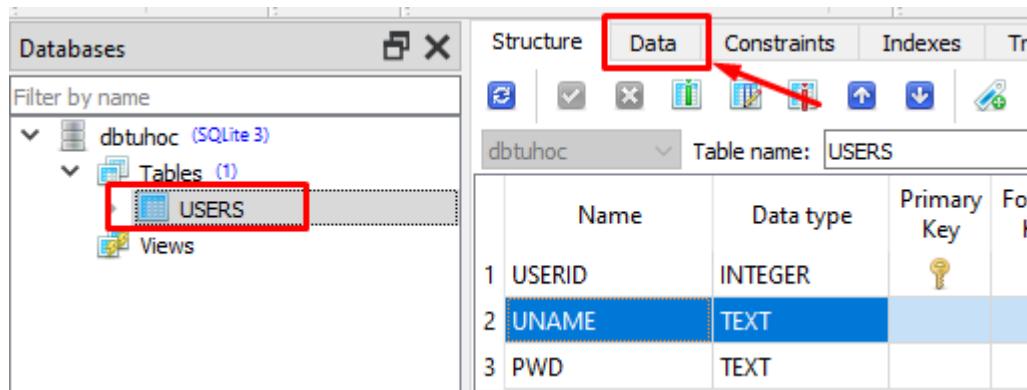
At the bottom, there is a checkbox "Don't show again" and two buttons: "OK" (highlighted with a red box) and "Cancel".



## 2

## SQLite Studio

## □ 6. Nhập dữ liệu ( Data ) :



Databases

Filter by name

dbtuhoc (SQLite 3)

Tables (1)

USERS

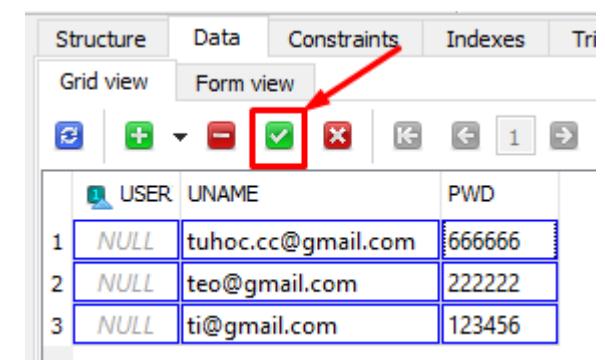
Structure Data Constraints Indexes Triggers

Table name: USERS

Name	Data type	Primary Key	Foreign Key
1 USERID	INTEGER	key	
2 UNAME	TEXT		
3 PWD	TEXT		



Click để xác nhận xong thêm data

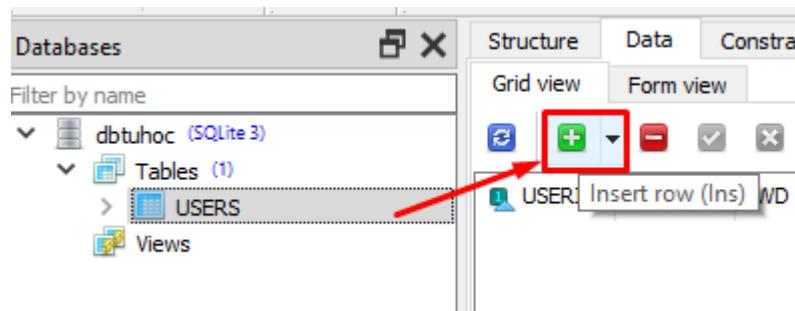


Structure Data Constraints Indexes Triggers

Grid view Form view

USER UNAME PWD

1	NULL	tuhoc.cc@gmail.com	666666
2	NULL	teo@gmail.com	222222
3	NULL	ti@gmail.com	123456



Databases

Filter by name

dbtuhoc (SQLite 3)

Tables (1)

USERS

Structure Data Constraints

Grid view Form view

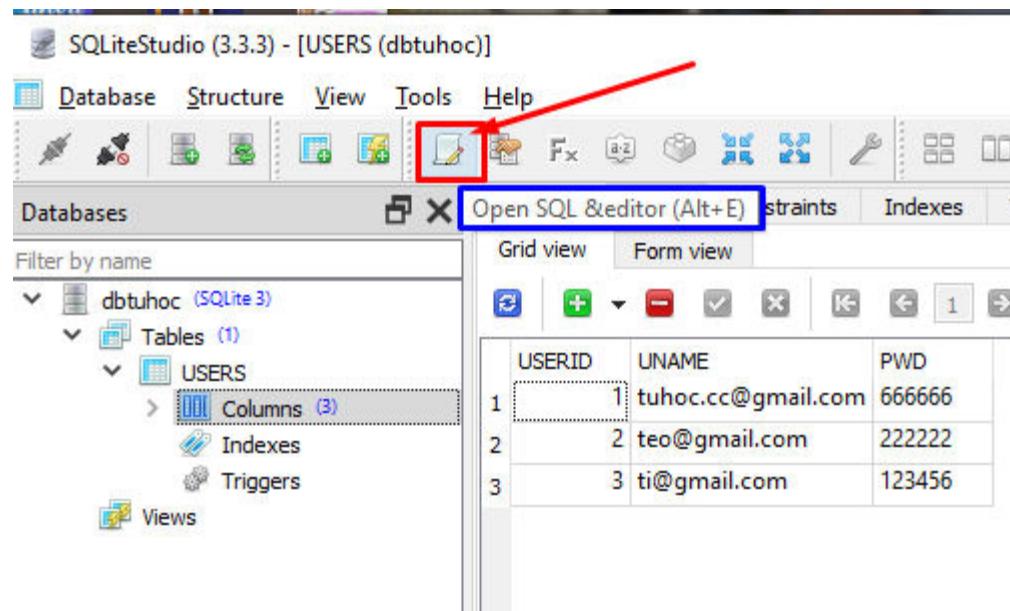
Insert row (Ins) WD



## 2

## SQLite Studio

## □ 7. Các lệnh truy vấn trực tiếp cơ bản :



The screenshot shows the SQLite Studio interface. The title bar reads "SQLiteStudio (3.3.3) - [USERS (dbtuhoc)]". The menu bar includes Database, Structure, View, Tools, and Help. A red arrow points to the "Open SQL & editor (Alt+E)" button in the toolbar, which is highlighted with a blue box. The left pane displays the database structure: "dbtuhoc (SQLite 3)" contains "Tables (1)", specifically "USERS" with "Columns (3)". The right pane shows a grid view of the "USERS" table with the following data:

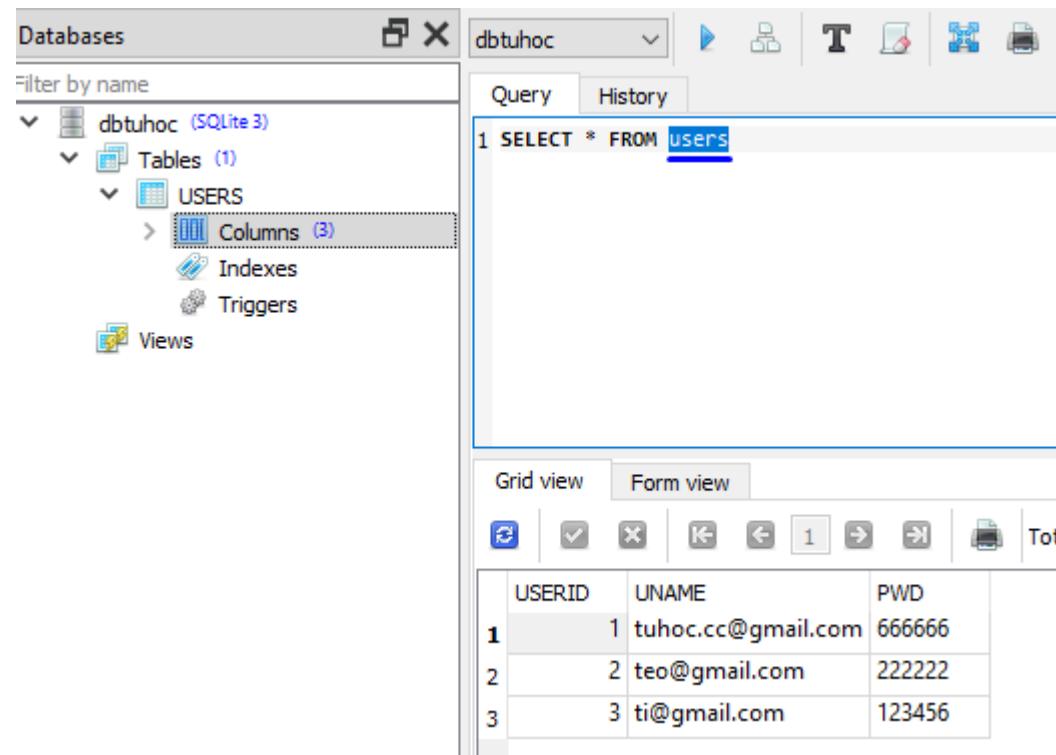
	USERID	UNAME	PWD
1	1	tuhoc.cc@gmail.com	666666
2	2	teo@gmail.com	222222
3	3	ti@gmail.com	123456



## 2

## SQLite Studio

- 7.1 `SELECT * FROM <table_name>`: xuất toàn bộ thông tin trong table\_name



The screenshot shows the SQLite Studio interface. On the left, the 'Databases' sidebar shows a database named 'dbtuhoc' containing one table, 'USERS'. The 'Tables' section of the sidebar is expanded, showing 'Columns (3)' as the selected item. In the main area, the 'Query' tab is active, displaying the SQL command 'SELECT \* FROM users'. Below the query window is a results grid. The grid has columns labeled 'USERID', 'UNAME', and 'PWD'. It contains three rows of data:

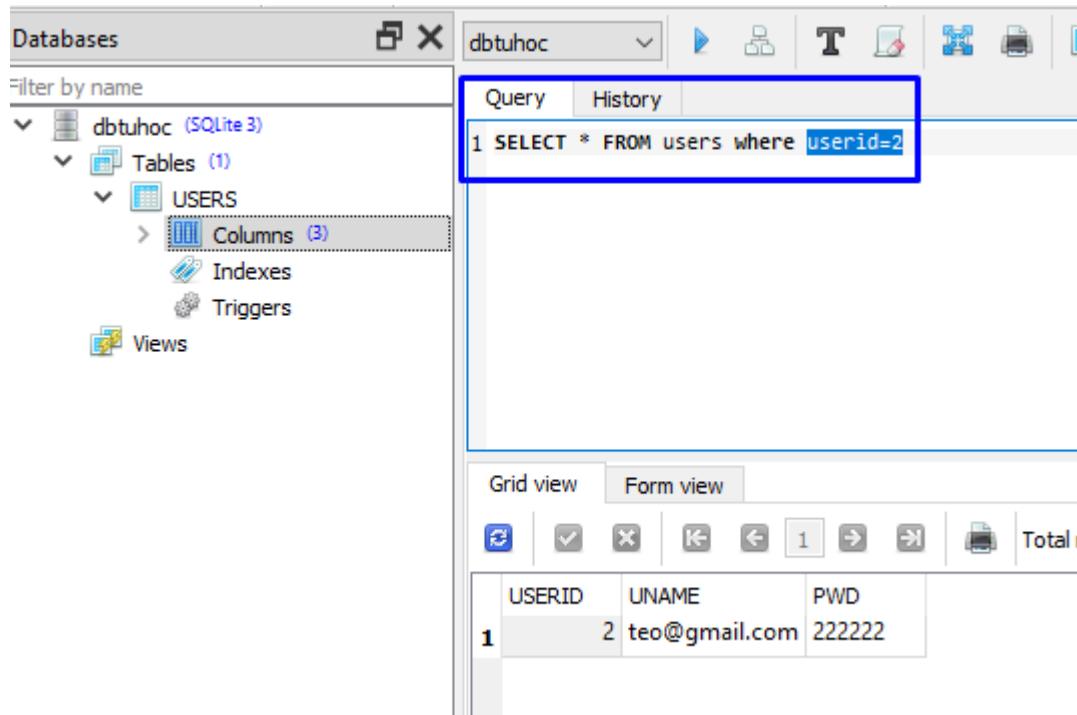
	USERID	UNAME	PWD
1	1	tuhoc.cc@gmail.com	666666
2	2	teo@gmail.com	222222
3	3	ti@gmail.com	123456

Lưu ý : SQLite không phân biệt  
chữ hoa chữ thường



## 2

## SQLite Studio

□ 7.2 **SELECT \* FROM <table\_name> WHERE <column = value> :**

The screenshot shows the SQLite Studio interface. On the left is a tree view of the database structure under 'dbtuhoc (SQLite 3)'. It includes 'Tables (1)' which contains 'USERS' with 'Columns (3)'. The main area has a 'Query' tab selected, displaying the SQL query:

```
1 SELECT * FROM users where userid=2
```

Below the query window is a 'Grid view' table with three columns: USERID, UNAME, and PWD. A single row is shown with values 1, teo@gmail.com, and 222222.

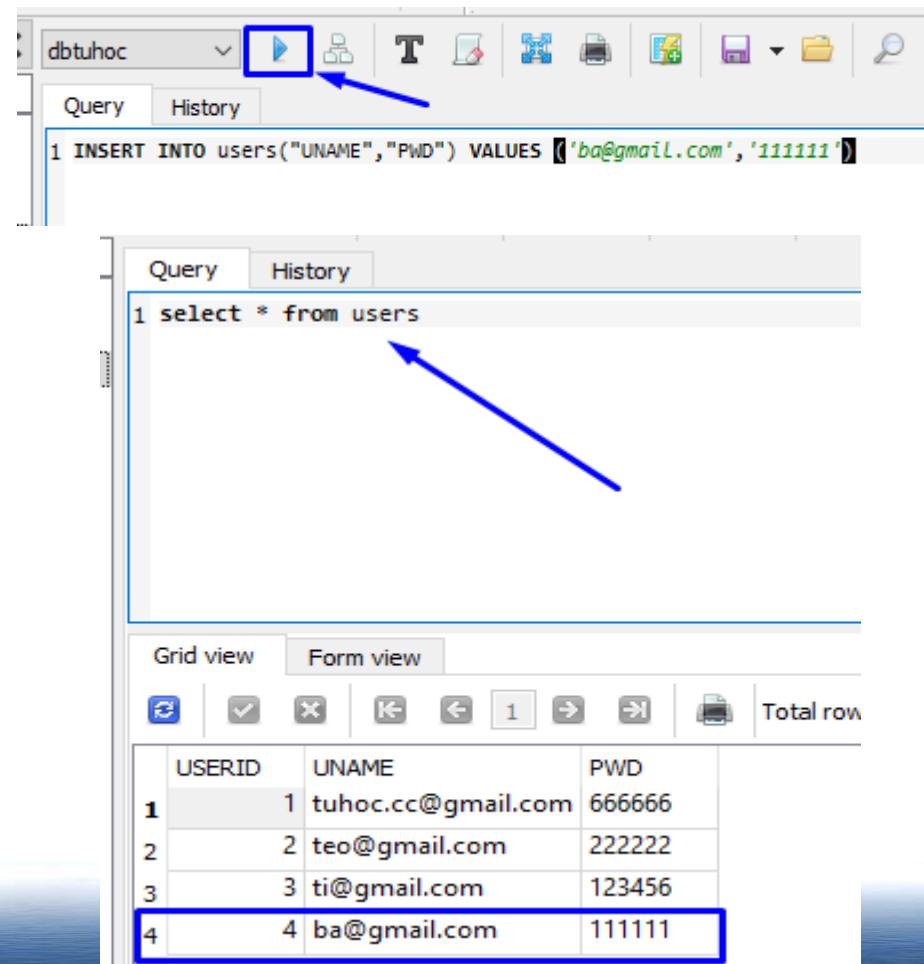
USERID	UNAME	PWD
1	teo@gmail.com	222222



## 2

## SQLite Studio

## □ 7.3 Thêm mới thông tin bằng lệnh :

**INSERT INTO users("UNAME","PWD") VALUES ('ba@gmail.com','111111')**

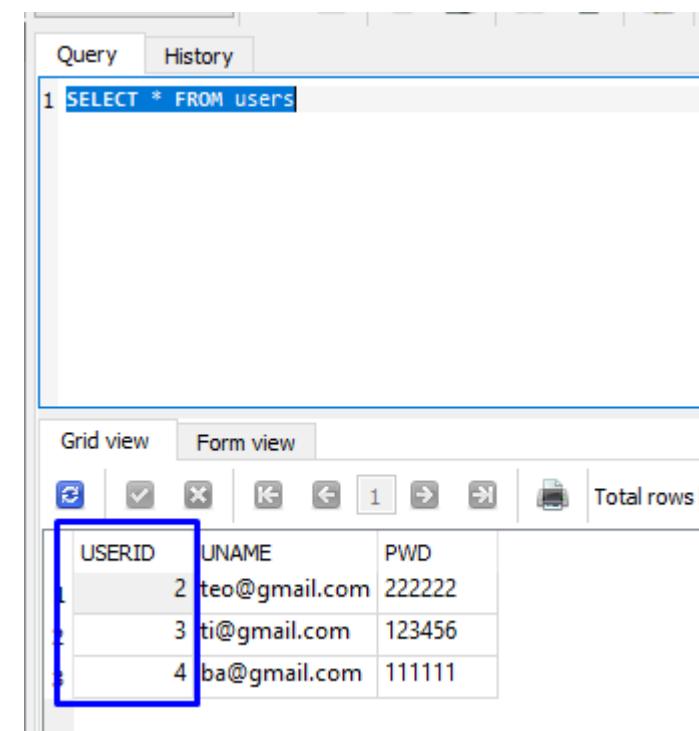
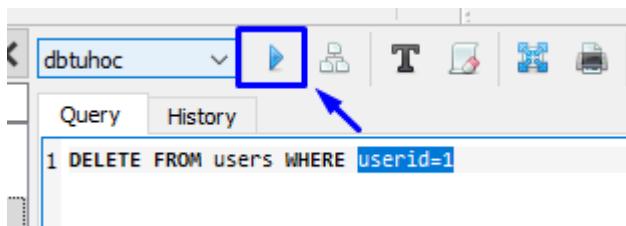
The screenshot shows the SQLite Studio interface. At the top, there's a toolbar with various icons. A blue arrow points to the first icon on the left, which is a play button (execute query). Below the toolbar, there are two tabs: 'Query' and 'History'. The 'Query' tab contains the SQL command: `1 INSERT INTO users("UNAME","PWD") VALUES ('ba@gmail.com','111111')`. The 'History' tab contains the SQL command: `1 select * from users`. Below these tabs, there are two views: 'Grid view' and 'Form view'. The 'Grid view' is selected. It displays a table with four columns: USERID, UNAME, PWD, and Total row. The table has four rows of data. Row 4 is highlighted with a blue border. The data is as follows:

USERID	UNAME	PWD	Total row
1	tuhoc.cc@gmail.com	666666	
2	teo@gmail.com	222222	
3	ti@gmail.com	123456	
4	ba@gmail.com	111111	



## 2

## SQLite Studio

 7.4 Xoá 1 item bằng code :**DELETE FROM users WHERE userid=1**

The screenshot shows the SQLite Studio interface after executing the DELETE query. At the top, the 'Query' tab is active, displaying the executed command '1 SELECT \* FROM users'. Below this, the 'Grid view' tab is selected, showing a table with three rows of data. The columns are labeled 'USERID', 'UNAME', and 'PWD'. The data is as follows:

USERID	UNAME	PWD
2	teo@gmail.com	222222
3	ti@gmail.com	123456
4	ba@gmail.com	111111

Id 1 đã bị xoá

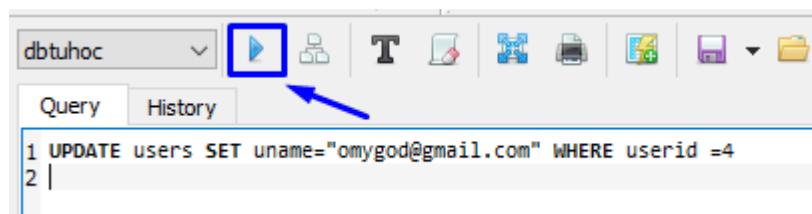


## 2

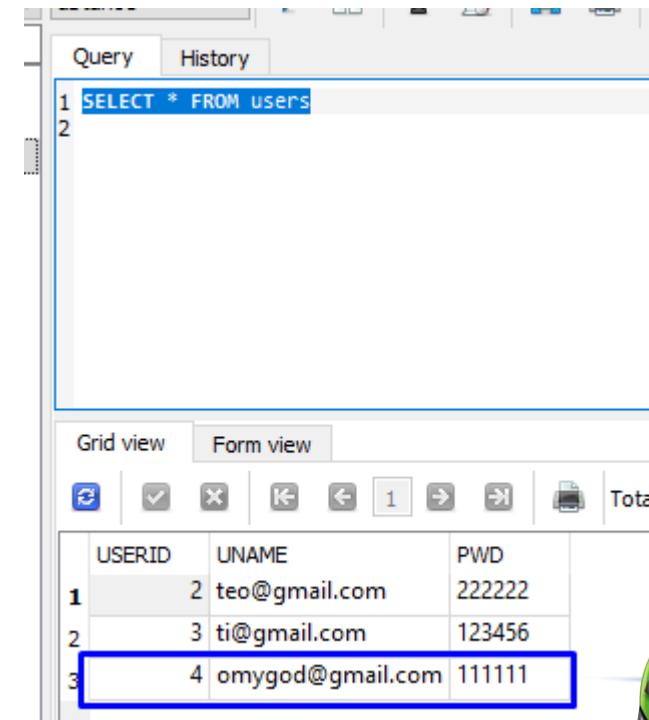
## SQLite Studio

## □ 7.5 update data cho 1 item :

**UPDATE users SET uname='omygod@gmail.com' WHERE userid =4**



```
1 UPDATE users SET uname="omygod@gmail.com" WHERE userid =4
2 |
```



```
1 SELECT * FROM users
2 |
```

	USERID	UNAME	PWD
1	2	teo@gmail.com	222222
2	3	ti@gmail.com	123456
3	4	omygod@gmail.com	111111

**Id 4 đã được update**



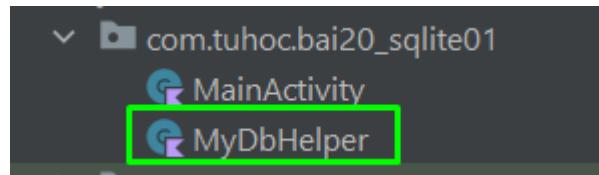


## SQLite SQLiteOpenHelper



## 1

## Tạo class Helper

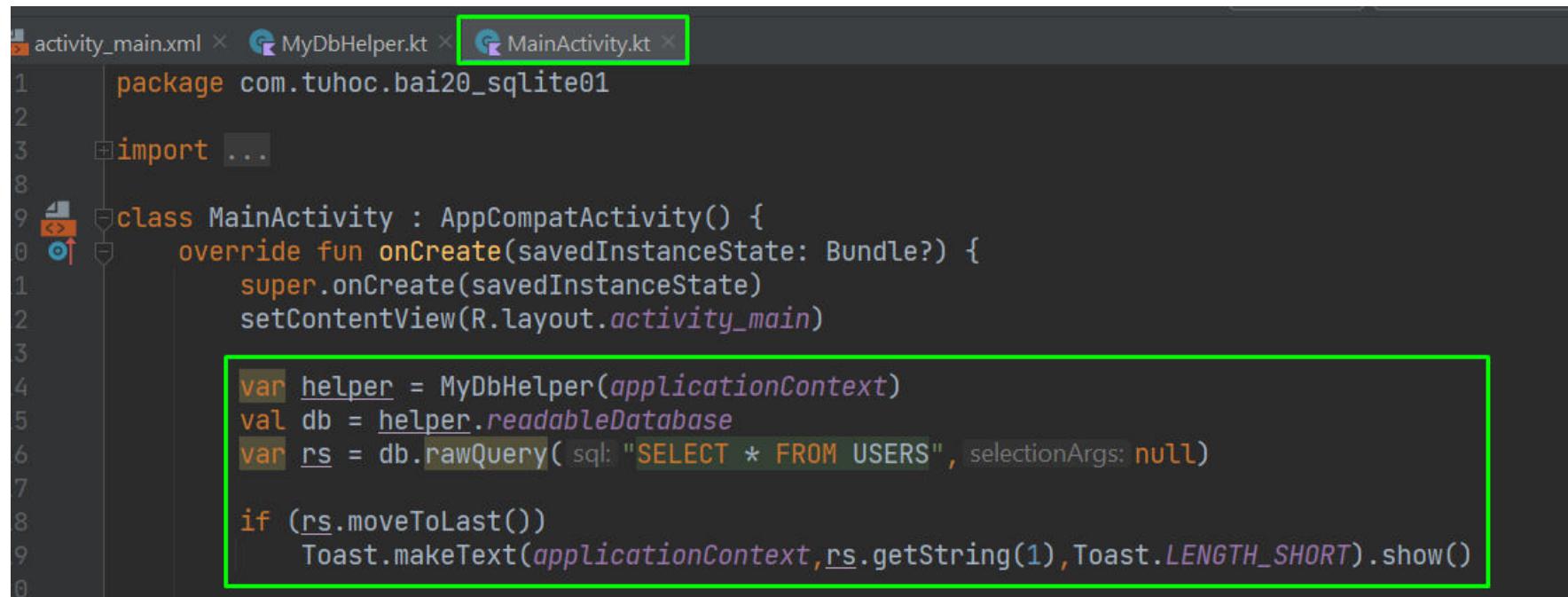


```
/*ctrl +P để xem hướng dẫn các constructors */
/*
name: Chuỗi ? : tên của db
factory SQLiteDatabase.CursorFactory ? : để sử dụng để tạo các đối tượng con trỏ, hoặc null cho mặc định
version Int : số của cơ sở dữ liệu (bắt đầu từ 1);
nếu cơ sở dữ liệu cũ hơn, onUpgradesẽ được sử dụng để nâng cấp cơ sở dữ liệu;
nếu cơ sở dữ liệu mới hơn, onDowngradesẽ được sử dụng để hạ cấp cơ sở dữ liệu*/
class MyDbHelper(context: Context) : SQLiteOpenHelper(context, name: "USERDB", factory: null, version: 1) {
    //ctrl+i để override
    override fun onCreate(p0: SQLiteDatabase?) {
        //tạo bảng
        p0?.execSQL( sql: "CREATE TABLE USERS(USERID INTEGER PRIMARY KEY AUTOINCREMENT,UNAME TEXT , PWD TEXT)")
        //add data
        p0?.execSQL( sql: "INSERT INTO USERS(UNAME,PWD) VALUES('teo@gmail.com','123456')")
        p0?.execSQL( sql: "INSERT INTO USERS(UNAME,PWD) VALUES('ti@gmail.com','654321')")
    }
    override fun onUpgrade(p0: SQLiteDatabase?, p1: Int, p2: Int) {
    }
}
```



## 2

## Code MainActivity



The screenshot shows the Android Studio interface with three tabs at the top: activity\_main.xml, MyDbHelper.kt, and MainActivity.kt. The MainActivity.kt tab is selected and highlighted with a green border. The code editor displays the following Kotlin code:

```
activity_main.xml × MyDbHelper.kt × MainActivity.kt ×

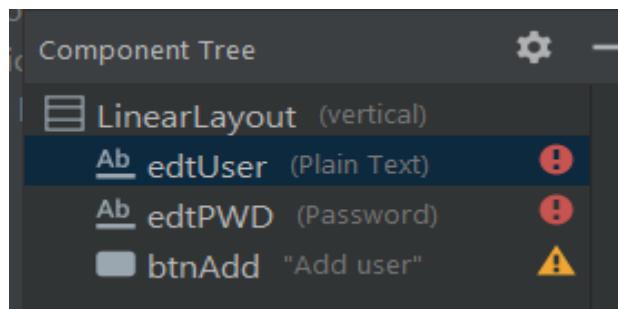
1 package com.tuhoc.bai20_sqlite01
2
3 import ...
4
5 class MainActivity : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         setContentView(R.layout.activity_main)
9
10        var helper = MyDbHelper(applicationContext)
11        val db = helper.readableDatabase
12        var rs = db.rawQuery("SELECT * FROM USERS", null)
13
14        if (rs.moveToFirst())
15            Toast.makeText(applicationContext, rs.getString(1), Toast.LENGTH_SHORT).show()
16    }
17}
```

The code uses the MyDbHelper class to query the 'USERS' table and display the first result using a Toast message.

