

# **Bài 03:**

## **GIỚI THIỆU CÁC THÀNH PHẦN ỨNG DỤNG CỦA ANDROID (TIẾP THEO)**

GVGD: ThS. Đặng Thế Hân

Biên soạn: ThS. Giang Hào Côn

# Mục tiêu

Cung cấp cho sinh viên kiến thức cơ bản về các thành phần ứng dụng của Android, và các kiểu lập trình sự kiện trong ngôn ngữ lập trình ứng dụng trên Android.

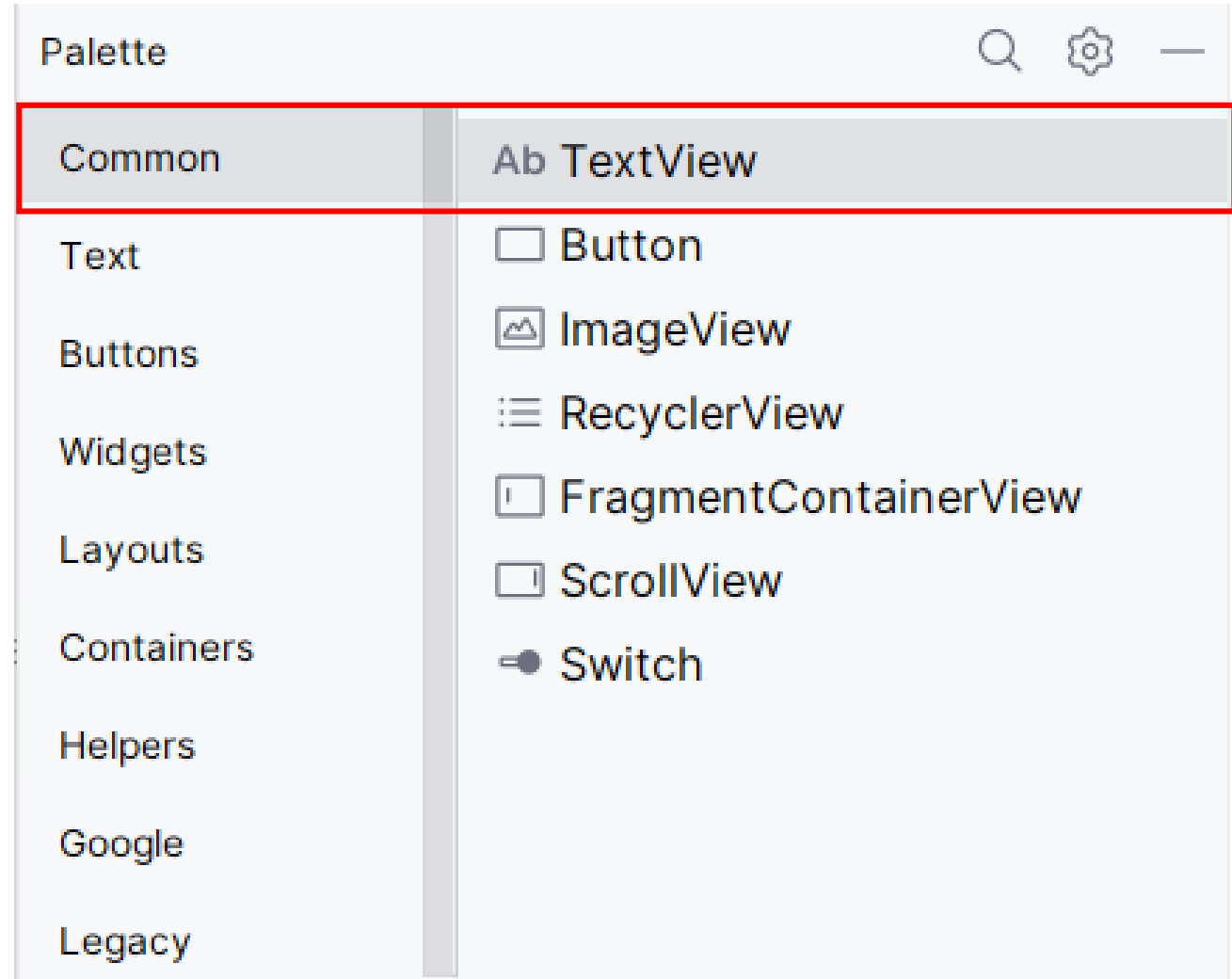
# Nội dung

- 1) Sử dụng TextView.
- 2) Sử dụng EditText.
- 3) Sử dụng Button.
- 4) Sử dụng Toast Notification.
- 5) Sử dụng Alert Dialog

# 3.1/ Sử dụng TextView, EditText, Button

## 1) TextView

**TextView** là view dùng để hiển thị dữ liệu không cho phép người sử dụng thay đổi dữ liệu. **TextView** thuộc nhóm Common và Text trong công cụ Palette:



# 3.1/ Sử dụng TextView, EditText, Button

## 1) TextView – Các thuộc tính quan trọng của TextView

Các thuộc tính	Mô tả
