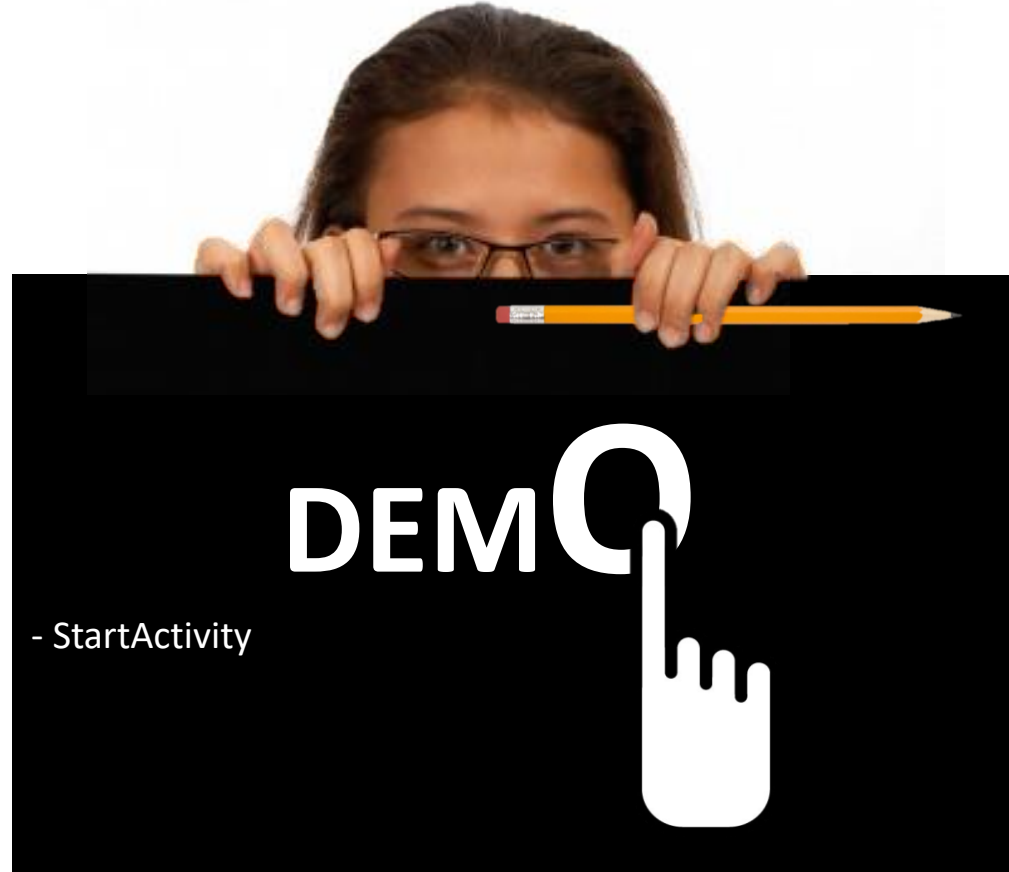


LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG

CHƯƠNG 3:

LAYOUT CỦA ỨNG DỤNG ANDROID

- ❖ Giảng viên: Ths. Nguyễn Trung Hiếu
- ❖ Email: hieun.tg@ptithcm.edu.vn
- ❖ Mobie: 0983051825



- ⊙ Kết thúc bài học này bạn có khả năng
 - ⊙ Sử dụng Layout của ứng dụng Android
 - ⊙ Sử dụng các widget cơ bản



Phần I: Layout của ứng dụng Android

 RelativeLayout

 LinearLayout

Phần II: Các widget cơ bản

 TextView

 EditText

 Button

 Image

 ListView



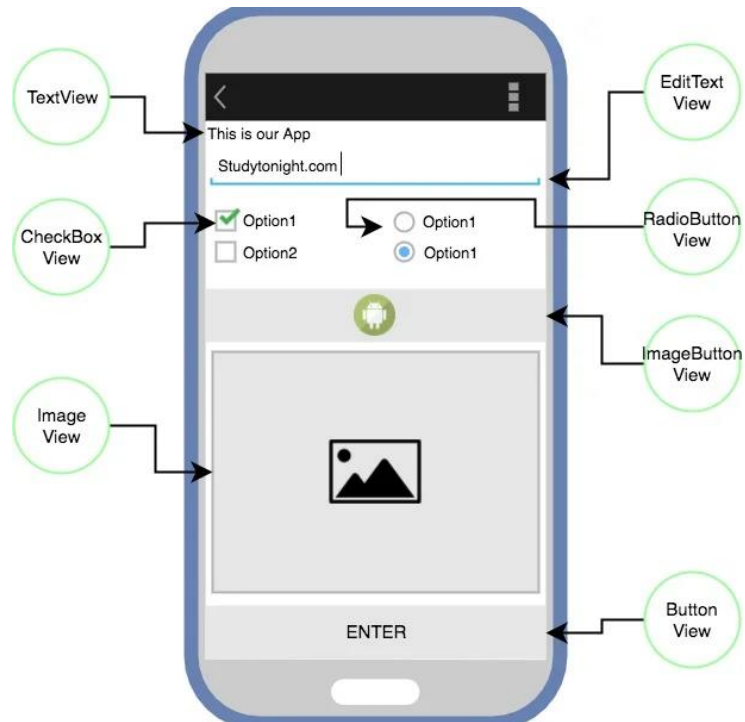
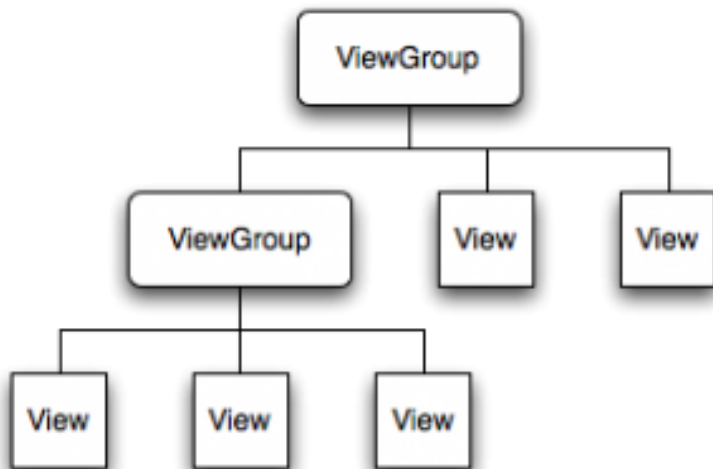
1

PHẦN I: Layout của ứng dụng Android

Thiết kế giao diện người dùng

- Càng đơn giản càng tốt
- Dành nhiều thời gian tìm hiểu nhu cầu của khách hàng về giao diện
- Sử dụng điều khiển giao diện chuẩn

Cây phân cấp View (View Hierarchy)



Cây phân cấp View (View Hierarchy)

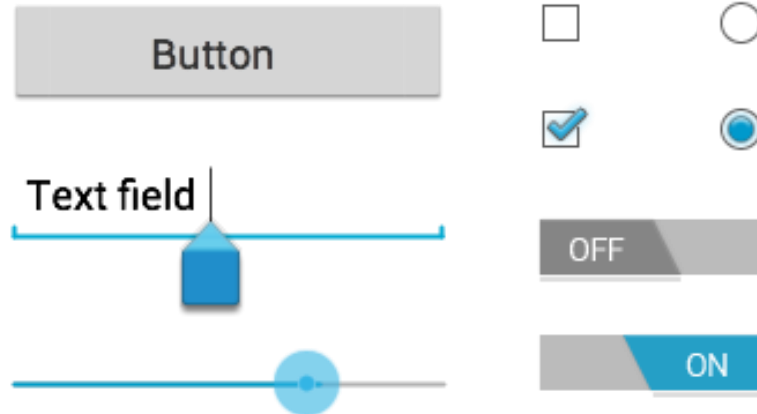
- View: đơn vị cơ bản của giao diện người dùng
 - Widgets: `android.widget.*`
 - Là lá của cây phân cấp View
- ViewGroup: chính là bố cục layout
 - Nằm trong `android.widget.*`
 - Định nghĩa nơi chứa các Views (hoặc View Group) con



View sẽ nằm trong ViewGroup. Khi ta xóa view thì Viewgroup thay đổi. Nhưng nếu xóa Viewgroup thì tất cả view trong viewgroup mất.

Ví dụ Widget (view)

- Button
- EditText
- CheckBox và RadioButton
- TextView, ImageView,...



LAYOUT CỦA ỨNG DỤNG ANDROID

Ví dụ về Layout (ViewGroup)

- LinearLayout
- RelativeLayout
- FrameLayout
- TableLayout
- Absolute Layout
- ScrollView Layout
-

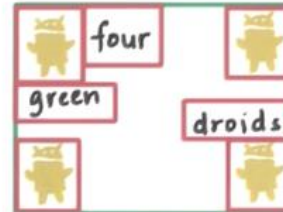
FrameLayout



LinearLayout



RelativeLayout



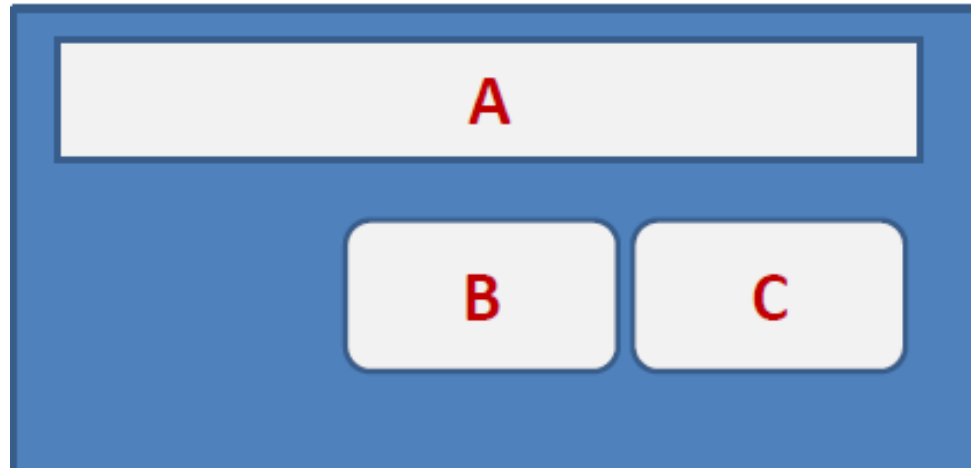
GridLayout



- RelativeLayout cho phép sắp xếp các control theo vị trí tương đối giữa các control khác trên giao diện (kể cả control chứa nó).
- Thường ta sẽ dựa vào Id của các control khác để sắp xếp theo vị trí tương đối.
- Do đó khi làm RelativeLayout phải chú ý là đặt Id control cho chuẩn xác, nếu sau khi Layout xong mà lại đổi Id của các control thì giao diện sẽ bị xáo trộn (do đó nếu đổi ID thì phải đổi luôn các tham chiếu khác sao cho khớp với Id mới đổi).

Ví dụ:

- A đứng trên đầu ,
- C bên dưới A và ở phía bên phải,
- B bên dưới A và bên trái C



Một số thuộc tính sắp xếp widget với layout chứa nó:

- **android:layout_alignParentTop**: chỉ ra rằng widget phải được đặt ở đầu của layout mà nó nằm.
- **android:layout_alignParentBottom** đặt ở dưới cùng
- **android:layout_alignParentLeft** đặt ở bên trái
- **android:layout_alignParentRight** : đặt ở bên phải
- **android:layout_centerInParent** : đặt ở trung tâm
- **android:layout_centerHorizontal**: đặt ở trung tâm theo chiều ngang
- **android:layout_centerVertical**: đặt ở trung tâm theo chiều dọc

Một số thuộc tính sắp xếp widget với các widget khác:

- **android:layout_above** chỉ ra rằng widget phải được đặt ở trên của widget tham chiếu.
- **android:layout_below** chỉ ra rằng widget phải được đặt ở dưới của widget tham chiếu.
- **android:layout_toLeftOf** chỉ ra rằng widget phải được đặt ở bên trái của widget tham chiếu.
- **android:layout_toRightOf** chỉ ra rằng widget phải được đặt ở bên phải của widget tham chiếu.