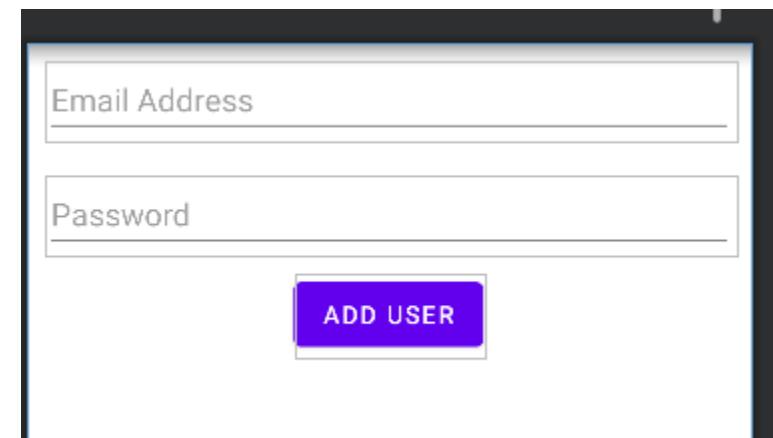
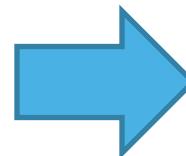


## 3

## Xây dựng activity\_main.xml



```
<EditText  
    android:id="@+id(edtUser"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_margin="10dp"  
    android:ems="10"  
    android:hint="Email Address"  
    android:inputType="textPersonName" />  
  
<EditText  
    android:id="@+id(edtPWD"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_margin="10dp"  
    android:ems="10"  
    android:hint="Password"  
    android:inputType="textPassword" />  
  
<Button  
    android:id="@+id(btnAdd"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:text="Add user" />
```



## 4

## Add data vào db

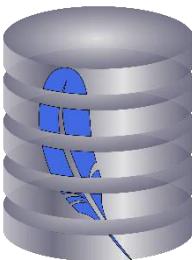
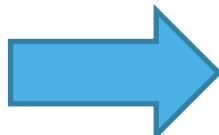
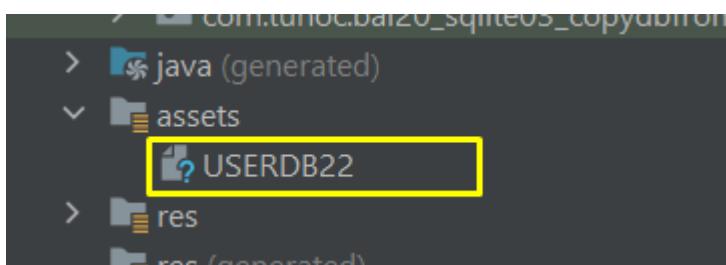
```
activity_main.xml x DBHelper.kt x MainActivity.kt x
1 package com.tuhoc.bai20_sqlite01
2
3 import ...
4
5 class MainActivity : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         setContentView(R.layout.activity_main)
9
10        var helper = DBHelper(applicationContext)
11        val db = helper.readableDatabase
12        var rs = db.rawQuery("SELECT * FROM USERS", selectionArgs: null)
13
14        if (rs.moveToFirst())
15            Toast.makeText(applicationContext, rs.getString(1), Toast.LENGTH_SHORT).show()
16
17
18        //lắng nghe sự kiện lên btnAdd
19        btnAdd.setOnClickListener { it: View!
20            var cv = ContentValues()
21            cv.put("UNAME", edtUser.text.toString())
22            cv.put("PWD", edtPWD.text.toString())
23            db.insert(table: "USERS", nullColumnHack: null, cv)
24
25            //reset 2 ô nhập liệu username và password
26            edtPWD.setText("")
27            edtUser.setText("")
28            edtUser.requestFocus() //cho con trỏ ở ô user
29
30
31
32    }
33
34 }
```





# SQLite

## Copy SQLite database from assets to androidstudio kotlin



1

## Tạo database

SQLiteStudio (3.3.3) - [TUHOC (TUHOCDB)]

Database Structure View Tools Help

Databases

Filter by name

TUHOCDB (SQLite 3)

Tables (2)

- android\_metadata
- TUHOC

Views

Table name: TUHOC  WITHOUT ROWID

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 _id	integer								NULL
2 user	TEXT								NULL
3 email	TEXT								NULL

SQLiteStudio.exe



1

## Add data vào bảng

SQLiteStudio (3.3.3) - [TUHOC (TUGOCDB)]

Database Structure View Tools Help

Databases Filter by name

TUGOCDB (SQLite 3)

Tables (1)

> TUHOC

Views

Structure Data Constraints Indexes Triggers DDL

Grid view Form view

+ - ✓ ✕

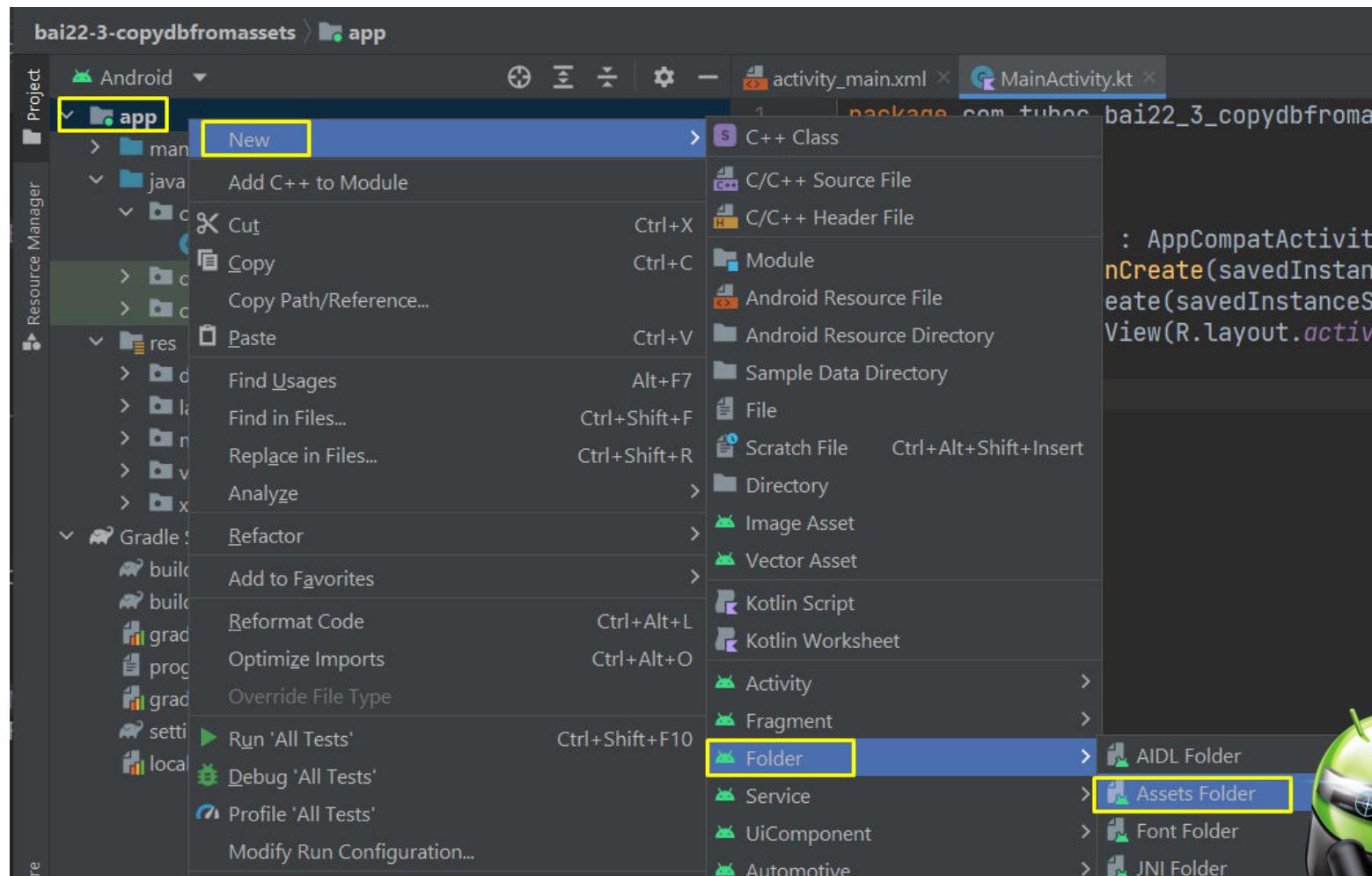
_id	user	email
1	NULL	ba@gmail.com
2	NULL	hai@gmail.com
3	NULL	mot@gmail.com

Total rows loaded: 0



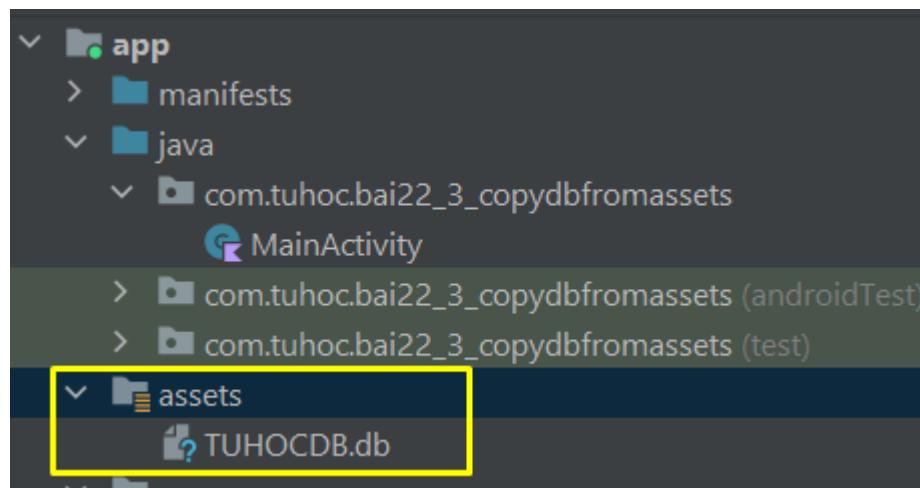
## 2

## Copy db vào assets



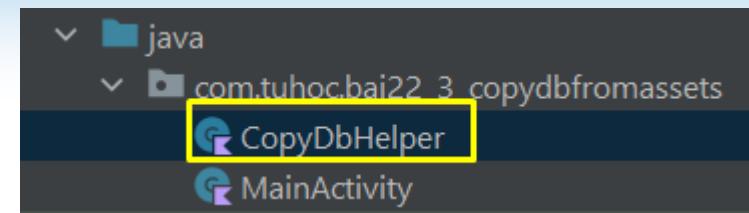
2

## Copy db vào assets



3

## Tạo class CopyDbHelper



```
class CopyDbHelper(private val context: Context) {
    companion object {
        private val DB_NAME = "TUHOCDB.db"
    }
    fun openDatabase(): SQLiteDatabase {
        val dbFile = context.getDatabasePath(DB_NAME)
        Log.wtf("tag: "wtf", dbFile.toString())
        val file = File(dbFile.toString())
        if (file.exists()) {
            //không làm gì cả
            Log.e("tag: "tuhoc", msg: "file đã tồn tại")
        } else{
            copyDatabase(dbFile)
        }
        return SQLiteDatabase.openDatabase(dbFile.path, factory: null, SQLiteDatabase.OPEN_READWRITE)
    }

    private fun copyDatabase(dbFile : File){
        val openDB = context.assets.open(DB_NAME)
        val outPutStream = FileOutputStream(dbFile)
        val buffer = ByteArray( size: 1024)
        while (openDB.read(buffer)>0){
            outPutStream.write(buffer)
            Log.wtf("tag: "DB", msg: "writing")
        }
        outPutStream.flush()
        outPutStream.close()
        openDB.close()
        Log.wtf("tag: "DB", msg: "copy DB completed")
    }
}
```



## 4

## Code MainActivity

```
activity_main.xml × MainActivity.kt × CopyDbHelper.kt ×
1 package com.tuhoc.bai22_3_copydbfromassets
2
3 import ...
4
5
6 class MainActivity : AppCompatActivity() {
7
8     private var db:CopyDbHelper?=null
9
10    override fun onCreate(savedInstanceState: Bundle?) {
11        super.onCreate(savedInstanceState)
12        setContentView(R.layout.activity_main)
13
14        //1.sao chép database
15        db= CopyDbHelper( context: this)
16        db?.openDatabase()
17    }
}
```



## 5

## Kiểm tra db và test thử

```
com.tuhoc.bai22_3_copydbfrom databases
├── com.tuhoc.bai22_3_copydbfrom
│   ├── cache
│   └── databases
│       ├── TUHOCDB.db
│       └── TUHOCDB.db-journal
```

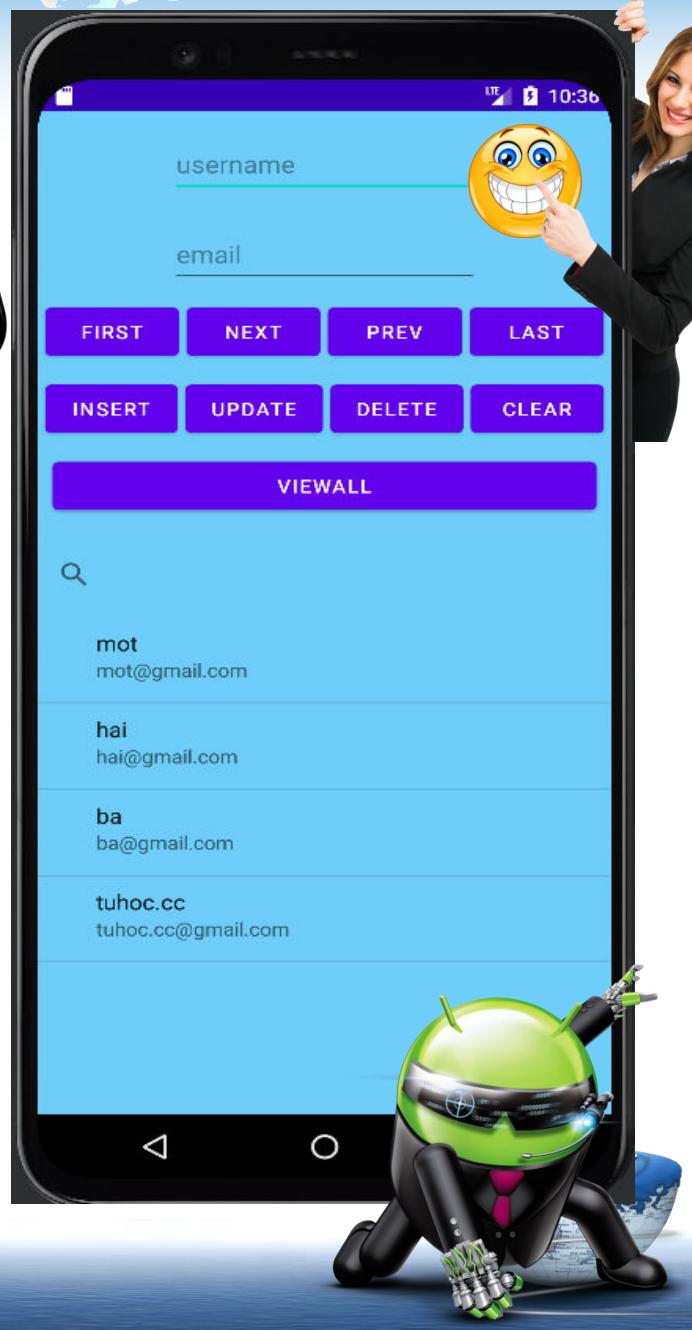
```
dbfroma Error Q
.tuhoc.bai22_3_copydbfromassets E/wtf: /data/user/0/com.tuhoc.bai22_3_copydbfromassets/databases/TUHOCDB.db
.tuhoc.bai22_3_copydbfromassets E/tuhoc: file đã tồn tại
```

File chỉ được copy lần đầu tiên khởi chạy, từ lần thứ 2 file đã tồn tại sẽ không bị copy đè



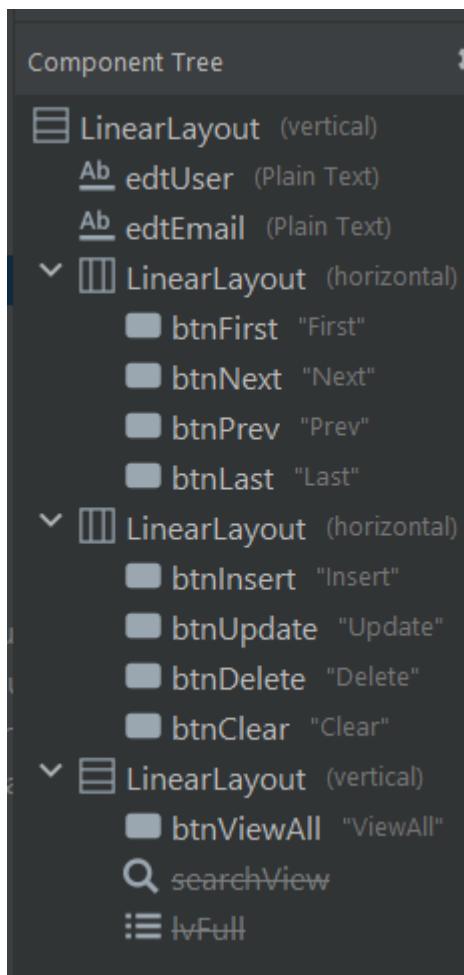


**SQLite CRUD**  
**insert, select, update,**  
**delete and display data**  
**in listview**



## 1

## Thiết kế layout ứng dụng



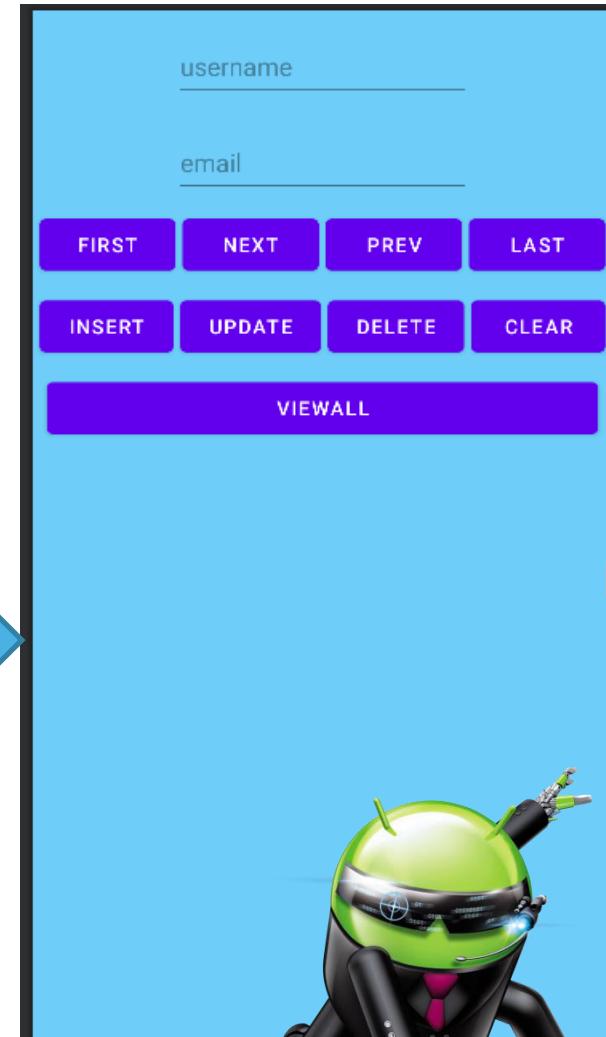
## Full code bài 22-4

<http://android.tuhoc.cc/>

22-4.bai22-4-crud-sqlite-tuhoc.cc.zip

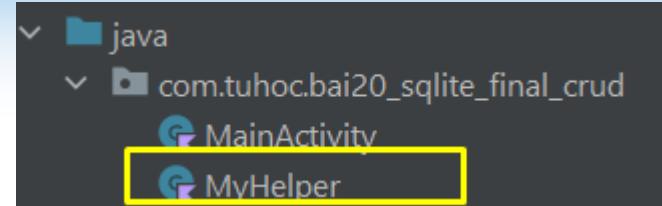
```
<SearchView  
    android:id="@+id/searchView"  
    android:layout_width="match_parent"  
    android:layout_height="63dp"  
    android:visibility="gone" />  
  
<ListView  
    android:id="@+id/lvFull"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:visibility="gone" />
```

Searchview và lvFull sẽ ẩn



2

## Tạo Class MyHelper



```
class MyHelper(context: Context) : SQLiteOpenHelper(context, name: "TUHOCDB", factory: null, version: 1) {
    //ctrl + i
    override fun onCreate(db: SQLiteDatabase?) {
        //tạo table , column
        db?.execSQL("CREATE TABLE TUHOC(_id integer primary key autoincrement,user TEXT, email TEXT)")
        //thêm data vào cơ sở dữ liệu
        db?.execSQL("Insert into TUHOC(user,email) values ('mot','mot@gmail.com')")
        db?.execSQL("Insert into TUHOC(user,email) values ('hai','hai@gmail.com')")
        db?.execSQL("Insert into TUHOC(user,email) values ('ba','ba@gmail.com')")
        db?.execSQL("Insert into TUHOC(user,email) values ('tuhoc.cc','tuhoc.cc@gmail.com')")
    }

    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
        TODO(reason: "Not yet implemented")
    }
}
```



## 3

## Code MainActivity

```
class MainActivity : AppCompatActivity() {  
  
    lateinit var db: SQLiteDatabase  
    lateinit var rs : Cursor  
    lateinit var adapter: SimpleCursorAdapter  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        var helper = MyHelper(applicationContext)  
        db = helper.readableDatabase  
        rs = db.rawQuery(sql: "SELECT * FROM TUHOC LIMIT 20", selectionArgs: null)  
        //sql : String chuỗi câu lệnh SQL  
        //String[] selectionArgs: Chon lọc dữ liệu với điều kiện mảng  
  
        //1. đưa thử dữ liệu của dòng đầu tiên trong db lên edtuser và edtEmail  
        if (rs.moveToFirst()){  
            edtUser.setText(rs.getString(columnIndex: 1))  
            edtEmail.setText(rs.getString(columnIndex: 2))  
        }  
    }  
}
```



## 3

## Code MainActivity

```
//2.code cho button first
btnFirst.setOnClickListener { it: View!
    if (rs.moveToFirst()){
        edtUser.setText(rs.getString(columnIndex: 1))
        edtEmail.setText(rs.getString(columnIndex: 2))
    }
    else
        Toast.makeText(applicationContext, text: "No data found", Toast.LENGTH_SHORT).show()
}

//3.code cho button next
btnNext.setOnClickListener { it: View!
    if (rs.moveToNext()){
        edtUser.setText(rs.getString(columnIndex: 1))
        edtEmail.setText(rs.getString(columnIndex: 2))
    }
    else if (rs.moveToFirst()){
        edtUser.setText(rs.getString(columnIndex: 1))
        edtEmail.setText(rs.getString(columnIndex: 2))
    }
    else
        Toast.makeText(applicationContext, text: "No data found", Toast.LENGTH_SHORT).show()
}
```



## 3

## Code MainActivity

```
//4.code cho prev
btnPrev.setOnClickListener { it: View!
    if (rs.moveToPrevious()){
        edtUser.setText(rs.getString( columnIndex: 1))
        edtEmail.setText(rs.getString( columnIndex: 2))
    }
    else if (rs.moveToLast()){
        edtUser.setText(rs.getString( columnIndex: 1))
        edtEmail.setText(rs.getString( columnIndex: 2))
    }
    else
        Toast.makeText(applicationContext, text: "No data found", Toast.LENGTH_SHORT).show()
}

//5.code cho button last
btnLast.setOnClickListener { it: View!
    if (rs.moveToLast()){
        edtUser.setText(rs.getString( columnIndex: 1))
        edtEmail.setText(rs.getString( columnIndex: 2))
    }
    else
        Toast.makeText(applicationContext, text: "No data found", Toast.LENGTH_SHORT).show()
}
```



## 3

## Code MainActivity

```
//6. code phần lvFull
adapter = SimpleCursorAdapter(
    applicationContext, android.R.layout.simple_expandable_list_item_2, rs,
    arrayOf("user", "email"), intArrayOf(android.R.id.text1, android.R.id.text2), flags: 0
)
lvFull.adapter = adapter
//7. code cho nút viewall
btnViewAll.setOnClickListener { it: View!
    searchView.visibility = View.VISIBLE
    lvFull.visibility = View.VISIBLE
    adapter.notifyDataSetChanged()
    searchView.queryHint ="Tìm kiếm trong ${rs.count} bản ghi"
}

//code cho phần tìm kiếm nội dung
searchView.setOnQueryTextListener(object : SearchView.OnQueryTextListener{
    //ctrl + i
    override fun onQueryTextSubmit(query: String?): Boolean {
        return false
    }

    override fun onQueryTextChange(newText: String?): Boolean {
        rs = db.rawQuery( sql: "SELECT * FROM TUHOC WHERE user LIKE '%$${newText}' or email LIKE '%$${newText}'",
            selectionArgs: null)
        adapter.changeCursor(rs)
        return true
    }
})
```



## 3

## Code MainActivity

```
//8. code cho nút insert
btnInsert.setOnClickListener { it: View! ->
    var cv = ContentValues()
    cv.put("user", edtUser.text.toString())
    cv.put("email", edtEmail.text.toString())
    db.insert(table: "TUHOC", nullColumnHack: null, cv)
    rs.requery()
    adapter.notifyDataSetChanged()

    var ad = AlertDialog.Builder(context: this)
    ad.setTitle("Add record")
    ad.setMessage("Add thành công")
    ad.setPositiveButton(text: "OK", DialogInterface.OnClickListener { dialog, which ->
        edtUser.setText("")
        edtEmail.setText("")
        edtUser.requestFocus()
    })
    ad.show()
}
```



## 3

## Code MainActivity

```
//9. code cho nút update
btnUpdate.setOnClickListener { it: View! ->
    var cv = ContentValues()
    cv.put("user", edtUser.text.toString())
    cv.put("email", edtEmail.text.toString())
    db.update(table: "TUHOC", cv, whereClause: "_id=?", arrayOf(rs.getString(columnIndex: 0)))
    rs.requery()
    adapter.notifyDataSetChanged()

    var ad = AlertDialog.Builder(context: this)
    ad.setTitle("Update record")
    ad.setMessage("Update thành công")
    ad.setPositiveButton(text: "OK", DialogInterface.OnClickListener { dialog, which ->
        edtUser.setText("")
        edtEmail.setText("")
        edtUser.requestFocus()
    })
    ad.show()
}
```



## 3

## Code MainActivity

```
//10. code nút clear
btnClear.setOnClickListener { it: View!
    edtUser.setText("")
    edtEmail.setText("")
    edtUser.requestFocus()
}

//11. code delete
btnDelete.setOnClickListener { it: View!
    db.delete( table: "TUHOC", whereClause: "_id=?", arrayOf(rs.getString( columnIndex: 0)))
    rs.requery()
    adapter.notifyDataSetChanged()
    //thông báo
    var ad = AlertDialog.Builder( context: this)
    ad.setTitle("Update record")
    ad.setMessage("Update thành công")
    ad.setPositiveButton( text: "OK", DialogInterface.OnClickListener { dialog, which ->
        if (rs.moveToFirst()){
            edtUser.setText("")
            edtEmail.setText("")
            edtUser.requestFocus()
        }
        else
        {
            edtUser.setText("No data found")
            edtEmail.setText("No data found")
        }
    })
    ad.show()
}
```



## 3

## Code MainActivity

```
//đăng ký sử dụng context menu cho listview lvFull
registerForContextMenu(lvFull)

}

override fun onCreateContextMenu(
    menu: ContextMenu?,
    v: View?,
    menuInfo: ContextMenu.ContextMenuItemInfo?
) {
    super.onCreateContextMenu(menu, v, menuInfo)
    menu?.add( groupId: 100, itemId: 11, order: 1, title: "DELETE")
    menu?.setHeaderTitle("Removing data")
}

override fun onContextItemSelected(item: MenuItem): Boolean {
    if (item.itemId==11)
    {
        //Toast.makeText(applicationContext,"Click lên item trên menu",Toast.LENGTH_SHORT).show()
        db.delete( table: "TUHOC", whereClause: "_id=?", arrayOf(rs.getString( columnIndex: 0)))
        rs.requery()
        adapter.notifyDataSetChanged()
        searchView.queryHint ="Tìm kiếm trong ${rs.count} bản ghi"
    }

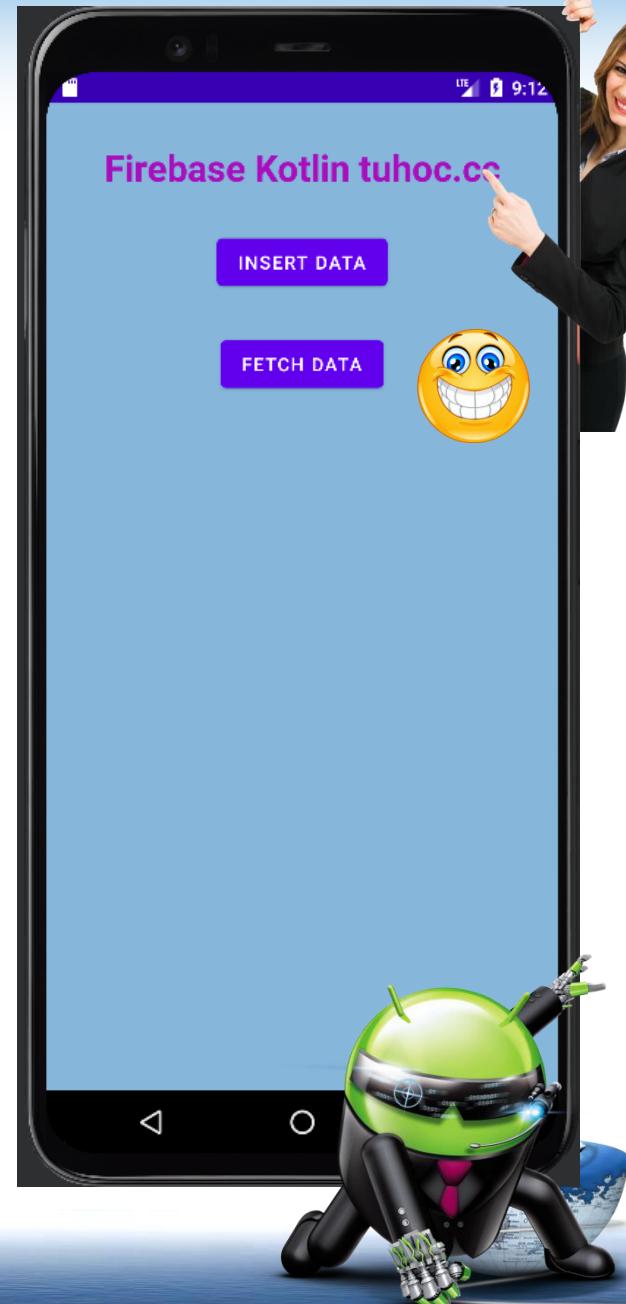
    return super.onContextItemSelected(item)
}
```



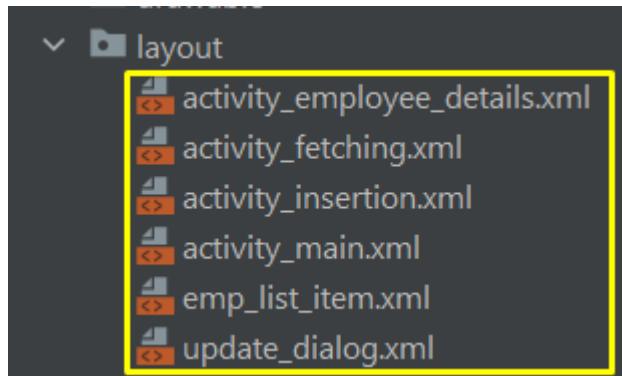


# **CRUD firebase kotlin**

## **Firebase Database with Kotlin**



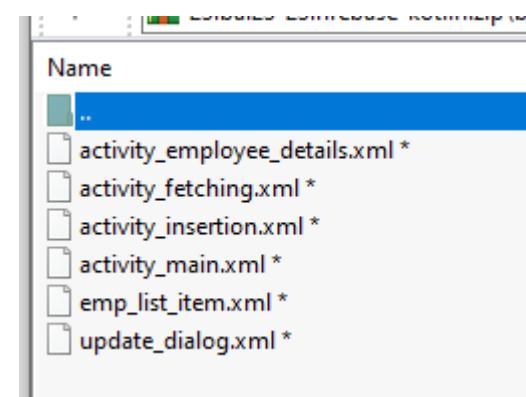
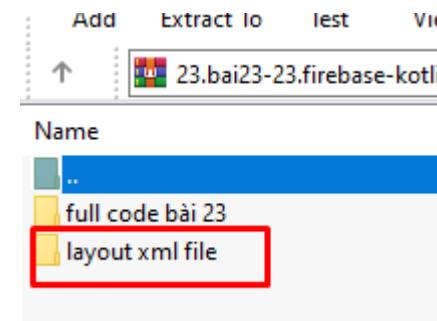
## □ 0 . File xml layout và code



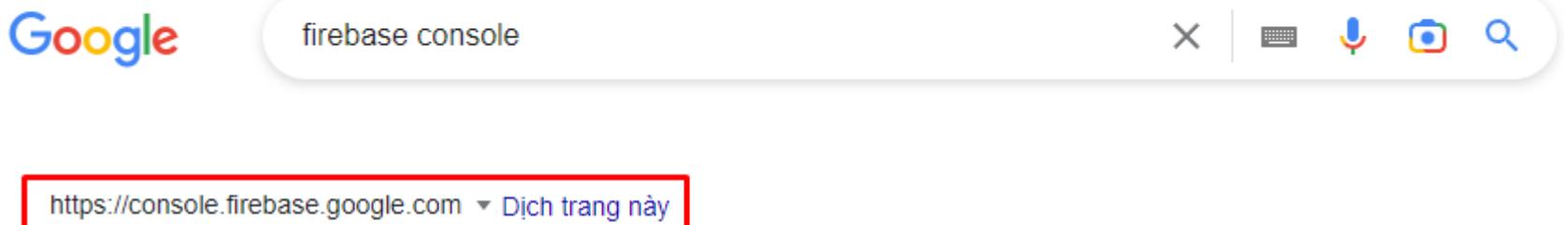
File xml layout  
và code toàn bộ bài 23  
**Mật khẩu giải nén: firebase**

<http://android.tuhoc.cc/>

▼ [23.bai23-23.firebaseio-kotlin.zip](#) 



## □ 1. Truy cập firebase :



### Firebase Console - Google

Forgot email? CAPTCHA image of text used to distinguish humans from robots.

Bạn đã truy cập trang này vào ngày 01/11/2022.

#### Sign in

Not your computer? Use Guest mode to sign in privately. Learn ...