android:id	Id của TextView
android:layout_width	Độ rộng
android:layout_height	Độ cao
android:text	Chữ hiển thị lên giao diện
android:textColor	Màu chữ
android:textSize	Cỡ chữ
android:background	Màu nền
android:hint	Chữ gợi ý khi android:text rỗng

## 3.1/ Sử dụng TextView, EditText, Button

### 1) TextView – Truy xuất TextView

TextView **<variable\_name>** = findViewById(**R.id.<id\_name>**);

Cấu trúc XML của TextView:

```
<TextView  
    android:id="@+id/txtMessage"  
    android:layout width="match parent"
```

Để truy xuất TextView này, trong Java code (.java):

```
TextView txtMessage = findViewById(R.id.txtMessage);
```

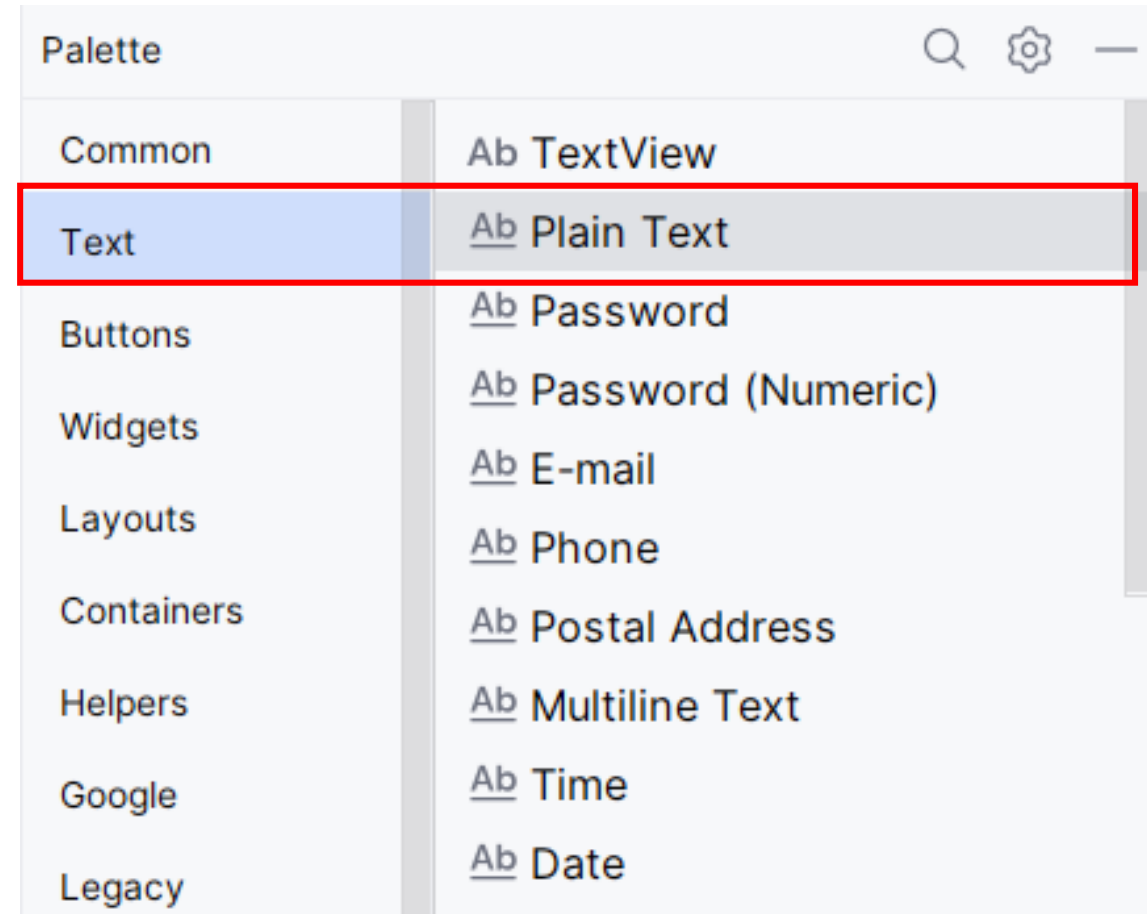
```
//Thiết lập nội dung hiển thị cho TextView
```

```
txtMessage.setText("Android chào các bạn");
```

