

# Bài 3: Android Layout Buổi 1

Giáo viên: Lê Quốc Anh

# Nội dung

---

1. View, ViewGroup, Layout
2. TextView
3. Button
4. EditText
5. CheckBox
6. RadioButton và RadioGroup

# 1. View, ViewGroup, Layout

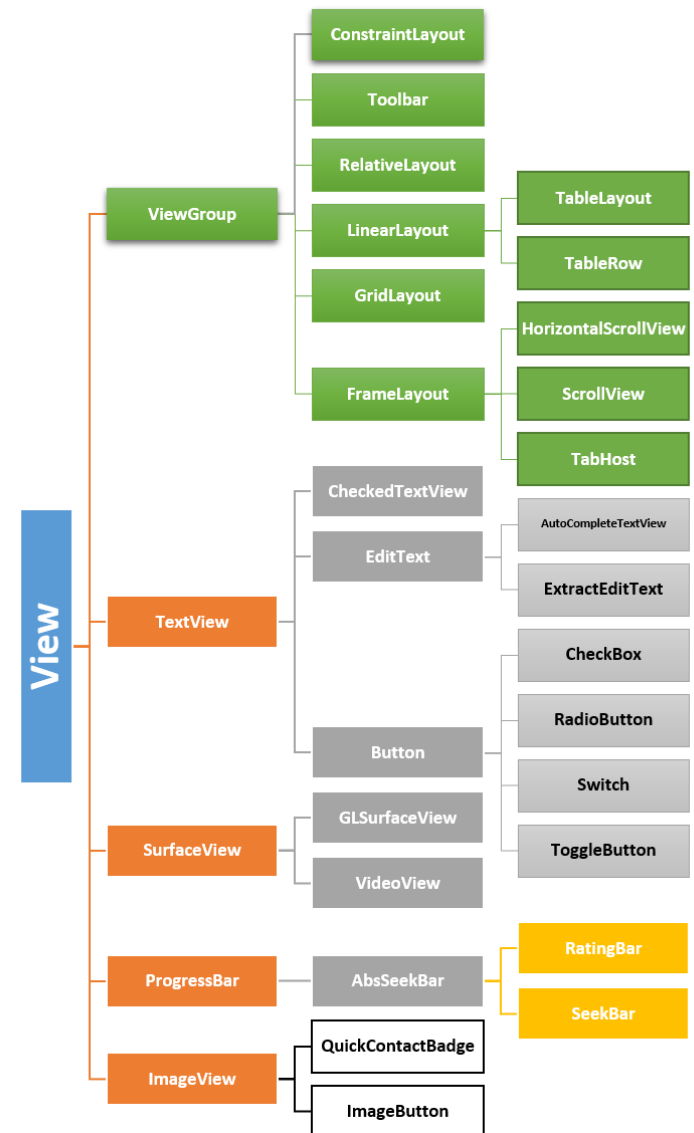
---

- Các thành phần cơ bản để xây dựng giao diện người dùng (UI) trong Android là View, ViewGroup và Layout. Từ các thành phần cơ sở đó kết hợp chúng với nhau tạo ra các loại UI phức tạp cho ứng dụng Android
- Để tạo ra một giao diện người dùng trong ứng dụng Android, chúng ta xây dựng từ sự kết hợp các thành phần có tên là View, ViewGroup, Layout.

# 1. View, ViewGroup, Layout

## ■ View và ViewGroup

- ❑ **View** biểu diễn một hình chữ nhật, trong đó nó hiện thị thông tin nào đó cho người dùng, và người dùng có thể tương tác với View. Một số View cơ bản đó là: TextView, ImageView, Button, ImageButton, EditText ...
- ❑ **ViewGroup** Có một lớp abstract kế thừa từ View là ViewGroup, nó cũng chính là một View nhưng có khả năng chứa các View khác bên trong (kể cả ViewGroup khác). Nó là lớp cơ sở để xây dựng nên các layout như trên sơ đồ ta thấy có: ConstraintLayout, RelativeLayout, LinearLayout, GridLayout, FrameLayout



