

CÁC WIDGET CƠ BẢN TRONG ANDROID

Android



Khái niệm Widget

- Trong Android, widget là một thành phần nhỏ của ứng dụng (gọi là "app widget") có thể đặt trên màn hình chính (home screen) của thiết bị để cung cấp thông tin hoặc chức năng tương tác nhanh mà không cần mở ứng dụng đầy đủ.
- Các widget cơ bản trong Android bao gồm:
 - ❖ TextView: Hiển thị văn bản đơn giản.
 - ❖ Button: Nút bấm tương tác.
 - ❖ ImageView: Hiển thị hình ảnh.
 - ❖ EditText: Ô nhập văn bản.
 - ❖ CheckBox: Hộp kiểm để chọn hoặc bỏ chọn.
 - ❖ RadioButton: Nút chọn (radio button) dùng để chọn một lựa chọn trong một nhóm.
 - ❖ ProgressBar: Hiển thị thanh tiến trình.
 - ❖ Spinner: Menu chọn từ một danh sách.
 - ❖ ListView: Hiển thị danh sách các mục theo chiều dọc.

TextView

❑ **TextView** là một View cho phép hiện thị các dòng chữ (text) trên màn hình, nó có nhiều thuộc tính tùy mục đích sử dụng mà áp dụng, như thiết lập cỡ chữ, font chữ, màu chữ,...

❑ **TextView** cung cấp cho ta một số thuộc tính:

- ❖ android:layout_width
- ❖ android:gravity
- ❖ android:text
- ❖ android:textColor
- ❖ android:textSize
- ❖ android:textAllCaps
- ❖ android:layout_height
- ❖ android:padding
- ❖ android:margin
- ❖ android:ellipsize
- ❖ android:background
- ❖ ...





TextView

- ❖ **android:gravity** – Canh chỉnh text của nó sao cho canh trái, phải, giữa,... so với không gian của chính nó.
- ❖ **android:text** – Hiển thị nội dung của văn bản cần hiển thị.
- ❖ **android:textColor** – Hiển thị màu cho text.
- ❖ **android:textSize** – Kích cỡ của text, kích cỡ này được tính theo đơn vị **sp**.
- ❖ **android:padding** – Set khoảng cách giữa biên của view đến các thành phần con của nó. Đơn vị tính của thuộc tính này cũng là dp hoặc dip. Nếu bạn muốn khoảng cách riêng cho từng cạnh biên có thể dùng tách biệt từng thuộc tính cụ thể của padding như **paddingTop**, **paddingBottom**, **paddingStart** (**paddingLeft**) và **paddingEnd** (**paddingRight**).