# 3.1/ Sử dụng TextView, EditText, Button

## 2) EditText

**EditText** được dùng thay đổi nội dung text, chứa tất cả thuộc tính của TextView nên TextView hoạt động như thế nào thì EditText hoạt động như vậy, nằm trong nhóm Text của Palette. Đặt Id nên bắt đầu là edt hoặc txt



# 3.1/ Sử dụng TextView, EditText, Button

## 2) EditText - Các thuộc tính quan trọng

Các thuộc tính	Mô tả
android:id	Id của EditText
android:layout_width	Độ rộng
android:layout_height	Độ cao
android:text	Chữ hiển thị lên giao diện
android:textColor	Màu chữ
android:textSize	Cỡ chữ
android:background	Màu nền
android:hint	Chữ gợi ý khi android:text rỗng
android:inputType	Thiết lập loại dữ liệu nhập để có bàn phím phù hợp



## 3.1/ Sử dụng TextView, EditText, Button

### 2) EditText - Cấu trúc XML của EditText

```
<EditText  
    android:id="@+id/edtPassword"
```

Để truy suất ta cũng dùng hàm findViewById như sau:

```
EditText matkhau=findViewById(R.id.edtPassword);
```

Để thay đổi nội dung hiển thị ta dùng:

```
matkhau.setText("obama")
```

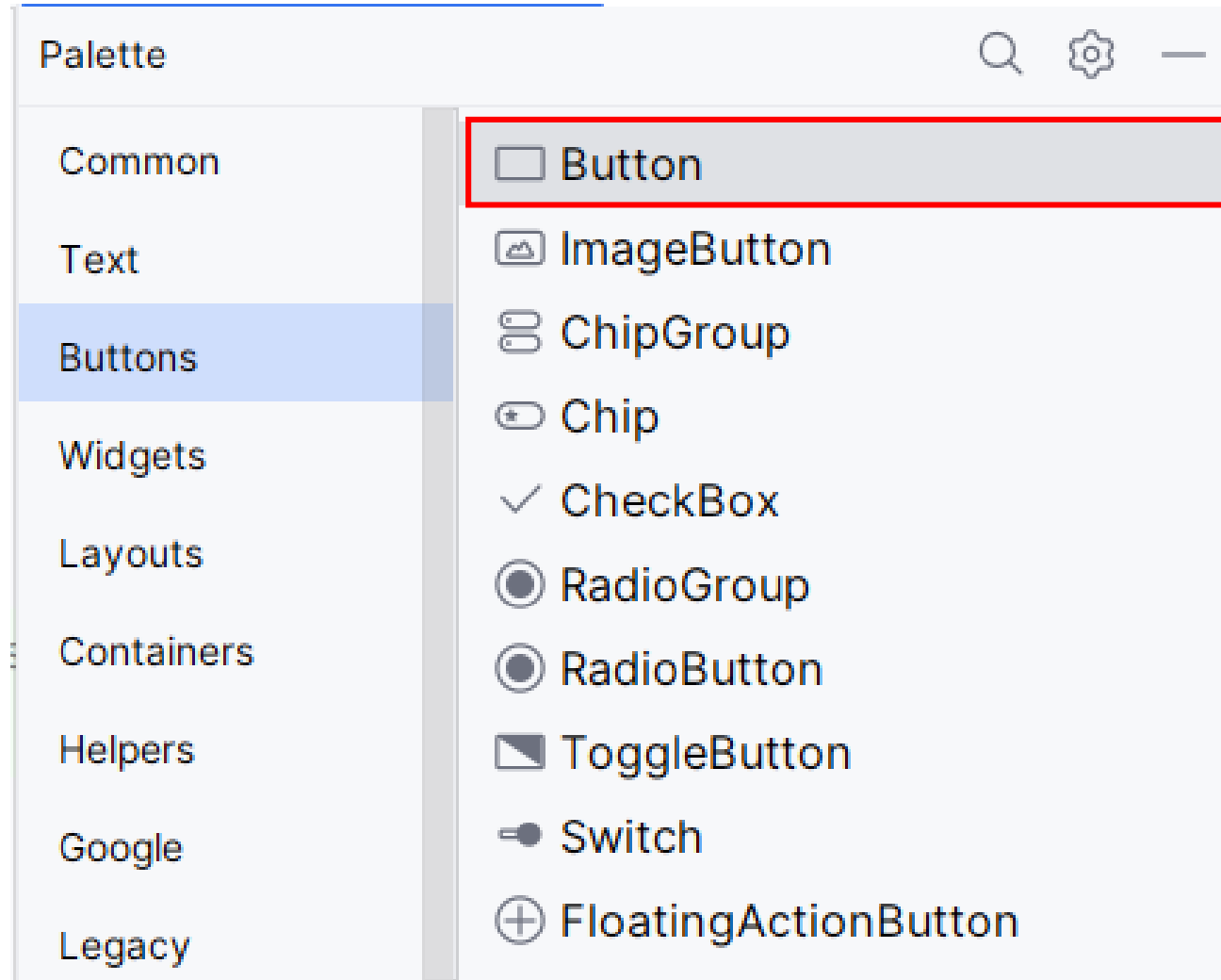
Để lấy nội dung người dùng nhập liệu:

```
matkhau.getText().toString()
```