# 1. View, ViewGroup, Layout

---

- **Layout** Các layout chính là các View (cụ thể nó kế thừa thừa ViewGroup) được thiết kế với mục đích chứa các View con và điều khiển, sắp xếp vị trí các View con đó trên màn hình, mỗi layout có cơ chế điều khiển vị trí View con riêng của mình.
  - **FrameLayout** đơn giản chỉ cung cấp một vùng màn hình, thường dùng nó để hiển thị một View con duy nhất.
  - **ConstraintLayout** (giới thiệu trong Android 7), sử dụng layout này được khuyến khích cho hầu hết trường hợp. ConstraintLayout cho phép điều khiển vị trí và ứng xử của các view con trong layout bằng cách gán ràng buộc đơn giản vào mỗi view con.
  - **LinearLayout** với layout này thì các view con được xếp nối tiếp nhau (linear) thành một hàng hay một cột (tùy vào lúc thiết kế thiết lập hướng xếp)

# 1. View, ViewGroup, Layout

---

- **RelativeLayout** cho phép các view con định vị căn vào liên hệ với các view con khác đồng thời liên hệ với view cha thông qua các tham số align và margin
- **GridLayout** chia ra thành lưới gồm một số hàng và một số cột để chứa các view con.
- **TableLayout** cung cấp khả năng bố trí các view con thành một lưới dạng bảng (gồm có hàng và cột). Một dòng của bảng biểu diễn bằng đối tượng view con TableRow, trong nó chứa có phần tử View con hiểu như các ô bảng.
- **CoordinatorLayout** nó được thiết kế nhằm mục đích có sự tương tác của các View con trong nó, đặc biệt sử dụng với ActionBar, FloatingActionButton, Snackbar ...

# 1. View, ViewGroup, Layout

---

- Xây dựng View trong Android
  - Code Java
  - XML
  - Kéo thả trong Android Studio

# Xây dựng view bằng code java

```
LinearLayout layout = new LinearLayout( context: this);
layout.setOrientation(LinearLayout.VERTICAL);
layout.setBackgroundColor(Color.GREEN);

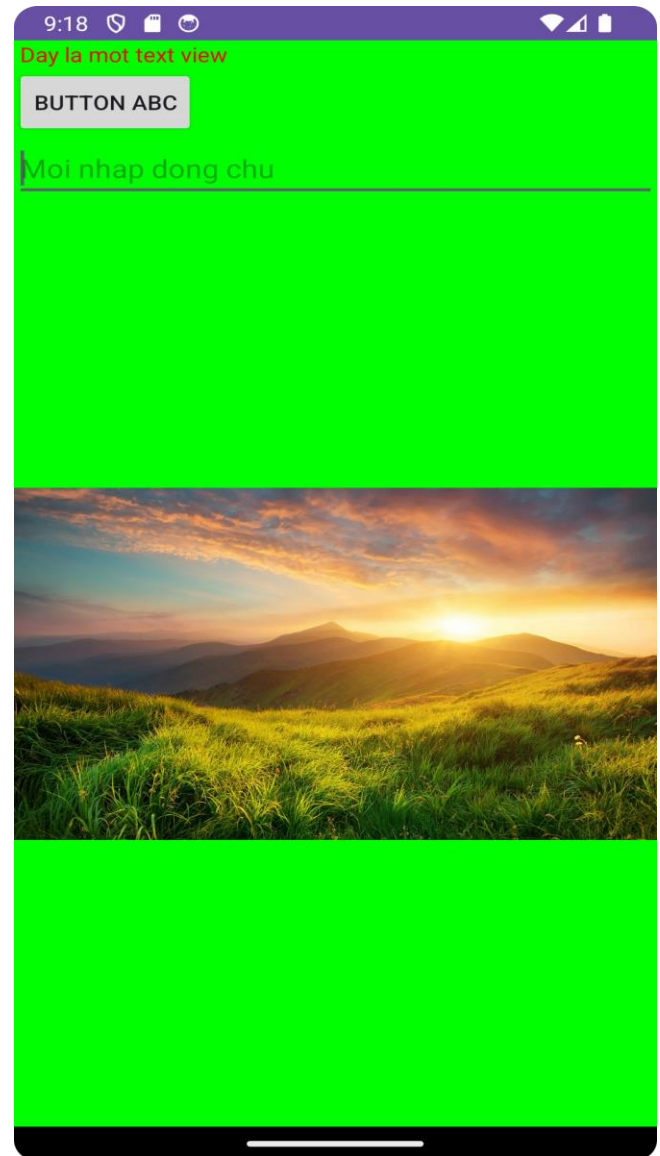
TextView textView = new TextView( context: this);
textView.setText(" Day la mot text view");
textView.setTextColor(Color.RED);
layout.addView(textView);

Button bt = new Button( context: this);
bt.setText("Button ABC");
LinearLayout.LayoutParams pr = new LinearLayout.LayoutParams(
    ViewGroup.LayoutParams.WRAP_CONTENT,
    ViewGroup.LayoutParams.WRAP_CONTENT);
bt.setLayoutParams(pr);
layout.addView(bt);

EditText editText = new EditText( context: this);
editText.setHint("Moi nhap dong chu");
layout.addView(editText);

ImageView imageView = new ImageView( context: this);
imageView.setImageResource(R.drawable.pc);
layout.addView(imageView);

setContentView(layout);
```