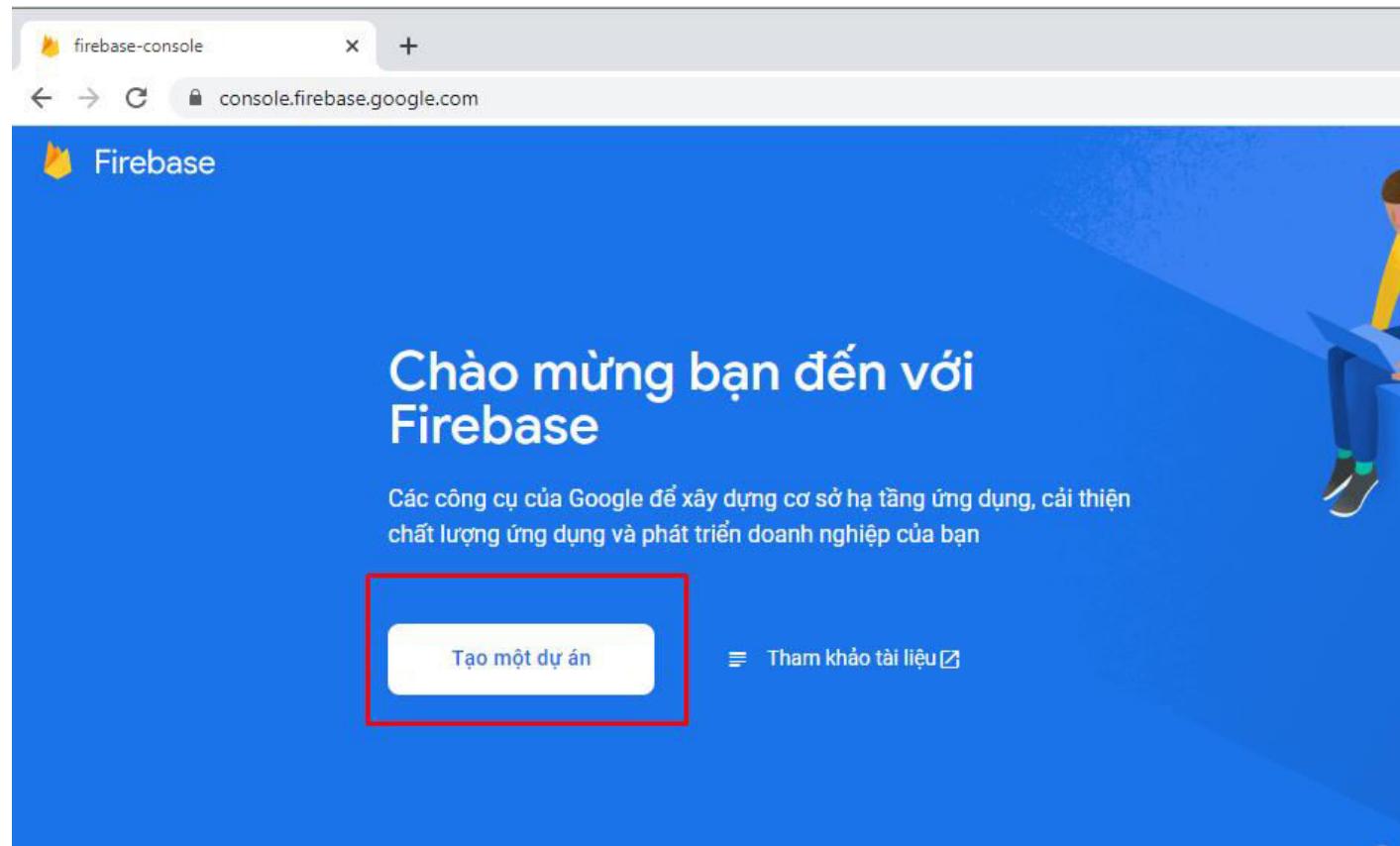
#### Use your Google Account

Forgot email? CAPTCHA image of text used to distinguish humans ...



## ❑ 2. Tạo dự án :

✓ Cần đăng nhập gmail trước khi tạo :



## □ 2. Tạo dự án :

Hãy bắt đầu bằng cách đặt  
tên cho dự án của bạn<sup>①</sup>

tên của dự án

tuhoc-Firebase

 my-awesome-project-id



Tôi đồng ý với [Điều khoản dịch vụ của Firebase](#)



Tôi xác nhận rằng tôi sẽ chỉ sử dụng Firebase cho các hoạt động thương mại, công ty, sáng tạo hoặc nghề nghiệp của mình.

Tiếp tục



## □ 2. Tạo dự án:

### Google Analytics cho dự án Firebase của bạn

Google Analytics là một giải pháp phân tích miễn phí, không giới hạn cho phép nhắm mục tiêu, báo cáo và hơn thế nữa trong Firebase Crashlytics, Cloud Messaging, Remote Config, A / B Testing, Cloud Functions và trong ứng dụng nhắn tin.

Google Analytics cho phép bạn sử dụng các tính năng sau:

-  Thử nghiệm A / B ?
  -  Phân khúc người dùng và nhắm mục tiêu trong các sản phẩm Firebase
  -  Người dùng không gặp sự cố ?
  -  Trình kích hoạt chức năng đám mây ?  
dựa trên sự kiện
  -  Báo cáo miễn phí và không giới hạn ?
- Bật Google Analytics cho dự án này  
sự giới thiệu

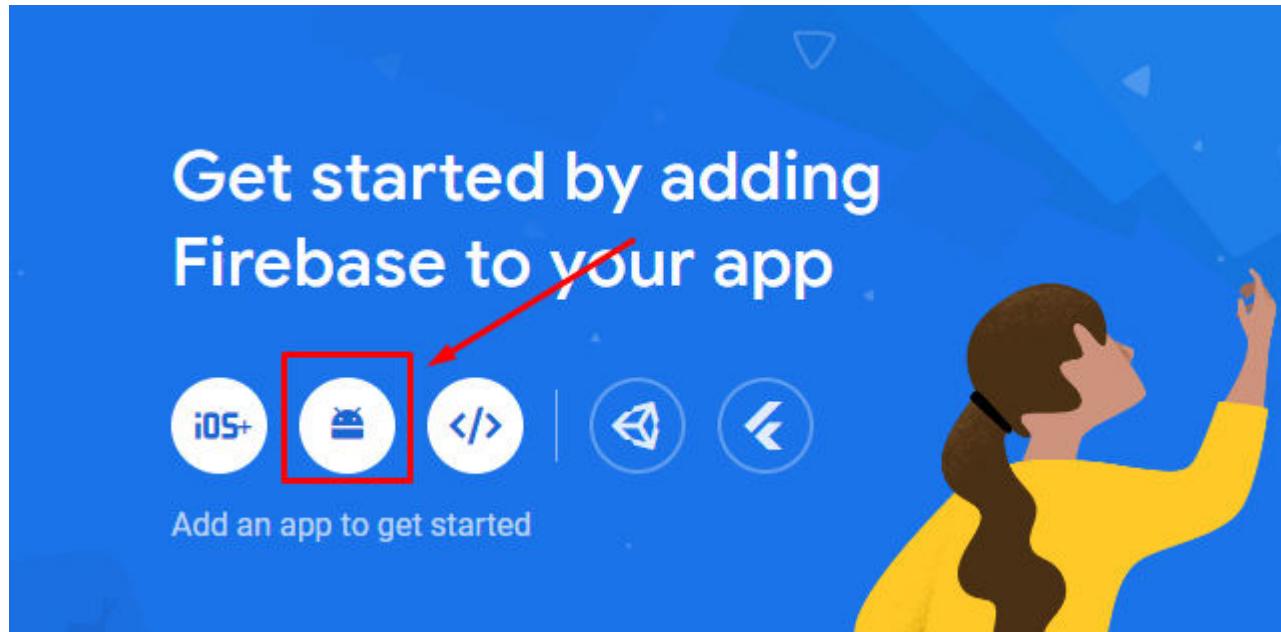
Trước

Tiếp tục



❑ 2. Tạo dự án :

✓ Chọn android :



## ❑ 2. Tạo dự án :

✓ *Register app :*

✗ Add Firebase to your Android app

1 Register app

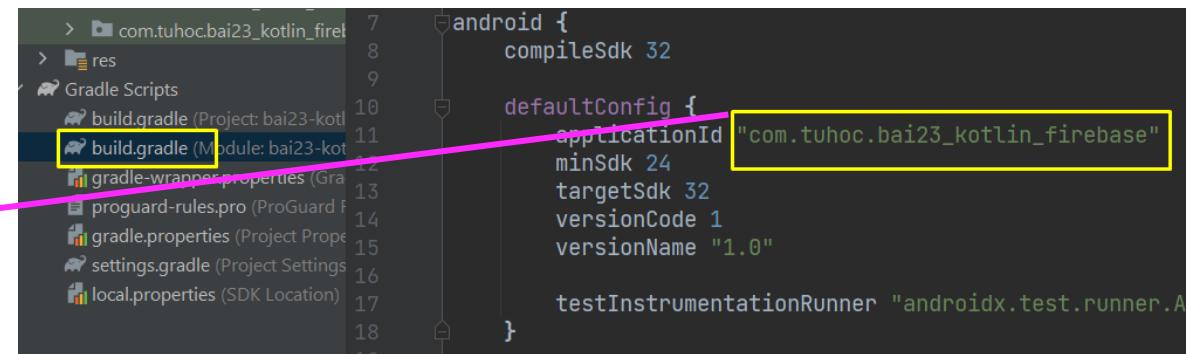
Android package name ②



App nickname (optional) ②

Debug signing certificate SHA-1 (optional) ②

ⓘ Required for Dynamic Links, and Google Sign-In or phone number support in Auth.  
Edit SHA-1s in Settings.



```
> com.tuhoc.bai23_kotlin_firebase
> res
Gradle Scripts
build.gradle (Project: bai23-kotlin)
build.gradle (Module: bai23-kotlin)
gradle-wrapper.properties (Gradle)
proguard-rules.pro (ProGuard Rules)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)

7 android {
8     compileSdk 32
9
10    defaultConfig {
11        applicationId "com.tuhoc.bai23_kotlin_firebase"
12        minSdk 24
13        targetSdk 32
14        versionCode 1
15        versionName "1.0"
16
17        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
18    }
19}
```

A yellow rectangular box highlights the line 'applicationId "com.tuhoc.bai23\_kotlin\_firebase"' in the build.gradle code. A pink arrow points from this box to the 'Android package name' input field on the left side of the screen.

1 Register app

Android package name ②

App nickname (optional) ②

Debug signing certificate SHA-1 (optional) ②

ⓘ Required for Dynamic Links, and Google Sign-In or phone number support in Auth.  
Edit SHA-1s in Settings.



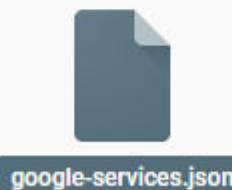
## □ 2. Tạo dự án :

- 2 Download and then add config file

 [Download google-services.json](#)

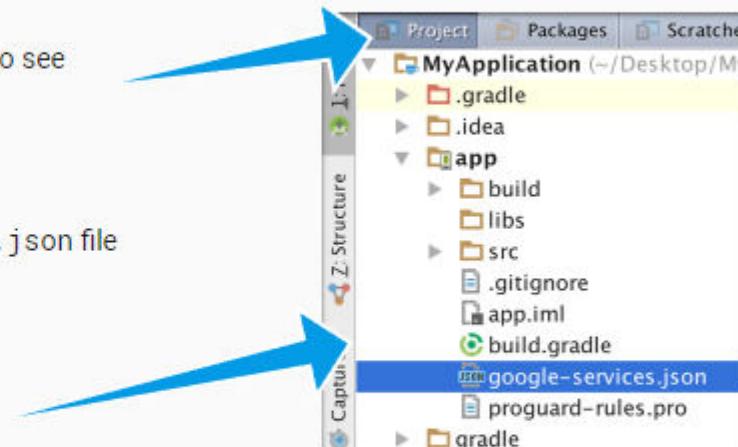
Switch to the Project view in Android Studio to see your project root directory.

Move your downloaded google-services.json file into your module (app-level) root directory.



Next

Instructions for Android Studio below | [Unity](#) [C++](#)



## □ 2. Tạo dự án :

4 Next steps

You're all set!

Make sure to check out the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

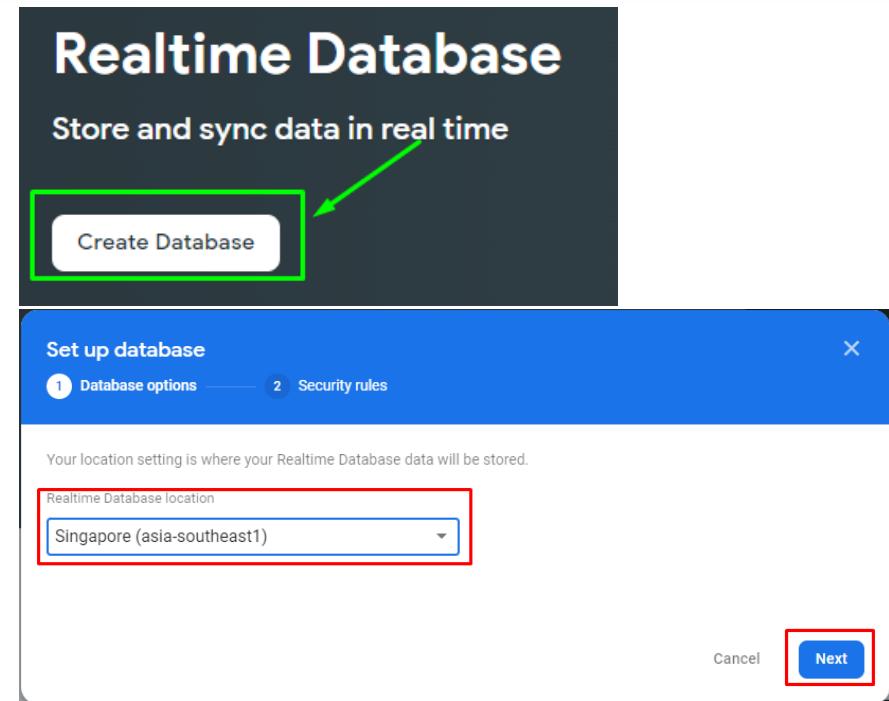
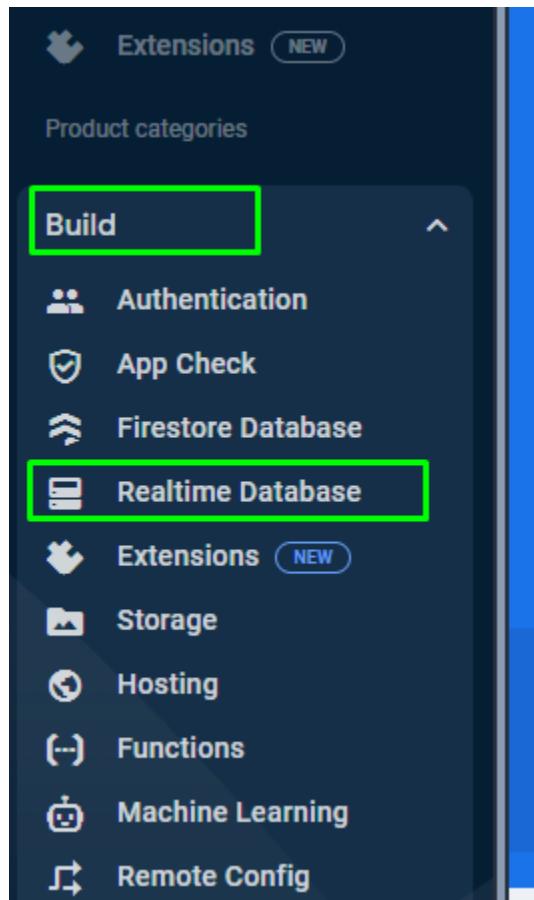
You can also explore [sample Firebase apps](#).

Or, continue to the console to explore Firebase.

Previous [Continue to console](#)



## □ 3. Firebase Realtime Database:



**Realtime Database**  
Store and sync data in real time

Create Database

**Set up database**

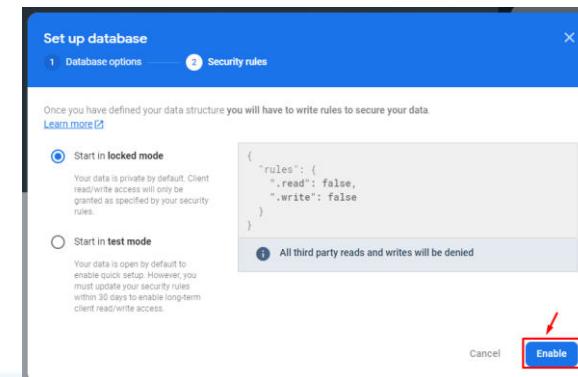
1 Database options    2 Security rules

Your location setting is where your Realtime Database data will be stored.

Realtime Database location

Singapore (asia-southeast1)

Cancel    Next



**Set up database**

1 Database options    2 Security rules

Once you have defined your data structure you will have to write rules to secure your data.  
[Learn more](#)

Start in **locked mode**  
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

Start in **test mode**  
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

{  
  "rules": {  
    ".read": false,  
    ".write": false  
  }  
}

All third party reads and writes will be denied

Cancel    Enable



❑ 3. Firebase Realtime Database:  
✓ Set Rule for *read* and *write* database:

Realtime Database

Data    **Rules**    Backups    Usage

Edit rules    Monitor rules

★ Default security rules are locked from access

```
1  {  
2    "rules": {  
3      ".read": false,  
4      ".write": false  
5    }  
6  }
```



ta    **Rules**    Backups    Usage

Edit rules    Monitor rules

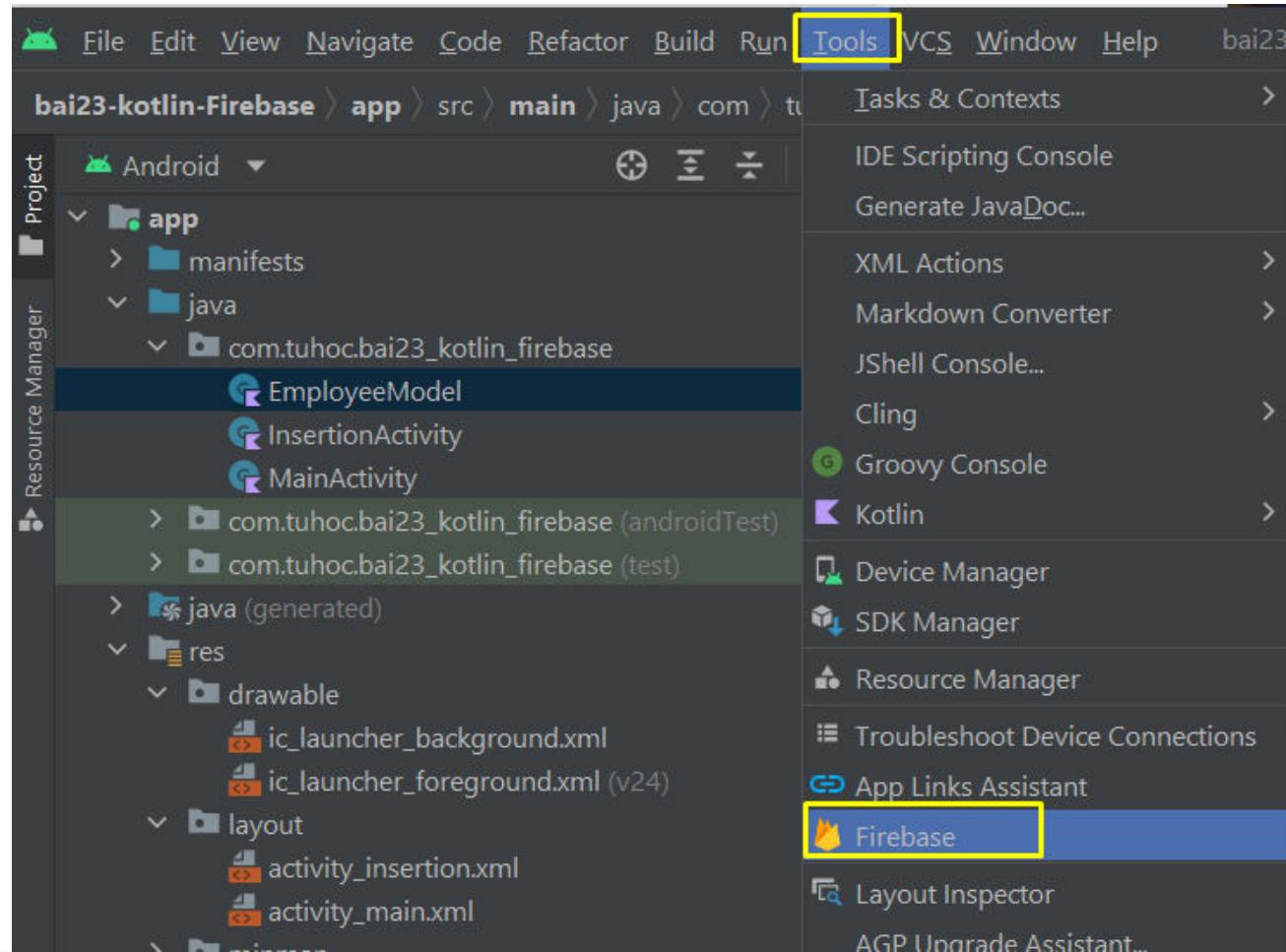
unpublished changes    **Publish**    Discard

★ Default security rules are locked from access

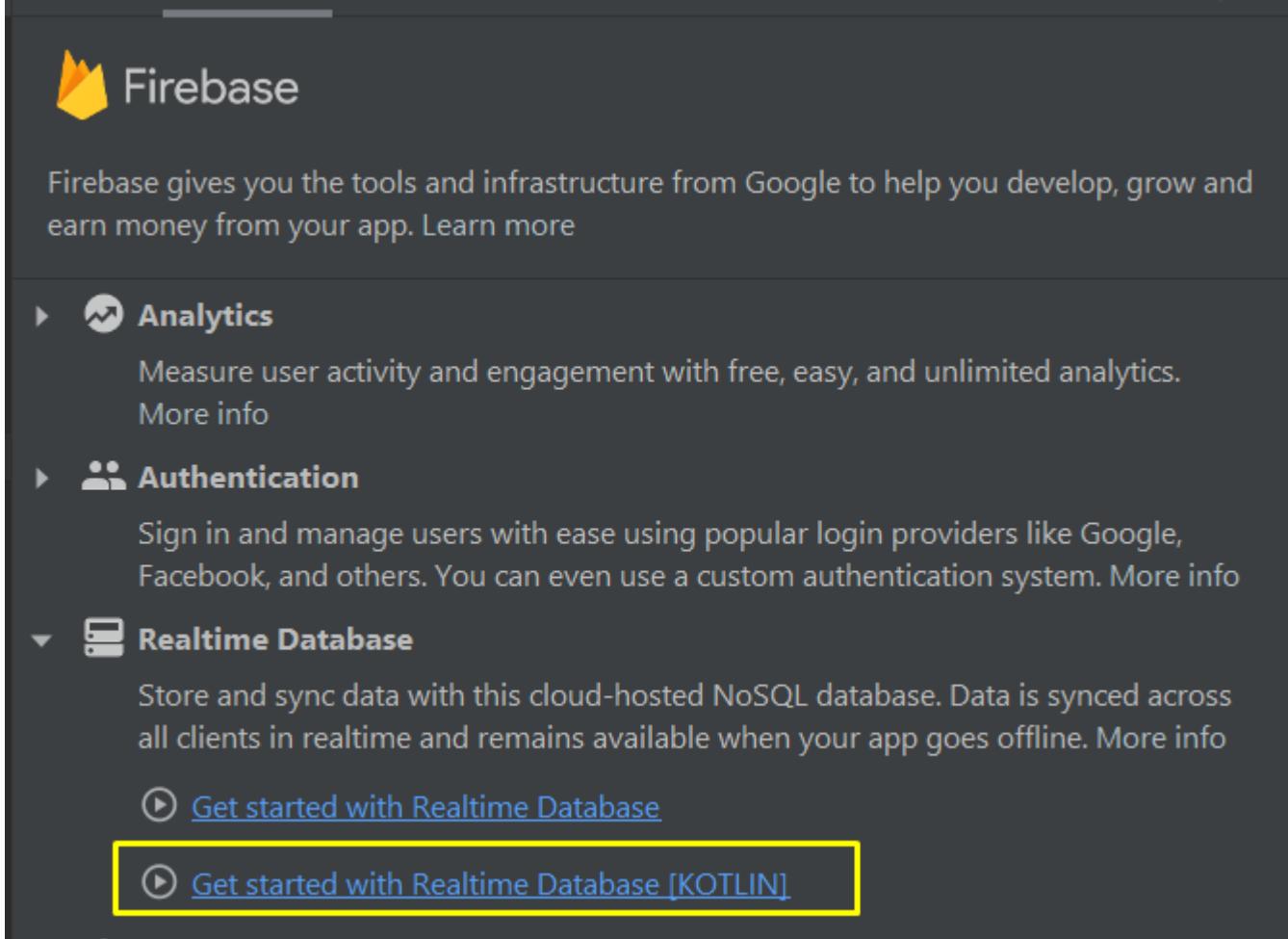
```
1  {  
2    "rules": {  
3      ".read": true,  
4      ".write": true  
5    }  
6  }
```



## □ 4. Connect firebase on AndroidStudio :



## □ 4. Connect firebase on AndroidStudio :



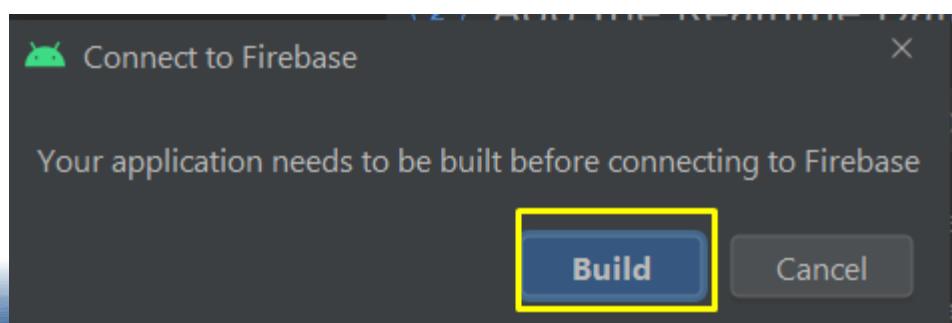
Firebase gives you the tools and infrastructure from Google to help you develop, grow and earn money from your app. Learn more

- ▶  **Analytics**  
Measure user activity and engagement with free, easy, and unlimited analytics.  
[More info](#)
- ▶  **Authentication**  
Sign in and manage users with ease using popular login providers like Google, Facebook, and others. You can even use a custom authentication system. [More info](#)
- ▼  **Realtime Database**  
Store and sync data with this cloud-hosted NoSQL database. Data is synced across all clients in realtime and remains available when your app goes offline. [More info](#)
  - ▶ [Get started with Realtime Database](#)
  - ▶ [Get started with Realtime Database \[KOTLIN\]](#)

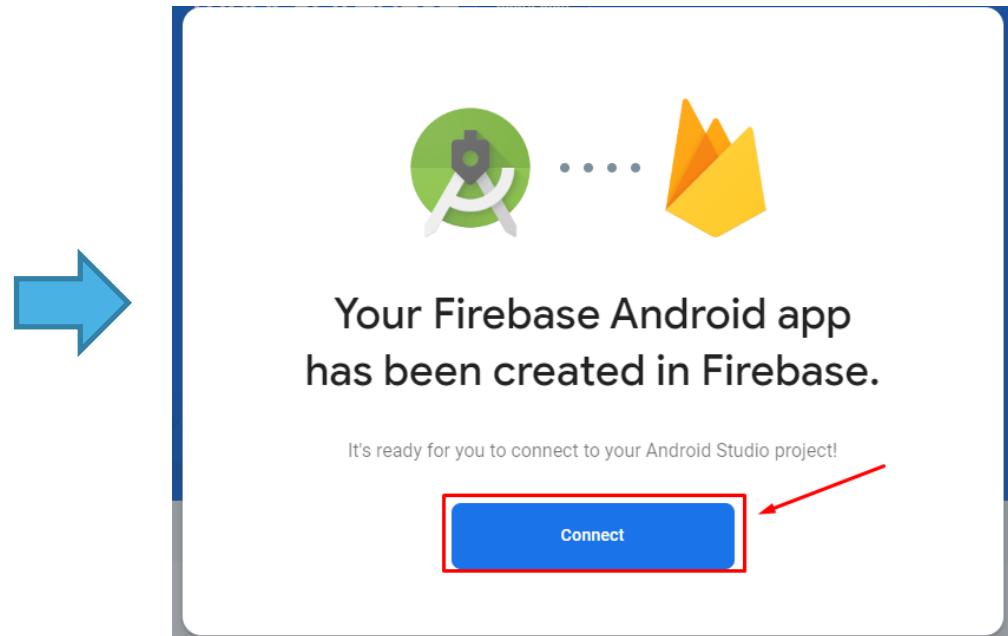
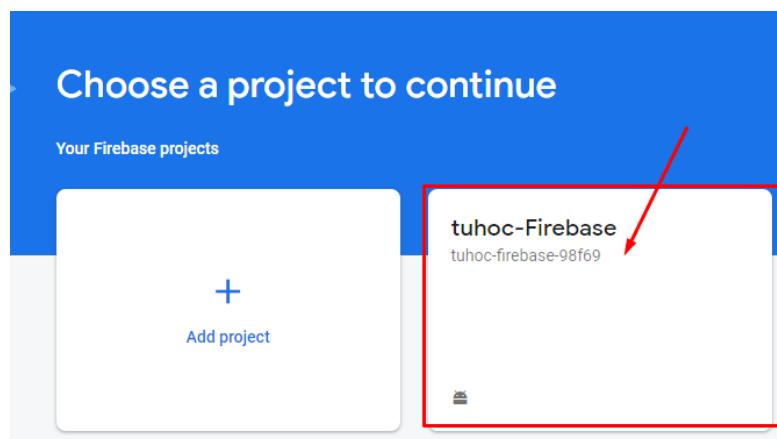


## □ 4. Connect firebase on AndroidStudio :

- ① Connect your app to Firebase  
[Connect to Firebase](#)
  
- ② Add the Realtime Database to your app  
[Add the Realtime Database SDK to your app](#)  
  
**NOTE:** After adding the SDK, here are some other helpful configurations to consider:
  - **Do you want an easier way to manage library versions?**  
You can use the Firebase Android BoM to manage your Firebase library



## □ 4. Connect firebase on AndroidStudio :

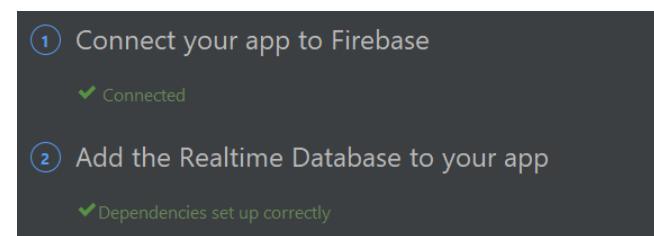
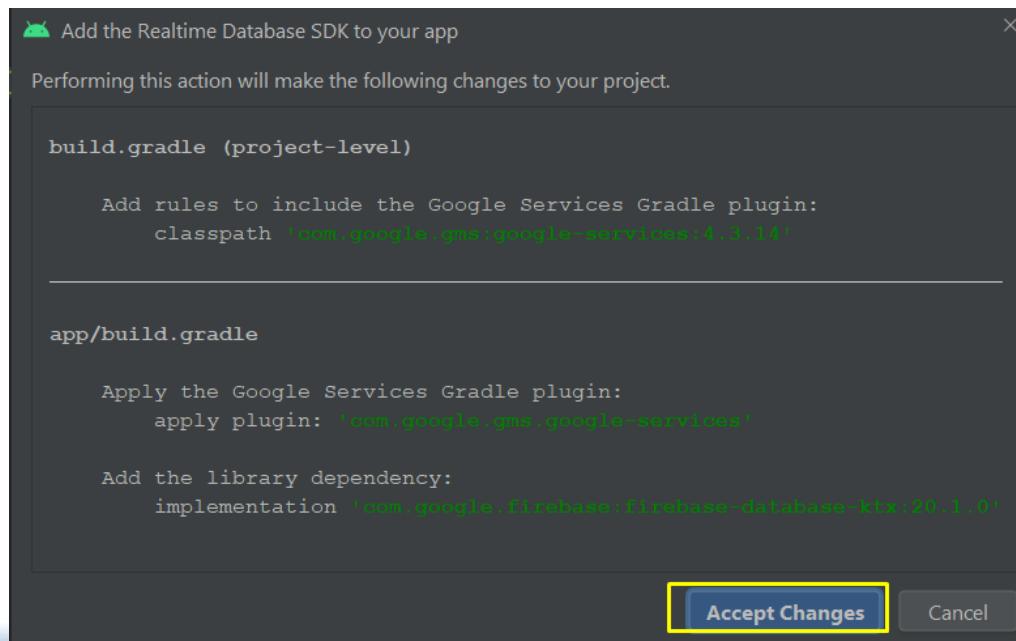
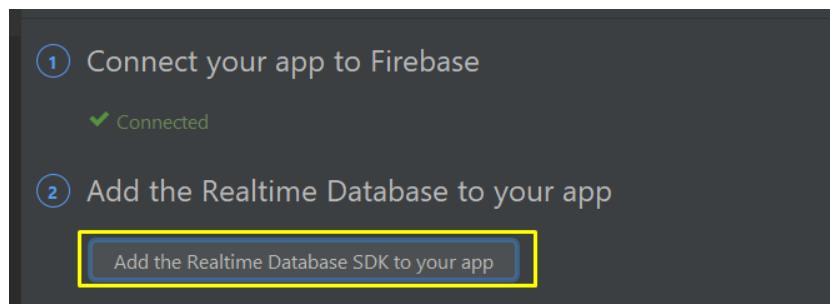


Your Android Studio project is connected  
to your Firebase Android app

You can now use Firebase in your project! Go back to Android Studio to start using one of the  
Firebase SDKs.