- **android:layout_alignTop:** làm cho top của widget này căn bằng với top của widget tham chiếu
- **android:layout_alignBottom** làm cho cạnh dưới của widget này căn bằng với cạnh dưới của widget tham chiếu
- **android:layout_alignLeft** làm cho cạnh trái của widget này căn bằng với cạnh trái của widget tham chiếu
- **android:layout_alignRight** làm cho cạnh phải của widget này căn bằng với cạnh phải của widget tham chiếu

Thuộc tính `android:layout_gravity` trong các View con

Khi các View nằm trong **FrameLayout** thì khi gán thuộc tính **`android:layout_gravity`** gán các giá trị ở bảng sau vị trí của nó thay đổi tương ứng:

GIÁ TRỊ	VỊ TRÍ CỦA VIEW CON
bottom	Nằm dưới FrameLayout
center	Nằm giữa FrameLayout
center_horizontal	Giữa theo chiều ngang
center_vertical	Giữa theo chiều đứng
end	Cuối FrameLayout
left	Bên trái
right	Bên phải
start	Bắt đầu FrameLayout
top	Trên FrameLayout

Để sắp xếp các Widget, ta cần phải làm những bước sau:

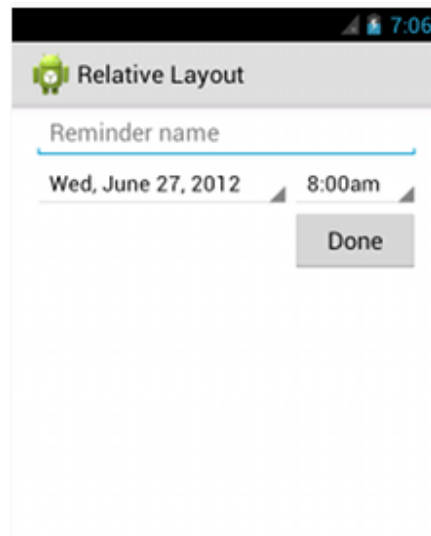
- Gán Id cho tất cả các phần tử control (**android:id**)
- Cú pháp: **@+id/...** để đặt id cho từng control, ví dụ cho thẻ EditText là **android:id = "@+id/editUserName"**
- Để bố trí control khác liên quan đến control này, ta cũng sẽ sử dụng giá trị (**@+id/...**). Ví dụ đặt một control dưới hộp EditText ở trên ta có câu lệnh:

android:layout_below = "@+id/editUserName"

RELATIVELAYOUT

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_cotent"
        android:layout_below="@id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentRight="true" />
```

```
<Button
    android:layout_width="96dp"
    android:layout_height="wrap_content"
    android:layout_below="@id/times"
    android:layout_alignParentRight="true"
    android:text="@string/done" />
</RelativeLayout>
```



<RelativeLayout

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:padding="18dp"
xmlns:android="http://schemas.android.com/apk/res/android">
```

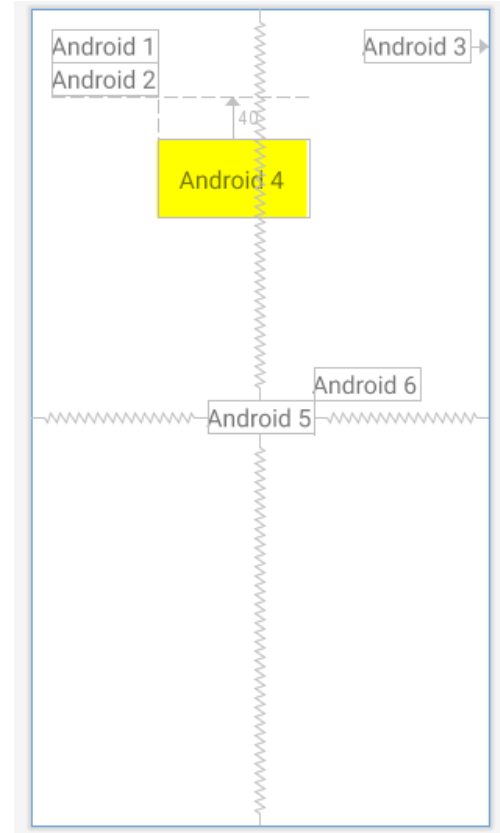
<TextView

```
android:id="@+id/txtView1"
android:textSize="22sp"
android:text="Android 1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
/>
```

<TextView

```
android:layout_below="@id/txtView1"
android:id="@+id/txtView2"
android:textSize="22sp"
android:text="Android 2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
/>
```

RELATIVELAYOUT



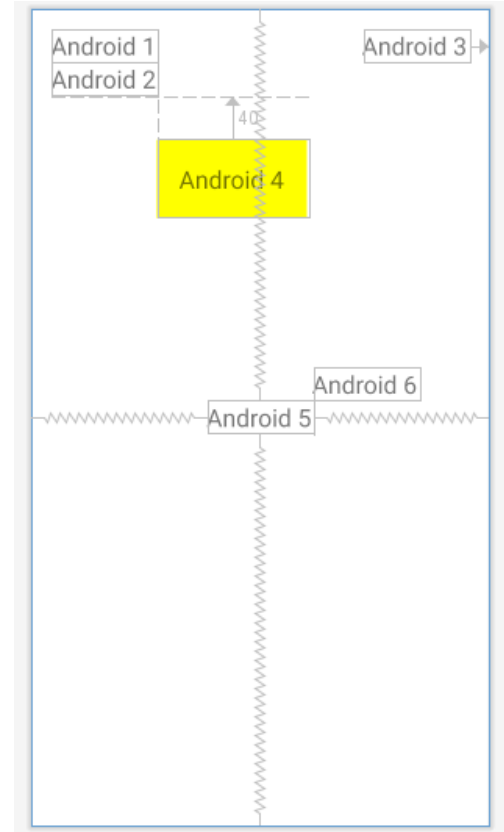
RELATIVELAYOUT

<TextView

```
android:id="@+id/txtView3"  
android:layout_alignParentRight="true"  
android:textSize="22sp"  
android:text="Android 3"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
>
```

<TextView

```
android:padding="20dp"  
android:background="#ff0"  
android:layout_below="@id/txtView2"  
android:layout_toRightOf="@id/txtView2"  
android:id="@+id/txtView4"  
android:layout_marginTop="40dp"  
android:textSize="22sp"  
android:text="Android 4"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
>
```



RELATIVELAYOUT

```
<TextView
```

```
    android:layout_centerInParent="true"
```

```
    android:id="@+id/txtView5"
```

```
    android:textSize="22sp"
```

```
    android:text="Android 5"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
>
```

```
<TextView
```

```
    android:layout_above="@id/txtView5"
```

```
    android:id="@+id/txtView6"
```

```
    android:textSize="22sp"
```

```
    android:text="Android 6"
```

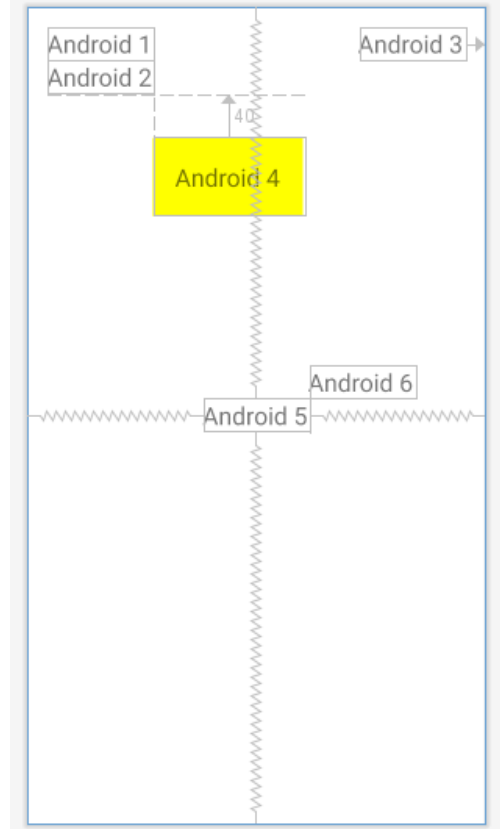
```
    android:layout_toRightOf="@id/txtView5"
```

```
    android:layout_width="wrap_content"
```

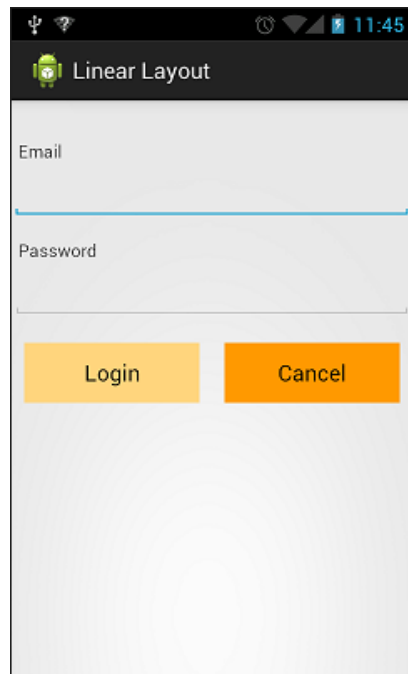
```
    android:layout_height="wrap_content"
```

```
>
```

```
</RelativeLayout>
```



- Là layout sắp xếp các View con trong nó theo duy nhất một chiều, ngang hoặc dọc theo giá trị của thuộc tính android:orientation
 - Orientation = vertical hoặc horizontal
- Có thể lồng nhiều layout phức tạp
 - Thông thường sử dụng cho form nhỏ

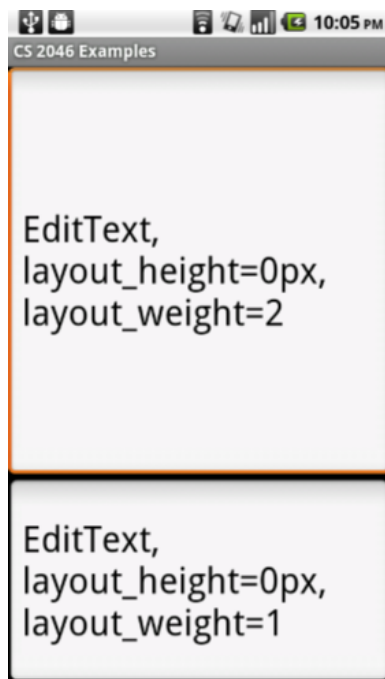


LinearLayout lồng nhau

- LinearLayout lồng nhau là một cách để dễ dàng tạo các giao diện chung
- Chú ý: nếu lồng nhau mà số cấp lớn hơn hoặc bằng 5 sẽ làm cho việc tải giao diện chậm hơn

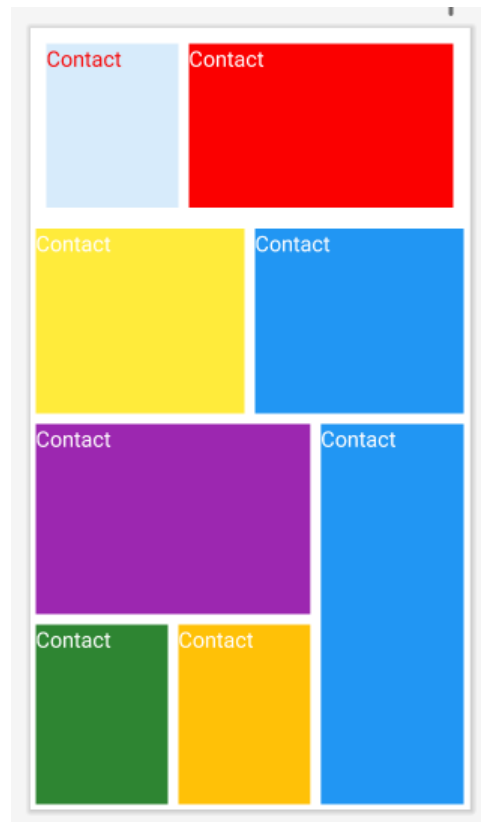
Trọng lượng của Layout (layout weight)

- Trọng lượng cho phép tạo LinearLayout với cỡ cân đối
- Default = 0 – không gian tối thiểu để hiển thị tất cả nội dung



Ví dụ

- Chúng ta sẽ định nghĩa layout cho giao diện sau như thế nào?



FrameLayout là loại View cơ sở, nó là loại Layout đơn giản nhất. Mặc dù nó có thể chứa nhiều View con bên trong, nhưng mục đích chính thiết kế ra nó để chứa một View, từ đó nó trở thành cơ sở để tạo ra các View khác phức tạp hơn. Khi thiết kế Layout chứa nhiều View thì không nên sử dụng layout này, vì nó quá đơn giản việc bố cục các View con trong nó rất khó khăn (nó không có các tính năng điều khiển vị trí View con sao cho việc độc lập về màn hình được đảm bảo).