# 3.1/ Sử dụng TextView, EditText, Button

## 3) Button

**Button** là một View nhưng chúng có thể cho phép ta **tương tác bằng cách bấm vào nó**. Nhưng trước khi ta muốn button nhận biết sự kiện click thì các ta phải bắt sự kiện click cho button đó.



# 3.1/ Sử dụng TextView, EditText, Button

## 3) Button – Các thuộc tính của Button

Các thuộc tính	Mô tả
android:id	Id của EditText
android:layout_width	Độ rộng
android:layout_height	Độ cao
android:text	Chữ hiển thị lên giao diện
android:textColor	Màu chữ
android:textSize	Cỡ chữ
android:background	Màu nền

## 3.1/ Sử dụng TextView, EditText, Button

### 3) Button – Cấu trúc XML của Button

<Button

android:id="@+id/btnDangNhap"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Đăng Nhập"

/>

# 3.1/ Sử dụng TextView, EditText, Button

## 3) Button – Các cách bắt sự kiện click của button

Giao diện có một nút lệnh như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<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:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/bnt_click"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Thực Hiện" />

</LinearLayout>
```

## 3.1/ Sử dụng TextView, EditText, Button

### 3) Button – Các cách bắt sự kiện click của button

Để truy xuất button ta phải tìm id của button trên giao diện XML ánh xạ vào biến button.