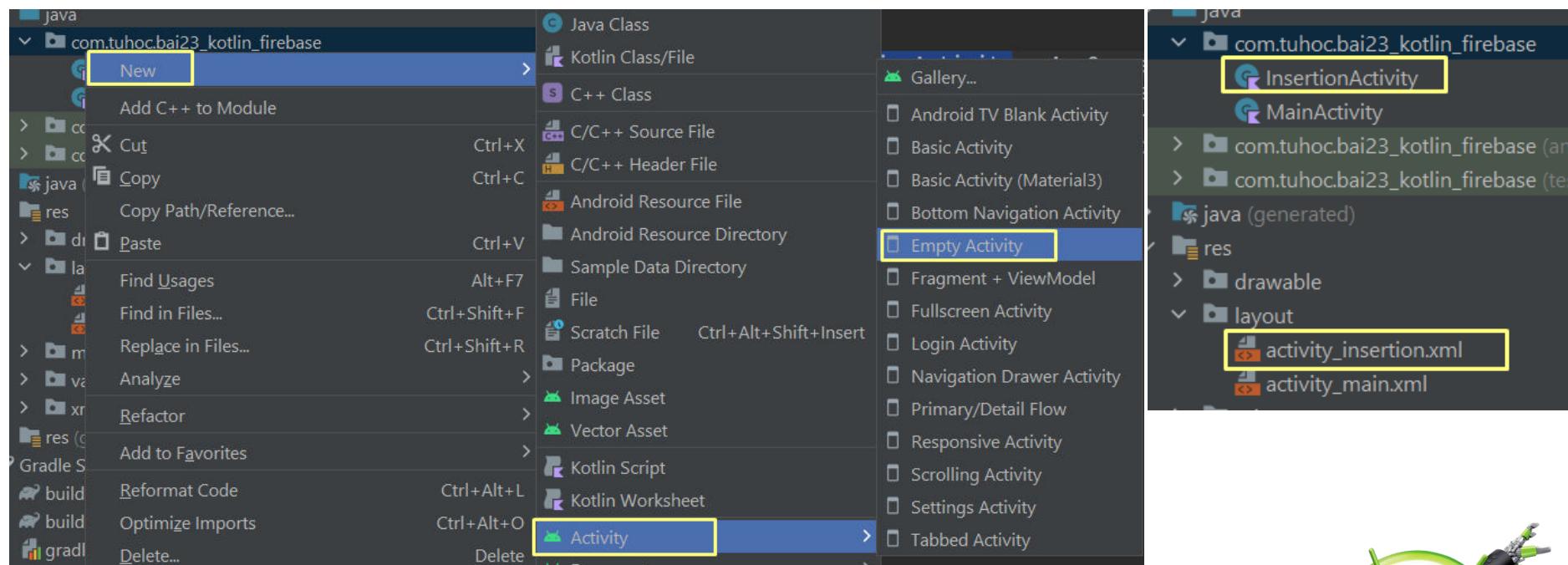


## □ 4. Connect firebase on AndroidStudio :



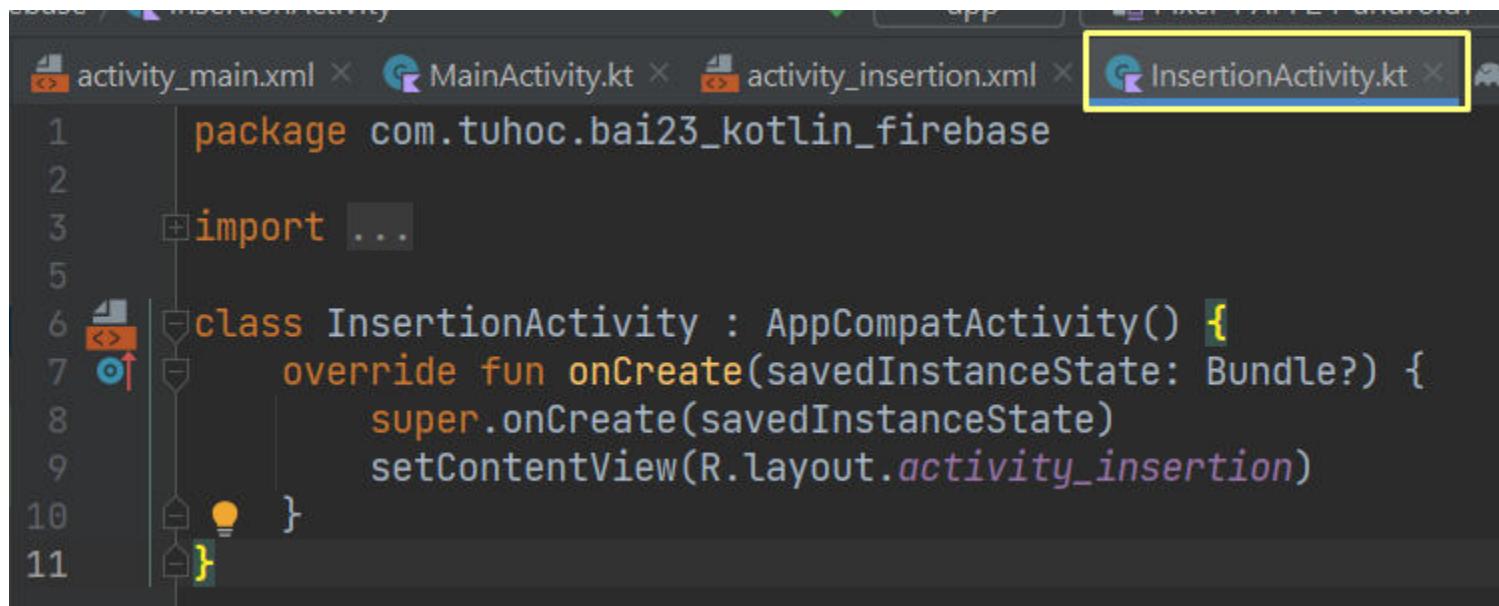
## □ 5. Tạo Activity insert data :

✓ Xem lại bài 14, 14.2 về intent :



## ❑ 5. Tạo Activity insert data :

✓ *Code InsertionActivity :*

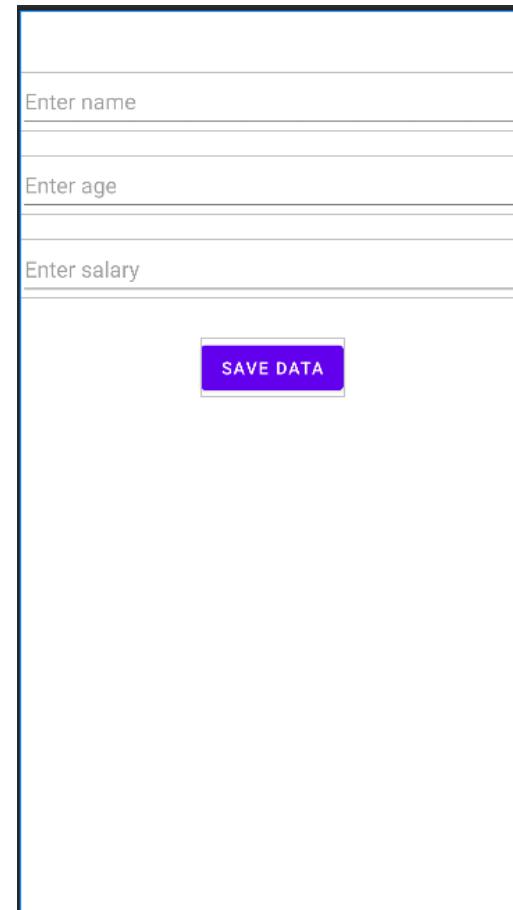
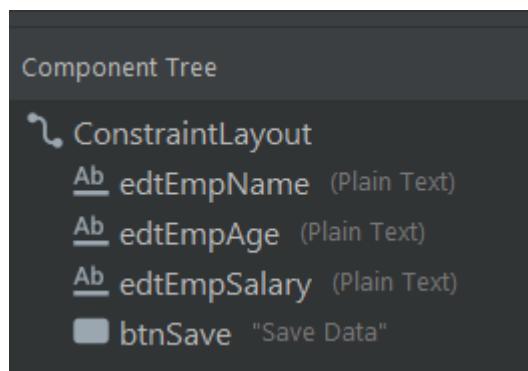


```
1 package com.tuhoc.bai23_kotlin_firebase
2
3 import ...
4
5
6 class InsertionActivity : AppCompatActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         setContentView(R.layout.activity_insertion)
10    }
11 }
```

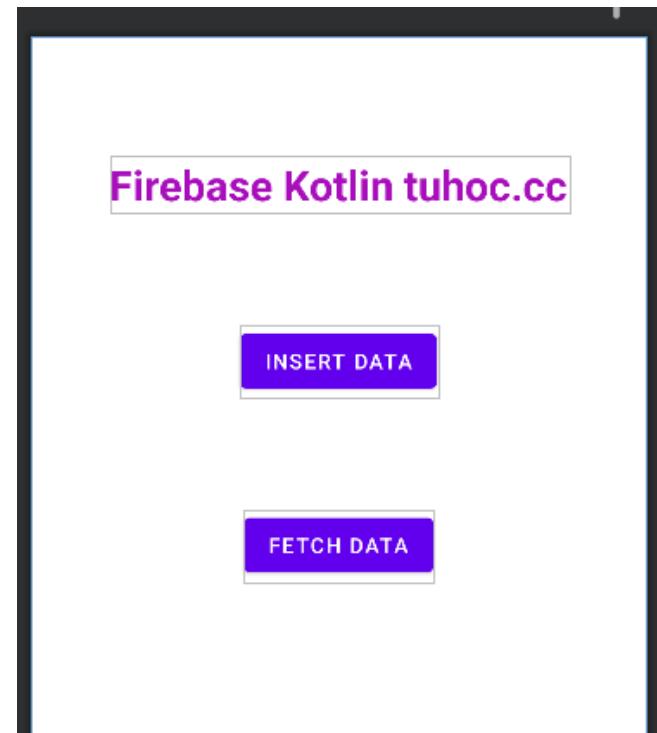
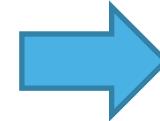
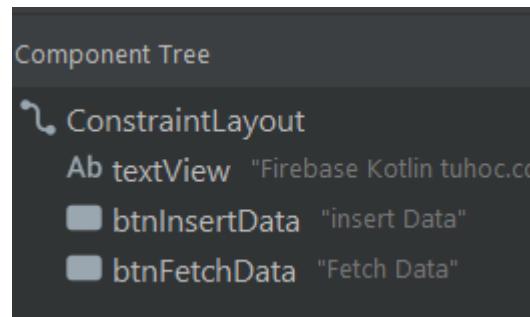


## ❑ 5. Tạo Activity insert data :

✓ *Giao diện Màn hình activity\_insertion.xml :*

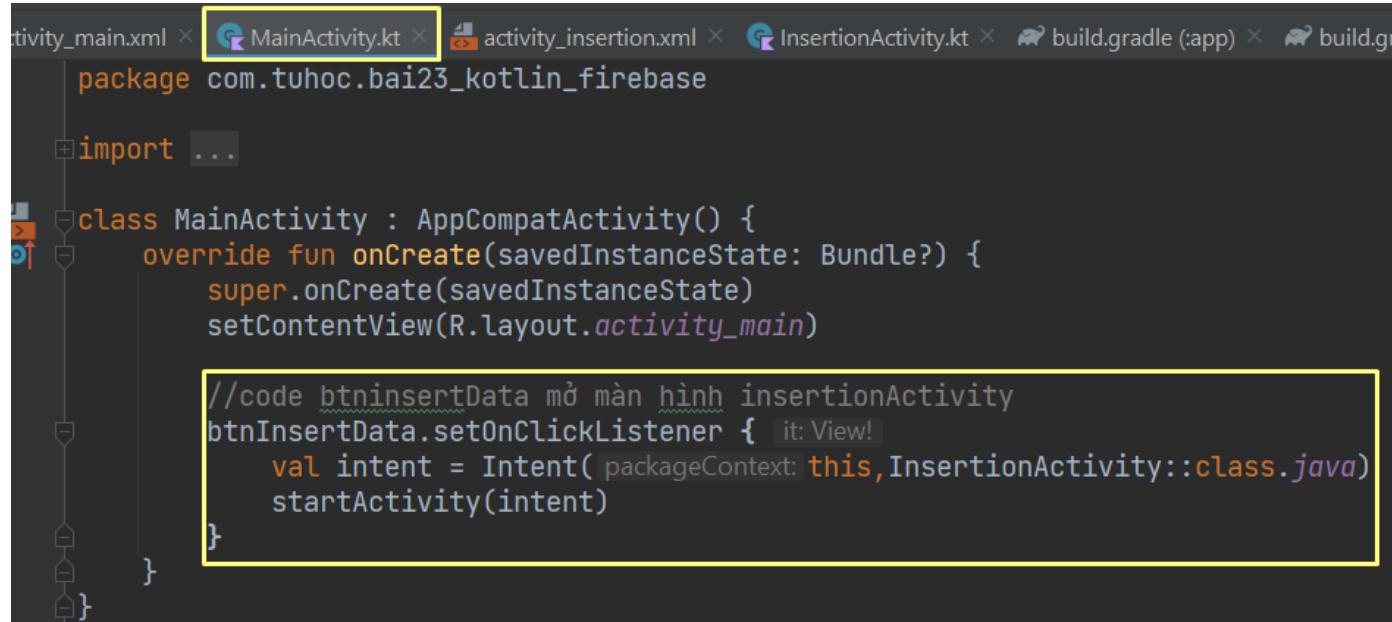


□ 6. Code giao diện chính **activity\_main.xml** :



## □ 7. Code MainActivity :

- ✓ *Gọi màn hình insertionActivity khi click vào button :*



```
activity_main.xml × MainActivity.kt × activity_insertion.xml × InsertionActivity.kt × build.gradle (:app) × build.gr
package com.tuhoc.bai23_kotlin_firebase

import ...

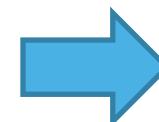
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        //code btninsertData mở màn hình insertionActivity
        btnInsertData.setOnClickListener { it: View!
            val intent = Intent(packageContext: this, InsertionActivity::class.java)
            startActivity(intent)
        }
    }
}
```

Firebase Kotlin tuhoc.cc

INSERT DATA

FETCH DATA

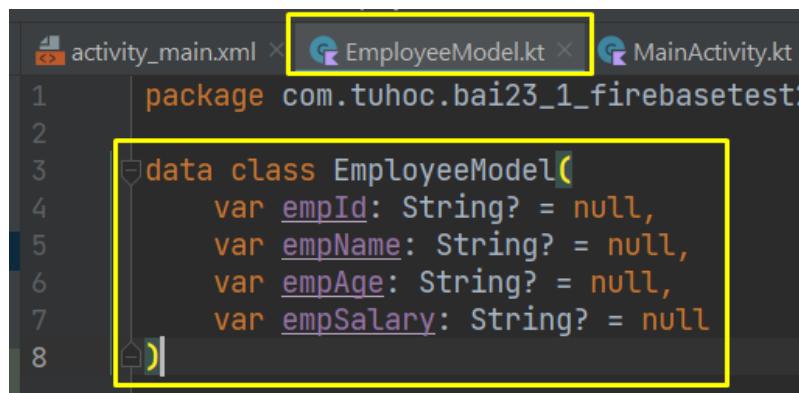
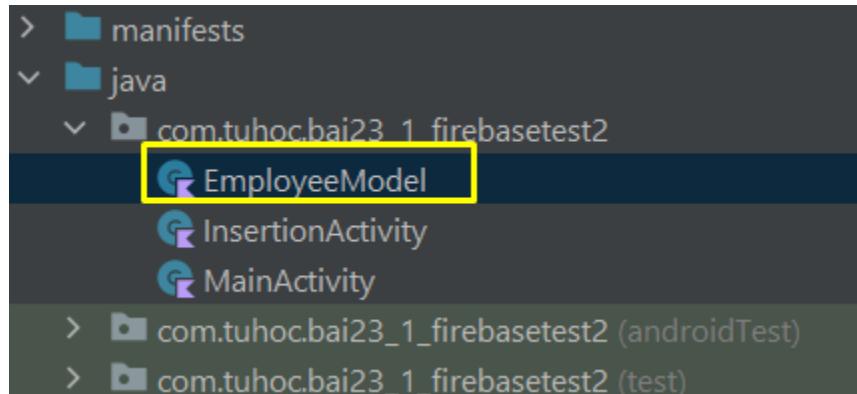


Enter name	<input type="text"/>
Enter age	<input type="text"/>
Enter salary	<input type="text"/>
<input type="button" value="SAVE DATA"/>	



## □ 8. add Class EmployeeModel:

- ✓ *Class sẽ gồm các thành phần trong table sau này cần thêm :*



The screenshot shows the 'EmployeeModel.kt' file in the Android Studio code editor. The code defines a data class 'EmployeeModel' with four nullable string properties: 'empId', 'empName', 'empAge', and 'empSalary'. The code is as follows:

```
1 package com.tuhoc.bai23_1_firebaseTest2
2
3 data class EmployeeModel(
4     var empId: String? = null,
5     var empName: String? = null,
6     var empAge: String? = null,
7     var empSalary: String? = null
8 )
```



## □ 8. Code InsertionActivity xử lý thêm dữ liệu vào database :

```
class InsertionActivity : AppCompatActivity() {  
    private lateinit var dbRef : DatabaseReference  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_insertion)  
  
        dbRef = FirebaseDatabase.getInstance().getReference( path: "Employees")  
        //xử lý sự kiện khi nhấn vào nút save  
        btnSave.setOnClickListener { it: View!  
            saveEmployeeData()  
        }  
    }  
}
```



## □ 8. Code InsertionActivity xử lý thêm dữ liệu vào database :

✓ *Code hàm saveEmployeeData() : hàm xử lý thêm dữ liệu vào database*

```
private fun saveEmployeeData() {
    //getting value
    val empName = edtEmpName.text.toString()
    val empAge = edtEmpAge.text.toString()
    val empSalary = edtEmpSalary.text.toString()
    //kiểm tra xem các ô giá trị nhập có bị bỏ trống
    if (empName.isEmpty()) {
        edtEmpName.error = "Please enter name"
        return
    }
    if (empAge.isEmpty()) {
        edtEmpAge.error = "Please enter age"
        return
    }
    if (empSalary.isEmpty()) {
        edtEmpSalary.error = "Please enter salary"
        return
    }

    //đẩy dữ liệu
    val empId = dbRef.push().key!!
    val employee = EmployeeModel(empId, empName, empAge, empSalary)
    dbRef.child(empId).setValue(employee)
        .addOnCompleteListener { it: Task<Void!>
            Toast.makeText(context: this, text: "Data insert thành công", Toast.LENGTH_SHORT).show()
        }.addOnFailureListener { err ->
            Toast.makeText(context: this, text: "Error ${err.message}", Toast.LENGTH_SHORT).show()
        }
}
```



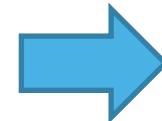
## □ 9. Test insert data on this app ☺ :

Data   Rules   Backups   Usage

Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://tuhoc-firebase-98f69-default.firebaseio.com/.json

https://tuhoc-firebase-98f69-default.firebaseio.com/.json/:null



tuhoc

18

1000000

**SAVE DATA**

https://tuhoc-firebase-98f69-default.firebaseio.com/.json

Employees

-NG64ikowRoBtxfZ4jHN

- empAge: "18"
- empId: "-NG64ikowRoBtxfZ4jHN"
- empName: "tuhoc"
- empSalary: "1000000"

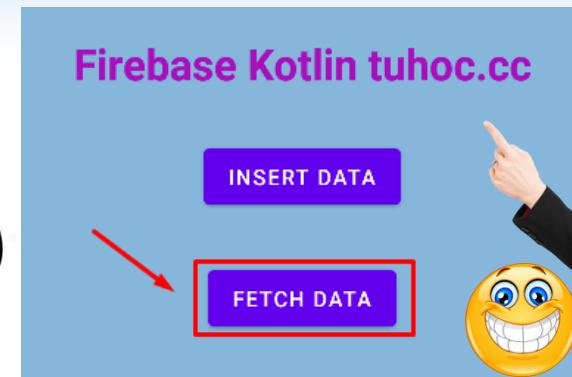


## □ 10. Xoá data các ô nhập liệu sau khi insert thành công :

- ✓ *Code trong hàm `saveEmployeeData()` : hàm xử lý thêm dữ liệu vào database*

```
//đẩy dữ liệu
val empId = dbRef.push().key!!
val employee = EmployeeModel(empId, empName, empAge, empSalary)
dbRef.child(empId).setValue(employee)
    .addOnCompleteListener { it: Task<Void!>
        Toast.makeText(context: this, text: "Data insert thành công", Toast.LENGTH_SHORT).show()
        //xoá trắng các ô nhập liệu
        edtEmpName.text.clear()
        edtEmpAge.text.clear()
        edtEmpSalary.text.clear()
    }
    .addOnFailureListener { err ->
        Toast.makeText(context: this, text: "Error ${err.message}", Toast.LENGTH_SHORT).show()
    }
}
```





**firebase kotlin part 2**

**View data on  
RecyclerView**

tuhoc.cc

jacky chan

ly lien kiet



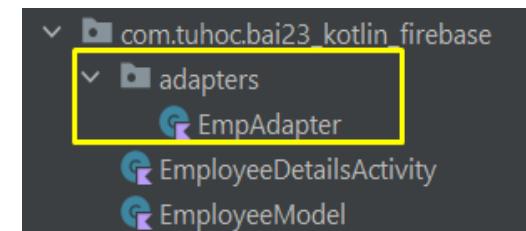
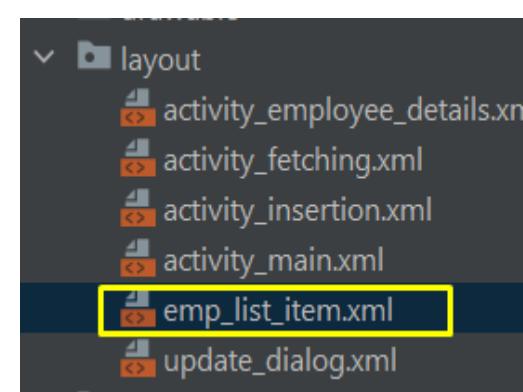
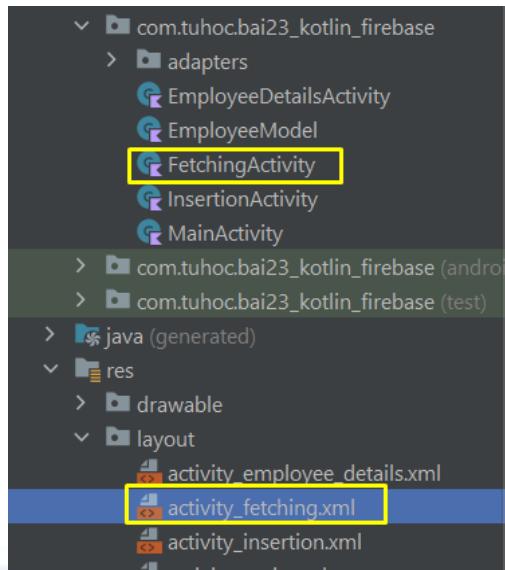
## □ 11. Hiển thị data lên RecyclerView :

✓ Các bước tiến hành

Step 1:  
Tạo màn hình  
**FetchingActivity**

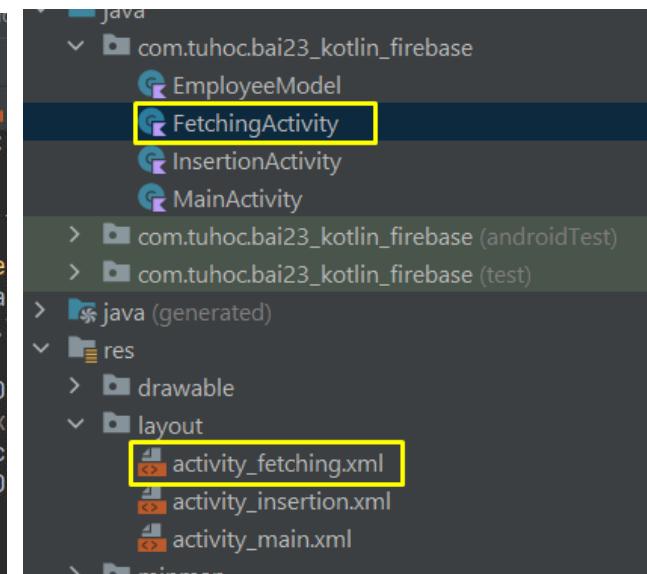
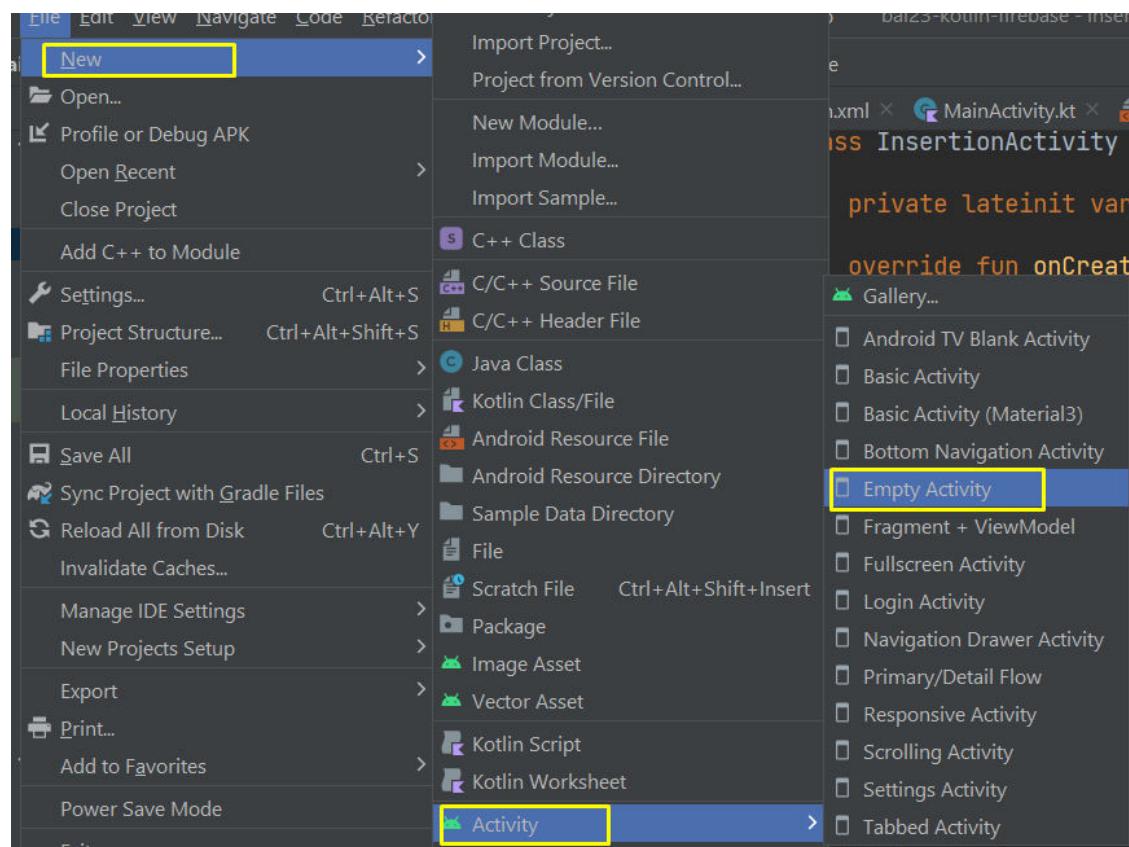
Step 2:  
Tạo layout mẫu 1  
item cho  
recyclerView  
**emp\_list\_item.xml**

Step 3:  
Make Class  
**EmpAdapter**



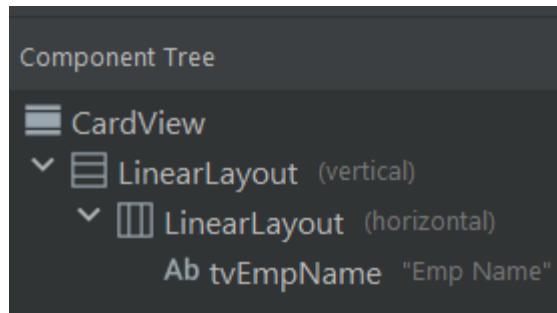
## ❑ 11. Hiển thị data lên RecyclerView :

### ✓ 11-1 Step 1: Create new Activity FetchingActivity



## □ 11. Hiển thị data lên RecyclerView :

- ✓ 11-2 Step2 Tạo layout cho 1 item **emp\_list\_item.xml**



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardElevation="8dp"
    app:cardCornerRadius="8dp"
    android:layout_margin="8dp">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:padding="16dp"
        android:layout_gravity="center"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical"
            android:orientation="horizontal">

            <TextView
                android:id="@+id/tvEmpName"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Emp Name"
                android:layout_gravity="center"
                android:gravity="center"
                android:textColor="@color/black"
                android:textSize="30sp"
                android:textStyle="bold"/>

        </LinearLayout>

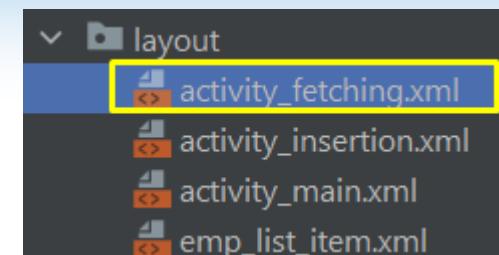
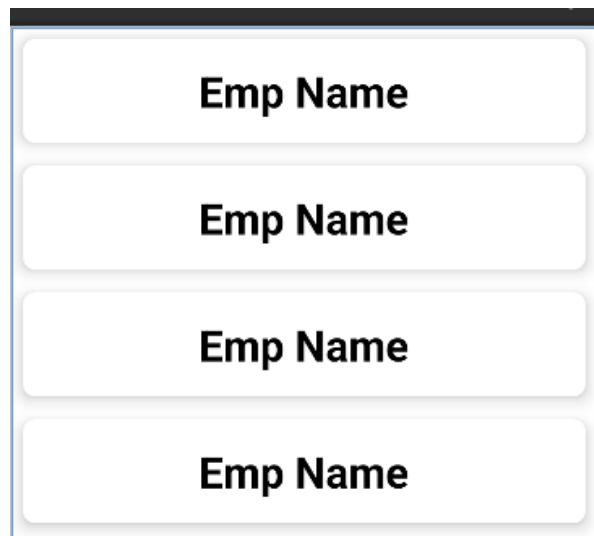
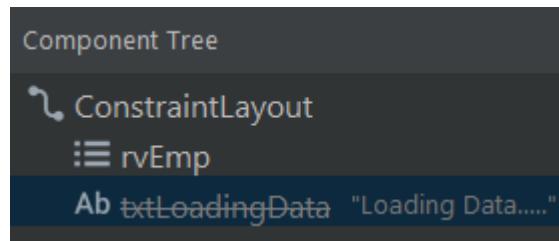
    </LinearLayout>