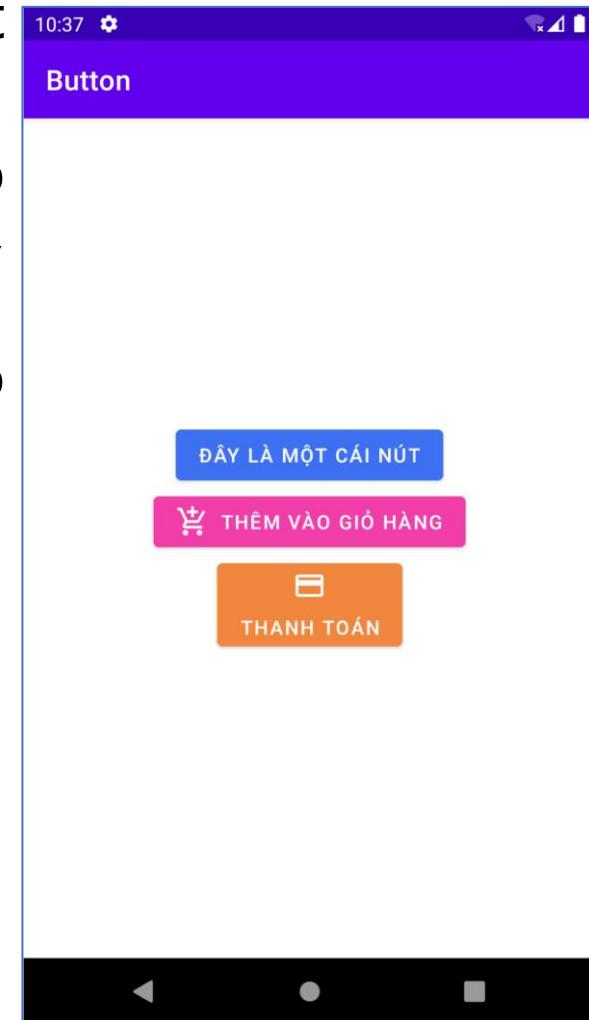
TextView

- **android:margin** – Set khoảng cách giữa biên của view đến các thành phần bên ngoài của nó. Nếu như padding giúp cho các thành phần con cách xa cạnh biên của view ra, thì margin lại giúp các thành phần bên ngoài cách xa cạnh biên của view. Đơn vị tính của thuộc tính này là dp hoặc dip. Tương tự bạn có thể set marginTop, marginBottom, marginStart (marginLeft) và marginEnd (marginRight) đối với từng cạnh của view.
- **android:ellipsize** – Dùng thuộc tính này khi bạn muốn text của mình sẽ bị cắt và hiển thị “...” khi không đủ không gian để chứa hết text đó.
- **android:background** – Dùng để set màu nền hoặc ảnh nền cho View.
- **android:textAllCaps** – Dùng để in hoa hết tất cả các ký tự của text nếu giá trị thuộc tính này là true.

Button

□ **Button** là một loại View, nó hiện thị nút bấm để chờ người dùng nhấn vào

- ❖ Do Button mở rộng từ TextView nên nó có đầy đủ các thuộc tính của TextView như: text, textColor, textSize, textStyle, textAllCaps,...
- ❖ Ngoài các thuộc tính được kế thừa còn có một số thuộc tính cần lưu ý:
 - android:drawableStart
 - android:drawableEnd
 - android:drawableTop
 - android:drawableBottom
 - android:drawablePadding
 - android:backgroundTint
 - android:minHeight
 - android:minWidth
 - android:maxLines





Button

Android

☐ Các thuộc tính

- ❖ android:drawableStart (android:drawableLeft),
 - ❖ android:drawableEnd (android:drawableRight),
 - ❖ android:drawableTop, android:drawableBottom để gán các ảnh Drawable vào biên trái, phải, trên, dưới của nút bấm
- ☐ android:drawablePadding - Thiết lập khoảng cách các ảnh Drawable đến vùng nội dung dùng thuộc tính
- ☐ android:minHeight="0dp" và android:minWidth="0dp" giảm kích thước của Button (loại bỏ khoảng giống padding của Button)
- ☐ android:maxLines="1" - Thiết lập Button chỉ hiện thị một dòng Text
- ☐ android:backgroundTint - Thiết lập màu nền cho nút



Button

- ❑ Bắt sự kiện click khi nhấn vào nút

//Ánh xạ widget

```
Button btnAddToCart = findViewById(R.id.btnAddToCart);
```

//Bắt sự kiện click

```
btnAddToCart.setOnClickListener(new View.OnClickListener() {
```

```
    public void onClick(View v) {
```

```
        //Thực hiện xử lý khi nhấn vào nút
```

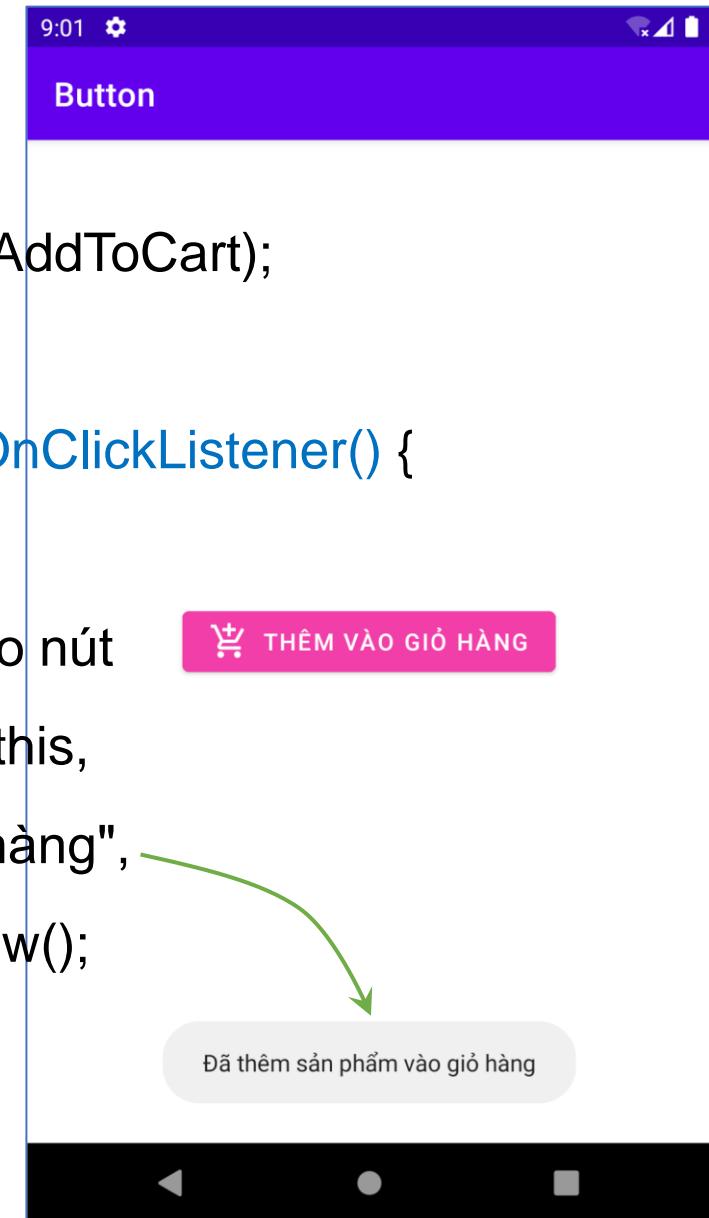
```
        Toast.makeText(MainActivity.this,
```

```
            "Đã thêm sản phẩm vào giỏ hàng",
```

```
            Toast.LENGTH_SHORT).show();
```

```
});
```