# Xây dựng view bằng XML

```
<LinearLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#0CE63B"
    xmlns:android="http://schemas.android.com/apk/res/android">

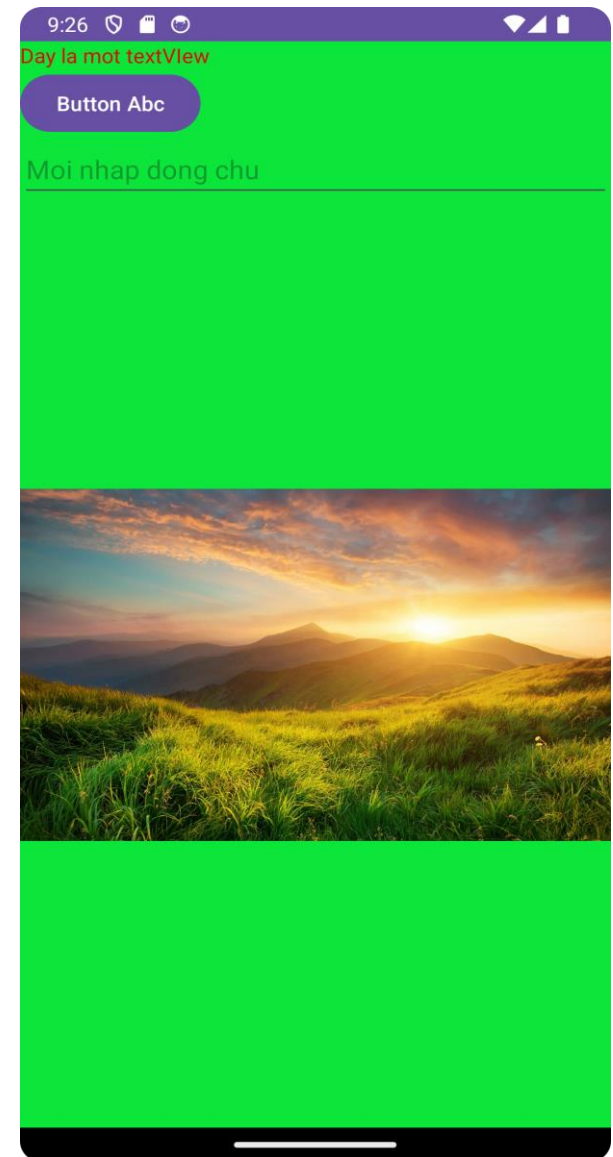
    <TextView
        android:text="Day la mot textVlew"
        android:textColor="#E40606"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

    <Button
        android:text="Button Abc"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

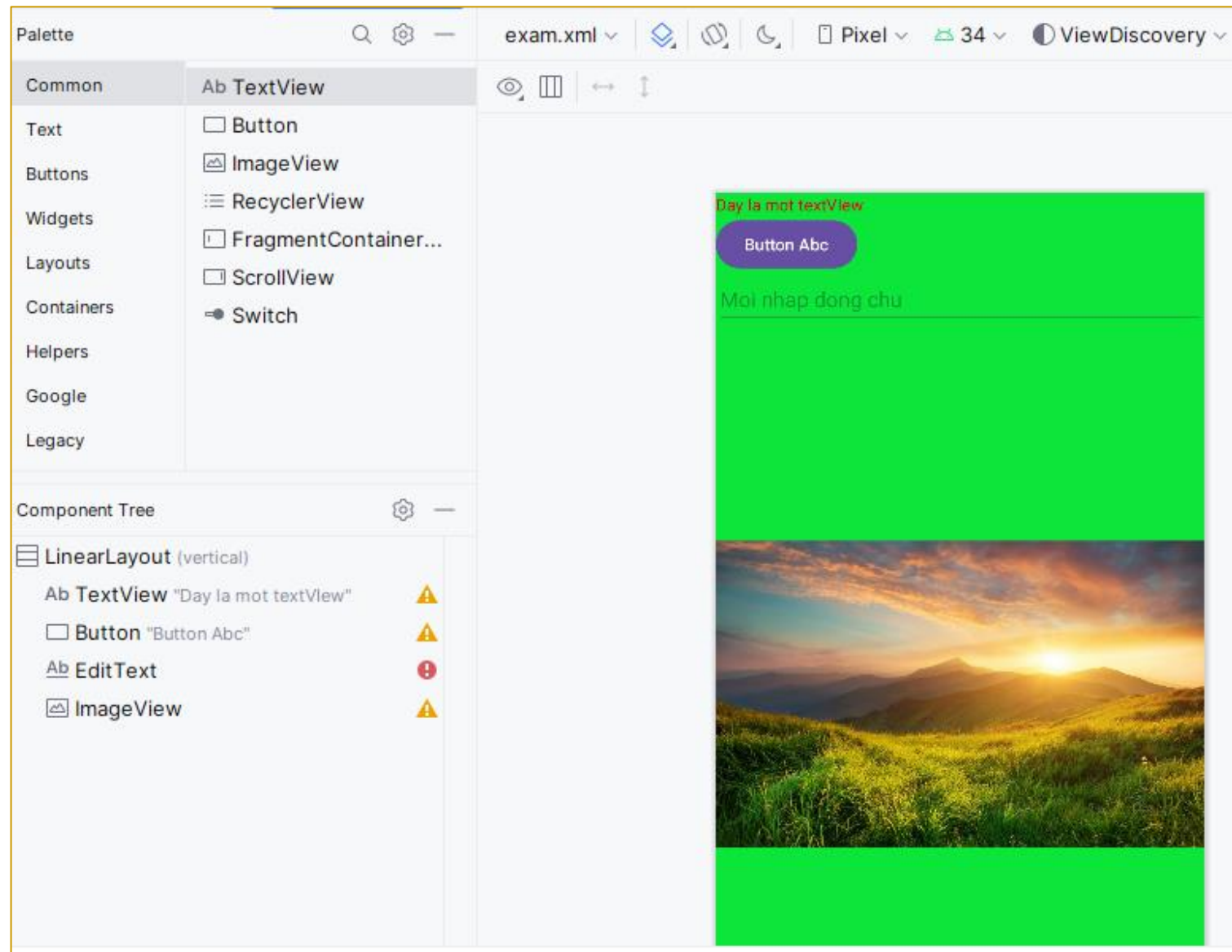
    <EditText
        android:hint="Moi nhap dong chu"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

    <ImageView
        android:src="@drawable/pc"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>
```