```
public class MainActivity extends AppCompatActivity {  
  
    Button button; //Khai báo một button  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Lấy id của Button trên giao diện XML gán cho biến button  
        button = findViewById(R.id.bnt_click);  
    }  
}
```

# 3.1/ Sử dụng TextView, EditText, Button

## 3) Button – Bắt trực tiếp

```
public class MainActivity extends AppCompatActivity {  
  
    Button button; //Khai báo một button  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        button = findViewById(R.id.bnt_click);  
  
        button.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                Toast.makeText(MainActivity.this, "Bạn đã click", Toast.LENGTH_SHORT).show();  
            }  
        });  
    }  
}
```



# 3.1/ Sử dụng TextView, EditText, Button

## 3) Button – Bắt trong giao diện xml

```
<?xml version="1.0" encoding="utf-8"?>
<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:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/bnt_click"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click"
        android:onClick="ThongBao" />

</LinearLayout>
```

Trong file .java  
phải có method:

```
public void ThongBao(View view) {

    //Code here

}
```



# 3.1/ Sử dụng TextView, EditText, Button

## 3) Button – Bắt sự kiện click cho nhiều nút

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener{

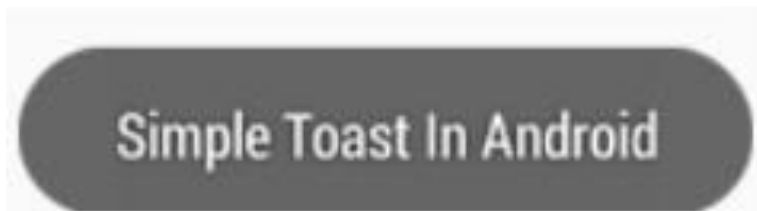
    Button button; //Khai báo một button
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        button = findViewById(R.id.bnt_click); //Tìm lại button
        button.setOnClickListener(this);
    }

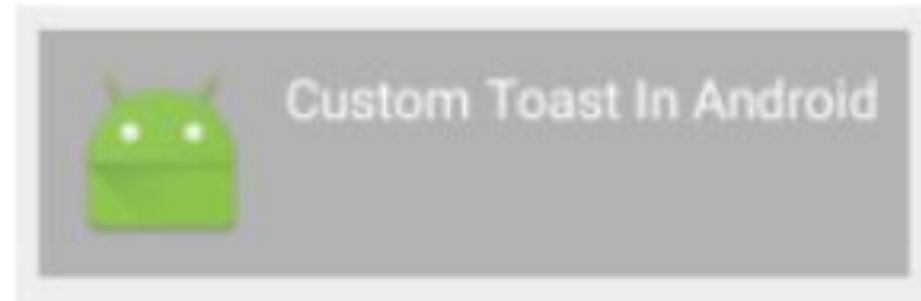
    @Override
    public void onClick(View view) {
        if(view.getId() == R.id.bnt_click){
            Toast.makeText(MainActivity.this, "Bạn đã click", Toast.LENGTH_SHORT).show();
        }
    }
}
```

## 3.2/ Toast

**Toast** là một loại thông báo (message) ngắn. Để hiển thị trong một khoảng thời gian nhất định và tự động mất dần. Nó giúp người lập trình gỡ rối (debugging) cho chương trình khi cần thiết. Trong lúc hiển thị, không ảnh hưởng đến activity khác và không bắt các sự kiện của người dùng.



**Toast**



**Custom Toast With Image**

## 3.2/ Toast

### Các Phương thức quan trọng của Toast.

1. `makeText(Context context, CharSequence text, int duration)`:  
Phương thức này thường sử dụng để hiển thị thông báo.
  - a) **Context context**: thường là `YourActivity.this` của bạn vào nếu như bạn đang sử dụng ở `Activity`, nếu như bạn sử dụng ở `Fragment` thì điền vào đó là `getActivity()`.
  - b) **CharSequence text**: đây chính là nội dung bạn muốn hiển thị.
  - c) **int duration**: Khoảng thời gian Toast cần hiển thị, nó là hằng số.  
Hằng số của Toast: `LENGTH_LONG` (Toast sẽ hiển thị trong 3,5 giây), `LENGTH_SHORT` (Toast sẽ hiển thị trong 2 giây).