</CardView>
```



## □ 11. Hiển thị data lên RecyclerView :

✓ 11-3 Code *activity\_fetching.xml*

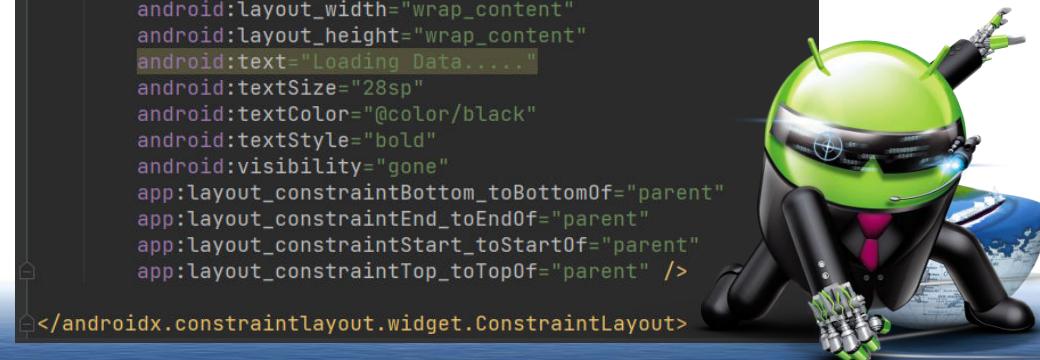


```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".FetchingActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvEmp"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="1dp"
        android:layout_marginTop="1dp"
        android:layout_marginEnd="1dp"
        android:layout_marginBottom="1dp"
        tools:listitem="@layout/emp_list_item"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

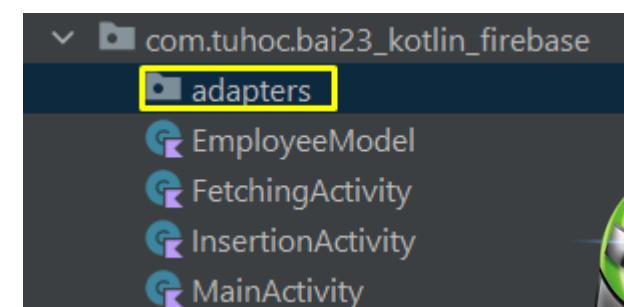
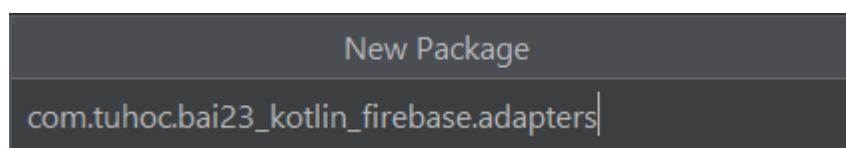
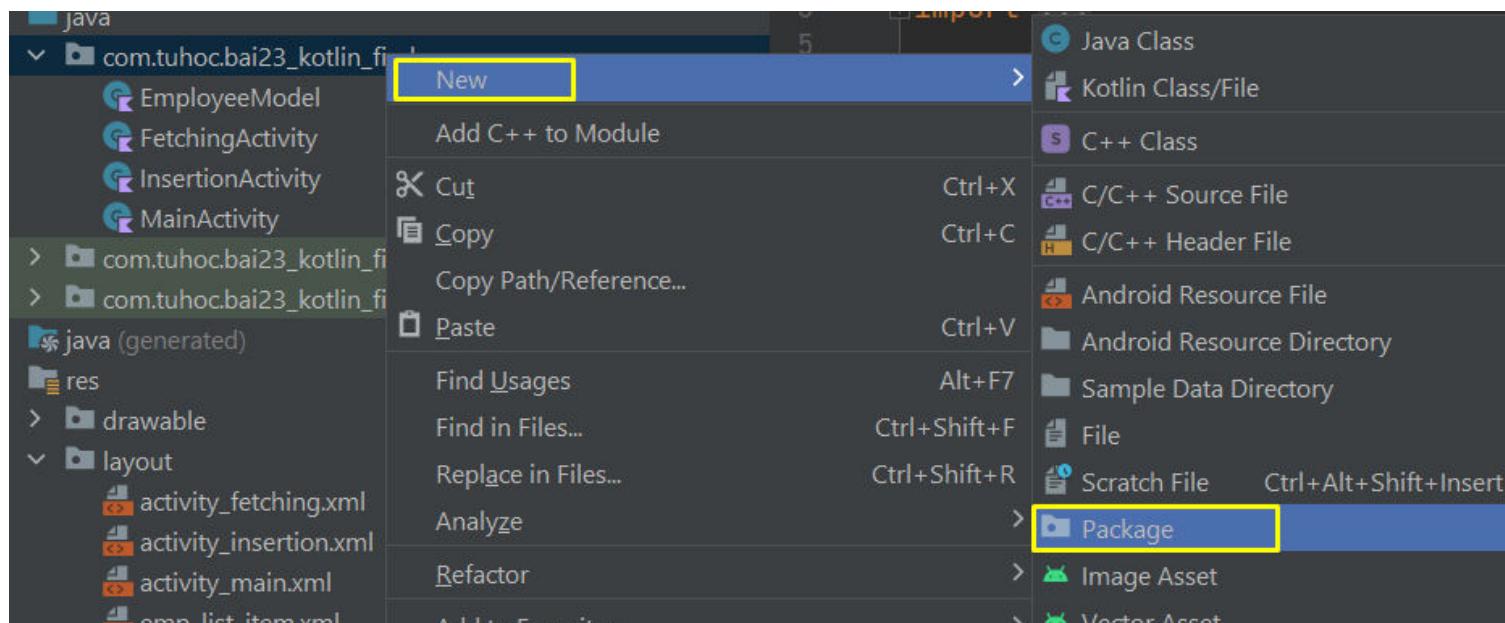
    <TextView
        android:id="@+id/txtLoadingData"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Loading Data....."
        android:textSize="28sp"
        android:textColor="@color/black"
        android:textStyle="bold"
        android:visibility="gone"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



## □ 11. Hiển thị data lên gridView :

- ✓ 11-4 Step 3: Tạo **adapter** cho recyclerView



## □ 11. Hiển thị data lên RecyclerView :

- ✓ 11-4 Step 3: Tạo **adapter** cho recyclerView

```
package com.tuhoc.bai23_kotlin_firebase.adapter

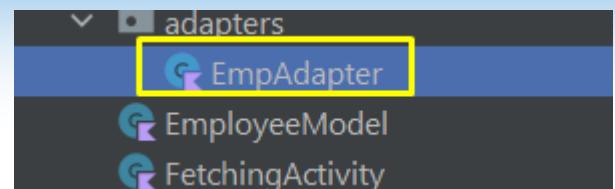
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.tuhoc.bai23_kotlin_firebase.EmployeeModel
import com.tuhoc.bai23_kotlin_firebase.R
import kotlinx.android.synthetic.main.emp_list_item.view.*

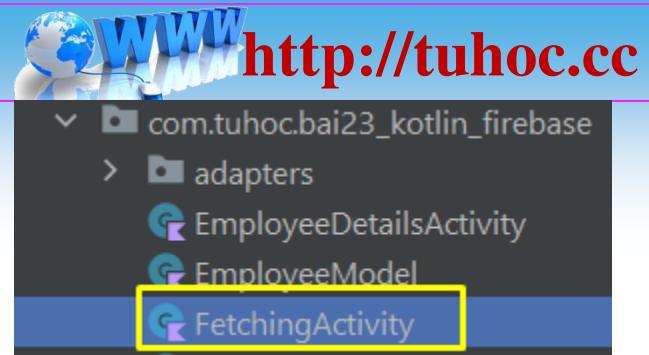
//bài 16
class EmpAdapter(private val ds:ArrayList<EmployeeModel>) :RecyclerView.Adapter<EmpAdapter.ViewHolder>() {
    //tạo class viewholder
    class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView)

    //ctrl + i
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        val itemView = LayoutInflater.from(parent.context)
            .inflate(R.layout.emp_list_item, parent, attachToRoot: false)
        return ViewHolder(itemView)
    }

    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        holder.itemView.apply {
            tvEmpName.text = ds[position].empName
        }
    }

    override fun getItemCount(): Int {
        return ds.size
    }
}
```



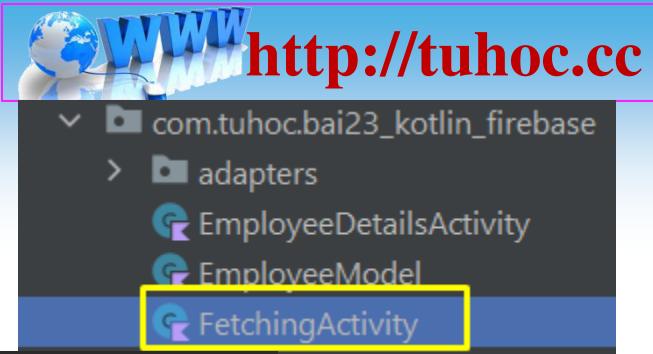


## □ 11. Hiển thị data lên RecyclerView :

✓ 11-5 Code *FetchingActivity.kt*

```
class FetchingActivity : AppCompatActivity() {  
  
    private lateinit var ds:ArrayList<EmployeeModel>  
    private lateinit var dbRef :DatabaseReference  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_fetching)  
  
        rvEmp.layoutManager = LinearLayoutManager(context: this)  
        rvEmp.setHasFixedSize(true)  
        ds = arrayListOf<EmployeeModel>()  
        GetThongTinNV() // alt + enter  
    }  
}
```





## □ 11. Hiển thị data lên RecyclerView :

- ✓ 11-5 Code *FetchingActivity.kt*
- GetThongTinNV()**

```
private fun GetThongTinNV() {
    rvEmp.visibility = View.GONE
    txtLoadingData.visibility = View.VISIBLE
    dbRef = FirebaseDatabase.getInstance().getReference( path: "Employees")
    //Để đọc dữ liệu tại một đường dẫn và lắng nghe các thay đổi,
    //hãy sử dụng addValueEventListener()
    //https://firebase.google.com/docs/database/android/read-and-write
    dbRef.addValueEventListener(object : ValueEventListener{
        //ctrl + i
        override fun onDataChange(snapshot: DataSnapshot) {
            ds.clear()
            if (snapshot.exists()){
                for(empSnap in snapshot.children){
                    val empData = empSnap.getValue(EmployeeModel::class.java)
                    ds.add(empData!!)
                }
                mAdapter = EmpAdapter(ds)
                rvEmp.adapter = mAdapter
                rvEmp.visibility = View.VISIBLE
                txtLoadingData.visibility = View.GONE
            }
        }
        override fun onCancelled(error: DatabaseError) {
            TODO(reason: "Not yet implemented")
        }
    })
}
```





# **firebase kotlin part 3**

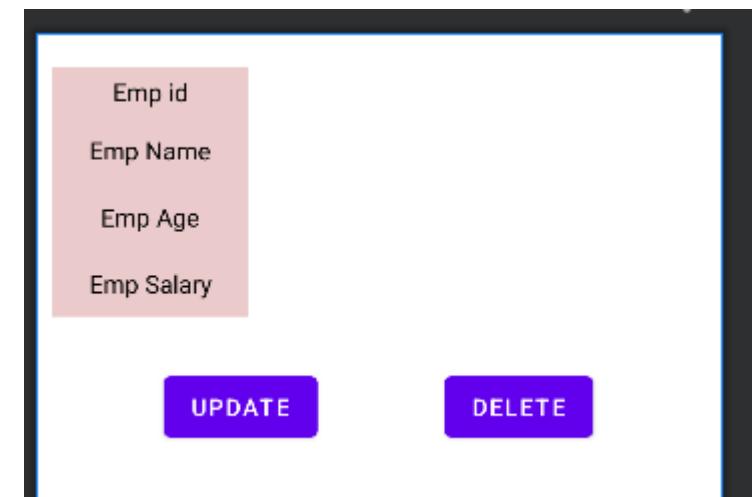
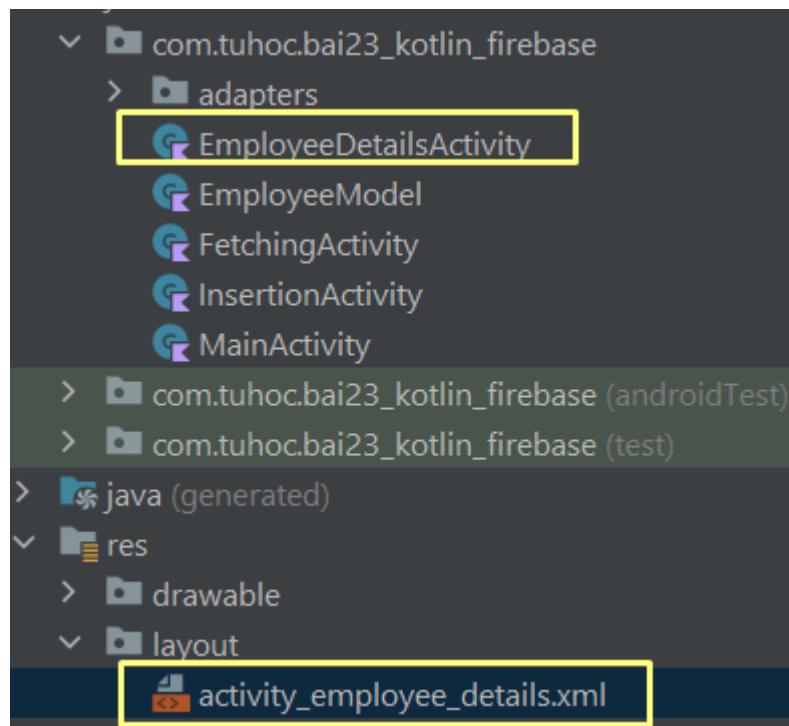
## **View detail database & Delete Database**

Emp id	-NGGZcsqXKq_qbwTlpyL
Emp Name	ly lien kiet
Emp Age	50
Emp Salary	555555

**UPDATE** **DELETE**

## □ 12. Hiển thị thông tin chi tiết data :

- ✓ 12-1 Tạo new Activity *EmployeeDetailsActivity*
- ✓ Thiết kế giao diện *activity\_employee\_details.xml*



## □ 12. Hiển thị thông tin chi tiết data :

### ✓ 12-2 Code class adapter *EmpAdapter*

```
//bai 16
class EmpAdapter(private val ds:ArrayList<EmployeeModel>) :RecyclerView.Adapter<EmpAdapter.ViewHolder>() {
    //code adapter lắng nghe sự kiện
    private lateinit var mClickListener: onItemClickListener
    interface onItemClickListener{
        fun onItemClick(position: Int)
    }
    fun setOnItemClickListener(clickListener: onItemClickListener){
        mClickListener = clickListener
    }

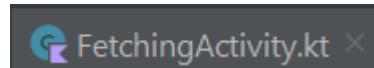
    //tạo class viewholder
    class ViewHolder(itemView: View,clickListener: onItemClickListener) : RecyclerView.ViewHolder(itemView){
        init {
            itemView.setOnClickListener { it: View!
                clickListener.onItemClick(adapterPosition)
            }
        }
    }

    //ctrl + i
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        val itemView = LayoutInflater.from(parent.context)
            .inflate(R.layout.emp_list_item,parent, attachToRoot: false)
        return ViewHolder(itemView, mClickListener)
    }
}
```



## □ 12. Hiển thị thông tin chi tiết data :

✓ 12-3 *setItemClickListener RecyclerView FetchingActivity.kt*



```
private fun GetThongTinNV() {
    rvEmp.visibility = View.GONE
    txtLoadingData.visibility = View.VISIBLE
    dbRef = FirebaseDatabase.getInstance().getReference(path: "Employees")
    //Để đọc dữ liệu tại một đường dẫn và lắng nghe các thay đổi,
    //hãy sử dụng addValueEventListener()
    //https://firebase.google.com/docs/database/android/read-and-write
    dbRef.addValueEventListener(object : ValueEventListener{
        //ctrl +i
        override fun onDataChange(snapshot: DataSnapshot) {
            ds.clear()
            if (snapshot.exists()){
                for(empSnap in snapshot.children){
                    val empData = empSnap.getValue(EmployeeModel::class.java)
                    ds.add(empData!!)
                }
                val mAdapter = EmpAdapter(ds)
                rvEmp.adapter = mAdapter
                //code lắng nghe sự kiện click lên item rv
                mAdapter.setOnItemClickListener(object : EmpAdapter.OnItemClickListener{
                    //ctrl +i
                    override fun onItemClick(position: Int) {
                        val intent = Intent(packageContext: this@FetchingActivity, EmployeeDetailsActivity::class.java)
                        //put extras
                        intent.putExtra(name: "empId", ds[position].empId)
                        intent.putExtra(name: "empName", ds[position].empName)
                        intent.putExtra(name: "empAge", ds[position].empAge)
                        intent.putExtra(name: "empSalary", ds[position].empSalary)
                        startActivity(intent)
                    }
                })
            }
        }
    })
}
```



## □ 12. Hiển thị thông tin chi tiết data :

- ✓ 12-4 Code *EmployeeDetailsActivity* nhận dữ liệu gửi từ *FetchingActivity*

```
class EmployeeDetailsActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_employee_details)  
  
        setValuesToViews()  
    }  
  
    private fun setValuesToViews() {  
        tvEmpId.text = intent.getStringExtra(name: "empId")  
        tvEmpName.text = intent.getStringExtra(name: "empName")  
        tvEmpAge.text = intent.getStringExtra(name: "empAge")  
        tvEmpSalary.text = intent.getStringExtra(name: "empSalary")  
    }  
}
```



## ❑ 13. Xoá bản ghi :

EmployeeDetailsActivity.kt

```
class EmployeeDetailsActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_employee_details)

        setValueToView()
        //code cho button delete
        btnDelete.setOnClickListener { it: View!
            //alt + enter
            deleteRecord(
                intent.getStringExtra(name: "empId").toString()
            )
        }
    }

    private fun deleteRecord(id: String) {
        val dbRef = FirebaseDatabase.getInstance().getReference( path: "Employees").child(id)
        val mTask = dbRef.removeValue()
        mTask.addOnSuccessListener { it: Void!
            Toast.makeText( context: this, text: "Employee data đã xoá", Toast.LENGTH_SHORT).show()
            val intent = Intent( packageContext: this,FetchingActivity::class.java)
            finish()
            startActivity(intent)
        }.addOnFailureListener { err->
            Toast.makeText( context: this, text: "Delete err ${err.message}", Toast.LENGTH_SHORT).show()
        }
    }
}
```





## firebase kotlin part 4 Update Database

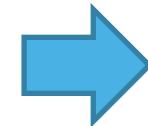
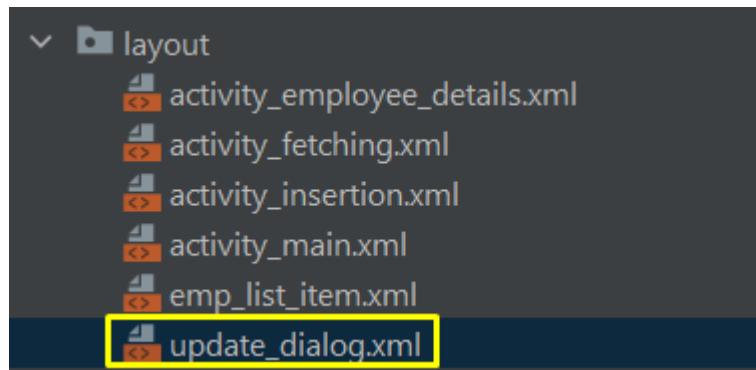
Emp id	-NGB2QJR_V8e8RSUOs-U
Emp Name	tuhoc.cc
Emp Age	18
Emp Salary	555555

**UPDATE**      **DELETE**



## ❑ 14. Update data :

- ✓ 14-1 Thiết kế giao diện *update\_dialog.xml*



Employee Name

Employee Age

Employee Salary

**UPDATE DATA**

Updating tuhoc.cc Record

tuhoc.cc

---

18

---

555555

---

**UPDATE DATA**



## ❑ 14. Update data :

- ✓ 14-2 Lắng nghe sự kiện click lên btnUpdate

EmployeeDetailsActivity.kt ×

```
class EmployeeDetailsActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_employee_details)

        setValuesToViews()

        btnUpdate.setOnClickListener { it: View!
            openUpdateDialog(
                intent.getStringExtra(name: "empId").toString(),
                intent.getStringExtra(name: "empName").toString()
            )
        }
    }
}
```



## □ 14. Update data :

- ✓ 14-3 code hàm *openUpdateDialog()*
- ✓ *Hiển thị thông tin lên Dialog*

```
private fun openUpdateDialog(  
    empId: String,  
    empName: String  
) {  
    val mDialog = AlertDialog.Builder( context: this)  
    val inflater = layoutInflater  
    val mDialogView = inflater.inflate(R.layout.update_dialog, root: null)  
    mDialog.setView(mDialogView)  
  
    val etEmpName = mDialogView.findViewById<EditText>(R.id.etEmpName)  
    val etEmpAge = mDialogView.findViewById<EditText>(R.id.etEmpAge)  
    val etEmpSalary = mDialogView.findViewById<EditText>(R.id.etEmpSalary)  
    val btnUpdateData = mDialogView.findViewById<Button>(R.id.btnUpdateData)  
  
    etEmpName.setText(intent.getStringExtra( name: "empName").toString())  
    etEmpAge.setText(intent.getStringExtra( name: "empAge").toString())  
    etEmpSalary.setText(intent.getStringExtra( name: "empSalary").toString())  
  
    mDialog.setTitle("Updating $empName Record")  
  
    val alertDialog = mDialog.create()  
    alertDialog.show()  
}
```



## □ 14. Update data :

- ✓ 14-4 code hàm *openUpdateDialog()*
- ✓ *Update data khi click button btnUpdateData*

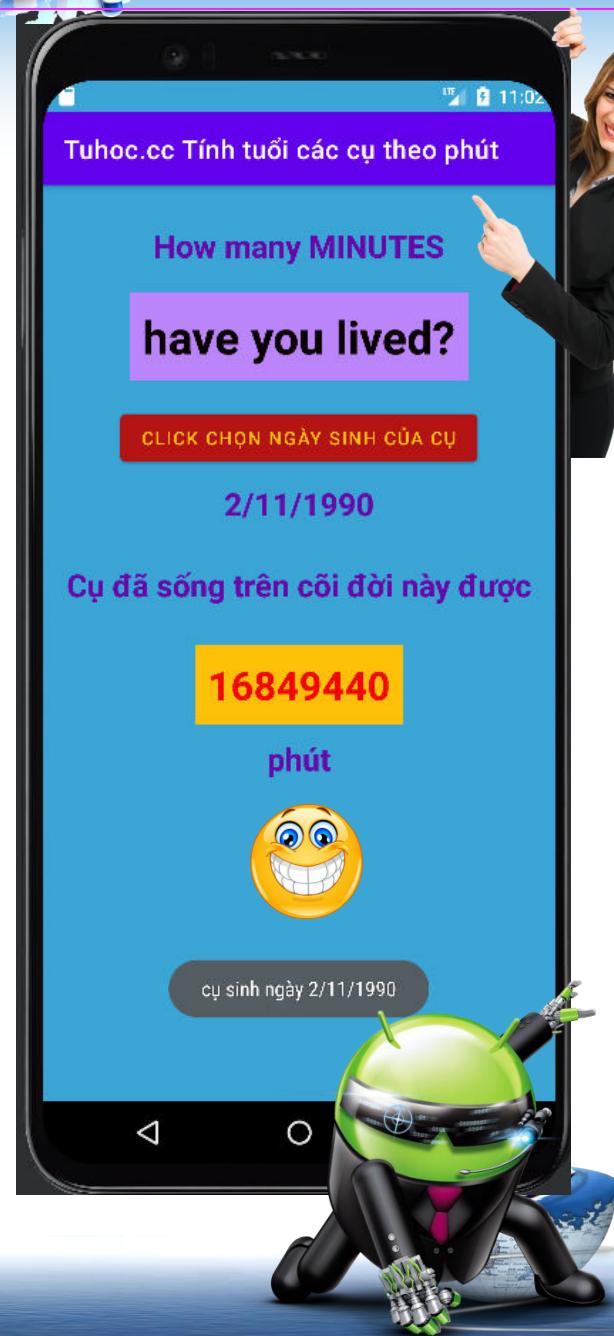
```
btnUpdateData.setOnClickListener { it: View!  
    updateEmpData(  
        empId,  
        etEmpName.text.toString(),  
        etEmpAge.text.toString(),  
        etEmpSalary.text.toString()  
    )  
  
    Toast.makeText(applicationContext, text: "Employee Data Updated", Toast.LENGTH_LONG).show()  
  
    //we are setting updated data to our textviews  
    tvEmpName.text = etEmpName.text.toString()  
    tvEmpAge.text = etEmpAge.text.toString()  
    tvEmpSalary.text = etEmpSalary.text.toString()  
  
    alertDialog.dismiss()  
}  
}  
private fun updateEmpData(  
    id: String,  
    name: String,  
    age: String,  
    salary: String  
) {  
    val dbRef = FirebaseDatabase.getInstance().getReference(path: "Employees").child(id)  
    val empInfo = EmployeeModel(id, name, age, salary)  
    dbRef.setValue(empInfo)  
}
```





## Simple App with kotlin

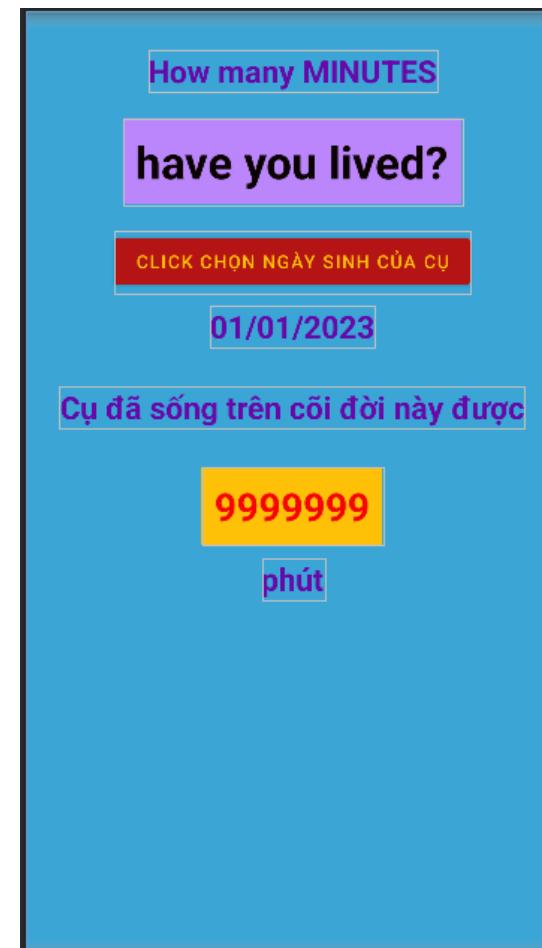
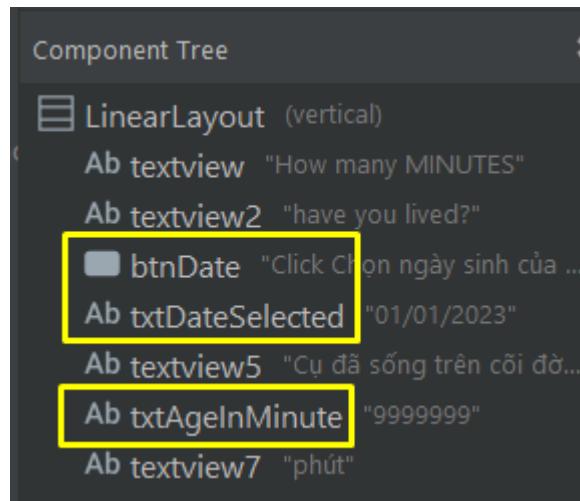
### App 01- Tính tuổi thọ theo phút



1

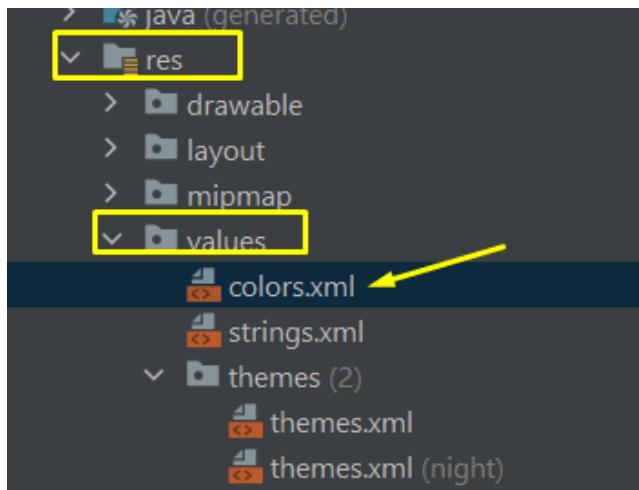
## App01-tính tuổi theo phút

## □ 1 . Thiết kế giao diện



## 1

## App01-tính tuổi theo phút

 2 . Quy định màu thường dùng trong colors.xml

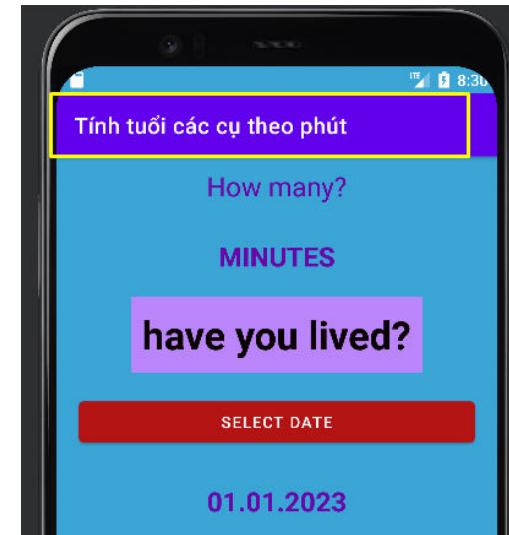
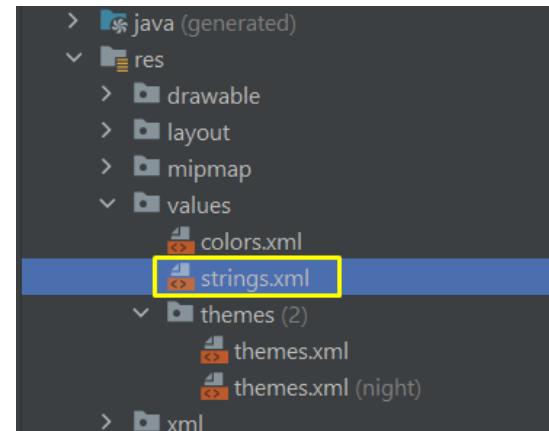
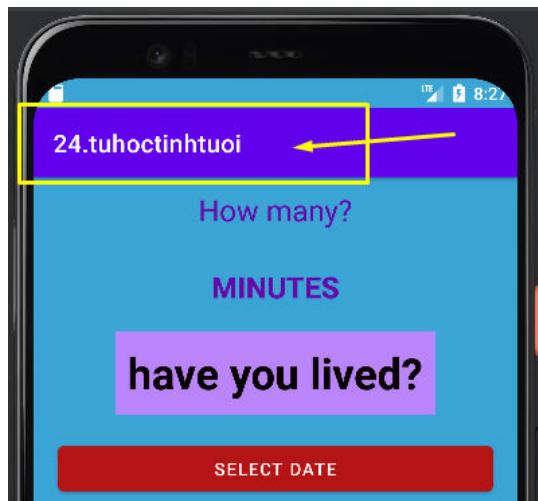
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFF</color>
    <color name="maucuatoi">#52CAA2</color>
</resources>
```



1

## App01-tính tuổi theo phút

## □ 3 . Thay đổi text tiêu đề ứng dụng