# Xây dựng view bằng kéo thả



# Đơn vị đo

---

- **in** – Inches
- **mm** – Millimeters
- **pt** – Points (1/72 of an inch)
- **Dp/dip** – Density-independent pixels: Không phụ thuộc vào mật độ pixel, 160dp = 1'' trên màn hình
- **sp** – Scale-independent pixels: thường dùng để set font size
- **px** – Actual screen pixels: là 1 đơn vị pixel thực tế trên màn hình, không nên dùng đơn vị này vì nó là đơn vị chính xác → Trên các màn hình khác nhau sẽ hiển thị không như ý muốn

## 2. TextView

- **TextView** là một View cho phép hiển 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
    android:id="@+id/textview1"
    android:layout_width="match_parent"
    android:layout_height="500px"
    android:background="#959595"

    android:drawableRight="@drawable/ico"
    android:drawableBottom="@drawable/ico"
    android:drawablePadding="10dp"
    android:fontFamily="monospace"
    android:gravity="center|bottom"

    android:hint="Noi dung la gi"
    android:padding="10dp"
    android:text="Day la dong chu trong textview"
    android:textAllCaps="true"

    android:textColor="#DC0707"
    android:textSize="14sp"
    android:textStyle="bold|italic" />
```



Từ Activity hoặc ViewGroup lấy đối tượng TextView bằng cách sử dụng **findViewById(id)**:

```
TextView textView1;
textView1 = findViewById(R.id.textview1);
```

```
textView1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        textView1.setText(null);
    }
});
```

# 3. Button

- **Button** là một loại View, nó hiện thị nút bấm để chờ người dùng bấm vào. Button kế thừa từ TextView nên các thuộc tính, thiết lập cho TextView ở phần trước là có hiệu quả như đối với Button

## Cú pháp khai báo Button trong XML

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

Từ Activity hoặc từ đối tượng ViewGroup chứa Button, để lấy Button và bắt sự kiện khi người dùng bấm vào thì làm như sau:

```
final Button button = findViewById(R.id.mybutton_id);
button.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        //Muốn làm gì khi bấm vào Button thì
        //Viết code ở đây - Ví dụ:
        Toast.makeText(MainActivity.this, "Hello!",
            Toast.LENGTH_SHORT).show();
    }
});
```

# 3. Button

```
<Button
    android:id="@+id/button1"
    android:text="Button1 "
    android:minWidth="0dp"
    android:minHeight="80dp"
    android:maxLines="2"
    android:backgroundTint="@drawable/bg"
    android:textColor="#0E57E8"
    android:textSize="20sp"
    android:textStyle="bold | italic"
    android:fontFamily="monospace"
    android:textAllCaps="false"
    android:layout_width="200dp"
    android:layout_height="wrap_content"/>
```



BUTTON 2

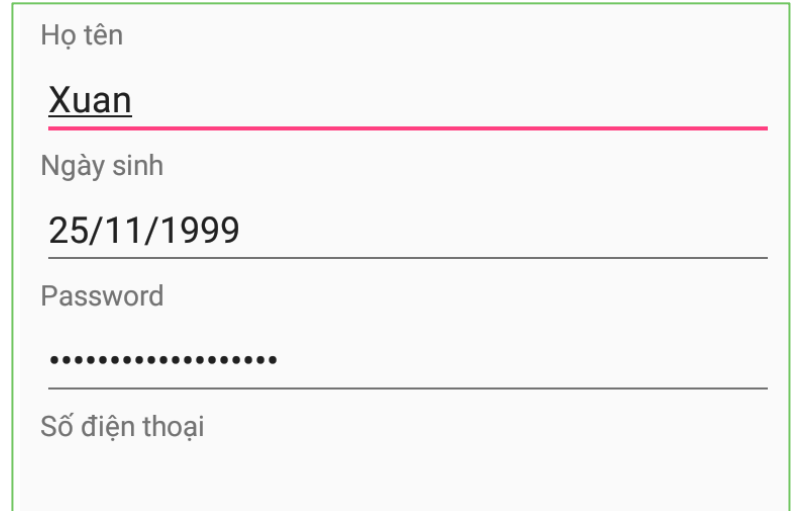
BUTTON 3

```
<Button
    android:id="@+id/button2"
    android:text="Button 2"
    style="@style/Widget.AppCompat.Button.Borderless"
    android:textColor="#111"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
<Button
    android:id="@+id/button3"
    android:text="Button 3"
    style="@style/TextAppearance.AppCompat.Widget.Button.Colored"
    android:backgroundTint="#5FDC1C"
    android:textColor="#990DA5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

```
button1 = findViewById(R.id.button1);
button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(context, MainActivity.this,
            text: "Bạn vừa bấm vào nút bấm",
            Toast.LENGTH_SHORT).show();
    }
});
```

# 4. EditText

- Sử dụng **EditText** và các lớp mở rộng từ EditText như **TextInputEditText**, **AutoCompleteTextView** để nhập dữ liệu, kiểm soát nhập dữ liệu với **TextWatcher**
- **EditText** (hoặc **AppCompatActivity** trong thư viện Support Design) là loại View hiển thị một hộp (chữ nhật) cho phép người dùng nhập dữ liệu (chữ, số ... có thể khống chế nhập liệu là text, số, phone, ngày tháng ...). Ví dụ màn hình dưới dùng EditText để người dùng nhập Tên, Ngày Sinh, Password ...
- Do **EditText** mở rộng chức năng từ **TextView**, nên các tùy chọn thiết lập trình bày ở TextView vẫn đúng cho EditText như: màu chữ, font chữ, màu nền, hint, gán drawable vào các biên