Nếu bạn vẫn sử dụng **FrameLayout** để thiết kế layout, thì cần lưu ý: Các View con đặt vào FrameLayout nằm chồng lên nhau theo thứ tự cái nào đưa vào sau thì hiện thị ở lớp trước, mỗi View con chỉ có thể điều chỉnh vị trí nó thông qua thuộc tính **android:layout_gravity** gán cho View con

Và có thể tinh chỉnh khoảng cách theo hướng cách cánh bằng các thuộc tính liên quan đến **margin** như: *android:layout_margin*, *android:layout_marginTop*, *android:layout_marginBottom*, *android:layout_marginLeft*, *android:layout_marginRight*, *android:layout_marginStart*, *android:layout_marginEnd*, *android:layout_marginHorizontal*, *android:layout_marginVertical*

<FrameLayout

```
android:layout_width="match_parent"  
android:layout_height="match_parent"  
xmlns:android="http://schemas.android.com/apk/res/android">
```

<ImageView

```
android:src="@drawable/backgroundAndroid"  
android:layout_width="match_parent"  
android:layout_height="match_parent">
```

</ImageView>

<TextView

```
android:layout_marginTop="40dp"  
android:layout_gravity="center_horizontal"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:textColor="#ffc"  
android:textSize="40dp"  
android:text="Lập trình Android"  
>
```

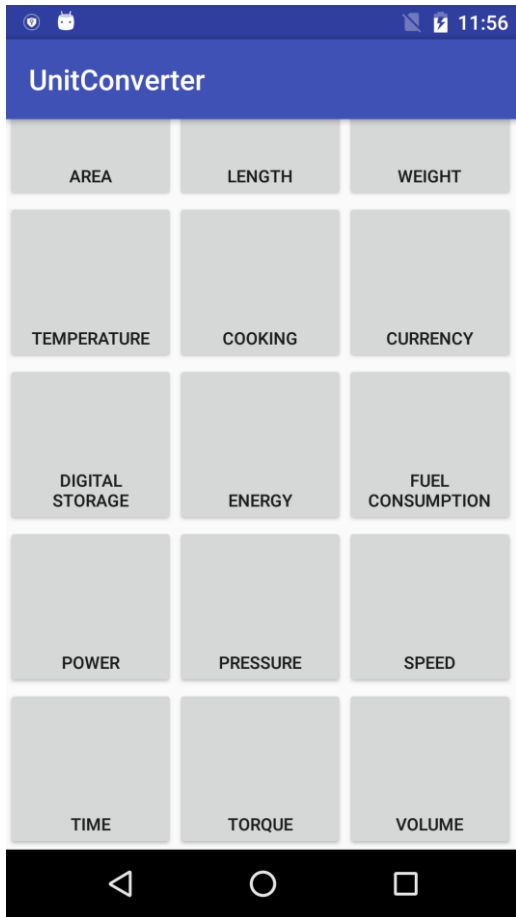
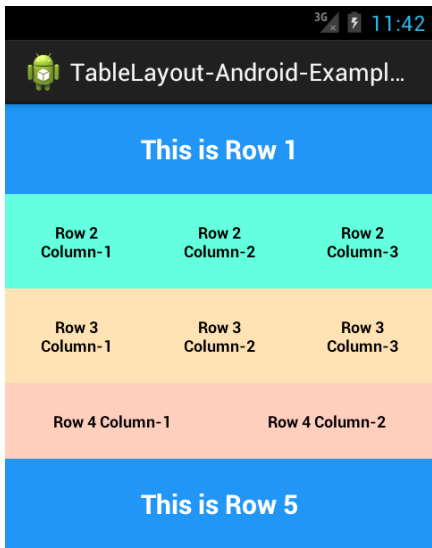
</TextView>

</FrameLayout>

FRAME LAYOUT



- ❖ Với **TableLayout** nó sẽ sắp xếp các View con bên trong thành dạng bảng.
- ❖ Mỗi hàng là một đối tượng view **TableRow** bên trong **TableRow** chứa các View con, mỗi View con này nằm ở vị trí một ô bảng (cell).
- ❖ Cột / hàng trong bảng bắt đầu từ số 0.



<TableLayout

```
    android:stretchColumns="1"
    android:shrinkColumns="true"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
```

<TableRow android:background="#2196F3">

<TextView

```
    android:layout_span="3"
    android:padding="5dp"
    android:gravity="center"
    android:textColor="#fff"
    android:textSize="15sp"
    android:text="This is row 1"
```

>

</TextView>

</TableRow>

<TableRow android:background="#A4E7EF" >

<TextView

```
    android:padding="5dp"
    android:gravity="center"
    android:textSize="15sp"
    android:text="Row 2 Column 1"
```

>

TABLELAYOUT

This is row 1		
Row 2 Column 1	Row 2 Column 2	Row 2 Column 3
Row 3 Column 1	Row 3 Column 2	Row 3 Column 3
Row 4 Column 1	Row 4 Column 2	
This is row 5		

<TextView

```
    android:padding="5dp"
    android:gravity="center"
    android:textSize="15sp"
    android:text="Row 2 Column 2"
>
```

</TextView>

<TextView

```
    android:padding="5dp"
    android:gravity="center"
    android:textSize="15sp"
    android:text="Row 2 Column 3"
>
```

</TextView>

</TableRow>

<TableRow android:background="#FFC107"

>

<TextView

```
    android:padding="5dp"
    android:gravity="center"
    android:textSize="15sp"
    android:text="Row 3 Column 1"
>
```

</TextView>

TABLELAYOUT

This is row 1		
Row 2 Column 1	Row 2 Column 2	Row 2 Column 3
Row 3 Column 1	Row 3 Column 2	Row 3 Column 3
Row 4 Column 1	Row 4 Column 2	
This is row 5		

TABLELAYOUT

```
<TextView  
    android:padding="5dp"  
    android:gravity="center"  
    android:textSize="15sp"  
    android:text="Row 3 Column 2"  
>
```

```
</TextView>
```

```
<TextView  
    android:padding="5dp"  
    android:gravity="center"  
    android:textSize="15sp"  
    android:text="Row 3 Column 3"  
>
```

```
</TextView>
```

```
</TableRow>
```

```
<TableRow  
    android:weightSum="2"  
    android:background="#FFE1FD" >
```

```
<TextView  
    android:layout_weight="1"  
    android:padding="5dp"  
    android:gravity="center"  
    android:textSize="15sp"  
    android:text="Row 4 Column 1"  
>
```

This is row 1		
Row 2 Column 1	Row 2 Column 2	Row 2 Column 3
Row 3 Column 1	Row 3 Column 2	Row 3 Column 3
Row 4 Column 1	Row 4 Column 2	
This is row 5		

TABLELAYOUT

```
<TextView  
    android:layout_weight="1"  
    android:padding="5dp"  
    android:gravity="center"  
    android:textSize="15sp"  
    android:text="Row 4 Column 2"  
>
```

```
</TextView>
```

```
</TableRow>
```

```
<TableRow android:background="#2196F3">
```

```
    <TextView  
        android:layout_span="3"  
        android:padding="5dp"  
        android:gravity="center"  
        android:textColor="#fff"  
        android:textSize="15sp"  
        android:text="This is row 5"  
    >
```

```
    </TextView>
```

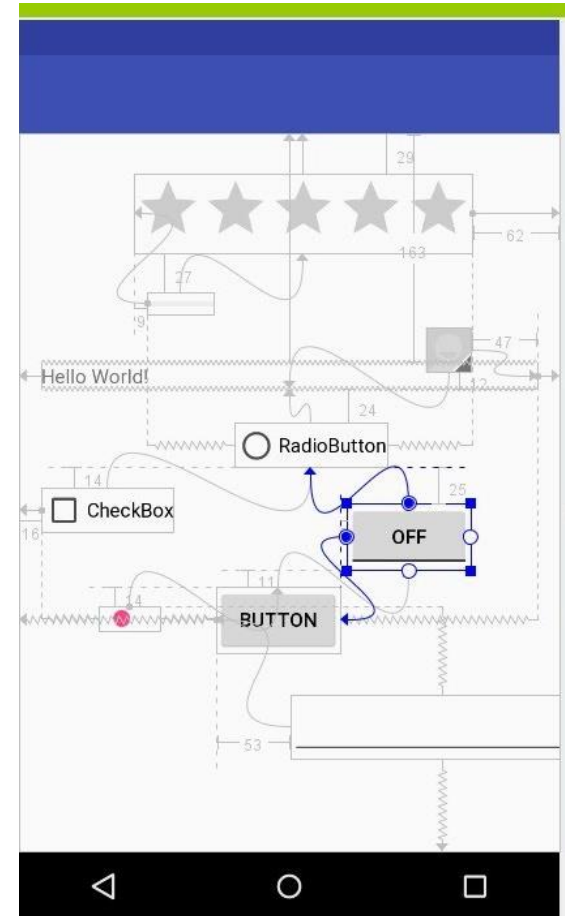
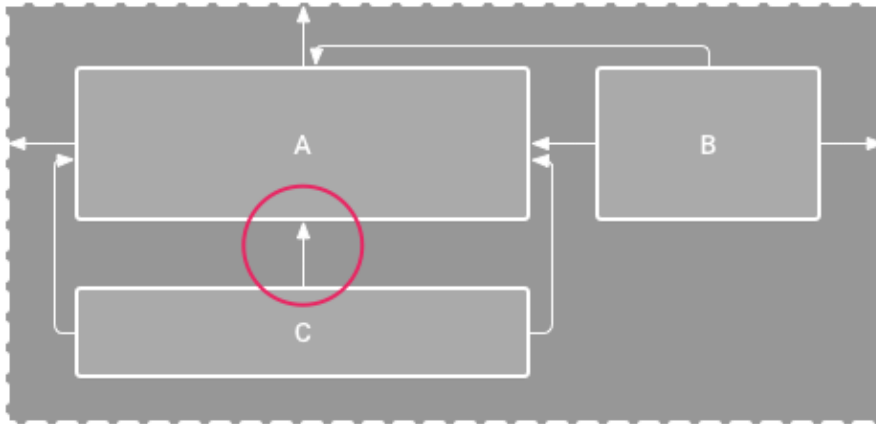
```
</TableRow>
```

```
</TableLayout>
```

This is row 1		
Row 2 Column 1	Row 2 Column 2	Row 2 Column 3
Row 3 Column 1	Row 3 Column 2	Row 3 Column 3
Row 4 Column 1	Row 4 Column 2	
This is row 5		

CONSTRAINTLAYOUT

ConstraintLayout tạo ra các giao diện phức tạp, mềm dẻo (hạn chế tối đa sử dụng các layout lồng nhau). Nó giúp định vị, sắp xếp các View con dựa trên sự ràng buộc liên hệ của các View con với View cha và sự liên hệ ràng buộc giữa các View con với nhau.



CONSTRAINTLAYOUT

Sự ràng buộc: Mỗi view trong **ConstraintLayout** để định vị được chính xác cần tối thiểu 2 ràng buộc, một theo phương ngang (X) và một theo phương đứng (Y). Mỗi ràng buộc của phần tử View sẽ hoặc hướng đến vị trí của nó theo trục X hoặc trục Y. Các View không có ràng buộc sẽ định vị ở góc trái - trên (tọa độ 0,0).

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

2

PHẦN II: Các widget cơ bản

1. TextView

- Trong Android, một label được sử dụng là **TextView**.
- Một **TextView** dùng để hiển thị thông tin và không cho phép người dùng chỉnh sửa.
- Một số thuộc tính của control:
 - Cần thiết lập id cho control.
 - layout_width, layout_height cũng nên thiết lập cho control (bắt buộc)
 - Để thay đổi màu nền dùng background, thay đổi màu chữ dùng textColor...