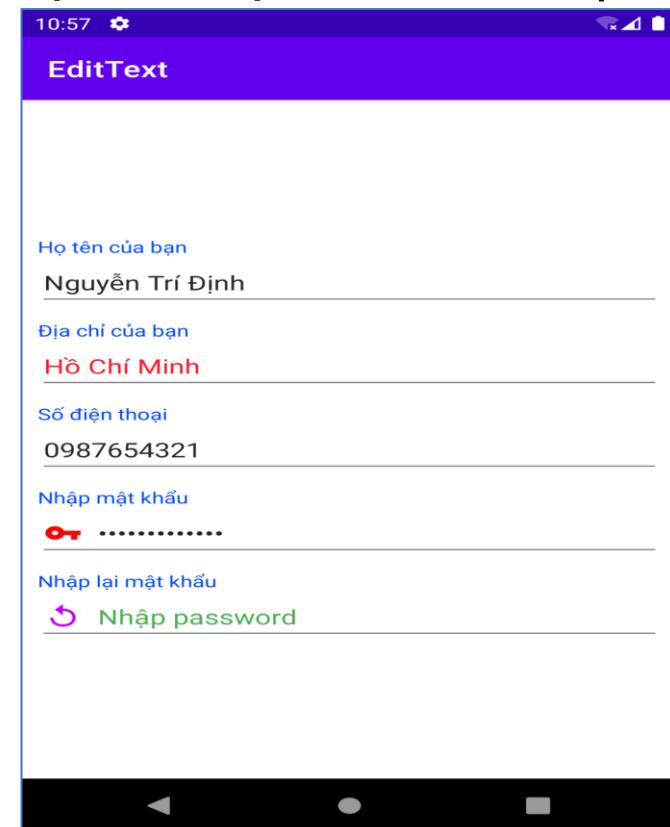
```
}
```



EDITTEXT

□ **EditText** là loại View cho phép người dùng nhập dữ liệu (chữ, số, số điện thoại, ngày tháng,...)

- ❖ Do EditText mở rộng chức năng từ TextView, nên nó có đầy đủ các thuộc tính của TextView như: màu chữ, font chữ, màu nền,...
- ❖ Ngoài các thuộc tính được kế thừa còn có một số thuộc tính cần lưu ý:
 - android:hint
 - android:inputType
 - android:maxLines
 - android:lines
 - android:digits
 - android:textColorHighlight
 - android:drawableLeft
 - android:drawableTint
 - android:textColorHint
 - android:drawablePadding





Android

EDITTEXT

- **android:hint** - Thiết lập gợi ý nội dung cần nhập vào EditText
- **android:maxLines** kết hợp **android:lines** - Giới hạn dòng chữ nhập trên một dòng
- **android:digits** - Giới hạn loại ký tự số nhập vào
- **android:textColorHighlight** - Chỉnh màu Highlight
- **android:drawableLeft** – Thêm drawable vào EditText
- **android:drawableTint** – Thay đổi màu sắc của drawable
- **android:textColorHint** – Điều chỉnh màu sắc của hint
- **android:drawablePadding** – Canh padding của drawable với nội dung



EDITTEXT

☐ **android:inputType** - Thiết lập kiểu nhập liệu.
Tương ứng với mỗi kiểu này EditText sẽ có cách hiển thị cũng như liên kết với loại bàn phím tương ứng.