```
<resources>
    <string name="app_name">Tính tuổi các cụ theo phút</string>
</resources>
```



## 1

## App01-tính tuổi theo phút

 4. Code MainActivity

- ✓ *Lắng nghe sự kiện click button btnDate*

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        btnDate.setOnClickListener { it: View!  
            selectDate()  
        }  
    }  
}
```



## 1

## App01-tính tuổi theo phút

## □ 3 . Code MainActivity

✓ *hàm selectDate()*

```
private fun selectDate() {
    val cal = Calendar.getInstance()
    //get toàn bộ thông tin date time hiện tại
    //check log để kiểm tra
    Log.wtf("tuhoc",cal.toString())
    val year = cal.get(Calendar.YEAR)
    val month = cal.get(Calendar.MONTH)
    val day = cal.get(Calendar.DAY_OF_MONTH)

    DatePickerDialog(context: this,
        DatePickerDialog.OnDateSetListener{
            view,sYear,sMonth, sDayOfMonth->
            Toast.makeText(context: this, text: "cụ sinh ngày $sDayOfMonth/${sMonth+1}/$sYear",
                Toast.LENGTH_SHORT).show()

            val dateBirth = "$sDayOfMonth/${sMonth+1}/$sYear"
            txtDateSelected.text= dateBirth

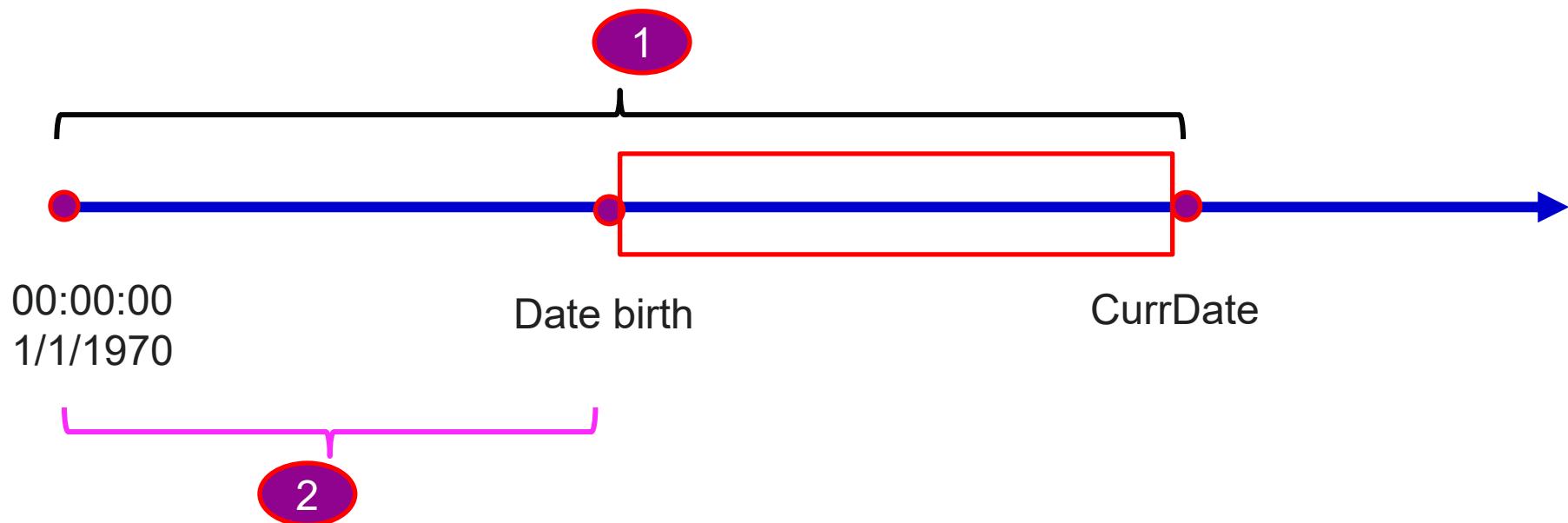
            //ép string sang định dạng ngày tháng đã học ở bài 17 phần cơ bản
            //17. [ Lập trình Android ] thư viện date time kotlin - lập trình android căn bản - tuhoc.cc
            //https://developer.android.com/reference/java/text/SimpleDateFormat
            val sdf = SimpleDateFormat(pattern: "dd/MM/yyyy", Locale.US)
            val ngaySinh = sdf.parse(dateBirth)
            val ngaySinhTheoPhut = ngaySinh.time/60000 //1 phút =1000ms

            //ngày hiện tại
            val currDate = sdf.parse(sdf.format(System.currentTimeMillis()))
            val currDateInMinute = currDate.time/60000

            //tính số phút = ngày hiện tại - ngày sinh
            val diff = currDateInMinute-ngaySinhTheoPhut
            txtAgeInMinute.text = diff.toString()
        },
        year,
        month,
        day).show()
}
```



❑ Bonus . Cách Tính tuổi theo gốc thời gian 0h00 1-1-1970



## 1

## App01-tính tuổi theo phút

 3. Code MainActivity

- ✓ *Tối ưu hoá code và set maxdate*

```
private fun selectDate() {
    val cal = Calendar.getInstance()
    //get toàn bộ thông tin date time hiện tại
    //check log để kiểm tra
    Log.wtf(tag: "tuhoc", cal.toString())
    val year = cal.get(Calendar.YEAR)
    val month = cal.get(Calendar.MONTH)
    val day = cal.get(Calendar.DAY_OF_MONTH)

    val dp = DatePickerDialog(context: this,
        DatePickerDialog.OnDateSetListener{...},
        year,
        month,
        day)
    dp.datePicker.maxDate = System.currentTimeMillis()-86400000
    dp.show()
}
```



## 1

## App01-tính tuổi theo phút

## ❑ 3. Code MainActivity

## ✓ Tối ưu hóa code

```
//...  
val sdf = SimpleDateFormat(pattern: "dd/MM/yyyy", Locale.US)  
val ngaySinh = sdf.parse(dateBirth)  
val ngaySinhTheoPhut = ngaySinh.time/60000 //1 phút =1000ms  
  
//ngày hiện tại  
val currDate = sdf.parse(sdf.format(System.currentTimeMillis()))  
val currDateinMinute = currDate.time  
  
//tính số phút = ngày hiện tại - ngày sinh  
val diff = currDateinMinute-ngaySinhTheoPhut  
txtAgeInMinute.text = diff.toString()
```

Unsafe use of a nullable receiver of type Date?

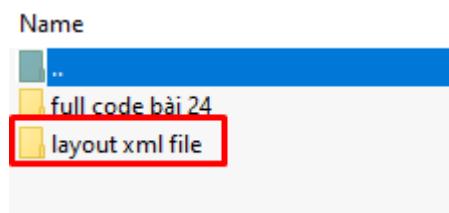
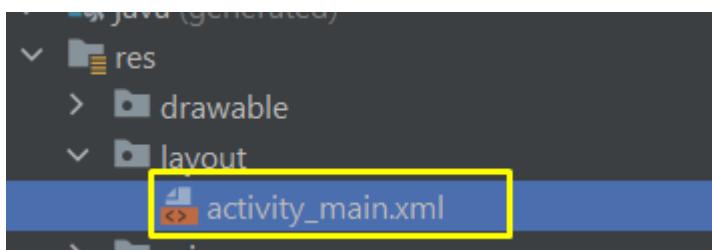
⋮

```
val sdf = SimpleDateFormat(pattern: "dd/MM/yyyy", Locale.US)  
val ngaySinh = sdf.parse(dateBirth)  
ngaySinh?.let { it: Date  
    val ngaySinhTheoPhut = ngaySinh.time/60000 //1 phút =1000ms  
  
    //ngày hiện tại  
    val currDate = sdf.parse(sdf.format(System.currentTimeMillis()))  
    currDate?.let { it: Date  
        val currDateinMinute = currDate.time/60000  
  
        //tính số phút = ngày hiện tại - ngày sinh  
        val diff = currDateinMinute-ngaySinhTheoPhut  
        txtAgeInMinute.text = diff.toString()  
    }  
}
```



1

## App01-tính tuổi theo phút

 4 . File xml layout và full code

File xml layout  
và code toàn bộ bài 24  
**Mật khẩu giải nén: app1**

<http://android.tuhoc.cc/>

 [24.bai24-app01-tinhluoi.zip](#) 





# BroadcastReceiver Kotlin

The screenshot shows a YouTube channel page for "Gà Lại Lập Trình". The channel banner at the top features a purple and blue abstract design with the text "GALAILAPTRINH" and "HTTP://TUHOC.CC". Below the banner is a circular profile picture of a smartphone. The channel name "Gà Lại Lập Trình" is displayed in bold black text, along with the subscriber count "6,65 N người đăng ký". A yellow emoji of a smiling face with a wide-open mouth is shown next to the channel name. A checkmark icon with the text "Nếu thấy video hay và bổ ích hãy like và chia sẻ để mình biết" is present. Below the channel information are navigation tabs: TRANG CHỦ, VIDEO, SHORTS, and DANH SÁCH PH. A message box in the center says "Không có kết nối Internet" with a download icon, followed by the text "Hãy xem các video đã tải xuống hoặc tìm video mới để tải xuống Thư viện của bạn." At the bottom, there are buttons for "Không, cảm ơn" and "Đi đến Video đã tải xuống". A cartoon Android character wearing a green helmet and black suit is visible at the bottom right. The overall theme is related to learning Java and Android development.

## □ 1 . Khái quát *BroadcastReceiver* :

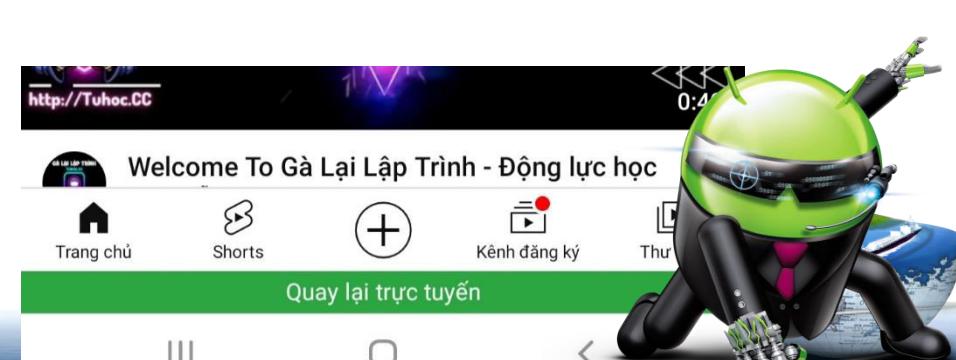
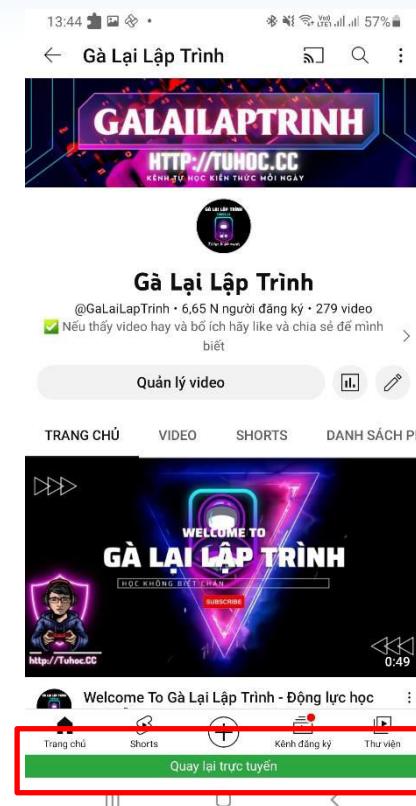
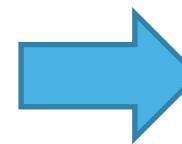
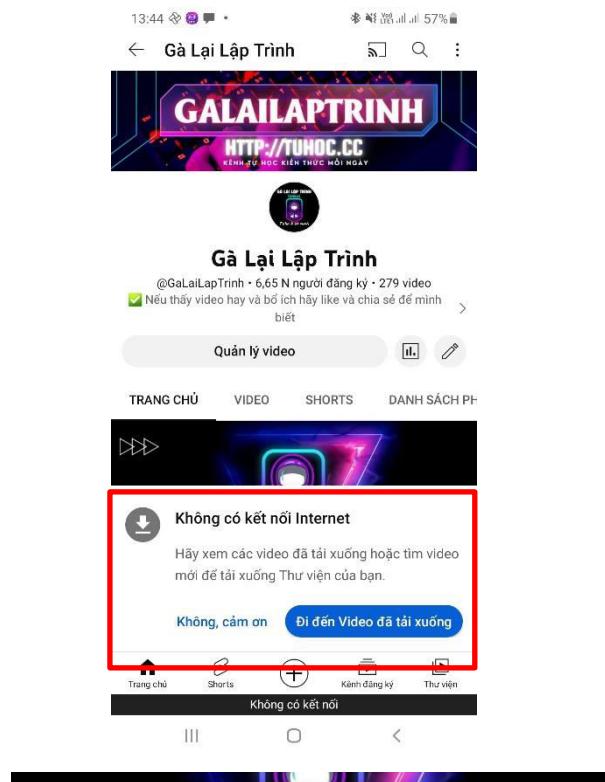
- ✓ *phụ trách việc gửi thông tin liên quan đến thiết bị (mức pin, chế độ máy bay, wifi, đang sạc, nhận tin nhắn....)*
- ✓ *1 ứng dụng A cũng có thể sử dụng BroadcastReceiver để gửi thông tin đến cho các ứng dụng khác (ví dụ: một số dữ liệu mới đã được tải xuống)*
- ✓ *Tài liệu: <https://developer.android.com/guide/components/broadcasts>*
- ✓ *Broadcast sử dụng intent để đóng gói và gửi thông tin*

## □ 2 . Ví dụ ứng dụng *BroadcastReceiver* :

- ✓ *VD1 : bạn phát triển 1 ứng dụng, và ứng dụng này giúp lưu ảnh của người dùng lên máy chủ, giúp cho người dùng có thể truy cập ảnh từ mọi nơi*
- ✓ *Nếu bạn gửi mọi ảnh đến máy chủ khi ảnh được chụp, nó sẽ nhanh chóng hút hết dung lượng 3G/ 4G của người dùng => Không ai muốn sử dụng ứng dụng kiểu thế này*
- ✓ *Cần sử dụng Broadcast, hệ điều hành sẽ gửi 1 tin nhắn đến ứng dụng thông báo khi wifi được bật, và lúc này ứng dụng mới gửi ảnh đến máy chủ*



## ❑ 2 . Ví dụ ứng dụng BroadcastReceiver :



## ❑ 2 . Ví dụ ứng dụng *BroadcastReceiver* :

**HƯỚNG DẪN TẢI & CÀI ĐẶT ỨNG DỤNG VÍ ĐIỆN TỬ MOMO**

Tải ứng dụng ví điện tử MoMo Chuyển nhận tiền với từ khóa "MoMo" trên 2 hệ điều hành: Android và iOS.

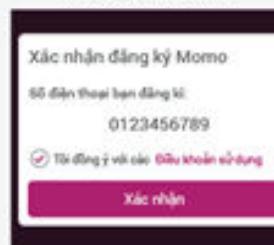
**1**

Mở Ứng dụng MoMo và nhập số điện thoại



**2**

Xác nhận số điện thoại đăng ký ví MoMo



**3**

Nhập mã xác thực trên Ứng dụng MoMo



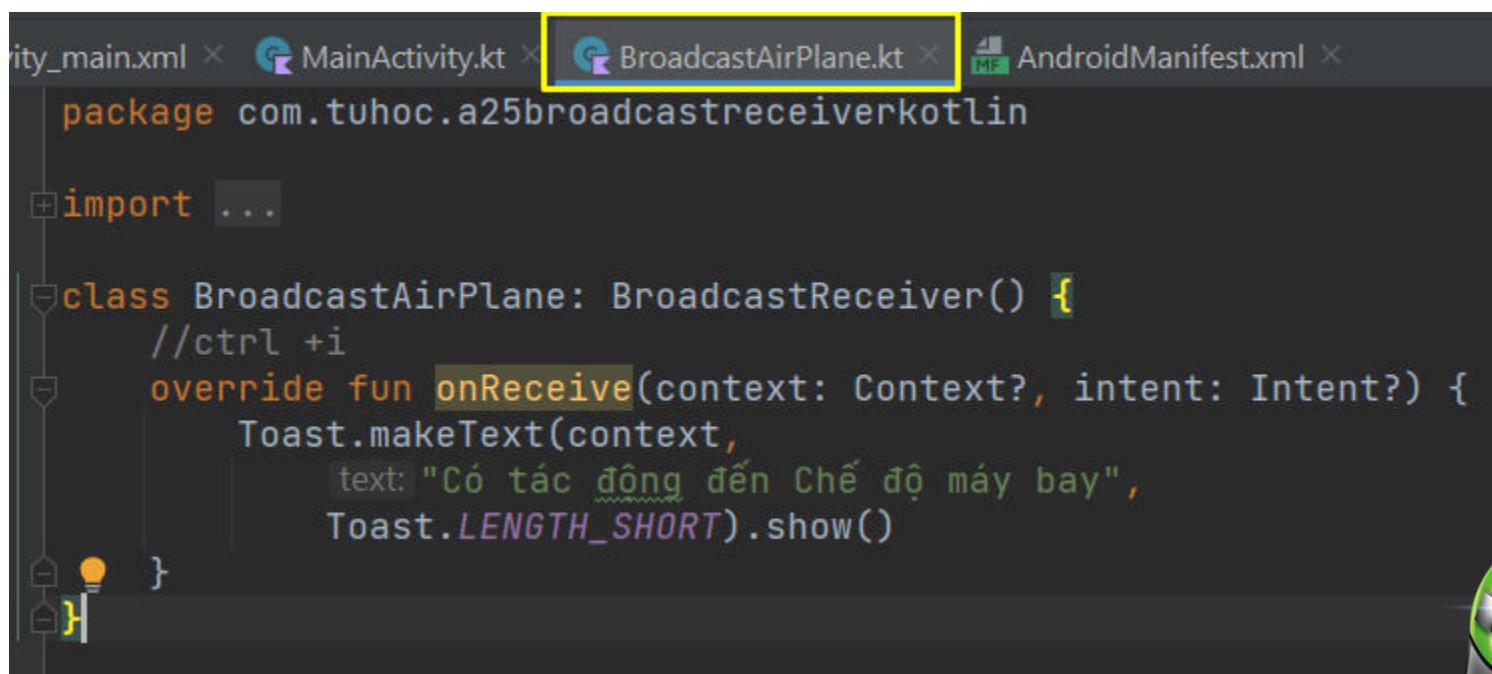
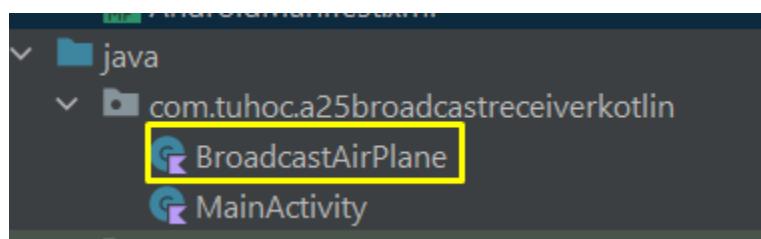
**4**

Tạo mật khẩu đăng nhập Ứng dụng MoMo





## ❑ 3 . Tạo Class *BroadcastReceiver* :



```
activity_main.xml × MainActivity.kt × BroadcastAirPlane.kt × AndroidManifest.xml ×
package com.tuhoc.a25broadcastreceiverkotlin

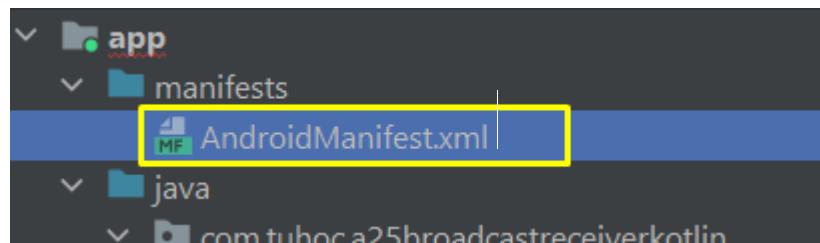
import ...

class BroadcastAirPlane: BroadcastReceiver() {
    //ctrl +i
    override fun onReceive(context: Context?, intent: Intent?) {
        Toast.makeText(context,
            text: "Có tác động đến Chế độ máy bay",
            Toast.LENGTH_SHORT).show()
    }
}
```

The screenshot shows the Android Studio code editor with the 'BroadcastAirPlane.kt' file open. The code defines a class 'BroadcastAirPlane' that extends 'BroadcastReceiver'. It overrides the 'onReceive' method to display a toast message indicating a change in flight mode.



## ❑ 4 . Cách 1: Đăng ký *BroadcastReceiver* vào *Manifest* :



```
11/20 15:32:28: Launching 'app' on Pixel 4 API 30.  
Installation did not succeed.  
The application could not be installed.
```

```
List of apks:  
[0] 'E:\TuhocAndroid\25.BroadcastReceiverKotlin\app\build\intermediates\apk\debug\app-debug.apk'  
Installation failed due to: 'device offline'  
Retry  
Failed to launch an application on all devices
```

Cách này Không hoạt động trên API 26 trở lên

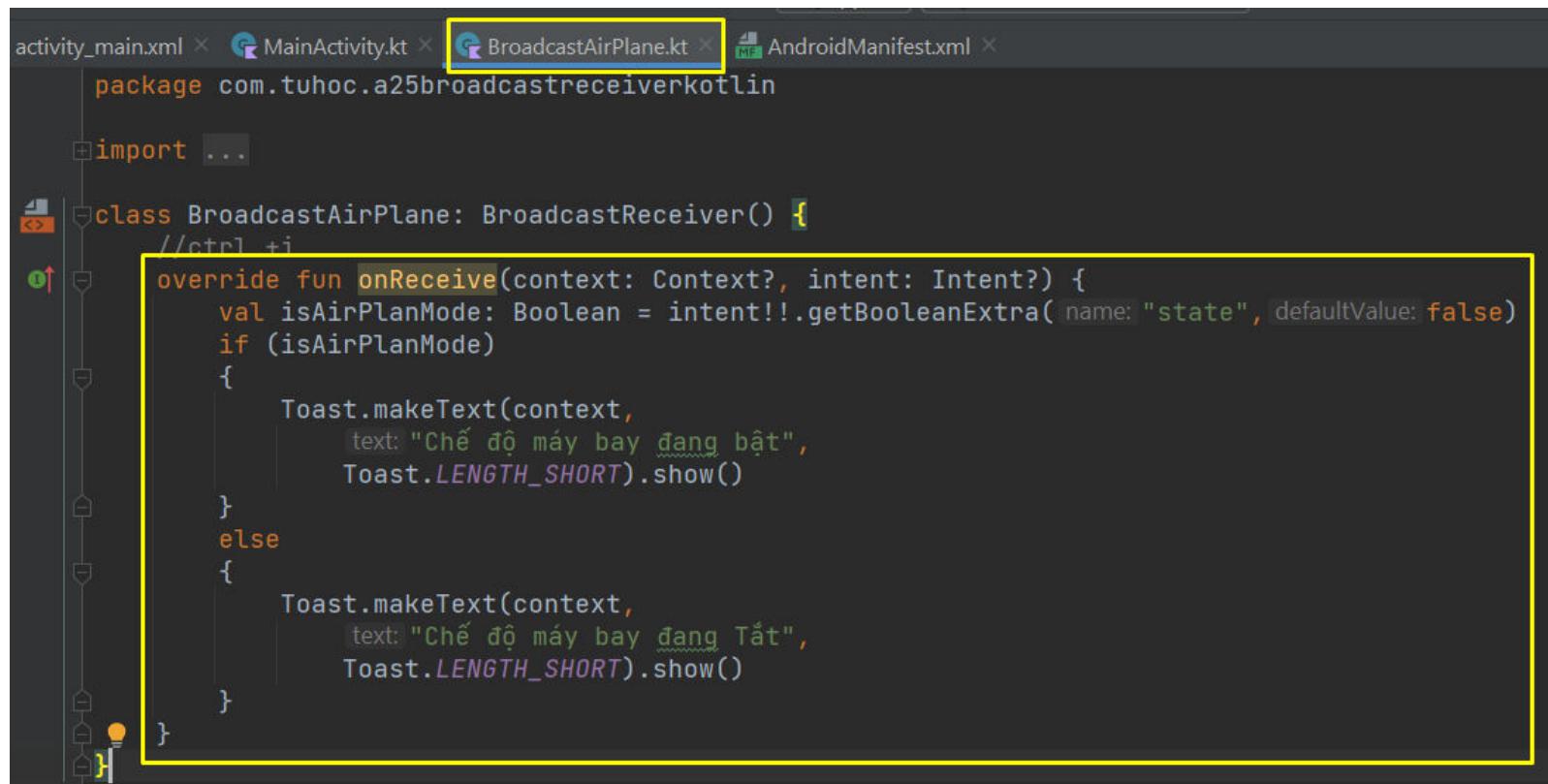
```
tools.targetApi= 31 >  
<activity  
    android:name=".MainActivity"  
    android:exported="true">  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
  
        <category android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
</activity>  
<receiver android:name=".BroadcastAirPlane"  
    android:exported="true">  
    <intent-filter>  
        <action android:name="android.intent.action.AIRPLANE_MODE"/>  
    </intent-filter>  
</receiver>
```

*true* : *Broadcast* có thể nhận các sự kiện được gửi bởi cùng một ứng dụng hoặc ứng dụng khác

*false*: *Broadcast* có thể nhận các sự kiện được gửi bởi cùng một ứng dụng



## ❑ 5 . Check airplan on off ( Sử dụng trên Class *BroadcastReceiver* ):



The screenshot shows the Android Studio interface with several files open in tabs: activity\_main.xml, MainActivity.kt, BroadcastAirPlane.kt (which is currently selected and highlighted with a yellow border), and AndroidManifest.xml. The BroadcastAirPlane.kt file contains the following Kotlin code:

```
activity_main.xml x MainActivity.kt x BroadcastAirPlane.kt x AndroidManifest.xml x
package com.tuhoc.a25broadcastreceiverkotlin

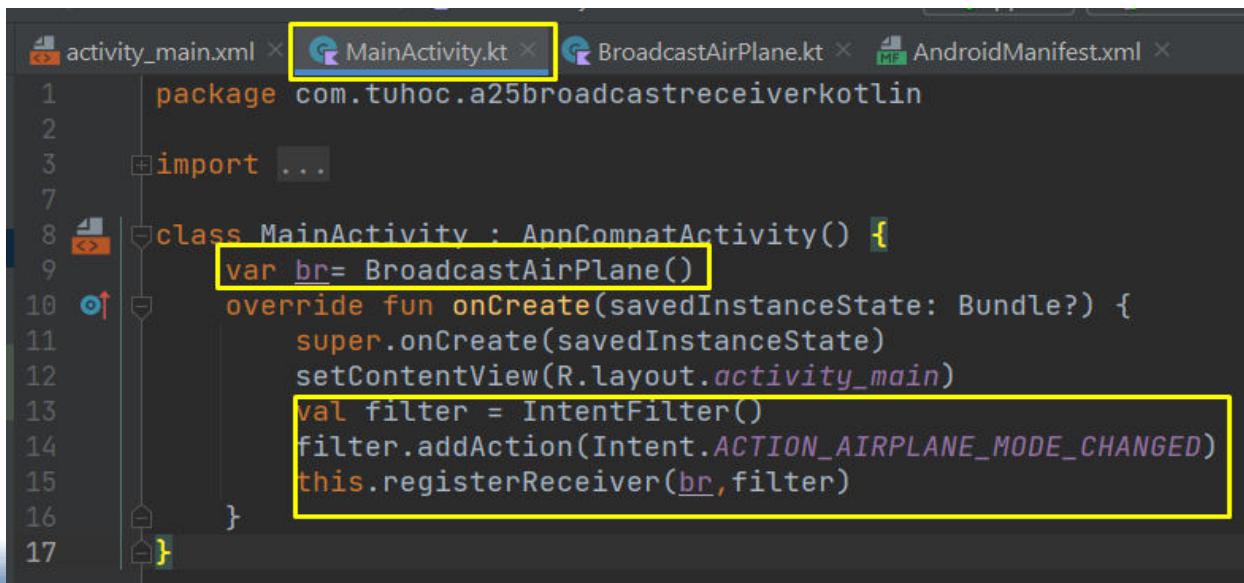
import ...

class BroadcastAirPlane: BroadcastReceiver() {
    //ctrl + i
    override fun onReceive(context: Context?, intent: Intent?) {
        val isAirPlanMode: Boolean = intent!!.getBooleanExtra( name: "state", defaultValue: false)
        if (isAirPlanMode)
        {
            Toast.makeText(context,
                text: "Chế độ máy bay đang bật",
                Toast.LENGTH_SHORT).show()
        }
        else
        {
            Toast.makeText(context,
                text: "Chế độ máy bay đang Tắt",
                Toast.LENGTH_SHORT).show()
        }
    }
}
```



## ❑ 6 . Cách 2: Đăng ký tại>MainActivity :

```
tools:targetApi="31">
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<receiver android:name=".BroadcastAirPlane"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.AIRPLANE_MODE" />
    </intent-filter>
</receiver>
```



```
activity_main.xml x MainActivity.kt x BroadcastAirPlane.kt x AndroidManifest.xml x
1 package com.tuhoc.a25broadcastreceiverkotlin
2
3 import ...
4
5 class MainActivity : AppCompatActivity() {
6     var br= BroadcastAirPlane()
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         setContentView(R.layout.activity_main)
10        val filter = IntentFilter()
11        filter.addAction(Intent.ACTION_AIRPLANE_MODE_CHANGED)
12        this.registerReceiver(br,filter)
13    }
14
15 }
```





## SharedPreferences Kotlin

# Shared Preferences

# Android



## □ 1 . Khái quát *SharedPreferences* :

1. Giả sử bạn đang chơi game, có cuộc gọi đến => ưu tiên số 1 của mobile vẫn là call

=> ứng dụng trò chơi của bạn sẽ chuyển sang phương thức tạm dừng (*onPause*) hoặc dừng *onStop*

Xem lại bài 15 activity lifecycle

=> Sử dụng *sharedPreferences* để lưu trữ dữ liệu => Khi bạn quay lại trò chơi một lần nữa, bạn sẽ có thể tiếp tục nơi bạn đã dừng lại Và cũng với số điểm cuối cùng của bạn.