Họ tên

Xuan

Ngày sinh

25/11/1999

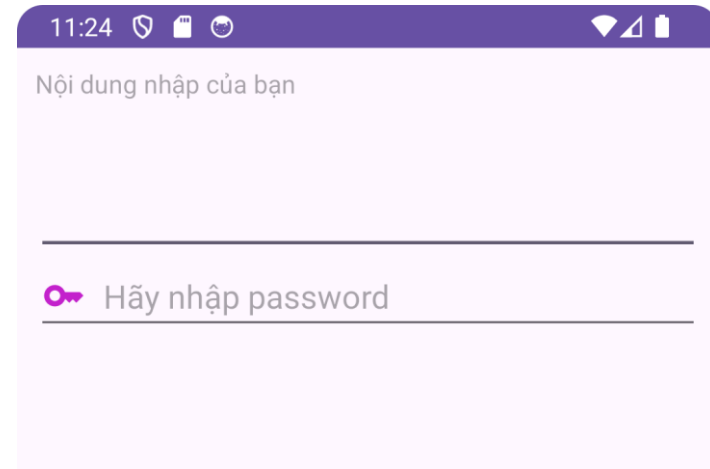
Password

.....

Số điện thoại

# 4. EditText

```
<TextView
    android:id="@+id/noidung"
    android:hint="Nội dung nhập của bạn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
<EditText
    android:id="@+id/edittext"
    android:textColor="#111"
    android:textColorHighlight="#3df107"
    android:inputType="textMultiLine"
    android:lines="3"
    android:maxLines="10"
    android:gravity="bottom|start"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
<EditText
    android:id="@+id/password"
    android:drawableLeft="@drawable/passico"
    android:drawableTint="#C321CC"
    android:drawablePadding="10dp"
    android:inputType="textPassword"
    android:hint="Hãy nhập password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```



```
editText = findViewById(R.id.edittext);
TextView noidung = findViewById(R.id.noidung);
editText.setText();
editText.getText();
editText.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        noidung.setText(s);
    }

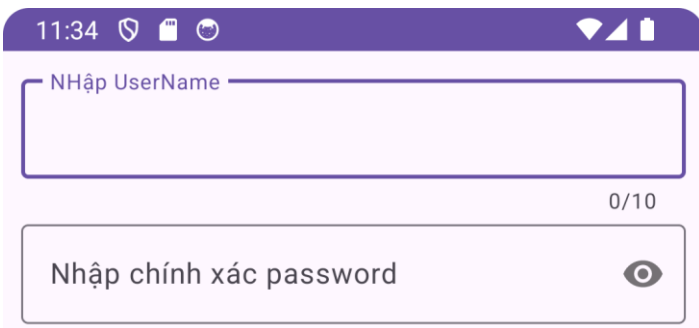
    @Override
    public void afterTextChanged(Editable s) {
    }
});
}
```



# 4. EditText

```
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    app:hintEnabled="true"
    app:counterEnabled="true"
    app:counterMaxLength="10"
    app:errorEnabled="true"
    android:layout_height="wrap_content">
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/user"
        android:maxLength="10"
        android:hint="NHập UserName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
</com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    app:hintEnabled="true"
    app:passwordToggleEnabled="true"
    android:layout_height="wrap_content">
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/password"
        android:inputType="textPassword"
        android:hint="Nhập chính xác password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
</com.google.android.material.textfield.TextInputLayout>
```