- ❖ Ví dụ như thiết lập nhập password thì dữ liệu nhập vào sẽ hiển thị bằng ký hiệu *, hay chọn dữ liệu số thì bàn phím xuất hiện là loại bàn phím số
- ❖ Các hằng số gán vào android:inputType có thể kết hợp nhiều loại với nhau, mỗi loại cách nhau bởi dấu |



EDITTEXT

- ☐ Một số giá trị thường sử dụng trong android:inputType

Giá trị	Ý nghĩa
date	Nhập ngày tháng
datetime	Nhập ngày tháng, giờ
number	Nhập số
numberDecimal	Nhập số thập phân
numberSigned	Nhập số nguyên không dấu
phone	Nhập số điện thoại
text	Nhập văn bản
textMultiLine	Chữ trên nhiều dòng
textPassword	Nhập password
textUri	Địa chỉ URL
time	Thời gian



EDITTEXT

☐ Lấy và gán dữ liệu cho EditText

//Ánh xạ widget

```
EditText edtName = findViewById(R.id.edtName);
```

//Lấy giá trị từ EditText

```
String name = edtName.getText().toString();
```

//Gán giá trị cho EditText

```
edtName.setText("Nguyễn Văn A");
```



EDITTEXT

□ Lấy thông tin từ EditText trong lúc nhập dữ liệu

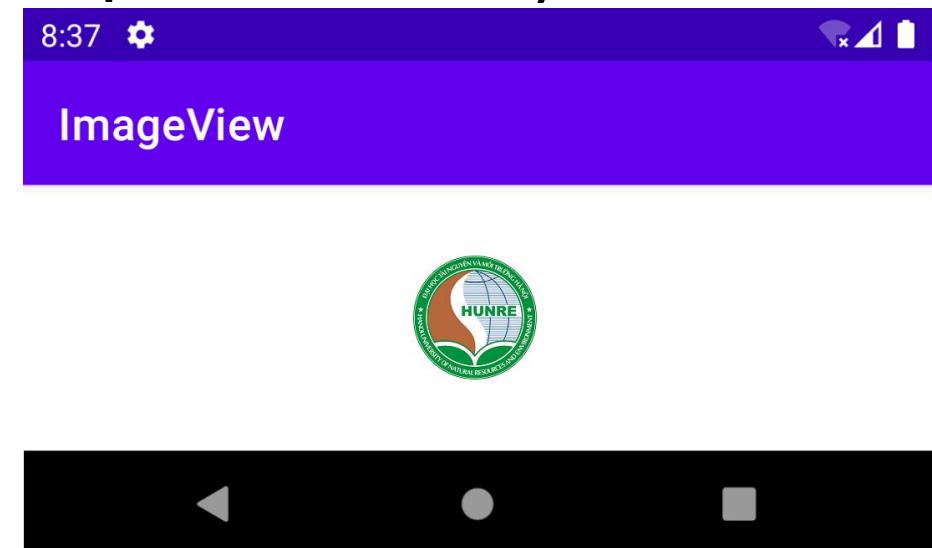
```
edtName.addTextChangedListener(new TextWatcher() {  
    @Override  
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {  
        }  
        //Gọi trước khi text thay đổi  
    @Override  
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {  
        }  
        //Gọi khi text thay đổi  
    @Override  
    public void afterTextChanged(Editable editable) {  
        //Gọi sau khi text thay đổi  
    }  
});
```

ImageView

❑ ImageView là loại View dùng để hiện 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 (tint) vào ảnh, co/kéo/cắt ảnh khi hiện thị trên View

❖ Trong ImageView có các thuộc tính cần lưu ý:

- android:src
- android:scaleType
- android:adjustViewBounds





ImageView

- ❑ android:src – Gán tài nguyên ảnh vào ImageView
- ❑ android:adjustViewBounds - Nếu nhận giá trị true thì các ImageView tự động co giãn vừa với ảnh. (cần có thiết lập chiều rộng hoặc cao là wrap_content)
- ❑ android:scaleType - Dùng để thiết lập thu phóng ảnh, nhận các giá trị như: fitXY, center, fitXY, ...