PHONE	
1 650-123	MOBILE
EMAIL	
Email	HOME
ADDRESS	
Address	HOME

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
  android:id="@+id/absLayout"
  android:layout_width="fill_parent" android:layout_height="fill_parent"
  xmlns:android="http://schemas.android.com/apk/res/android"
>
```

<TextView

```
  android:id="@+id/myTextView1"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:background="#ff0000ff"
  android:padding="3px"
  android:text="Enter User Name"
  android:textSize="16sp"
  android:textStyle="bold"
  android:gravity="center"
  android:layout_x="20px"
  android:layout_y="22px" >
```

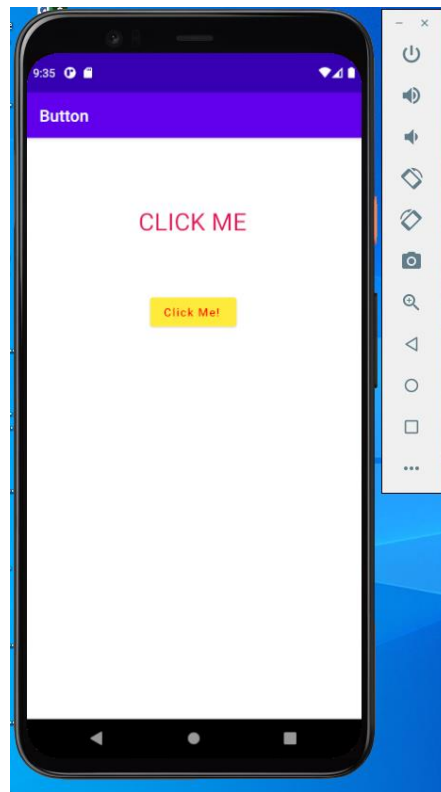
</TextView>

```
</AbsoluteLayout>
```

2. Button

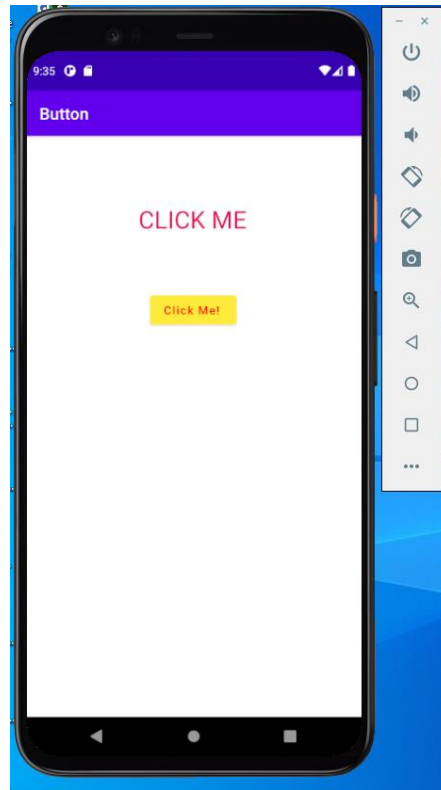
<Button

```
    android:id="@+id/btClickMe"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="68dp"  
    android:text="Click Me! "  
    android:textColor="#FF0000"  
    android:backgroundTint="#FFEB3B"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.498"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/txtShowNoiDung"  
    android:textAllCaps="false"  
>
```



1. Button

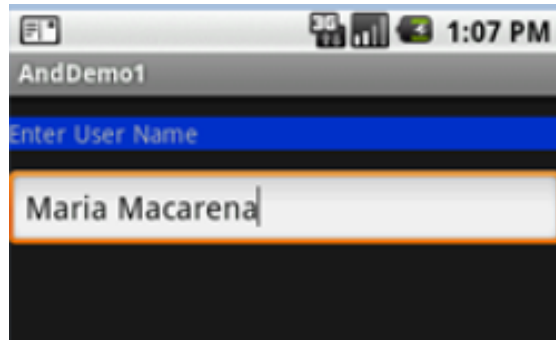
```
TextView txtShowNoiDung ;  
Button btClickMe;  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    txtShowNoiDung = (TextView) findViewById(R.id.txtShowNoiDung);  
    btClickMe = (Button) findViewById(R.id.btClickMe);  
    btClickMe.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            txtShowNoiDung.setText("Lập trình Android");  
        }  
    });  
}
```



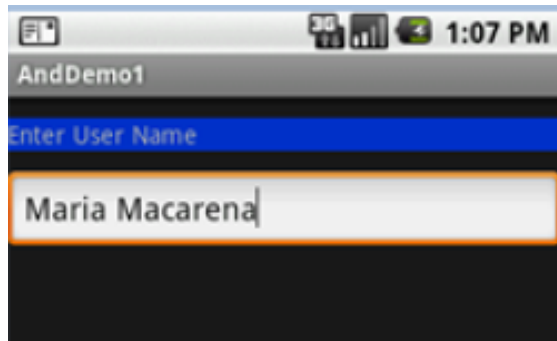
3. EditText

- Thẻ EditText (hoặc textBox) là một trường hợp mở rộng của TextView, cho phép cập nhật, chỉnh sửa dữ liệu

```
<EditText  
android:id="@+id/txtUserName"  
android:layout_width="fill_parent"  
android:layout_height="wrap_content"  
android:textSize="18sp"  
android:autoText="true"  
android:capitalize="words"  
android:hint="First Last Name"  
>  
</EditText>
```



```
<EditText  
    android:id="@+id/txtUserName"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:textSize="18sp"  
    android:autoText="true"  
    android:capitalize="words"  
    android:hint="First Last Name"  
>  
</EditText>
```



- Lấy control theo Id:

```
EditText txtbox=(EditText) findViewById(R.id.editText1);
```

- Thiết lập giá trị cho EditText

```
txtBox.setText("some text")
```

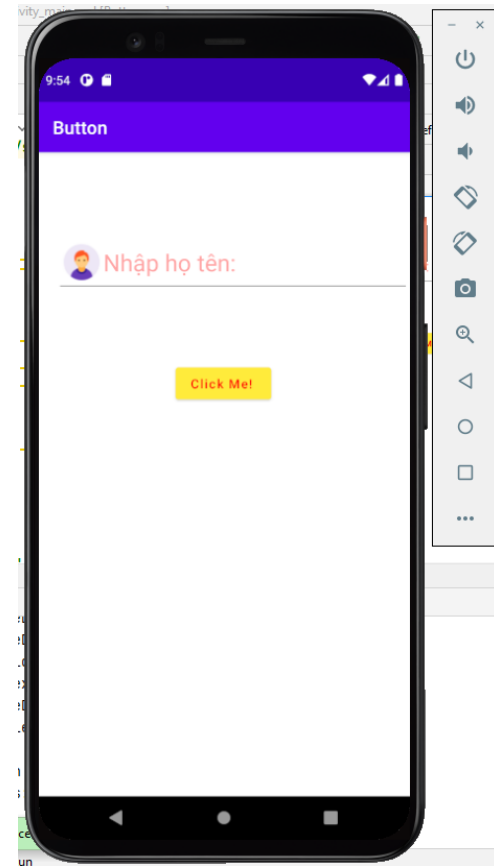
- Lấy dữ liệu bên trong EditText:

```
String msg=txtBox.getText().toString()
```

CÁC WIDGET CƠ BẢN

<EditText

```
android:id="@+id/editTextTextPersonName"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginLeft="20dp"  
android:layout_marginTop="88dp"  
android:drawableLeft="@drawable/user"  
android:ems="10"  
android:hint="Nhập họ tên:"  
android:inputType="textPersonName"  
android:textColor="#F80000"  
android:textColorHint="#FFA5A5"  
android:textSize="26sp"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.497"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

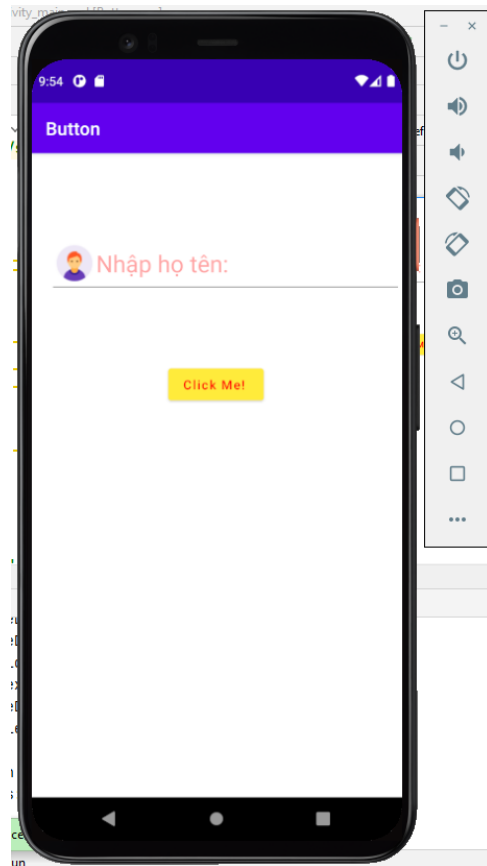


CÁC WIDGET CƠ BẢN

```
android:drawableBottom
android:drawableEnd
android:drawableLeft
android:drawablePadding
android:drawableRight
android:drawableStart
android:drawableTint
android:drawableTintMode
android:drawableTop
android:drawingCacheQuality
android:scrollbarAlwaysDrawHorizontalTrack
android:scrollbarAlwaysDrawVerticalTrack
Press Enter to insert, Tab to replace Next Tip
```

Cách đặt tên cho hình trong Android:

- ❖ Không viết chữ In/Hoa
- ❖ Không có khoảng trắng
- ❖ Không có ký tự đặt biệt trừ ký tự “_”
- ❖ Không bắt đầu là con số: 123abc -> abc123



4. Image

- **ImageView** là loại View dùng để hiển thị tài nguyên hình ảnh như các ảnh Bitmap, các ảnh Drawable. Nó cũng cung cấp các chức năng tùy biến khác nhau như đổ màu nhuộm (tint) vào ảnh, co kéo/cắt ảnh khi hiển thị trên View.
- Tài nguyên ảnh cần hiển thị gán vào thuộc tính **android:src** hoặc dùng phương thức phù hợp với loại ảnh hiển thị như: **setImageBitmap(Bitmap bm)**, **setImageDrawable(Drawable drawable)**, **setImageResource(int resId)**, **setImageURI(Uri uri)**

```
<ImageView
    android:scaleType="center"
    android:id="@+id/imageView"
    android:layout_width="301dp"
    android:layout_height="490dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.326"
    tools:srcCompat="@tools:sample/backgrounds/scenic"
/>
```

4. Image

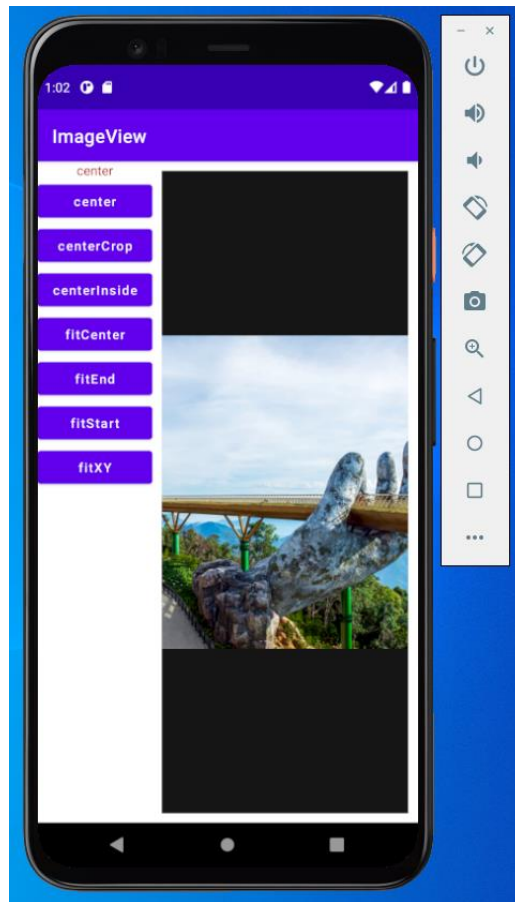
Thay đổi tỷ lệ / căn chỉnh ảnh hiển thị trong ImageView

- Thuộc tính **android:scaleType** dùng để thiết lập thu phóng ảnh, nhận các giá trị như: *fitXY*, *center*, *fitXY* ví dụ như **android:scaleType="center"**
- Code Java: `imageView.setScaleType(ImageView.ScaleType.CENTER);`

GIÁ TRỊ	SỬ DỤNG
center	ImageView.ScaleType.CENTER : Đặt ảnh vào giữa ImageView, không có thay đổi tỷ lệ ảnh.
centerCrop	ImageView.ScaleType.CENTER_CROP: Đặt ảnh vào giữa ImageView, có thu phóng ảnh (nhưng giữ nguyên tỉ lệ cao / rộng) sao cho ảnh phủ kín hết cả ImageView (phần thừa bị cắt)
centerInside	ImageView.ScaleType.CENTER_INSIDE: Đặt ảnh vào giữa ImageView, có thu phóng ảnh (nhưng giữ nguyên tỉ lệ cao / rộng) sao cho toàn bộ các phần của ảnh hiển thị trên ImageView.
fitCenter	ImageView.ScaleType.FIT_CENTER: Đặt ảnh vào giữa ImageView, có thu phóng ảnh (nhưng giữ nguyên tỉ lệ cao / rộng) sao cho toàn bộ các phần của ảnh hiển thị trên ImageView.
fitEnd fitStart	ImageView.ScaleType.FIT_END, ImageView.ScaleType.FIT_START: Co ảnh vừa View, vị trí ảnh ở cuối (ở đầu) ImageView
fitXY	ImageView.ScaleType.FIT_XY: Co ảnh vừa khít cả chiều rộng và cao.