```
TextInputEditText editText = findViewById(R.id.user);
editText.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
    }
    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        if(s.length()==0){
            editText.setError("Bạn phải nhập UserName ");
        }else{
            editText.setError(null);
        }
    }
    @Override
    public void afterTextChanged(Editable s) {
    }
});
```

# 5. CheckBox

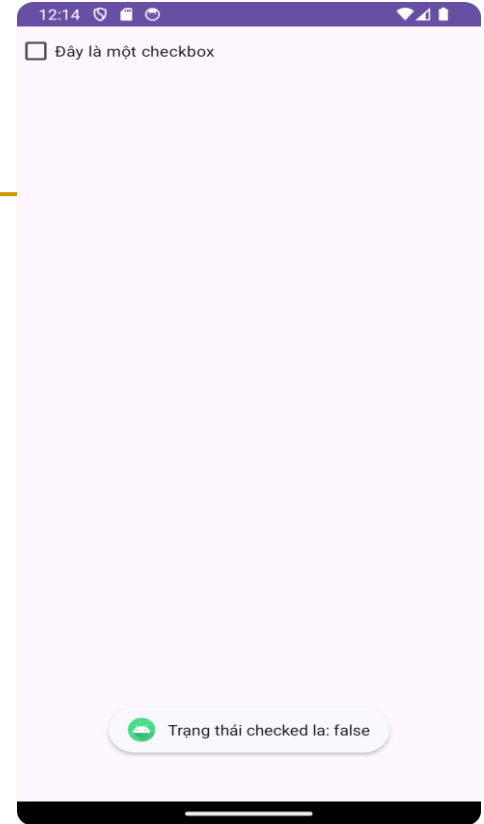
- **CompoundButton** là một lớp cơ abstract mở rộng từ TextView (Button), từ lớp này nó được mở rộng để xây dựng các View là: **CheckBox**, **RadioButton**, **Switch**, **ToggleButton**
- Các lớp mở rộng **CheckBox**, **RadioButton**, **Switch**, **ToggleButton** lưu lại các thuộc tính, ứng xử tương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



# 5. CheckBox

<CheckBox

```
    android:checked="false"
    android:id="@+id/checkbox1"
    android:text="Đây là một checkbox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```



```
checkbox1 = findViewById(R.id.checkbox1);
boolean checked = checkbox1.isChecked();
checkbox1.setChecked(true);
checkbox1.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        Toast.makeText(context, "Trạng thái checked là: "+isChecked,
            Toast.LENGTH_SHORT).show();
    }
});
```

# 5. CheckBox

<TextView

android:text="Loại biến nào được dùng đặt tên cho 1 thành phố"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<CheckBox

android:id="@+id/pa1"

android:text="int"

android:button="@drawable/checkboxicon"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<CheckBox

android:id="@+id/pa2"

android:text="double"

android:button="@drawable/checkboxicon"

android:buttonTint="#BB13C5"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<CheckBox

android:id="@+id/pa3"

android:text="String"

android:button="@drawable/checkboxicon"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<CheckBox

android:id="@+id/pa4"

android:button="@drawable/checkboxicon"

android:text="Tất cả các phương án"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<Button

android:id="@+id/kiemtra"

android:text="Kiểm tra"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"/>

<Button

android:id="@+id/goi y"

android:text="Gợi ý"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"/>

12:21

Loại biến nào được dùng đặt tên cho 1 thành phố

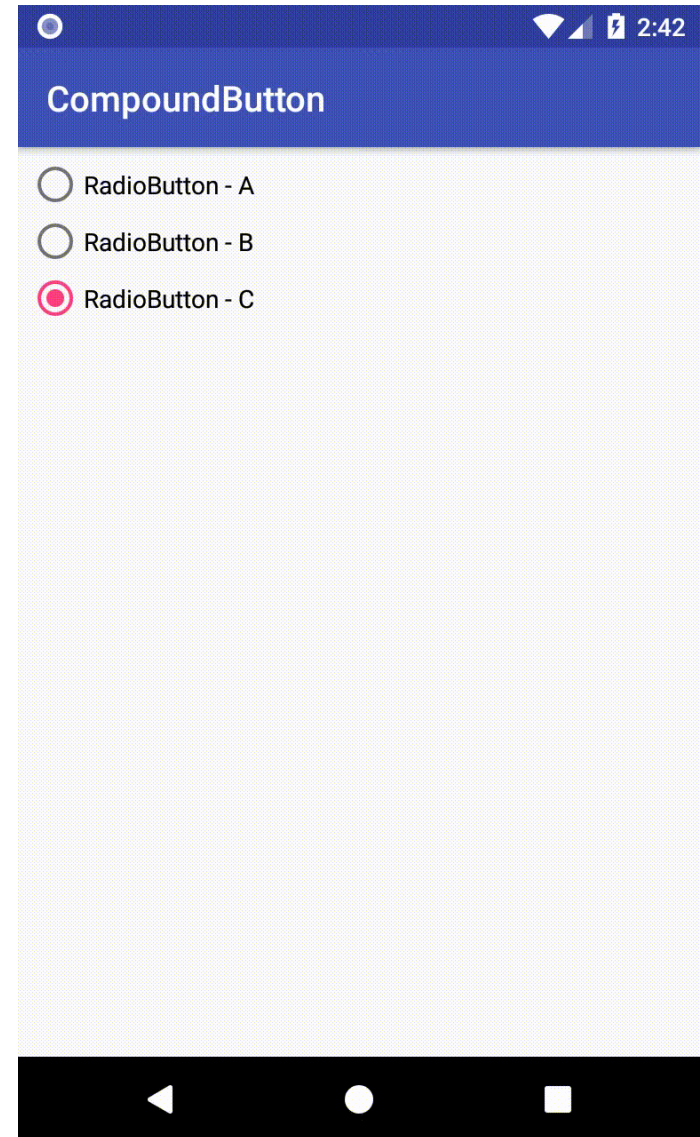
- ☐ int
- ☐ double
- ☐ String
- ☐ Tất cả các phương án