ImageView

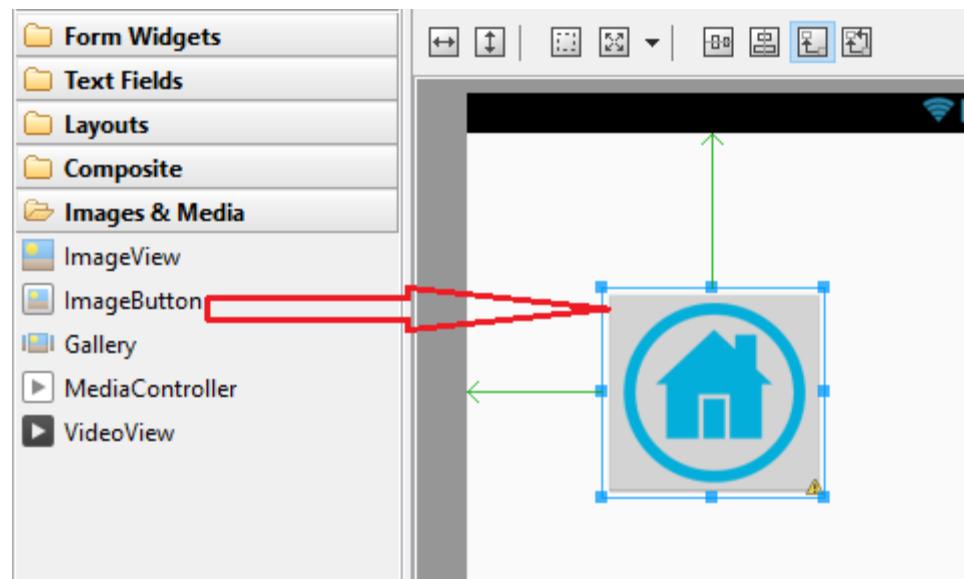
☐ Một số giá trị thường sử dụng trong android:scaleType

Giá trị	Ý nghĩa
center	Đặt ảnh vào giữa ImageView, không có thay đổi tỷ lệ ảnh.
centerCrop	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	Đặt ảnh vào giữa ImageView, có thu phóng ảnh (nhưng giữ nguyên tỉ lệ cao / rộng) sao cho toàn bộ các phần của ảnh hiện thị trên ImageView.
fitCenter	Đặt ảnh vào giữa ImageView, có thu phóng ảnh (nhưng giữ nguyên tỉ lệ cao / rộng) sao cho toàn bộ các phần của ảnh hiện thị trên ImageView.
fitEnd fitStart	Co ảnh vừa View, vị trí ảnh ở cuối (ở đầu) ImageView
fitXY	Co ảnh vừa khít cả chiều rộng và cao.



ImageButton

❑ ImageButton kế thừa từ ImageView nhưng nó hiển thị như một nút bấm, có một chút khác với Button đã biết là nó hiện thị ảnh ở giữa nút bấm thay vì dòng chữ.