2. Bạn cũng có thể sử dụng lớp này để lưu lại dữ liệu, chẳng hạn như điểm cuối cùng, điểm cao nhất....



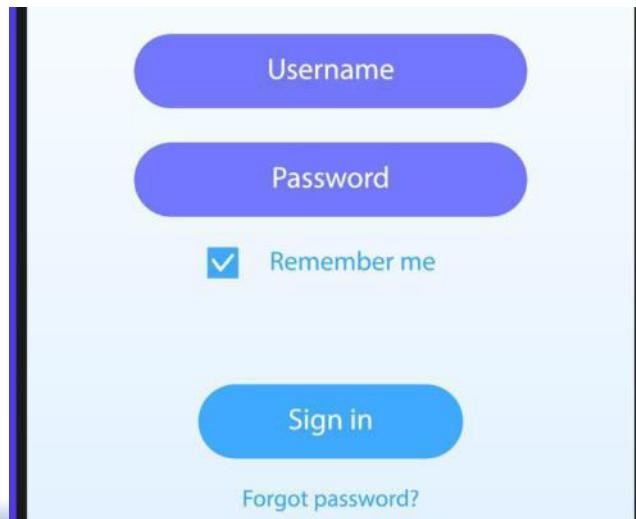
## □ 1 . Khái quát Shared Preferences :

3. *Bạn có thể sử dụng lớp này để lưu log đăng nhập, nhật ký người dùng.*

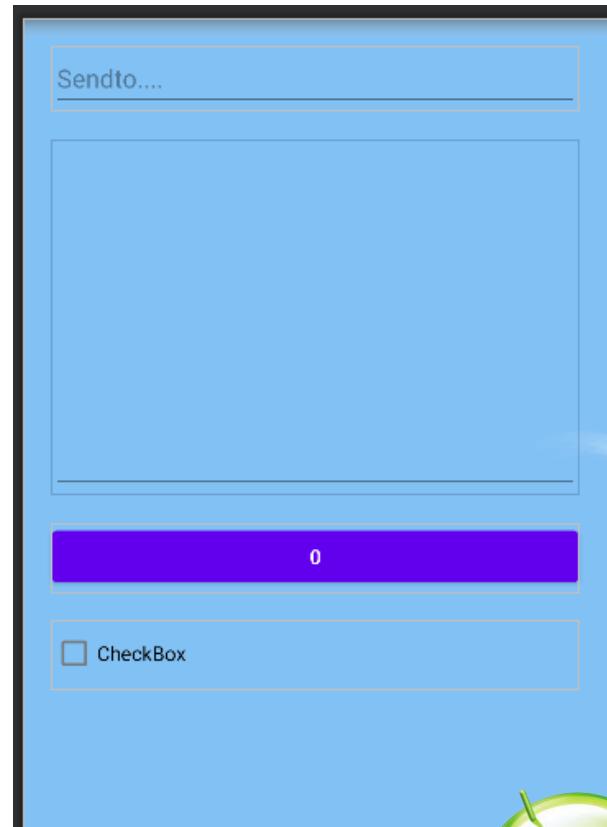
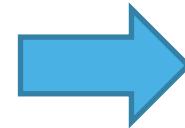
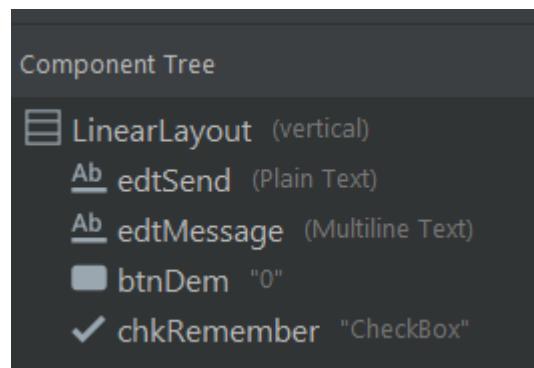
*Ví dụ: hầu hết các ứng dụng đều có trang đăng nhập. Trên trang này, người dùng phải nhập tên người dùng và mật khẩu.*

*Nếu người dùng không muốn nhập thông tin này mỗi lần, lập trình viên có thể code thêm phần tích chọn Ghi nhớ và sau khi chọn hộp kiểm, ứng dụng sẽ lưu dữ liệu của người dùng.*

*Và khi người dùng quay lại ứng dụng, người dùng sẽ không phải mất thời gian nhập lại username và password.*

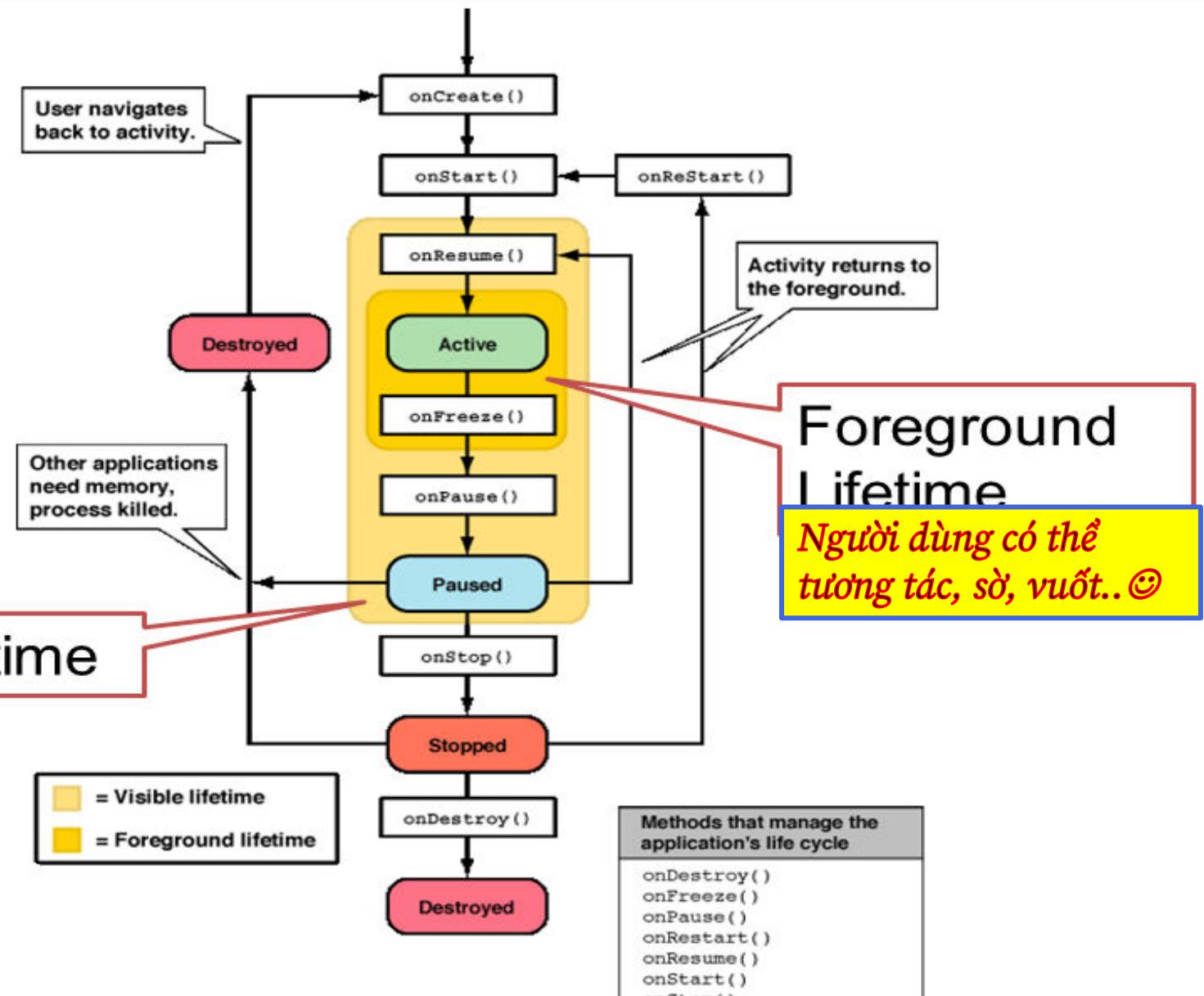


## ❑ 2. Thiết kế ứng dụng demo :



## ❑ 3 . Lưu trạng thái khi onPause:

Xem lại bài 15 activity lifecycle



## ❑ 3 . Lưu trạng thái khi onPause:

Xem lại bài 15 activity lifecycle

```
class MainActivity : AppCompatActivity() {  
    //khai báo biến đếm:  
    var dem=0  
    var send:String? = null  
    var message: String? = null  
    var isChecked: Boolean? = null  
    lateinit var sharedpreferences: SharedPreferences  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        btnDem.setOnClickListener { it: View!  
            dem++  
            btnDem.setText(""+dem)  
            //tương đương với cách viết  
            //btnDem.setText(dem.toString())  
        }  
    }  
    override fun onPause() {  
        super.onPause()  
        saveData()  
    }  
}
```



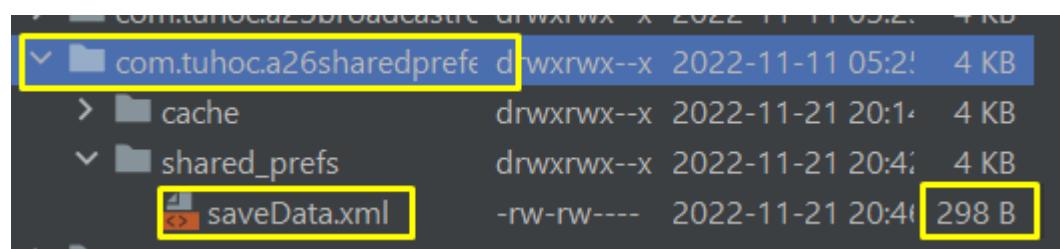
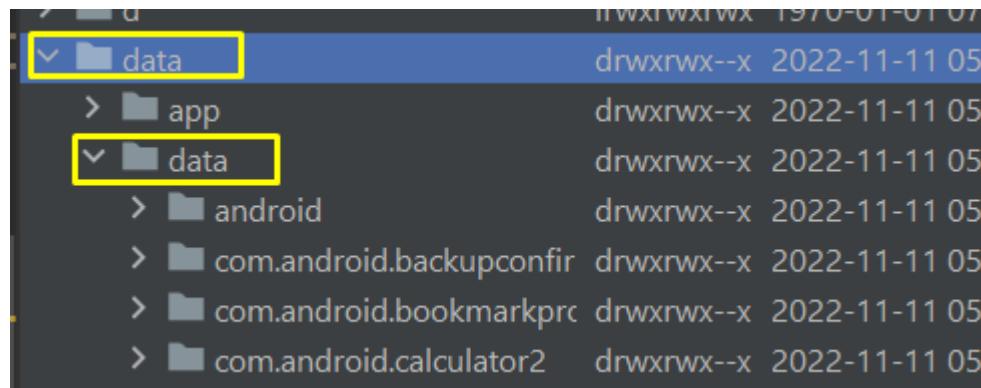
## □ 3 . Lưu trạng thái khi onPause (*code hàm saveData*) :

```
fun saveData(){
    sharedPreferences = this.getSharedPreferences( name: "saveData", Context.MODE_PRIVATE)
    //MODE_PRIVATE sẽ cấp quyền truy cập để lưu dữ liệu từ mọi nơi trong ứng dụng
    //còn bên ngoài ứng dụng sẽ không được cấp quyền
    send = edtSend.text.toString()
    message = edtMessage.text.toString()
    isChecked = chkRemember.isChecked

    //note: key phải là duy nhất
    val editor = sharedPreferences.edit()
    editor.putString("key_send", send)
    editor.putString("key_mes", message)
    editor.putInt("key_dem", dem)
    editor.putBoolean("key_remember", isChecked!!)
    editor.apply()
    Toast.makeText(applicationContext, text: "Data đã được lưu", Toast.LENGTH_SHORT).show()
}
```



## □ 4 . Kiểm tra vị trí lưu trữ của *sharedPreferences* :



## □ 5 . Gọi lại dữ liệu đã lưu trữ trong *sharedPreferences* :

MainActivity.kt ×

```
fun trieuHoiData(){
    sharedpreferences=this.getSharedPreferences( name: "saveData", Context.MODE_PRIVATE)
    send = sharedpreferences.getString( key: "key_send", defValue: null)
    // defValue: giá trị mặc định nếu o tìm dc key
    message=sharedpreferences.getString( key: "key_mes", defValue: null)
    dem=sharedpreferences.getInt( key: "key_dem", defValue: 0)
    isChecked = sharedpreferences.getBoolean( key: "key_remember", defValue: false)

    edtSend.setText(send)
    edtMessage.setText(message)
    btnDem.setText(""+dem)
    chkRemember.isChecked = isChecked!!
}

override fun onResume() {
    super.onResume()
    trieuHoiData()
}
```





## Multiple Language Kotlin



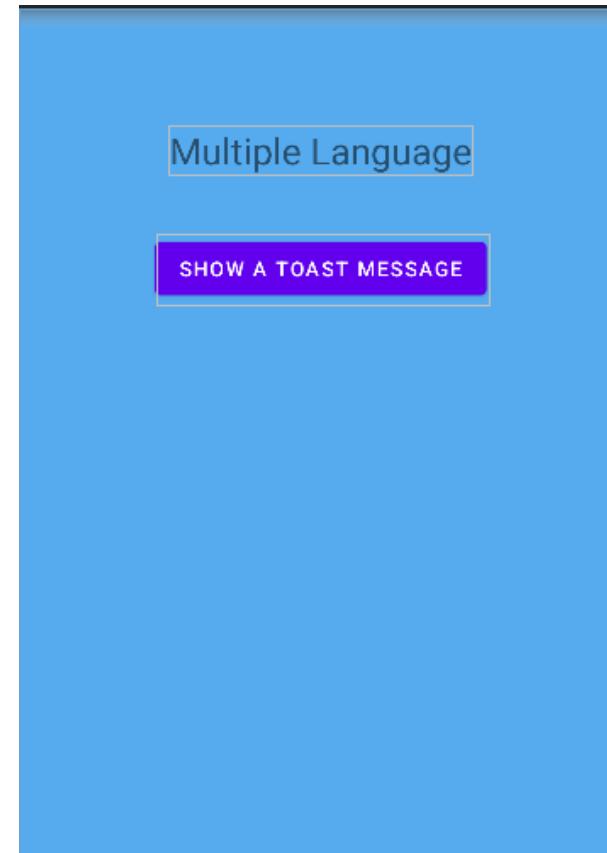
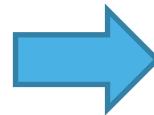
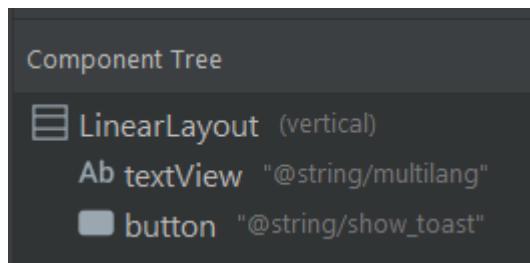
## □ 1 . Khái quát *Multiple Language*

*Android là một trong số những hệ điều hành có số lượng người dùng rất lớn ở nhiều quốc gia và sử dụng nhiều ngôn ngữ khác nhau*

*=> Đa ngôn ngữ sẽ giúp tiếp cận được nhiều tập khách hàng tại các nơi trên thế giới => Tăng thu nhập*

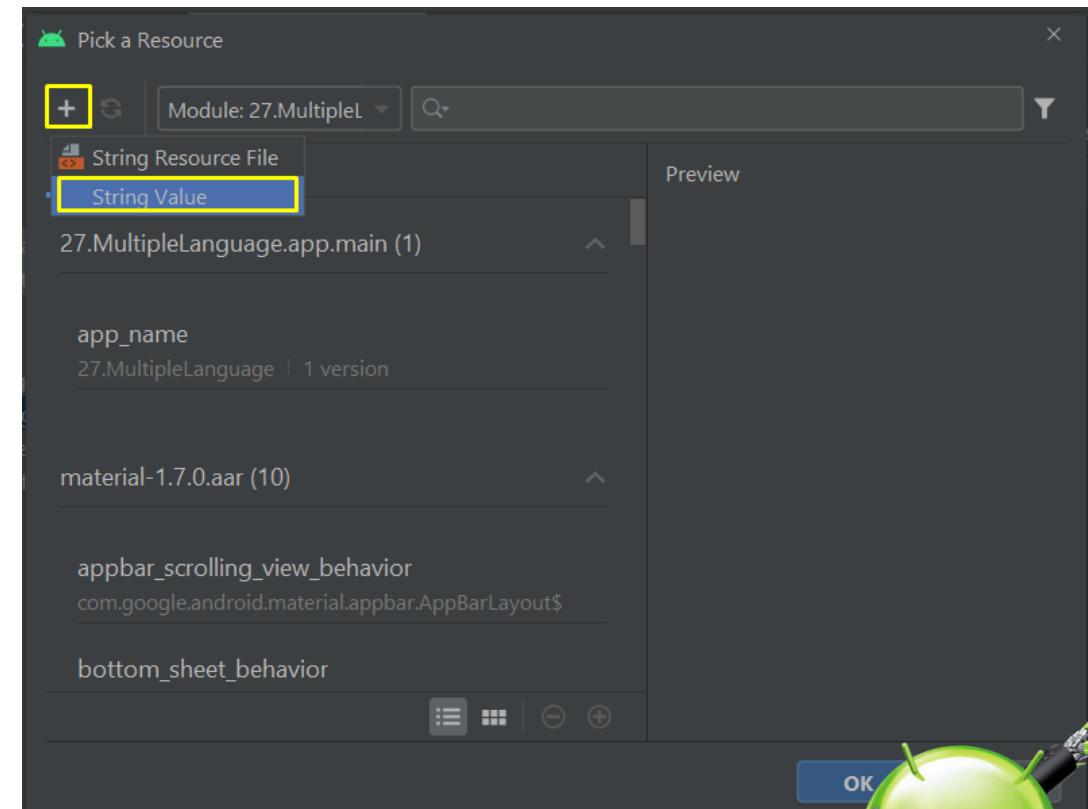
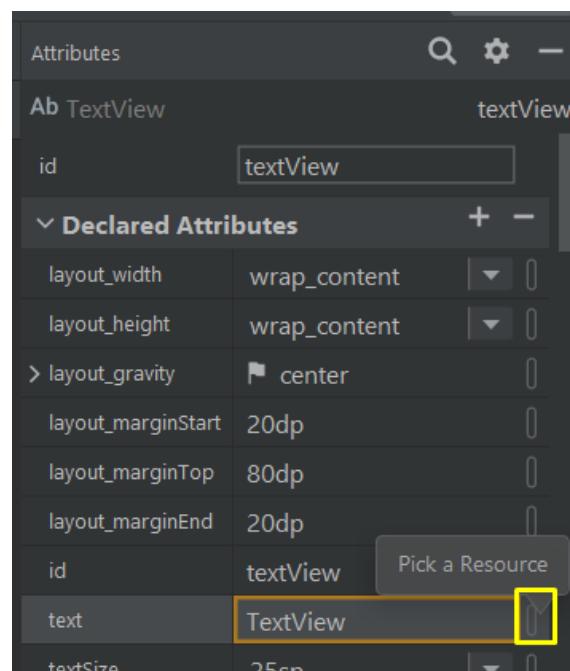


❑ 2. Thiết kế ứng dụng demo :



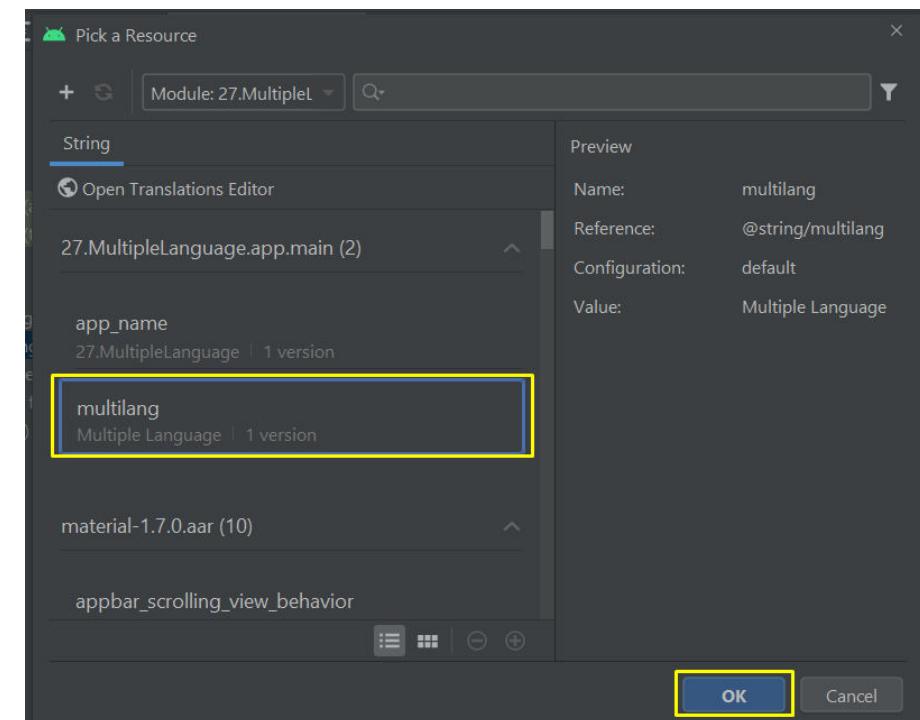
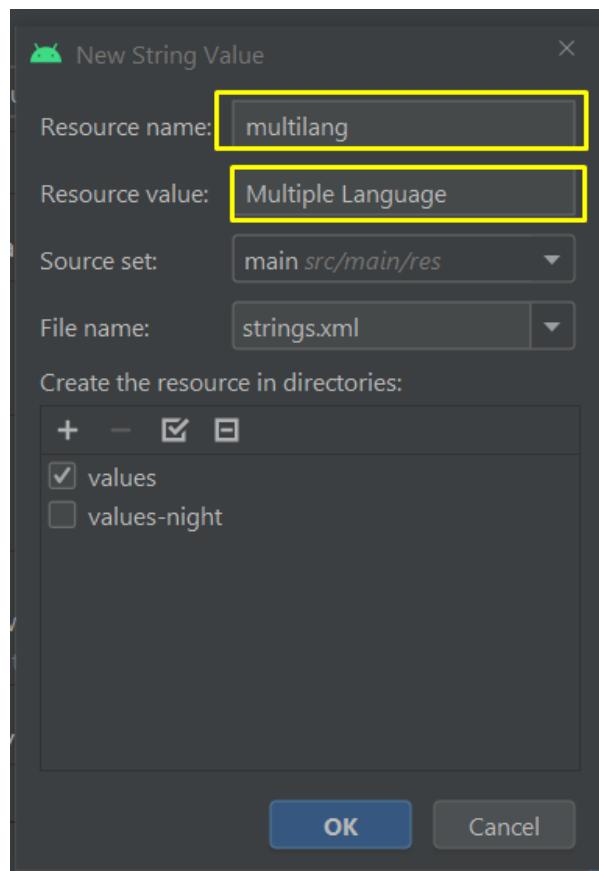
## ❑ 2 . Tạo string value cách 1 :

### a. Tạo cho textView



## ❑ 2 . Tạo string value cách 1 :

### a. Tạo cho textView

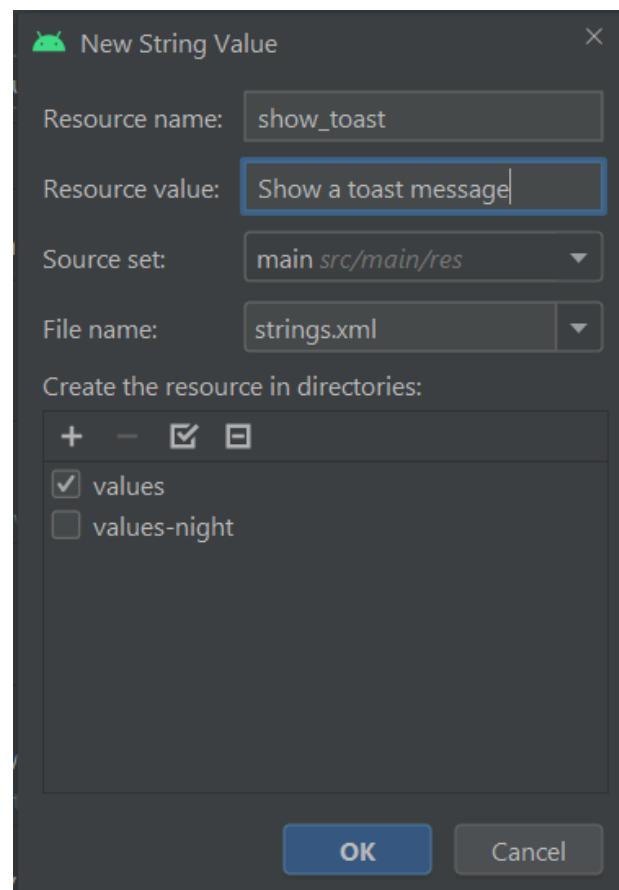


text @string/multilang

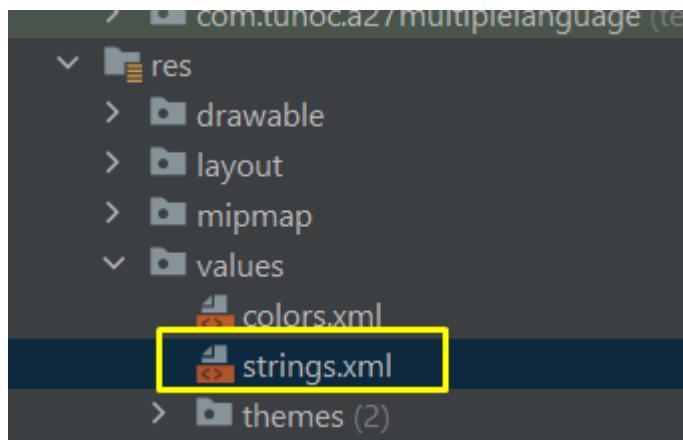


## ❑ 2 . Tạo string value cách 1 :

### b. Tạo cho button



## □ 3 . Tạo string value cách 2 :

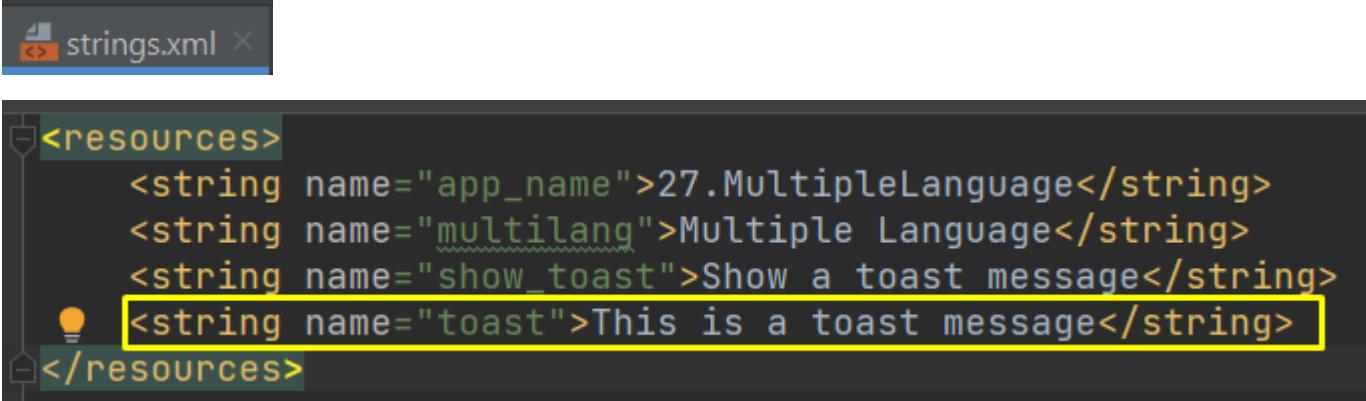


```
<resources>
    <string name="app_name">27.MultipleLanguage</string>
    <string name="multilang">Multiple Language</string>
    <string name="show_toast">Show a toast message</string>
</resources>
```



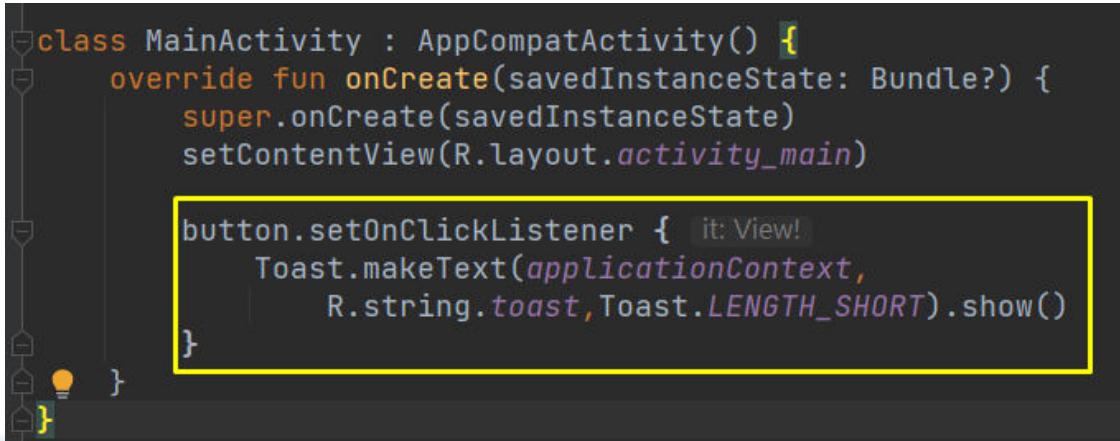
## □ 4 . Code main activity (*code cho button*) :

*Step 1. add string xml*



```
<resources>
    <string name="app_name">27.MultipleLanguage</string>
    <string name="multilang">Multiple Language</string>
    <string name="show_toast">Show a toast message</string>
    <string name="toast">This is a toast message</string>
</resources>
```

*Step 2. Code MainActivity*

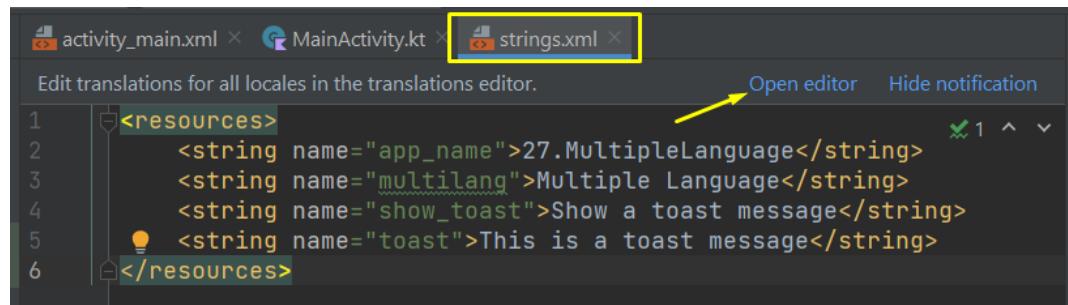


```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        button.setOnClickListener { it: View!
            Toast.makeText(applicationContext,
                R.string.toast,Toast.LENGTH_SHORT).show()
        }
    }
}
```

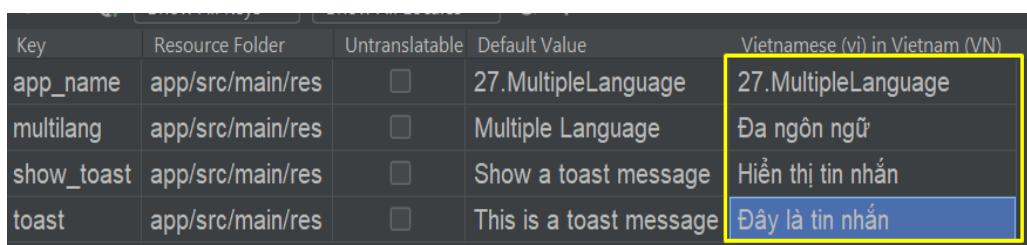
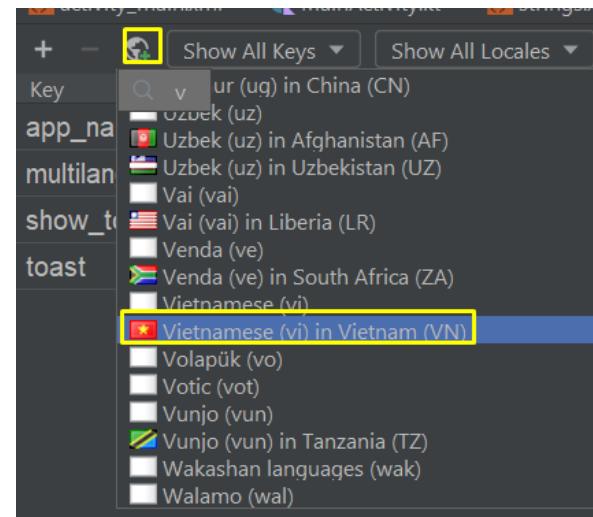


## ❑ 5 . Add new language :



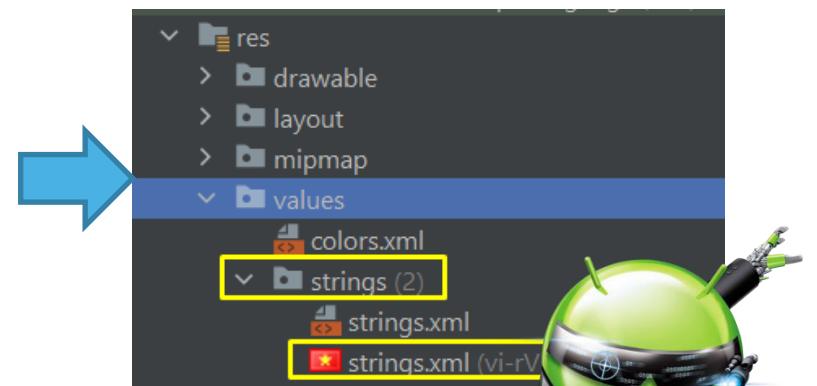
The screenshot shows the Android Studio interface with the strings.xml file open. The code includes string resources for app\_name, multilang, show\_toast, and toast. A yellow box highlights the strings.xml tab in the top navigation bar. An arrow points from the 'Open editor' button to the right.

```
<resources>
    <string name="app_name">27.MultipleLanguage</string>
    <string name="multilang">Multiple Language</string>
    <string name="show_toast">Show a toast message</string>
    <string name="toast">This is a toast message</string>
</resources>
```



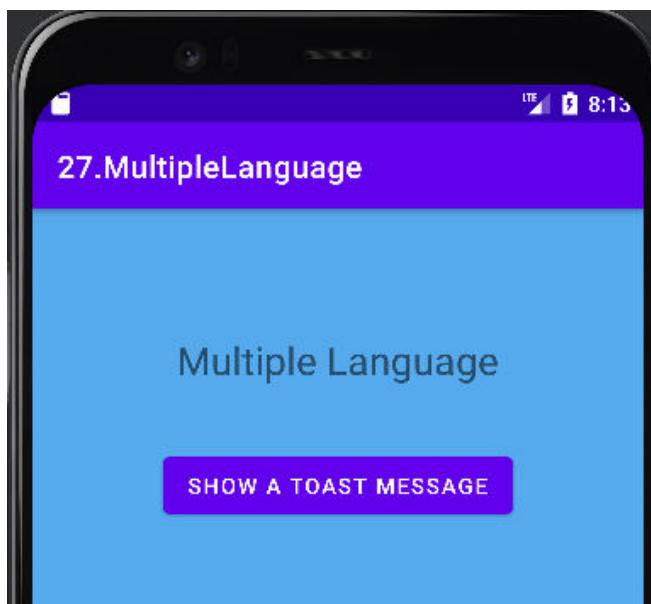
The screenshot shows the Translations Editor with the Vietnamese translation for each string resource. A blue arrow points from the table to the file structure on the right.