CÁC WIDGET CƠ BẢN

```
ImageView imageView;  
TextView textView;  
Button btn_center;  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    imageView = findViewById(R.id.image);  
    textView = findViewById(R.id.textview);  
    btn_center = findViewById(R.id.btn_center);  
    btn_center.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            imageView.setScaleType(ImageView.ScaleType.CENTER);  
            textView.setText(btn_center.getText().toString());  
        }  
    });  
}
```



4. Image

- **ImageView** và **ImageButton** thường được dùng để nhúng hình ảnh cho ứng dụng.
- Cả hai thẻ này đều tương tự như **TextView** và **Button**.
- Các thẻ này đều sử dụng thuộc tính **android:src** hoặc **android:background** để hiển thị ảnh.
- Ảnh được sử dụng sẽ được tham chiếu tới thư mục **/res/drawable/**
- **ImageButton** là một lớp con của **ImageView** và thêm một số chuẩn của **Button** để thực hiện các sự kiện Click



COMPOUNDBUTTON

CompoundButton: là một lớp cơ abstract mở rộng từ TextView (Button), từ lớp này nó được mở rộng để xây dựng các View là: **CheckBox, RadioButton, Switch, ToggleButton**. Các lớp mở rộng CheckBox, RadioButton, Switch, ToggleButton lưu lại các thuộc tính, ứng xử tương tự như TextView, Button nhưng có thêm hai trạng thái là checked và unchecked

Một số thuộc tính, phương thức chung cho CheckBox, RadioButton, Switch, ToggleButton

Trong XML

- android:checked thiết lập trạng thái checked, unchecked với giá trị "true" hay "false"
- android:button để gán Drawable vào View (vẽ trạng thái cho CheckBox, RadioButton ...)
- android:buttonTint để gán màu Tint

Trong code Java:

- isChecked() để kiểm tra trạng thái là checked (true) hay unchecked (false)
- setOnCheckedChangeListener bắt sự kiện khi trạng thái (checked/unchecked) chuyển đổi:

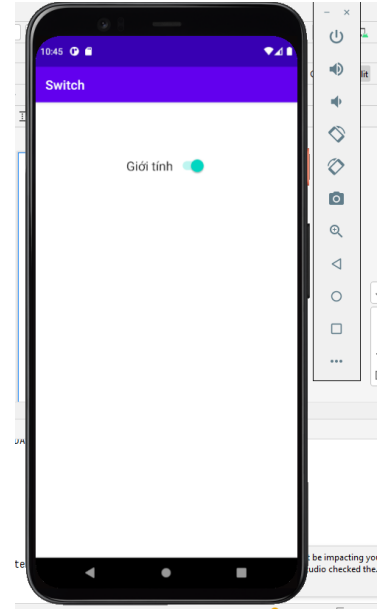
```
button.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChanged(CompoundButton compoundButton, boolean b) {  
        //b - trạng thái  
        //code làm gì khi chuyển trạng thái ở đây }  
    });
```

SWITCH

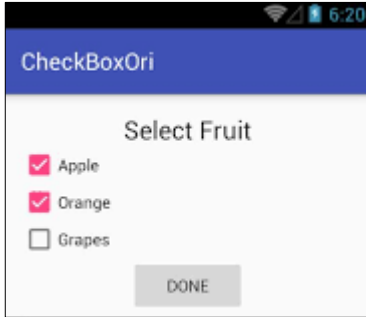


- **Switch** là loại View, mở rộng từ [CompoundButton](#), nên nó có các thiết lập, thuộc tính, triển khai code và XML tương tự như [TextView](#), [Button](#), [CheckBox](#).
- Cách viết code có các phương thức như: **isChecked**, **setChecked**, **setOnCheckedChangeListener**, **setOnClickListener**

```
swt = (Switch) findViewById(R.id.swtGener);
swt.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if(isChecked) {
            Toast.makeText(MainActivity.this, "Nam", Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(MainActivity.this, "Nữ", Toast.LENGTH_LONG).show();
        }
    }
});
```

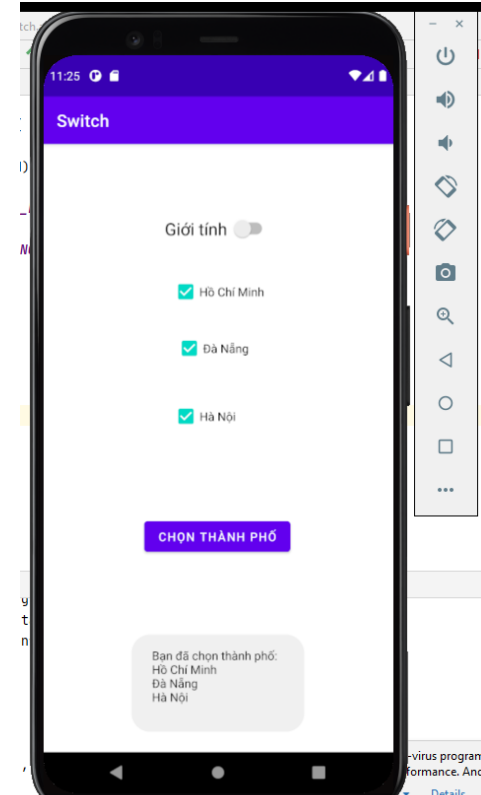


CHECKBOX

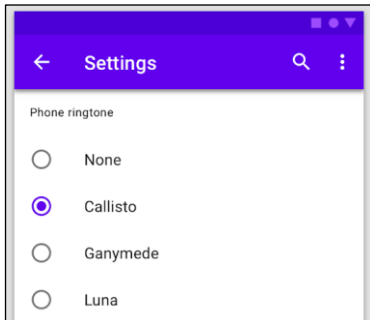


- CheckBox mở rộng từ CompoundButton tạo là một loại Button cho phép hiện thị hộp kiểm, hiện thị thông tin trạng thái checked hay unchecked, mã để tùy biến CheckBox

```
btXacnhan.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        String chonCity = "Bạn đã chọn thành phố: \n";  
        if(cbHCM.isChecked()){  
            chonCity += cbHCM.getText() + "\n";  
        }  
        if(cbDaNang.isChecked()){  
            chonCity += cbDaNang.getText() + "\n";  
        }  
        if(cbHaNoi.isChecked()){  
            chonCity += cbHaNoi.getText() + "\n";  
        }  
        Toast.makeText(MainActivity.this, chonCity, Toast.LENGTH_LONG).show();  
    }  
});
```

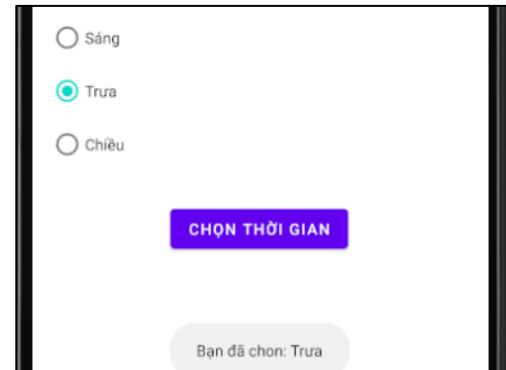


RADIOBUTTON VÀ RADIOGROUP

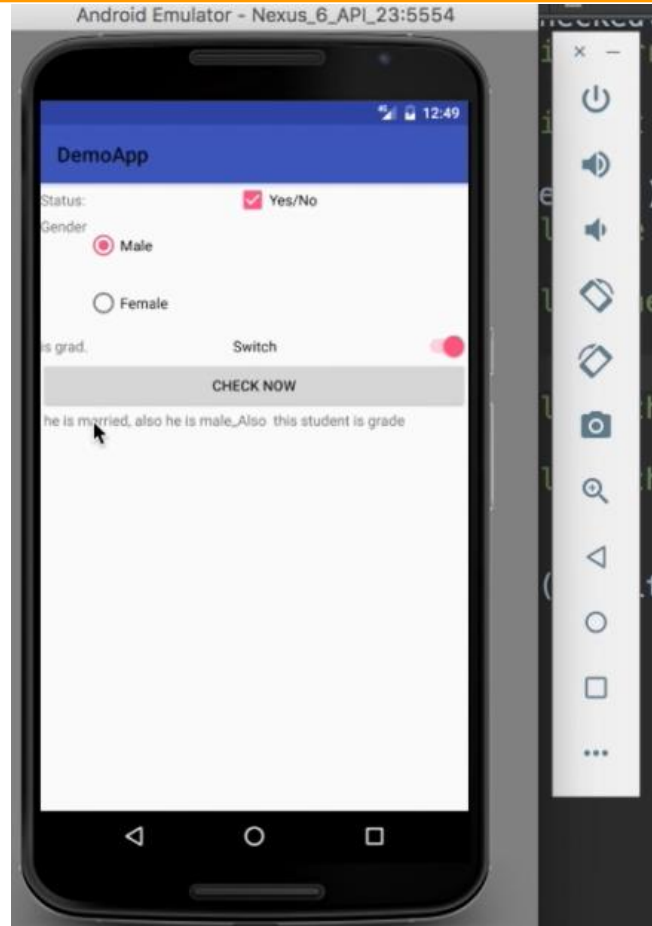


- RadioButton và RadioGroup: RadioButton cũng là loại control biểu diễn trạng thái checked/unchecked. Có điểm khác với CheckBox, Switch đó là khi người dùng nhấn vào chọn nó, nó sẽ chuyển sang checked nếu đang là unchecked, nhưng chiều ngược lại nếu nó đang là checked thì không thể bấm vào nó để chuyển sang trạng thái unchecked (Tuy nhiên có thể thiết lập bằng code)
- Thường RadioButton sẽ sử dụng cùng với RadioGroup, lúc đó sẽ có vài RadioButton bên trong RadioGroup và ở một thời điểm người dùng chỉ có thể chọn một

```
btChonTG.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if(rbSang.isChecked())  
        {  
            Toast.makeText(MainActivity.this, "Bạn đã chọn: " + rbSang.getText(), Toast.LENGTH_LONG).show();  
        }  
        if(rbTrua.isChecked())  
        {  
            Toast.makeText(MainActivity.this, "Bạn đã chọn: " + rbTrua.getText(), Toast.LENGTH_LONG).show();  
        }  
        if(rbChieu.isChecked())  
        {  
            Toast.makeText(MainActivity.this, "Bạn đã chọn: " + rbChieu.getText(), Toast.LENGTH_LONG).show();  
        }  
    }  
});
```



CHECKBOX



- ListView là phần tử View được dùng để hiển thị dữ liệu là một danh sách (mảng) từ một nguồn cấp gọi là Adapter, các bước để tạo và sử dụng ListView gồm có:
1. Khai báo ListView trong Layout

<ListView

android:id="@+id/listproduct"

android:layout_width="match_parent"

android:layout_height="match_parent" />

Khai báo Adapter – ràng buộc nội dung động vào View trong ListView với code java:

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>  
(this, android.R.layout.simple_list_item_1, Subjects);
```

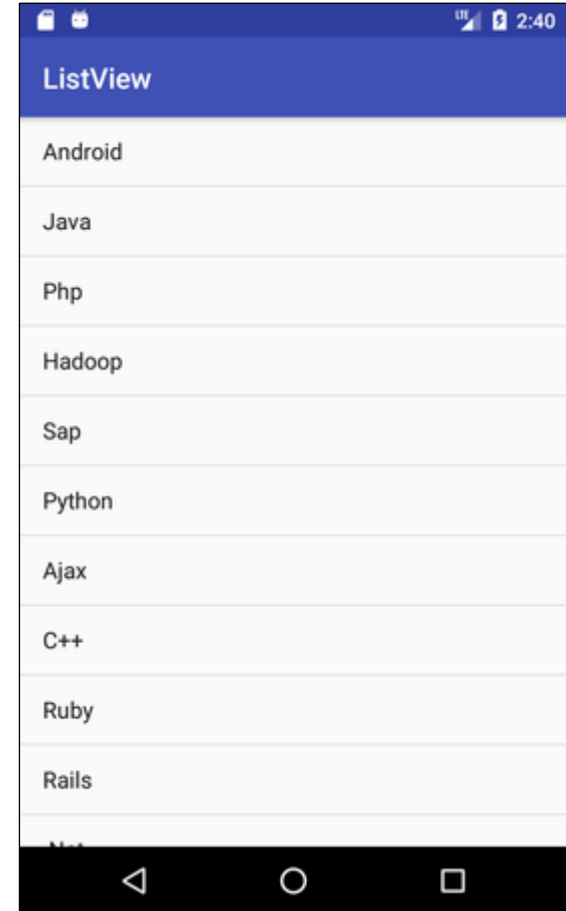


2. Gán cho ListView một Adapter là nguồn cấp dữ liệu. Xây dựng một Adapter cho ListView, sau khi có Adapter cần thiết lập cho ListView, ví dụ trong onCreate của Activity:

```
listViewProduct = findViewById(R.id.listproduct);  
listViewProduct.setAdapter(productListViewAdapter)
```

3. Khi đã gán Adapter vào ListView, thì ListView sẽ dùng Adapter này xác định cần hiện thị bao nhiêu phần tử, mỗi phần tử có View như thế nào do Adapter tạo và đính vào ListView, mỗi khi dữ liệu Adapter quản lý thay đổi, cần thông báo cho ListView biết mà cập nhật bằng cách gọi:

```
adapter.notifyDataSetChanged();
```



Các bước tạo ListView

- Tạo Listview trong Layout
- Tạo Array hoặc ArrayList chứa dữ liệu

```
String[] arr = new String[]{"Apple", "Samsung", "LG"}
```

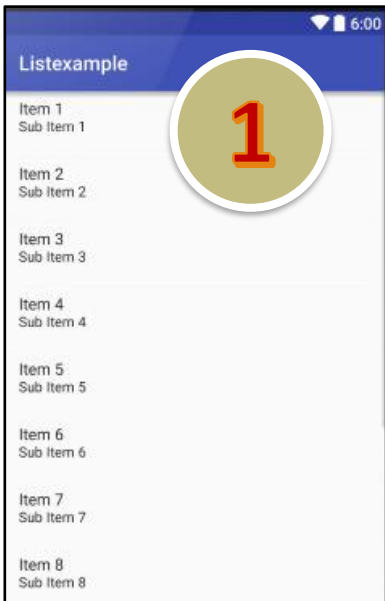
- Tạo Adapter ràng buộc nội dung động vào Array

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>  
(this, android.R.layout.simple_list_item_1, arr);
```

- Show dữ liệu trong mảng lên ListView

```
ListView.setAdapter(adapter)
```


Các bước tạo ListView cơ bản



Các bước tạo ListView cơ bản

1

```
<ListView
    android:id="@+id/lvMonhoc"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:layout_editor_absoluteX="1dp"
    tools:layout_editor_absoluteY="1dp" />
```

2

```
ArrayList<String> arrayMonhoc;
arrayMonhoc = new ArrayList<>();
arrayMonhoc.add("Android");
arrayMonhoc.add("Java");
arrayMonhoc.add("PHP");
arrayMonhoc.add("Hadoop");
arrayMonhoc.add("Sap");
arrayMonhoc.add("Python");
arrayMonhoc.add("Ajax");
arrayMonhoc.add("C++");
arrayMonhoc.add("Ruby");
arrayMonhoc.add("Rails");
```

3

```
ArrayAdapter adapter = new ArrayAdapter(MainActivity.this,
    android.R.layout.simple_list_item_1, arrayMonhoc);
```

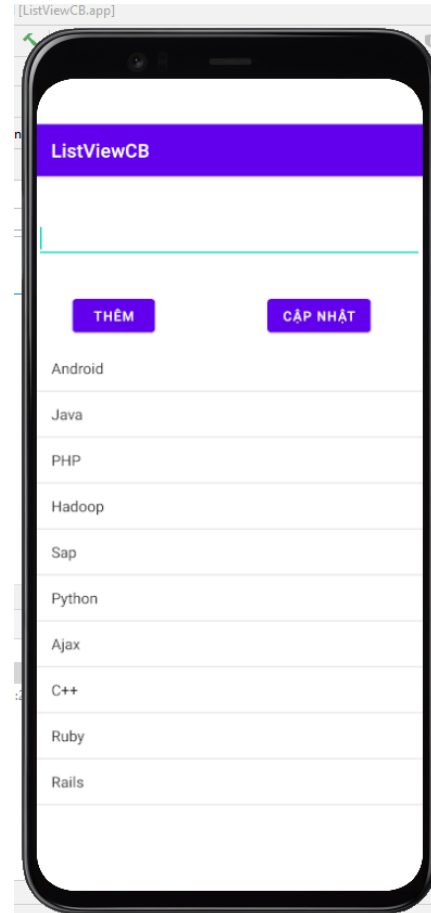
4

```
lvMonhoc.setAdapter(adapter);
```



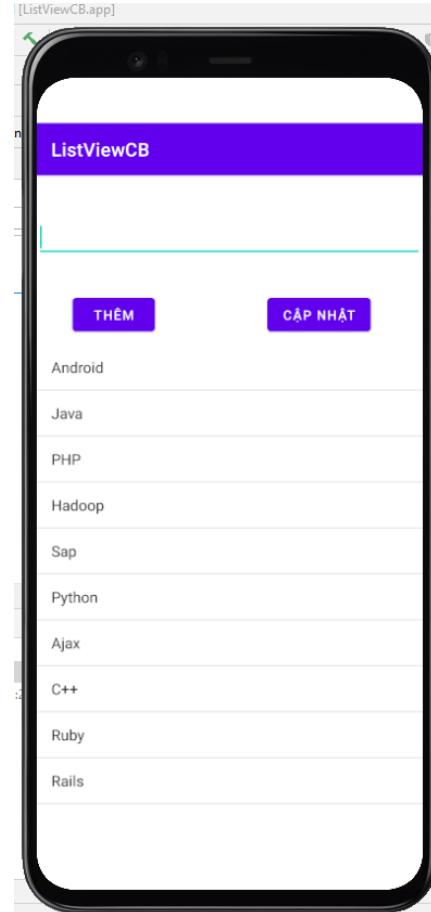
```
btThem.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        arrayMonhoc.add(edMonhoc.getText().toString());  
        adapter.notifyDataSetChanged();  
        edMonhoc.setText("");  
    }  
});
```

```
btCapnhat.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
  
        if(pos != -1) {  
            arrayMonhoc.set(pos, edMonhoc.getText().toString());  
            adapter.notifyDataSetChanged();  
        }  
        pos = -1;  
        edMonhoc.setText("");  
    }  
});
```

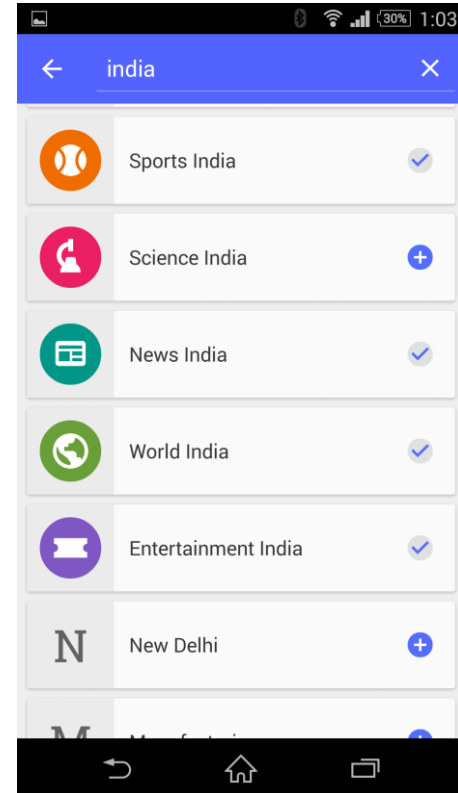
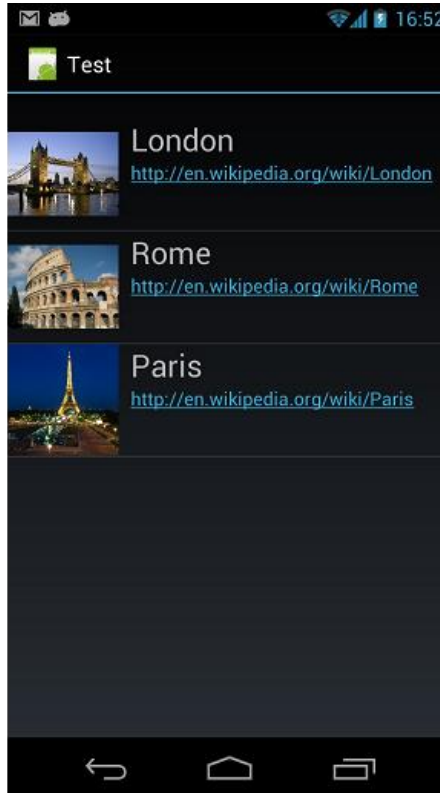


```
lvMonhoc.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int position, long id)  
    {  
        edMonhoc.setText(arrayMonhoc.get(position));  
        pos = position;  
    }  
});
```

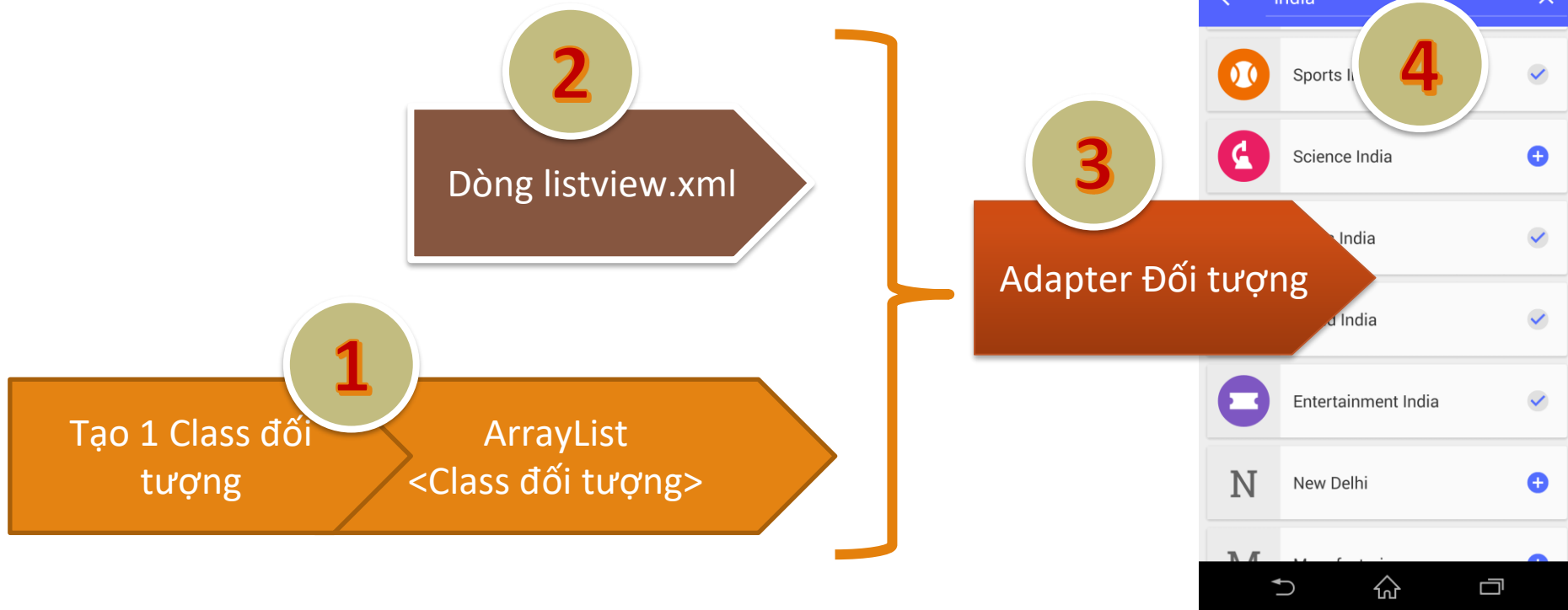
```
lvMonhoc.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {  
    @Override  
    public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {  
        Toast.makeText(MainActivity.this, "âsâs", Toast.LENGTH_LONG).show();  
        return false;  
    }  
});
```



Listview nâng cao:



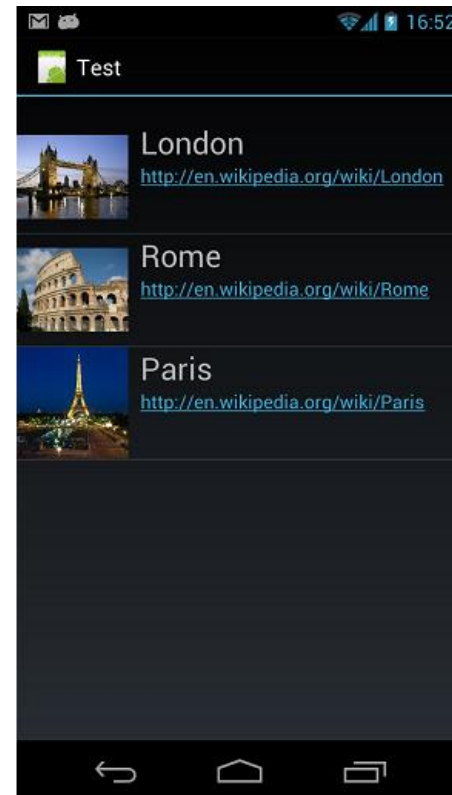
Các bước tạo ListView nâng cao



Các bước tạo ListView nâng cao

1

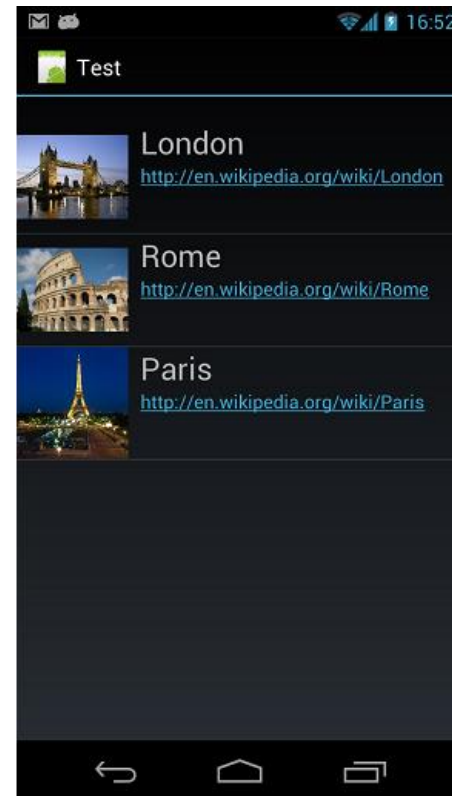
```
public class City {  
    private String NameCity;  
    private int Hinh;  
    private String linkWiki;  
  
    public City(String nameCity, int hinh, String linkWiki) {  
        NameCity = nameCity;  
        Hinh = hinh;  
        this.linkWiki = linkWiki;  
    }  
    getter/setter  
}
```



Các bước tạo ListView nâng cao

1

```
ArrayList<City> cityArrayList;  
cityArrayList.add(new City("New York",R.drawable.newyork,"http://1.com"));  
cityArrayList.add(new City("Paris",R.drawable.paris,"http://2.com"));  
cityArrayList.add(new City("Saigon",R.drawable.saigon,"http://3.com"));
```



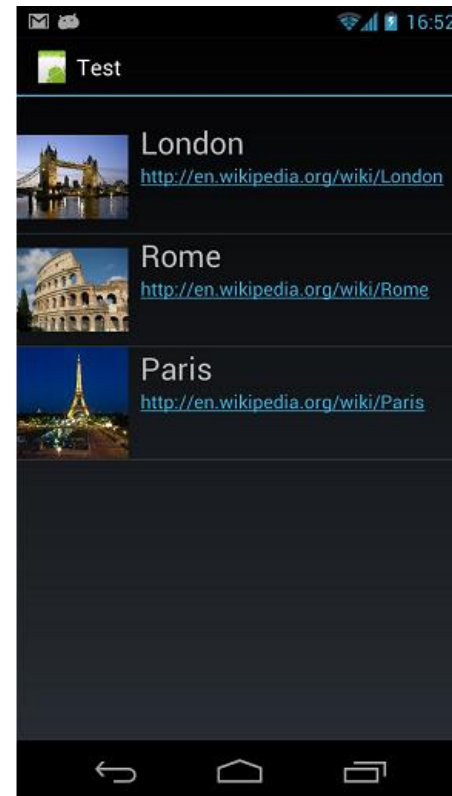
Các bước tạo ListView nâng cao

2

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:id="@+id/imgHinh"
        android:layout_width="120dp"
        android:layout_height="120dp"
        android:padding="10dp"/>

    <TextView
        android:id="@+id/txtTen"
        android:textSize="24dp"
        android:layout_toRightOf="@+id/imgHinh"
        android:text="Tên"
        android:textColor="#E91E63"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

    <TextView
        android:textColor="#3F51B5"
        android:layout_toRightOf="@+id/imgHinh"
        android:layout_below="@+id/txtTen"
        android:id="@+id/txtlink"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="link wiki"/>
</RelativeLayout>
```



```
public class CityAdapter extends BaseAdapter {
```

```
    private Context context;
```

```
    private int layout;
```

```
    private List<City> cityList;
```

```
    public CityAdapter(Context context, int layout, List<City> cityList) {
```

```
        this.context = context;
```

```
        this.layout = layout;
```

```
        this.cityList = cityList;
```

```
    }
```

```
    @Override
```

```
    public int getCount() {
```

```
        return cityList.size();
```

```
    }
```

```
    @Override
```

```
    public Object getItem(int position) {
```

```
        return null;
```

```
    }
```

```
    @Override
```

```
    public long getItemId(int position) {
```

```
        return 0;
```

```
    }
```

```
    @Override
```

```
    public View getView(int position, View convertView, ViewGroup parent) {
```

```
        LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
```

```
        convertView = inflater.inflate(layout, null);
```

```
        TextView txtTen = convertView.findViewById(R.id.txtTen);
```

```
        TextView txtlin = convertView.findViewById(R.id.txtlink);
```

```
        ImageView imgHinh = convertView.findViewById(R.id.imgHinh);
```

```
        City city = cityList.get(position);
```

```
        txtTen.setText(city.getNameCity());
```

```
        txtlin.setText(city.getLinkWiki());
```

```
        imgHinh.setImageResource(city.getHinh());
```

```
        return convertView;
```

```
    }
```

```
}
```

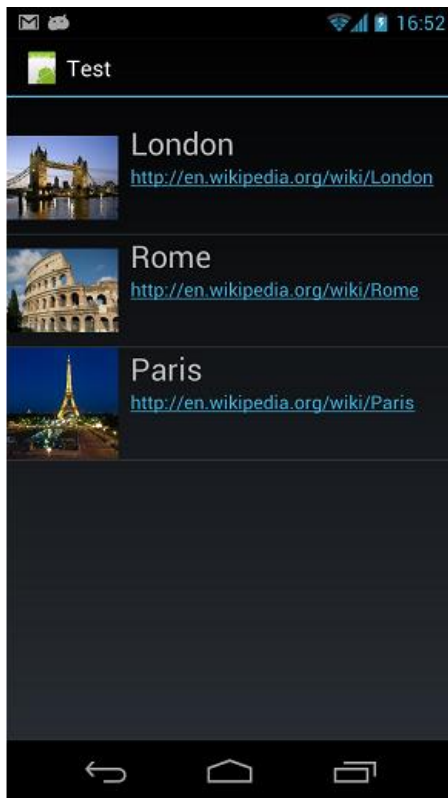
3

LISTVIEW

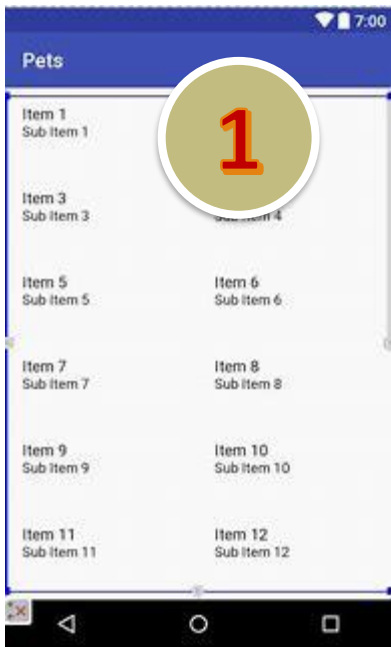


Các bước tạo ListView nâng cao

```
ListView lvCity;  
ArrayList<City> cityArrayList;  
CityAdapter adapter;  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    lvCity = findViewById(R.id.lvCity);  
    cityArrayList.add(new City("New York", R.drawable.newyork, "http://1.com"));  
    cityArrayList.add(new City("Paris", R.drawable.paris, "http://2.com"));  
    cityArrayList.add(new City("Saigon", R.drawable.saigon, "http://3.com"));  
    adapter = new CityAdapter(this, R.layout.dong_thanh_pho, cityArrayList);  
    lvCity.setAdapter(adapter);  
}
```



Các bước tạo GridView cơ bản



Các bước tạo GridView cơ bản

1

```
<GridView
    android:textAlignment="center"
    android:numColumns="4"
    android:id="@+id/gvTen"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:layout_editor_absoluteX="1dp"
    tools:layout_editor_absoluteY="1dp" />
```

2

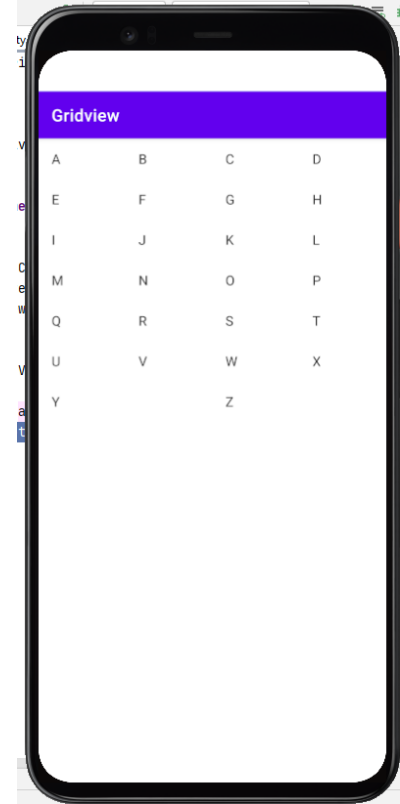
```
String[] arrayName =
{"A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N",
 "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "", "Z"};
```

3

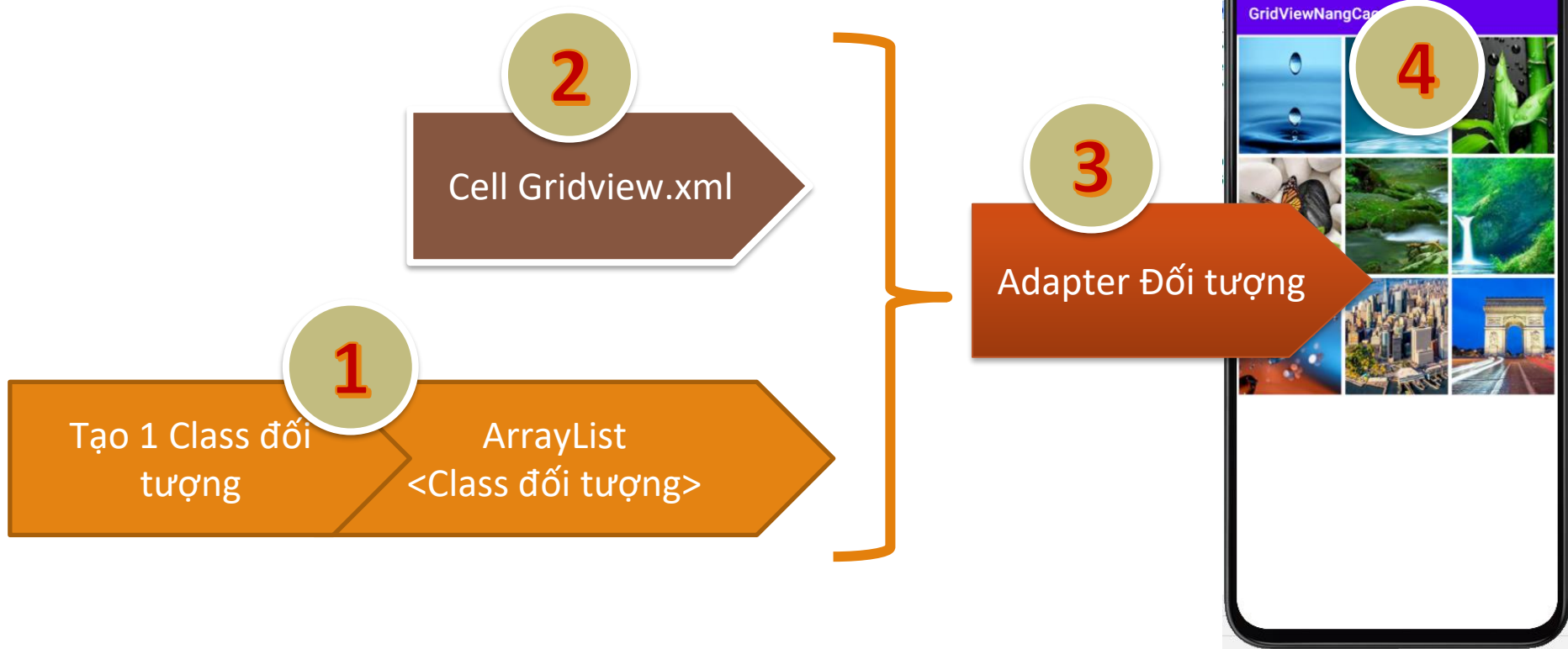
```
ArrayAdapter adapter = new ArrayAdapter(this,
    android.R.layout.simple_list_item_1, arrayName);
```

4