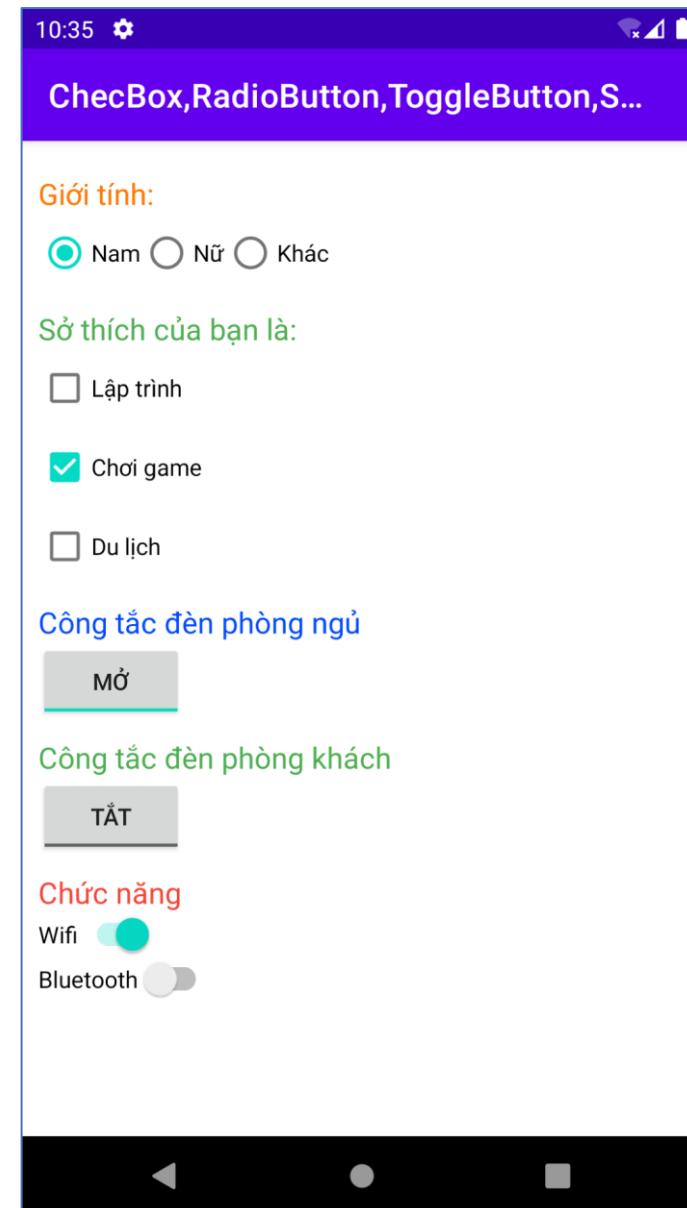




CHECBOX, RADIobutton, TOGGLEBUTTON, SWITCH

Android

- ❑ 4 view này xử lý khá giống nhau đều có 2 trạng thái **checked** và **unchecked**
- ❑ Checkbox cho phép chọn nhiều lựa chọn
- ❑ RadioButton cho phép chọn 1 trong nhiều lựa chọn (phải đưa vào một RadioGroup)
- ❑ ToggleButton và Switch chỉ khác nhau về hình dạng. Có thêm 2 thuộc tính là **textOn** và **TextOff** cho text ở 2 trạng thái
- ❑ Gán trạng thái mặc định trong layout dùng: android:**checked="true"**





CHECKBOX, RADIOPBUTTON, TOGGLEBUTTON, SWITCH

- ☐ Xét trạng thái hiện tại:

```
if(cb.isChecked() == true){  
}  
}
```

- ☐ Gán trạng thái hiện tại `cb.setChecked(true);`
- ☐ Đảo trạng thái hiện tại: `cb.toggle();`

- ☐ Bắt sự kiện khi có sự thay đổi trạng thái:

```
cb.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChanged(CompoundButton compoundButton, boolean b) {  
    }  
});|
```

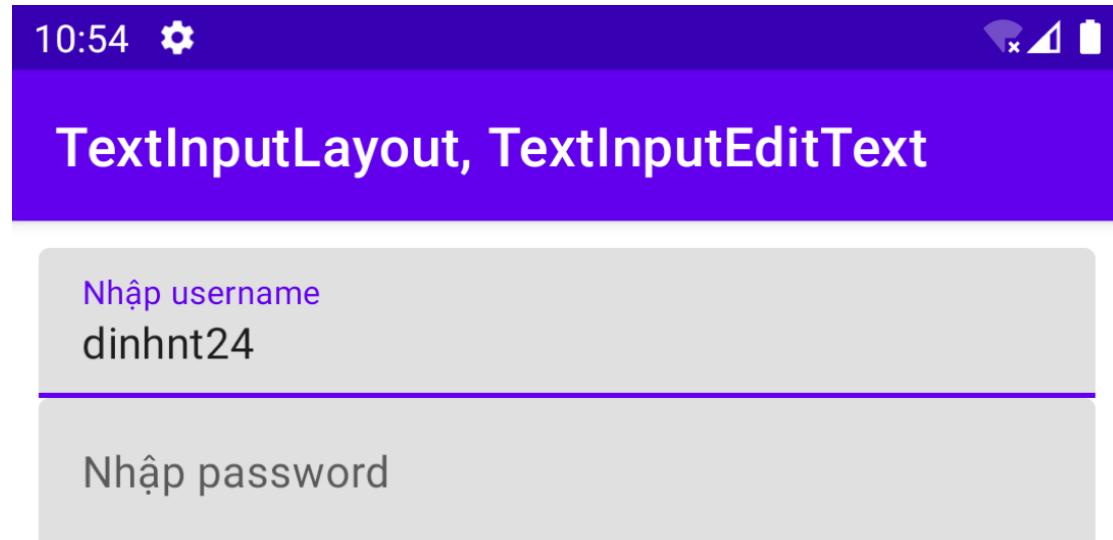
- ☐ Xét trạng thái theo RadioGroup:

```
RadioGroup group= (RadioGroup) findViewById(R.id.radiogroup1);  
int idChecked = group.getCheckedRadioButtonId();  
switch (idChecked)  
{  
    case R.id.radioButton: break;  
}
```