## 3.2/ Toast

### Các Phương thức quan trọng của Toast.

#### 2. show():

Phương thức này hiển thị thông báo ra màn hình. Ví dụ:

```
Toast toast = Toast.makeText(getApplicationContext(), "Ví dụ về Toast", Toast.LENGTH_LONG);  
toast.show(); // Hiển thị Toast
```

Hiển thị:



Ví dụ về Toast

## 3.2/ Toast

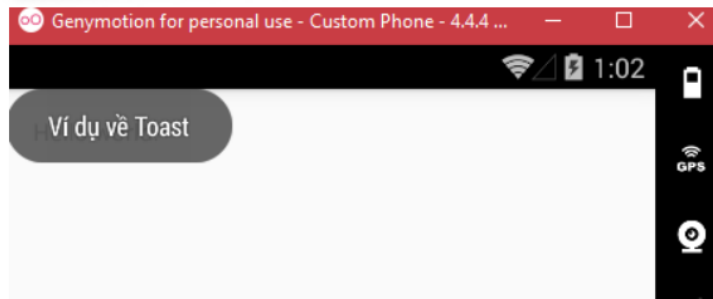
### Các Phương thức quan trọng của Toast.

#### 3. `setGravity(int,int,int)`:

Mặc định Toast sẽ hiển thị ở dưới cùng của màn hình theo chiều ngang, nhưng bạn có thể thay đổi vị trí này bằng cách `setGravity(int, int, int)`. Ví dụ:

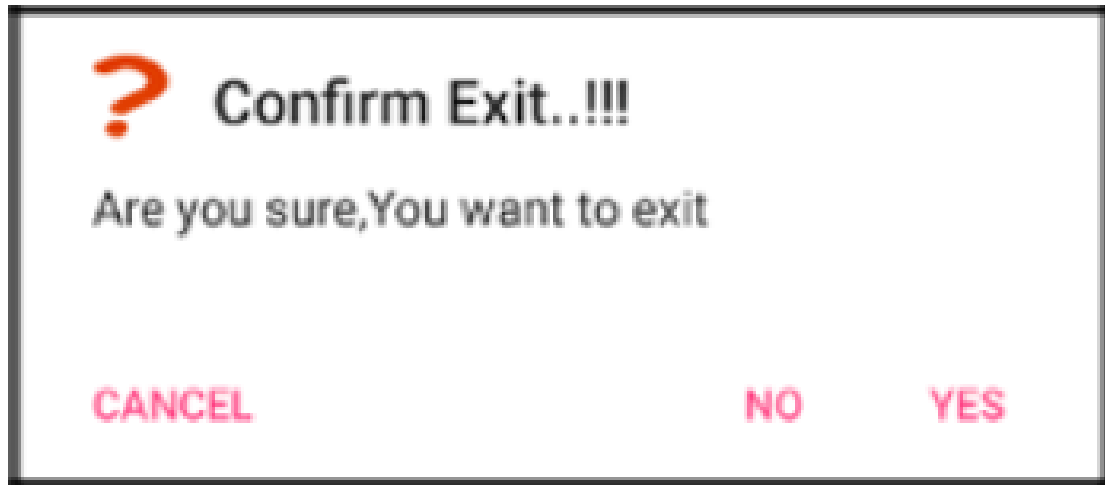
```
Toast toast = Toast.makeText(getApplicationContext(), "Ví dụ về Toast", Toast.LENGTH_LONG);  
toast.setGravity(Gravity.TOP | Gravity.LEFT, 0, 0);    // set gravity for the Toast.  
toast.show(); // Hiển thị Toast
```

Hiển thị:



## 3.3/ Alert Dialog

- **Dialog** có thể coi là một thông báo mà người dùng có thể tương tác trực tiếp được. Ví dụ khi các bạn muốn xóa một tập tin quan trọng hay muốn thoát một chương trình nào đấy thì việc hiển thị một thông báo để người dùng chắc chắn về hành vi của mình là rất quan trọng.