Key	Resource Folder	Untranslatable	Default Value	Vietnamese (vi) in Vietnam (VN)
app_name	app/src/main/res	<input type="checkbox"/>	27.MultipleLanguage	27.MultipleLanguage
multilang	app/src/main/res	<input type="checkbox"/>	Multiple Language	Đa ngôn ngữ
show_toast	app/src/main/res	<input type="checkbox"/>	Show a toast message	Hiển thị tin nhắn
toast	app/src/main/res	<input type="checkbox"/>	This is a toast message	Đây là tin nhắn

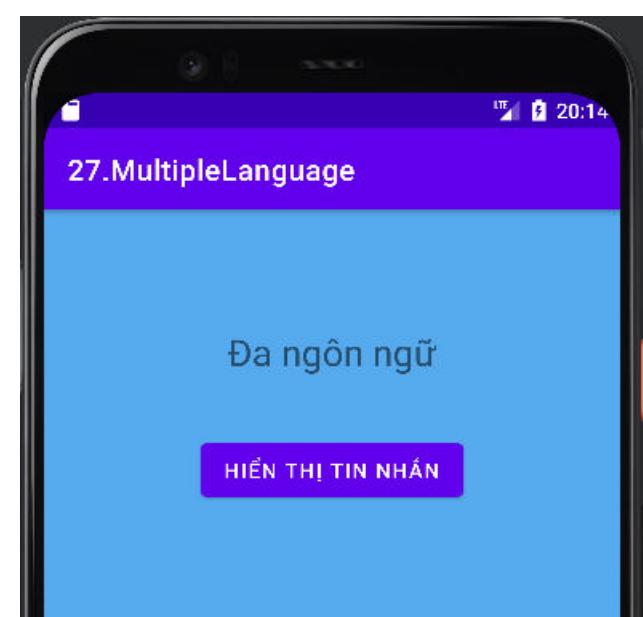


❑ 6 . Kết quả :

Khi máy sử dụng ngôn ngữ English

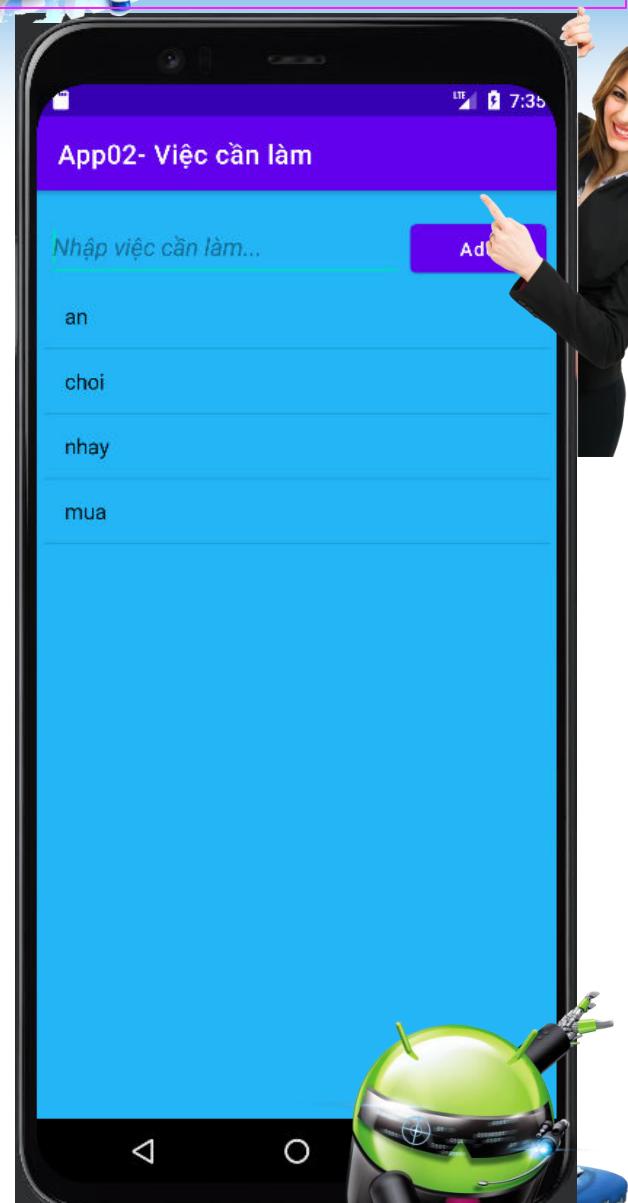


Khi máy sử dụng ngôn ngữ VietNam

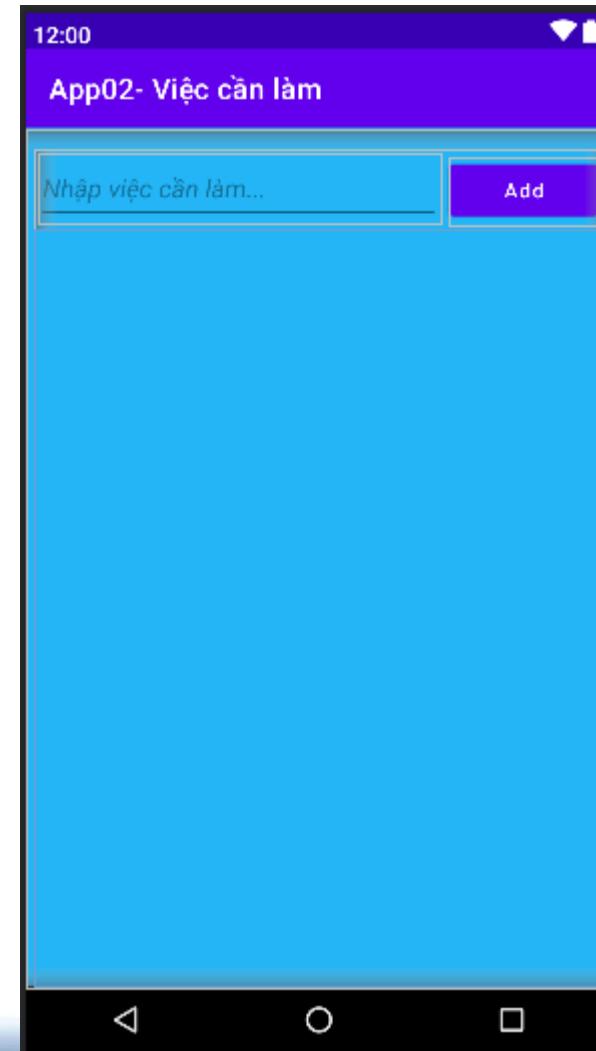
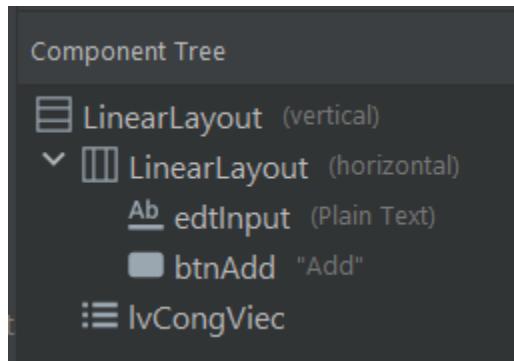




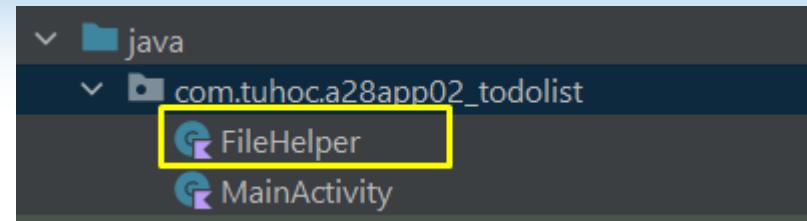
## Kotlin tutorial app 02 To Do List



## □ 1 . Tạo giao diện app



## □ 2 . Tạo class hỗ trợ lưu, đọc file



```
class FileHelper {
    val FILENAME = "listinfo.dat"
    //ghi tệp
    fun writeData(item: ArrayList<String>, context: Context){
        val fos: FileOutputStream = context.openFileOutput(FILENAME, Context.MODE_PRIVATE)
        val oas = ObjectOutputStream(fos)
        oas.writeObject(item)
        oas.close()
    }
    //đọc tệp
    fun readData(context: Context): ArrayList<String>{
        var itemList : ArrayList<String>
        try {
            val fis : FileInputStream = context.openFileInput(FILENAME)
            val ois = ObjectInputStream(fis)
            itemList = ois.readObject() as ArrayList<String>
        }catch (e:FileNotFoundException){
            itemList = ArrayList()
        }
        return itemList
    }
}
```



## □ 3 . Code MainActivity ghi file

```
class MainActivity : AppCompatActivity() {

    var itemList = ArrayList<String>()
    var fileHelper = FileHelper()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        itemList = fileHelper.readData(context: this)
        var arrayAdapter = ArrayAdapter(context: this,
            android.R.layout.simple_list_item_1,
            android.R.id.text1,
            itemList)
        lvCongViec.adapter = arrayAdapter

        btnAdd.setOnClickListener { it: View!
            var itemName = edtInput.text.toString()
            itemList.add(itemName)
            edtInput.setText("")
            fileHelper.writeData(itemList, applicationContext)
            arrayAdapter.notifyDataSetChanged()
        }
    }
}
```



## □ 4 . Code MainActivity xoá item sử dụng alert dialog

```
//xoá item sau khi đã hoàn thành
lvCongViec.setOnItemClickListener { parent, view, position, id ->
    var alert = AlertDialog.Builder(context: this)
    alert.setTitle("Delete")
    alert.setMessage("Bạn có chắc muốn xoá cv này? ")
    alert.setCancelable(true)
    alert.setNegativeButton(text: "No", DialogInterface.OnClickListener { dialog, which ->
        dialog.cancel()
    })
    alert.setPositiveButton(text: "Yes", DialogInterface.OnClickListener { dialog, which ->
        itemList.removeAt(position)
        arrayAdapter.notifyDataSetChanged()
        //ghi danh sách mới vào tệp
        fileHelper.writeData(itemList, applicationContext)
    })
    alert.create()
    alert.show()
}
```

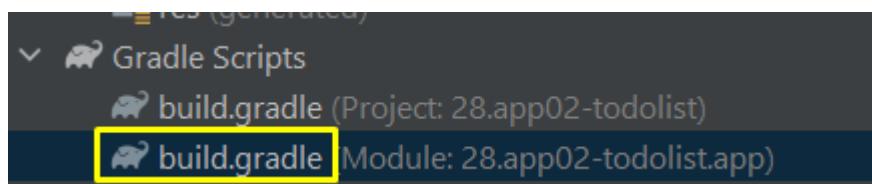




## Build Release APK - AAB file in Android Studio



## □ 1. Quản lý phiên bản ứng dụng



```
defaultConfig {  
    applicationId "com.tuhoc.a28app02_todolist"  
    minSdk 24  
    targetSdk 32  
    versionCode 1  
    versionName "1.0"  
}
```

**1. *versionCode*:** Số nguyên dương từ 1- 2100000000

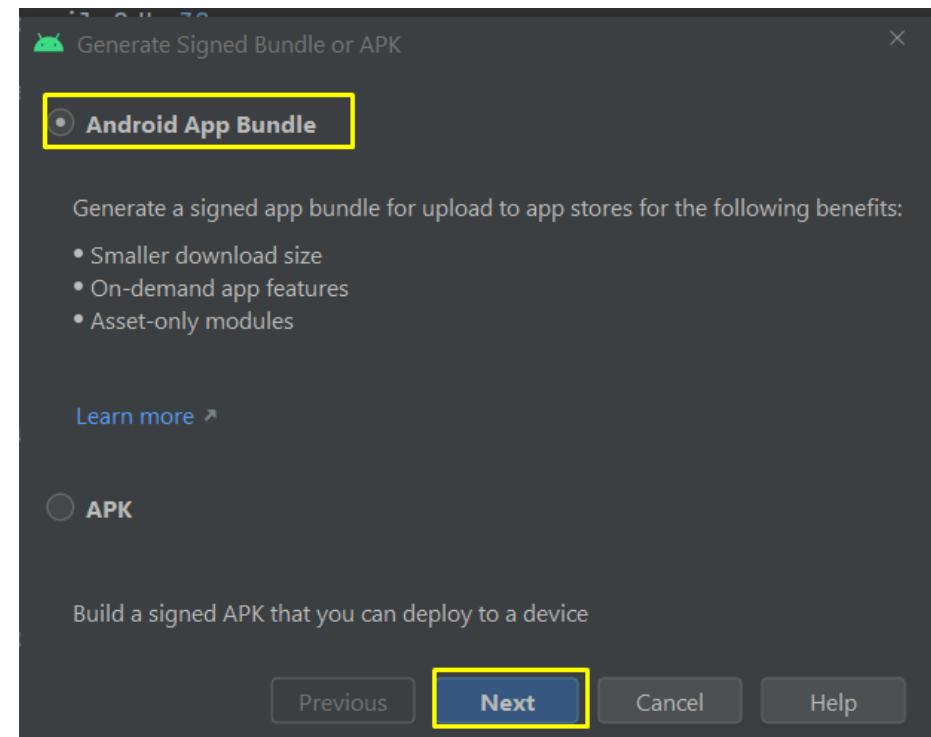
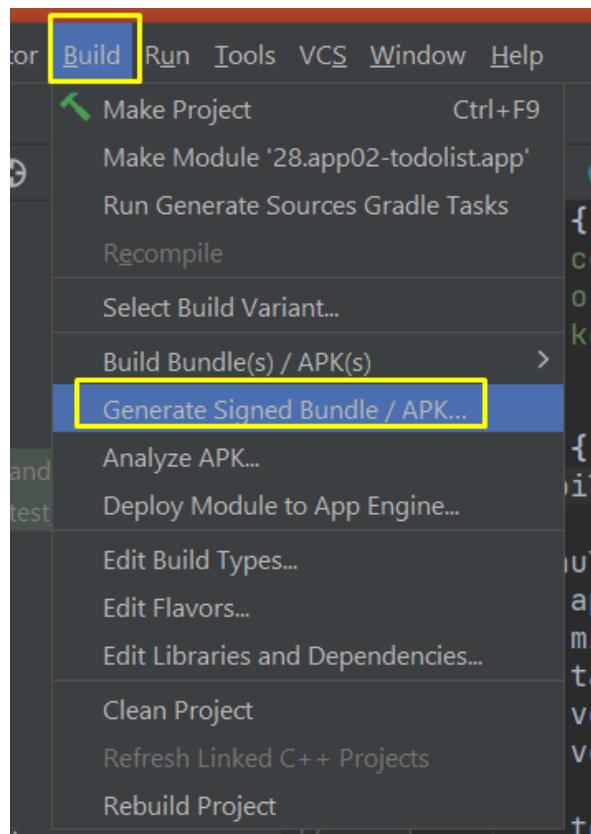
Là số phiên bản, google sẽ kiểm tra mã này để thông báo update nếu thiết bị cuối đang sử dụng ver thấp hơn

Bạn không thể tải APK lên Cửa hàng Play bằng một *versionCode* <= phiên bản hiện tại đang có trên google play

**2. *versionName*:** là 1 chuỗi. Thông tin *version* hiển thị giao tiếp và thông báo cho người dùng, google không quan tâm đến cái này.



## □ 2 . Tạo gói APK



## □ 2 . Tạo gói APK

Generate Signed Bundle or APK

Module: 28.app02-todolist.app

Key store path:  (highlighted)

Key store password:

Key alias:  

Key password:

Remember passwords

Export encrypted key for enrolling published apps in Google Play App Signing

Encrypted key export path: C:/Users/HACOM/Desktop 

Previous Next Cancel Help



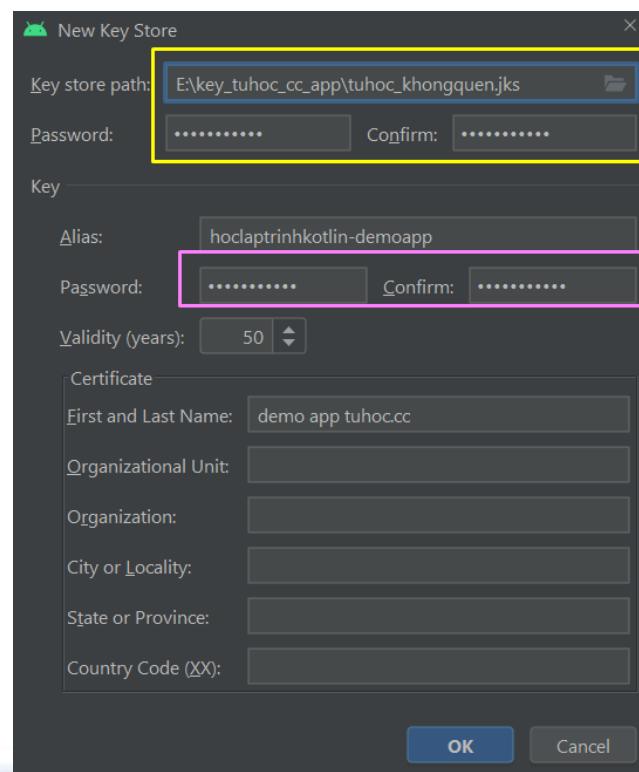
## □ 2 . Tạo gói AP

### Xác định nơi cần lưu AP

Lưu ý file này chỉ cấp 1 lần duy nhất, bạn phải lưu nó để dùng khi update ứng dụng  
Lưu nó tại ít nhất 3 nơi lưu trữ online

Không có nó ứng dụng bạn sẽ không bao giờ được chấp nhận cập nhật phiên bản

Mật khẩu cũng được tạo duy nhất 1 lần, không cấp lại ☺



Phần này có thể dùng chung cho tất cả các ứng dụng

Mật khẩu riêng của ứng dụng, mỗi ứng dụng có 1 mật khẩu riêng, có thể khác nhau giữa các ứng dụng khác nhau



## □ 2 . Tạo gói AP

Generate Signed Bundle or APK

Module: 28.app02-todolist.app

Key store path: E:\key\_tuhoc\_cc\_app\tuhoc\_khongquen.jks

Create new... Choose existing...

Key store password: .....  
Key alias: hoclaptrinhkotlin-demoapp  
Key password: .....

Remember passwords

Export encrypted key for enrolling published apps in Google Play App Signing ↗

Encrypted key export path: C:/Users/HACOM/Desktop

Previous Next Cancel Help



## □ 2 . Tạo gói AP

Generate Signed Bundle or APK

**Android App Bundle**

Generate a signed app bundle for upload to app stores for the following benefits:

- Smaller download size
- On-demand app features
- Asset-only modules

[Learn more ↗](#)

**APK**

Build a signed APK that you can deploy to a device

Previous **Next** Cancel Help



## □ 2 . Tạo gói AP

Generate Signed Bundle or APK

Module

Key store path  [Create new...](#) [Choose existing...](#)

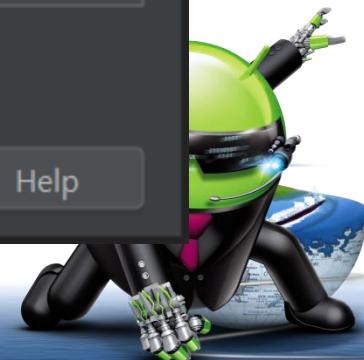
Key store password

Key alias  

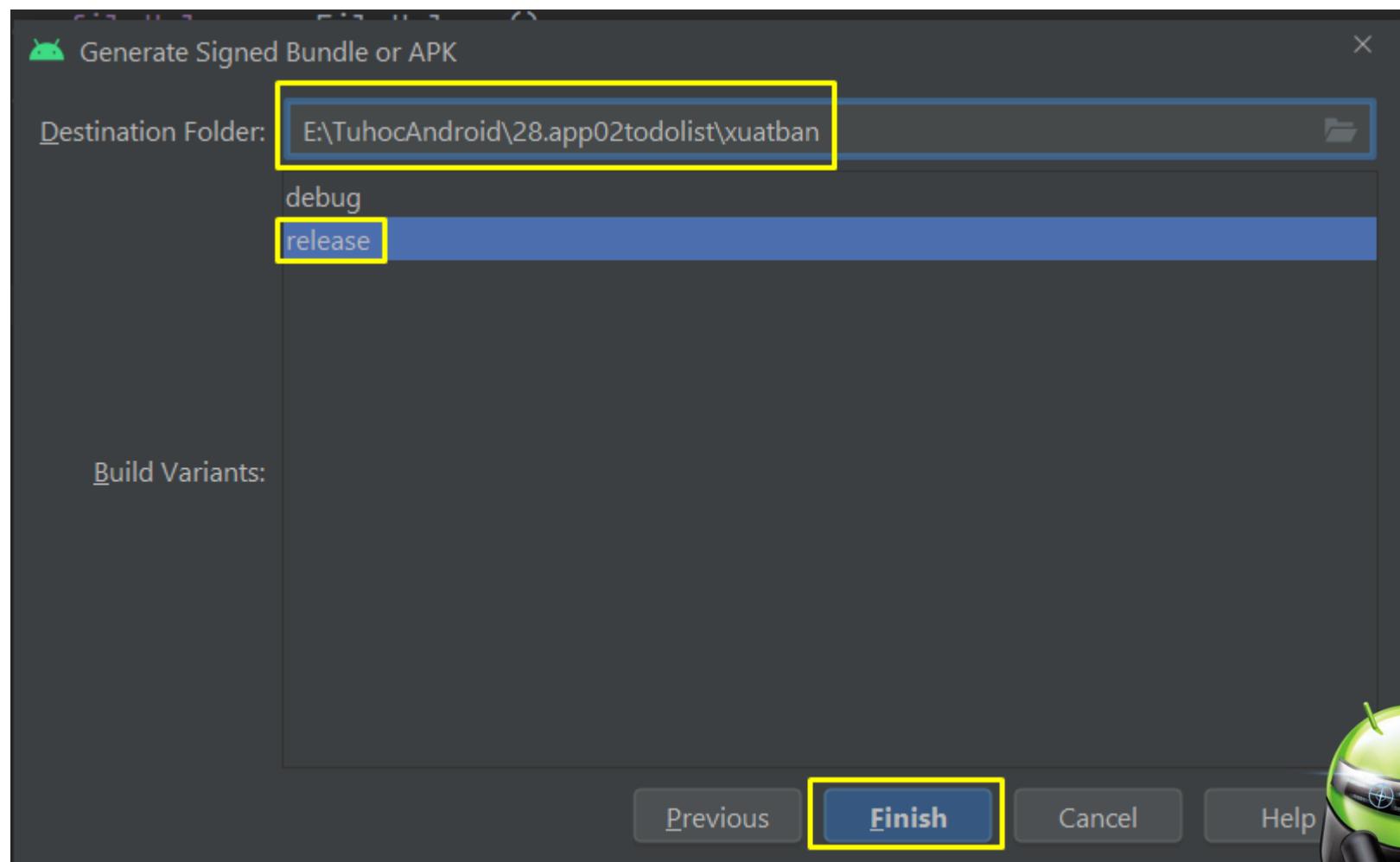
Key password

Remember passwords

[Previous](#) [Next](#) [Cancel](#) [Help](#)



## □ 2 . Tạo gói AP



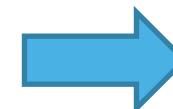
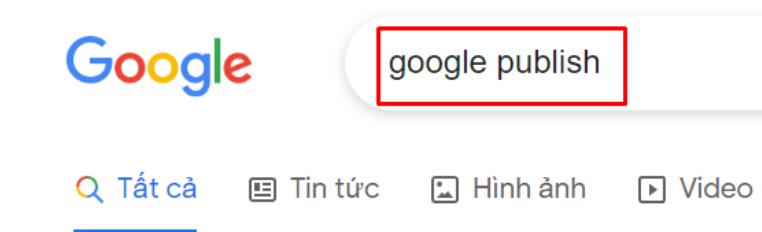


## Register Google Play Developer Account

A screenshot of the Google Play Console interface. On the left, there is a sidebar with links: "All applications", "Game services", "Download reports", "Alerts", and "Settings". The main area is titled "Welcome" and features two cards: one with the Google Play logo and a "PUBLISH AN ANDROID APP ON GOOGLE PLAY" button (which has a red circle around it), and another with a trophy and game controller icon labeled "USE GOOGLE PLAY GAME SERVICES". At the bottom left, there is a large orange triangle with the number "25\$". At the bottom, there is a green banner with the text "Buy Play Developer Account" in white.



## □ 1 . Đăng ký GooglePlay Developer Account



## Google Play Console

Publish your apps and games with the Google Play Console and grow your business on Google Play. Benefit from features that help you improve your app's quality, engage your audience, earn revenue, and more.

[Sign in](#) [Learn More](#)

<https://developer.android.com/distribute> → Dịch trang này

## Google Play Console - Android Developers

Publish your apps and games with the Google Play Console and grow your business on Google Play. Benefit from features that help you improve your app's ...



## □ 1 . Đăng ký GooglePlay Developer Account

Trước tiên, hãy chọn một loại tài khoản

Bạn đang tạo tài khoản cho ai? [Tim hiểu thêm về loại tài khoản cần chọn](#)

Bản thân

Chọn loại này nếu tài khoản của bạn được sử dụng cho mục đích cá nhân. Ví dụ: Nếu bạn là sinh viên hoặc người theo đuổi sở thích phát triển ứng dụng. Bạn vẫn có thể kiếm tiền trên Google Play và mời người khác tham gia tài khoản của mình.

**Bắt đầu →**



## □ 1 . Đăng ký GooglePlay Developer Account

*Phần giới thiệu các bạn đọc và làm theo, google đã hướng dẫn rất kỹ*

### Giới thiệu về bạn

Trải nghiệm của bạn với Play Console và Android

Hãy cho chúng tôi biết về trải nghiệm với Play Console và Android.  
Chỉ bạn mới thấy thông tin này.

Ví dụ: bạn có thể cho chúng tôi biết:

- bất kỳ trải nghiệm nào trong việc sử dụng Play Console
- bất kỳ ứng dụng nào bạn đã phát triển, quản lý hoặc phát hành
- bất kỳ trải nghiệm phát triển nào trên Android

Bạn có thể thêm các đường liên kết đến thông tin hỗ trợ.

0 / 5000



## □ 1 . Đăng ký GooglePlay Developer Account

### *Phản tài khoản nhà phát triển, đọc và làm theo*

#### Tài khoản nhà phát triển

Tên tài khoản nhà phát triển

Nhập tên công khai cho tài khoản nhà phát triển này. Thông tin này sẽ xuất hiện trên Google Play.

Nhập tên tài khoản nhà phát triển

0 / 50

Tên liên hệ

Đây là người chúng tôi sẽ liên hệ khi cần trao đổi về tài khoản của bạn. Thông tin này sẽ không xuất hiện trên Google Play.

Địa chỉ email liên hệ

Chúng tôi chỉ dùng địa chỉ email này để liên hệ với bạn và địa chỉ này sẽ không xuất hiện trên Google Play. Địa chỉ email này có thể khác với địa chỉ email liên kết với Tài khoản Google của bạn.



## □ 1 . Đăng ký GooglePlay Developer Account

### Tiếp tục đọc và làm theo

- ✓ Loại tài khoản
- ✓ Giới thiệu về bạn
- ✓ Tài khoản nhà phát triển
- **Ứng dụng**

#### Điều khoản

#### Ứng dụng

##### Số lượng ứng dụng

Bạn dự định phát hành bao nhiêu ứng dụng trên Google Play trong 12 tháng tới?

Nếu bạn không chắc chắn, hãy cung cấp con số ước chừng. Câu trả lời này sẽ không ảnh hưởng đến số lượng ứng dụng mà bạn có thể phát hành.

6 – 10

##### Kiếm tiền trên Google Play

Bạn có dự định kiếm tiền nhờ các ứng dụng phát hành trên Google Play không? Thông tin này sẽ không ảnh hưởng đến khả năng kiếm tiền nhờ ứng dụng cũng như những tính năng và dịch vụ mà bạn có thể tiếp cận.

Không

Có

Quay lại

**Tiếp theo**



## □ 1 . Đăng ký GooglePlay Developer Account

*Đồng ý với điều khoản của google ( mất phí 1 lần duy nhất là 25\$ )*

- ✓ Loại tài khoản
- ✓ Giới thiệu về bạn
- ✓ Tài khoản nhà phát triển
- ✓ Ứng dụng
- Điều khoản

### Điều khoản

- Tôi xác nhận rằng tôi đã đọc và đồng ý với [Thoả thuận phân phối dành cho nhà phát triển trên Google Play](#). Tôi xác nhận rằng tôi đã đủ 18 tuổi trở lên.
- Tôi xác nhận rằng tôi đã đọc và đồng ý với [Điều khoản dịch vụ của Google Play Console](#)

Tiếp tục nghĩa là bạn cam đoan và đảm bảo rằng bạn: (i) có đầy đủ quyền hạn pháp lý để ràng buộc tổ chức/cá nhân nêu trên tuân thủ Thoả thuận phân phối dành cho nhà phát triển trên Google Play và Điều khoản dịch vụ của Play Console; (ii) đã đọc và hiểu các Thoả thuận này; và (iii) thay mặt công ty/cá nhân đồng ý với các Thoả thuận này

-  Để tạo tài khoản, bạn cần thanh toán khoản phí đăng ký một lần là **25 USD**. Bạn có thể cần xác minh danh tính của mình bằng giấy tờ tùy thân hợp lệ để hoàn tất quá trình đăng ký tài khoản. Nếu không thể xác minh danh tính của bạn, chúng tôi sẽ không hoàn lại khoản phí đăng ký.

Quay lại

**Tạo tài khoản và thanh toán**



## □ 2 . Xác minh Tài khoản google developer

### Tất cả Ứng dụng

Xem tất cả các ứng dụng và trò chơi bạn có quyền truy cập trong tài khoản nhà phát triển của mình



#### Đang xác minh ID



Mã nhận dạng của bạn đang được xác minh. Quá trình này có thể mất đến 2 ngày làm việc. Bạn có thể phát hành ứng dụng sau khi quá trình xác minh hoàn tất.

[Tim hiểu thêm](#)