TextInputLayout và TextInputEditText

- ❑ **TextInputEditText** là view được mở rộng từ chính **EditText** nên có thể dùng nó giống hoàn toàn như **EditText**.
- ❑ Tuy nhiên **TextInputEditText** được thiết kế với mục đích chính là làm phần tử con của phần tử **TextInputLayout**, lúc đó **TextInputLayout** sẽ bao bọc lấy **TextInputEditText** và hiện thị các thông tin như: gợi ý (hint), đếm số ký tự, hiện thị thông báo lỗi,...





TextInputLayout và TextInputEditText

```
<com.google.android.material.textfield.TextInputLayout  
    android:id="@+id/tilUsername"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:hintEnabled="true">  
  
<com.google.android.material.textfield.TextInputEditText  
    android:id="@+id/tieUsername"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="Nhập username"  
    android:inputType="textEmailAddress"  
    android:maxLength="10" />  
  
</com.google.android.material.textfield.TextInputLayout>
```





TextInputLayout và TextInputEditText

- ☐ Hiển thị số ký tự nhập vào thiết lập: app:counterEnabled="true"

```
<com.google.android.material.textfield.TextInputLayout  
    android:id="@+id/tilUsername"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:counterEnabled="true"  
    app:hintEnabled="true">
```



```
<com.google.android.material.textfield.TextInputEditText  
    android:id="@+id/tieUsername"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="Nhập username"  
    android:inputType="textEmailAddress"  
    android:maxLength="10" />
```

```
</com.google.android.material.textfield.TextInputLayout>
```

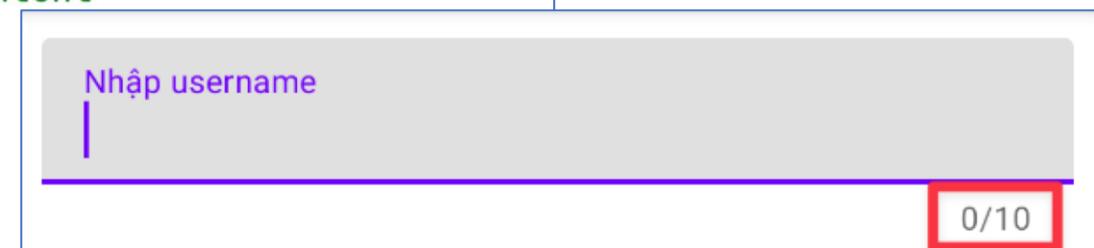


TextInputLayout và TextInputEditText

Android

- Để thiết lập số ký tự lớn nhất, ví dụ 10 ký tự thêm vào `app:counterMaxLength="10"` đồng thời ở `TextInputEditText` thiết lập

```
<com.google.android.material.textfield.TextInputLayout  
    android:id="@+id/tilUsername"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:counterEnabled="true"  
    app:counterMaxLength="10"  
    app:hintEnabled="true">
```



```
<com.google.android.material.textfield.TextInputEditText  
    android:id="@+id/tieUsername"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="Nhập username"  
    android:inputType="textEmailAddress"  
    android:maxLength="10" />
```

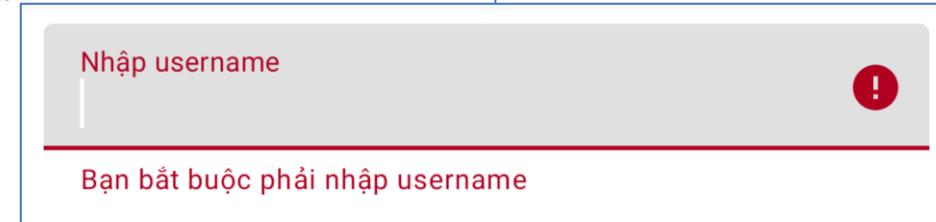
```
</com.google.android.material.textfield.TextInputLayout>
```



TextInputLayout và TextInputEditText

- Để hiện thị dòng thông báo lỗi, thiết lập app:errorEnabled="true" và dòng thông báo lỗi được thiết lập bằng code Java: setError()

```
<com.google.android.material.textfield.TextInputLayout  
    android:id="@+id/tilUsername"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:errorEnabled="true"  
    app:hintEnabled="true">
```



```
<com.google.android.material.textfield.TextInputEditText  
    android:id="@+id/tieUsername"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="Nhập username"  
    android:inputType="textEmailAddress"  
    android:maxLength="10" />
```