```
gvTen.setAdapter(adapter);
```



Các bước tạo GridView nâng cao



Các bước tạo GridView nâng cao

1

```
public class HinhAnh {  
  
    private int Hinh;  
    private String Ten;  
  
    public HinhAnh(int hinh, String ten) {  
        Hinh = hinh;  
        Ten = ten;  
    }  
    getter/setter  
}
```



Các bước tạo GridView nâng cao

1

```
arrayHinhAnh = new ArrayList<>();  
arrayHinhAnh.add(new HinhAnh(R.drawable.android1, "Hình số 1"));  
arrayHinhAnh.add(new HinhAnh(R.drawable.android2, "Hình số 2"));  
arrayHinhAnh.add(new HinhAnh(R.drawable.android3, "Hình số 3"));  
arrayHinhAnh.add(new HinhAnh(R.drawable.android4, "Hình số 4"));  
arrayHinhAnh.add(new HinhAnh(R.drawable.android5, "Hình số 5"));  
arrayHinhAnh.add(new HinhAnh(R.drawable.android6, "Hình số 6"));  
arrayHinhAnh.add(new HinhAnh(R.drawable.android7, "Hình số 7"));  
arrayHinhAnh.add(new HinhAnh(R.drawable.android8, "Hình số 8"));  
arrayHinhAnh.add(new HinhAnh(R.drawable.android9, "Hình số 9"));
```



Các bước tạo GridView nâng cao

2

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <ImageView
    android:id="@+id/imgHinhAnh"
    android:layout_width="150dp"
    android:layout_height="150dp"/>
</LinearLayout>
```



```

public class HinhAnhAdapter extends BaseAdapter {

    private Context context;
    private int layout;
    private List<HinhAnh> hinhAnhList;
    public HinhAnhAdapter(Context context, int layout, List<HinhAnh> hinhAnhList) {
        this.context = context;
        this.layout = layout;
        this.hinhAnhList = hinhAnhList;
    }
    @Override
    public int getCount() {
        return hinhAnhList.size();
    }
    @Override
    public Object getItem(int position) {
        return null;
    }
    @Override
    public long getItemId(int position) {
        return 0;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        convertView =inflater.inflate(layout,null);
        ImageView imageView = convertView.findViewById(R.id.imgHinhAnh);
        HinhAnh hinhAnh= hinhAnhList.get(position);
        imageView.setImageResource(hinhAnh.getHinh());
        return convertView;
    }
}

```



Các bước tạo GridView nâng cao

```
GridView gvHinhAnh;  
ArrayList<HinhAnh> arrayHinhAnh;  
HinhAnhAdapter adapter;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    gvHinhAnh = (GridView) findViewById(R.id.gvHinhAnh);  
    adapter = new HinhAnhAdapter(this, R.layout.dong_hinh_anh, arrayHinhAnh);  
    gvHinhAnh.setAdapter(adapter);  
}
```



Tài nguyên Drawable - Tài nguyên **Drawable** là một khái niệm chung ám chỉ đối tượng đồ họa nào đó có thể được vẽ ra trên màn hình.

- Trong ứng dụng Android với Android Studio thường các tài nguyên đồ họa này đặt trong thư mục **res/drawable**, sau đó nó được sử dụng bằng cách lấy ra từ trong code bằng các hàm API như **ContextCompat.getDrawable**, **ResourcesCompat.getDrawable** hay gán vào thuộc tính phù hợp trong XML như **android:drawable**, **android:icon**, **android:button** ...
- Lớp cơ sở biểu diễn Drawable là lớp **Drawable** (**android.graphics.drawable.Drawable**), từ lớp này mà có các kiểu Drawable khác nhau như:

Truy cập Drawable:

- Truy cập đến Drawable trong XML : "**@drawable/filename**"
- Truy cập đến Drawable trong code:

1. Lấy Drawable bằng ContextCompat:

Drawable drawable = ContextCompat.getDrawable(context,id)

context Là một đối tượng Context như Activity hoặc lấy từ các view: **getContext()**

id là ID trỏ đến Drawable, sử dụng lớp R để truy cập như **R.drawable.example**

2. Lấy Drawable bằng ResourcesCompat:

Drawable drawable = ResourcesCompat.getDrawable(res, id, null)

res là đối tượng Resources, lấy được bằng phương thức **getResources()** của View hay Activity

id là ID trỏ đến Drawable, sử dụng lớp R để truy cập như **R.drawable.example**

DRAWABLE SHAPE

```
<?xml version="1.0" encoding="utf-8"?>
<shape
xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">

    <solid android:color="@color/purple_200"/>
    <size android:height="50dp" android:width="100dp"/>

    <corners android:bottomLeftRadius="50dp"
        android:topLeftRadius="50sp"
        android:topRightRadius="50dp"
        android:bottomRightRadius="50dp"/>
    <gradient android:angle="90"

        android:startColor="@color/purple_200"
        android:endColor="#FF0000"/>

    <stroke android:color="@color/black" android:width="1dp" />
</shape>
```

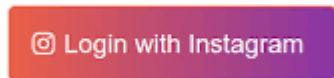
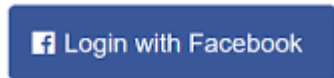
```
<Button
    android:id="@+id/button"

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="128dp"
    android:textSize="30dp"
    android:background="@drawable/custom_button"
    android:drawableLeft="@drawable/ic_untitled_1"
    android:text="Facebook"
    android:textAllCaps="false"
    app:backgroundTintMode="screen"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

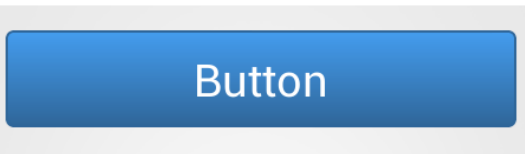
```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="40dp"
    android:height="40dp"
    android:viewportWidth="135"
    android:viewportHeight="135">
    <path
        android:pathData="M93.77,87.05l3,-19.51l78.05,67.54l78.05,54.83c0,-5.34 2.61,-
10.54 11,-10.54h8.51l97.56,27.68a103.8,103.8 0,0 0,-15.11 -1.32c-15.42,0 -25.5,9.34 -
25.5,26.26v14.87l39.83,67.49l39.83,87.05h17.14v47.17a68.21,68.21 0,0 0,21.09
0l78.06,87.05Z"
        android:fillColor="#FFFFFF"/>
</vector>
```



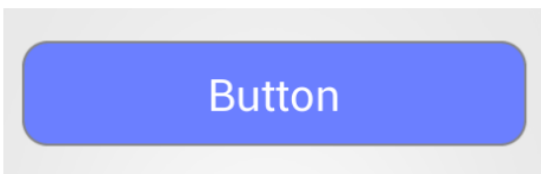
DRAWABLE SHAPE



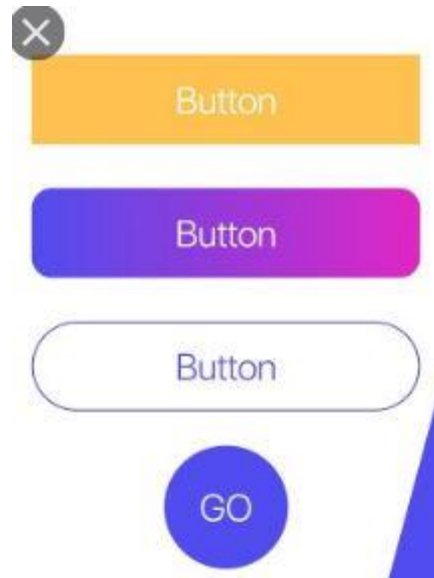
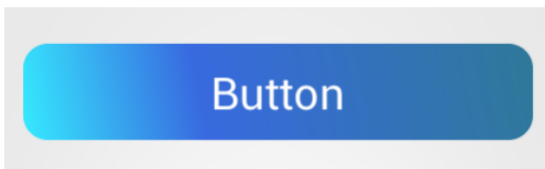
State : Default



State : focused



State : Pressed



Tài nguyên StateList Drawable <selector>:

StateListDrawable là loại Drawable mà nó chứa nhiều phần tử <item> bên trong, mỗi phần tử này biểu diễn một Drawable, chúng biểu diễn một vài trạng thái nào đó của View khi làm đối tượng đồ họa cụ thể trong từng loại View như làm nền, làm biểu tượng lựa chọn ...

Cú pháp xml tạo ra StateListDrawable như sau:

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

  <item>
    <!--Drawable của item-->
  </item>
  <!-- ... -->
</selector>
```

Thuộc tính của <item> trong LayerDrawable

<item> (lớp) của LayerDrawable bên trong nó có chứa phần tử con là bất kỳ loại Drawable nào đã biết như bitmap, vector ... và có thể thiết lập các thuộc tính như sau:

Thuộc tính	Ý nghĩa
android:state_pressed android:state_checked android:state_selected android:state_activated android:state_focused android:state_active android:state_enabled	<p>Thiết lập bằng true thì Drawable của Item biểu diễn trạng thái tương ứng. Nếu thiết lập bằng false biểu diễn không kích hoạt trạng thái có tên tương ứng. Nếu không thiết lập thì không có diễn tả gì cho trạng thái đó.</p> <p>Ví dụ:</p> <ul style="list-style-type: none">❖ <code>android:state_pressed="true"</code> Drawable biểu thị trạng thái bị nhấn❖ <code>android:state_pressed="false"</code> Drawable biểu thị trạng thái không bị nhấn❖ Không có thiết lập <code>android:state_pressed</code> thì khi nhấn / không nhấn không sử dụng đến Drawable này

DRAWABLE STATELIST

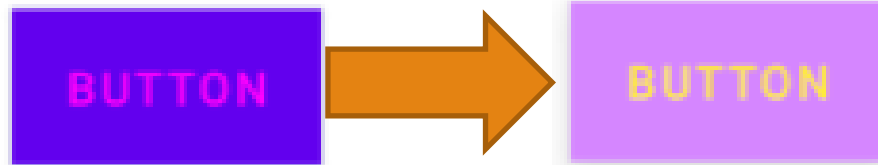
```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:drawable="@color/purple_200"
    android:state_pressed="true"
  />
  <item android:drawable="@color/black" />
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">

  <item android:state_pressed="true"
    android:color="#FFEB3B"/>

  <item android:drawable="@color/white"
  />
</selector>
```

```
<Button
  android:id="@+id/button"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_marginTop="100dp"
  android:background="@drawable/custom_button"
  android:textColor="@drawable/custom_color"
  android:text="Button"
  app:backgroundTintMode="screen"
  app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintHorizontal_bias="0.498"
  app:layout_constraintStart_toStartOf="parent"
  app:layout_constraintTop_toTopOf="parent" />
```



Phần I: Layout của ứng dụng Android

 RelativeLayout

 LinearLayout

Phần II: Các widget cơ bản

 TextView

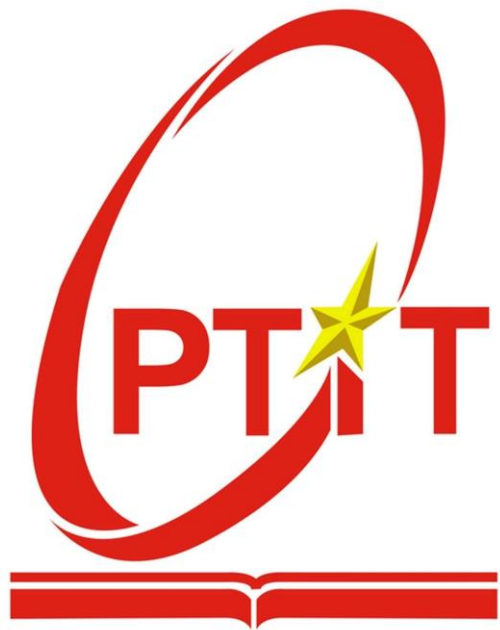
 EditText

 Button

 Image

 ListView





ptit.edu.vn