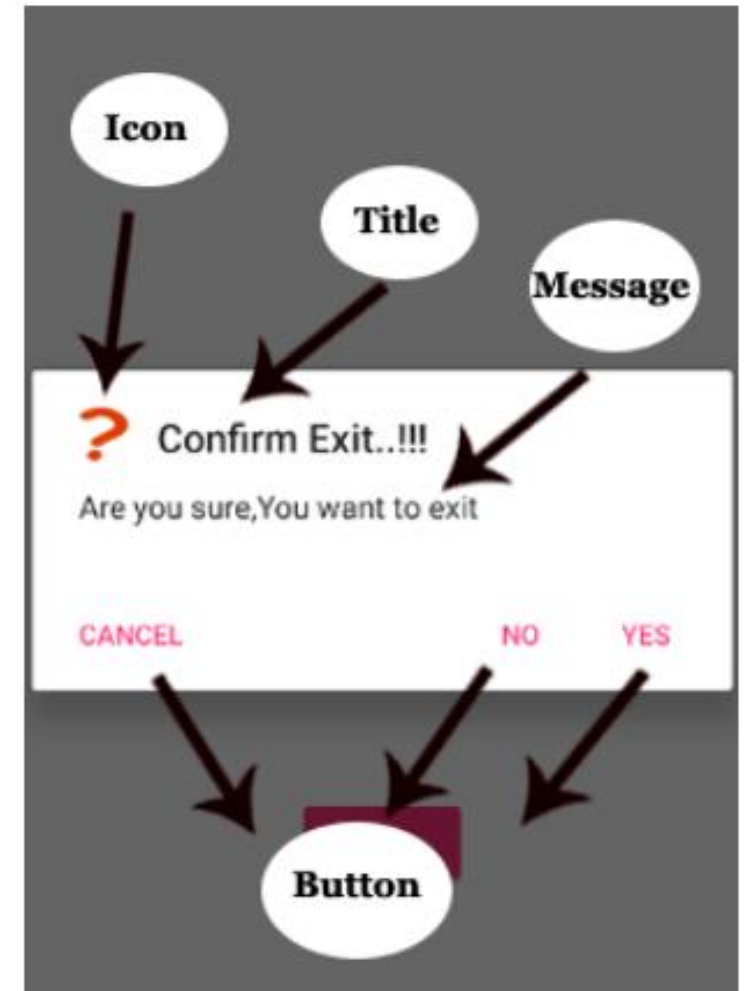
**Android AlertDialog** gồm có 3 vùng: Tiêu đề (title), nội dung(content) và các button hành động.

## 3.3/ Alert Dialog

**Để tạo Alert Dialog thường sử dụng thành phần AlertDialog.Builder.**

AlertDialog.Builder được sử dụng để tạo ra một giao diện là một hộp thoại cảnh báo trong Android. Thường thì chúng ta phải thiết lập các vùng: Tiêu đề, nội dung, image, button và xử lý sự kiện cho button.

```
AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
```





## 3.3/ Alert Dialog

### Cách phương thức thường dùng trong Alert

1. `setTitle(CharSequence title)` – Phương thức này thường dùng để thiết lập tiêu đề cho Alert Dialog. Phương thức này tùy chọn.

```
//Thiết lập tiêu đề cho alert dialog  
AlertDialogBuilder.setTitle("Xác nhận để thoát..!!!");
```

2. `setIcon(Drawable icon)` – Phương thức này dùng để thêm icon trước tiêu đề. Icon phải được lưu trong thư mục drawable.

```
// Icon Of Alert Dialog  
AlertDialogBuilder.setIcon(R.drawable.question);
```