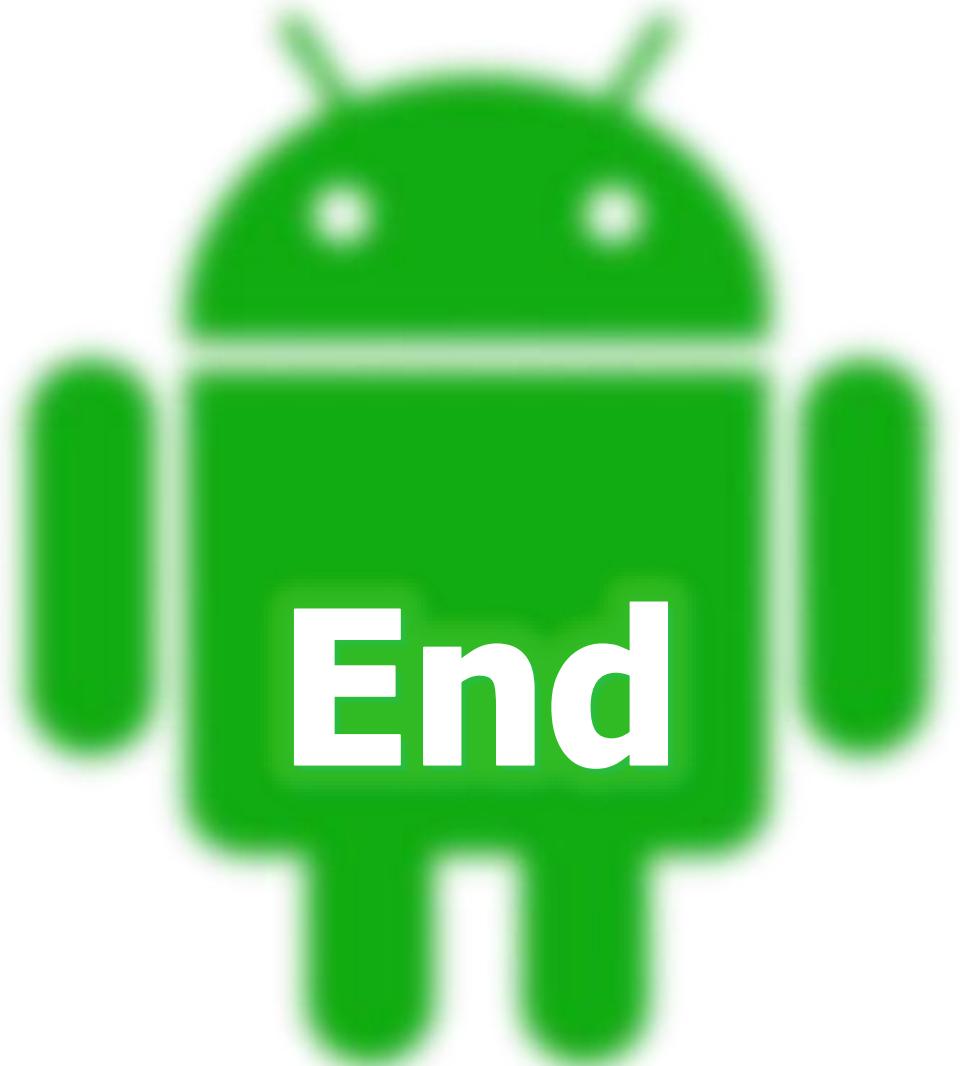
```
</com.google.android.material.textfield.TextInputLayout>
```



TextInputLayout và TextInputEditText

- Để hiện thị dòng thông báo lỗi, thiết lập `app:errorEnabled="true"` và dòng thông báo lỗi được thiết lập bằng code Java: `setError()`

```
textInputEditTextUsername.addTextChangedListener(new TextWatcher() {  
    @Override  
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {}  
  
    @Override  
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {  
        if (charSequence.length() == 0) {  
            textInputLayoutUername.setError("Bạn bắt buộc phải nhập username");  
        } else {  
            textInputLayoutUername.setError(null);  
        }  
    }  
  
    @Override  
    public void afterTextChanged(Editable editable) {}  
});
```

A large, semi-transparent green Android robot icon serves as the background for the text.

End

Android