Kiểm tra

Gợi ý

# 6. 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.



# 6. RadioButton và RadioGroup

<RadioButton

```
    android:id="@+id/radiobutton"  
    android:text="Radio Button"  
    android:checked="false"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```

```
RadioButton radioButton = findViewById(R.id.radiobutton);  
radioButton.isChecked();  
radioButton.setChecked(false);  
  
radioButton.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
        Toast.makeText(context: MainActivity.this,  
            text: "Trạng thái checked "+radioButton.isChecked(),  
            Toast.LENGTH_SHORT).show();  
    }  
});
```



# 6. RadioButton và RadioGroup

```
<TextView
```

```
    android:id="@+id/textView"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:padding="10dp"  
    android:text="Chọn các giá trị biểu thức sau:  
    Math.abs(Math.min(-6,3))" />
```

```
<RadioGroup
```

```
    android:layout_width="match_parent"  
    android:padding="10dp"  
    android:layout_height="wrap_content">
```

```
    <RadioButton
```

```
        android:id="@+id/radio_a"  
        android:text="3"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"/>
```

```
    <RadioButton
```

```
        android:id="@+id/radio_b"  
        android:text="6"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"/>
```

```
    <RadioButton
```

```
        android:id="@+id/radio_c"  
        android:text="-6"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"/>
```

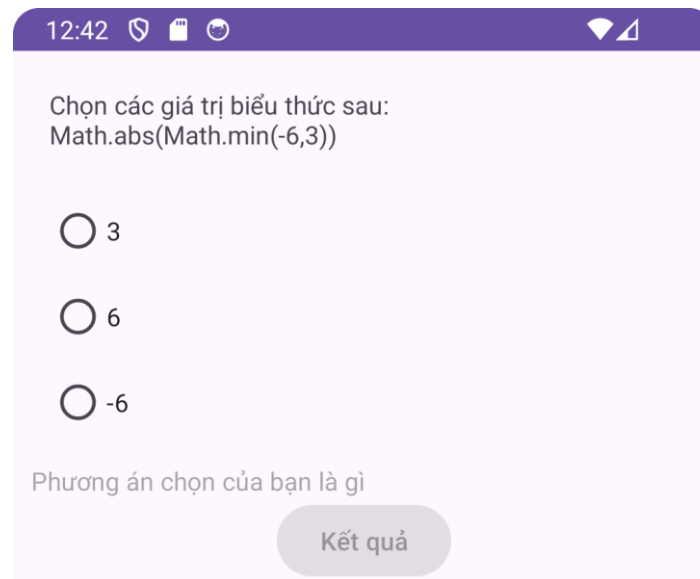
```
</RadioGroup>
```

```
<TextView
```

```
    android:id="@+id/mgs"  
    android:hint="Phương án chọn của bạn là gì"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"/>
```

```
<Button
```

```
    android:id="@+id/test"  
    android:enabled="false"  
    android:text="Kết quả"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```



---

**THANK YOU**  
**for**  
**YOUR ATTENTION**