## 3.3/ Alert Dialog

### Cách phương thức thường dùng trong Alert

3. `setMessage(CharSequence message)` – Phương thức này dùng hiển thị nội dung trong Alert Dialog. Nội dung này bắt buộc phải có.

```
// Setting Alert Dialog Message  
alertDialogBuilder.setMessage("Bạn có chắc thoát?");
```

4. `setCancelable(boolean cancelable)` – Phương thức này có 2 giá trị true/false. Nếu thiết lập false thì khi show dialog lên người dùng click ra bên ngoài dialog thì nó vẫn không bị mất, nếu set true thì sẽ mất khi click vào bất kì đâu ngoài dialog.

```
alertDialogBuilder.setCancelable(false);
```

## 3.3/ Alert Dialog

### Cách phương thức thường dùng trong Alert

5. `setPositiveButton(CharSequence text, DialogInterface.OnClickListener listener)` – Phương thức này thiết lập một Button Positive (dương), mà dương thì nó nằm bên phải.

```
alertDialogBuilder.setPositiveButton("yes", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface arg0, int arg1) {  
        // Đóng activity hiện tại  
        finish();  
    }  
});
```

## 3.3/ Alert Dialog

### Cách phương thức thường dùng trong Alert

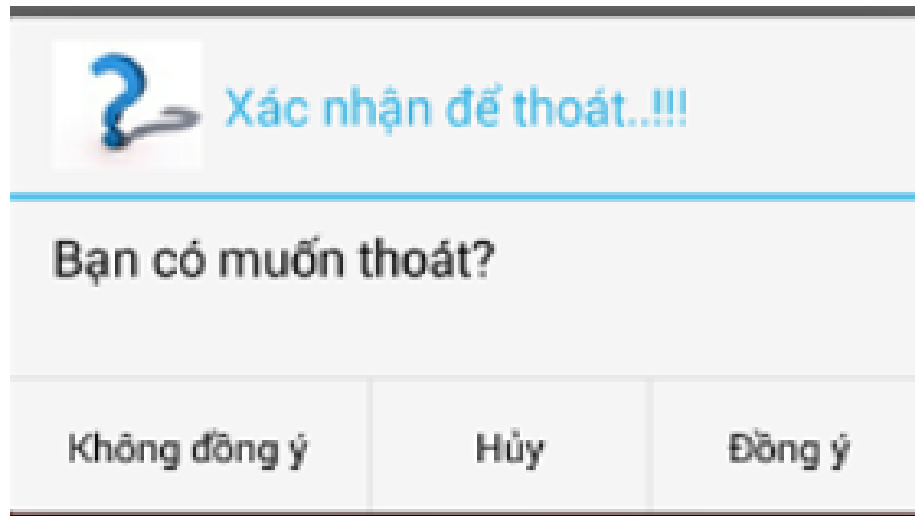
6. `setNegativeButton(CharSequence text, DialogInterface.OnClickListener listener)`  
– Phương thức này thiết lập một Button Negative (âm), mà âm thì nó nằm bên trái..

```
alertDialogBuilder.setNegativeButton("No", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialog, int which) {  
        Toast.makeText(MainActivity.this, "You clicked over No",  
            }  
});
```

## 3.3/ Alert Dialog

### Cách phương thức thường dùng trong Alert

7. `setNeutralButton(CharSequence text, DialogInterface.OnClickListener listener)` – Phương thức này đơn giản là thêm một Button mới, với button này chúng ta có thể xử lý sự kiện cho nó. Ví dụ 2 phương thức trên bạn đã thiết lập 2 Button "Yes" và "No" trong phương thức này chúng ta thiết lập Button "Cancel".



# Câu hỏi Thảo luận

- 1) Trình bày ứng dụng của TextView, EditText và Button trong lập trình giao diện Android.
- 2) Trình bày các cách bắt sự kiện cho Button trong Android
- 3) Phân biệt Toast và Alert Dialog trong ứng dụng Android