

Chương 3 :

XỬ LÝ GIAO DIỆN NGƯỜI DÙNG



TS. Huỳnh Hữu Nghĩa

luckerrhuynhvn@gmail.com

Nội dung:



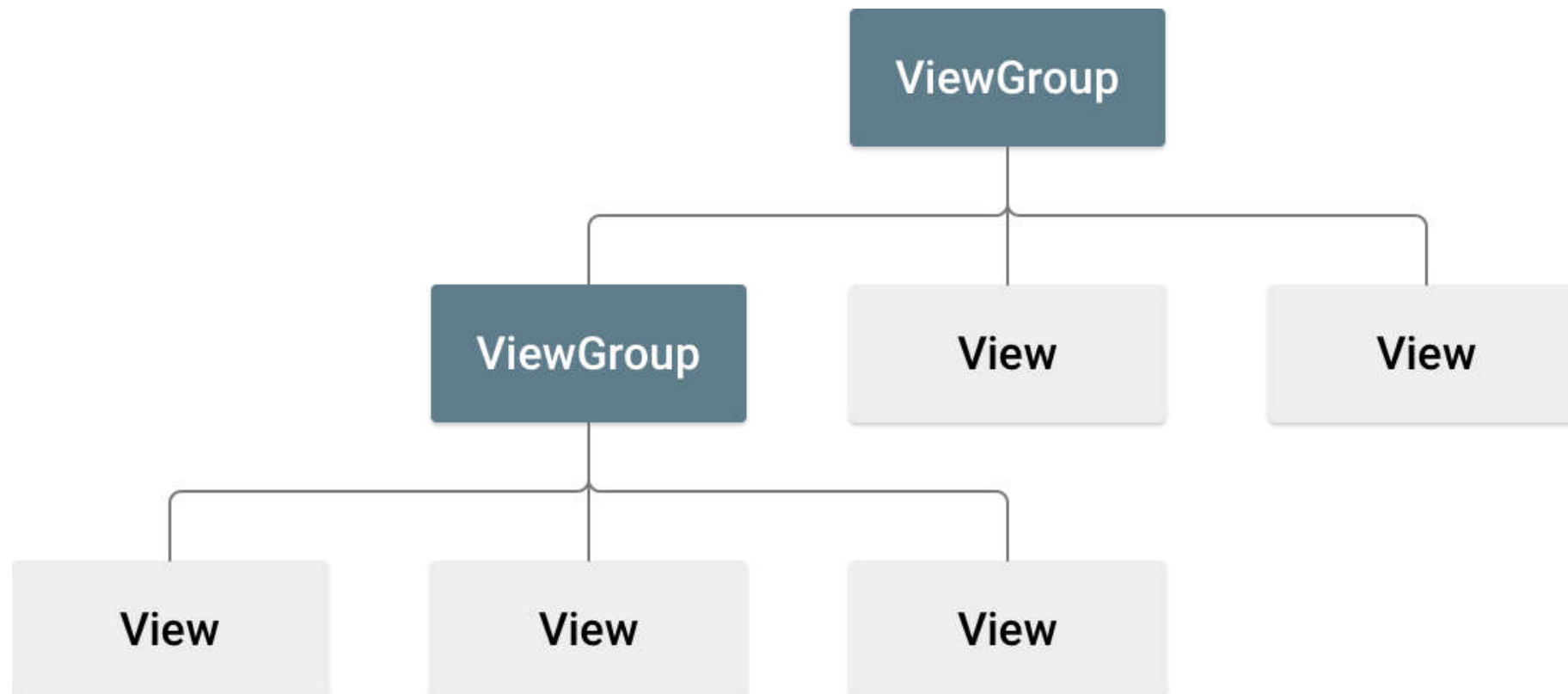
- Hiểu về các thành phần của giao diện
- Lập trình sự kiện trong Android
- Một số control thông dụng và nâng cao
- Webkit
- Cài đặt đa ngôn ngữ trên ứng dụng Android



Các thành phần của giao diện

- ❖ Giao diện (UI) của ứng dụng là những cái gì mà người dùng có thể nhìn thấy và tương tác.
- ❖ Android cung cấp nhiều thành phần UI được xây dựng sẵn như: các đối tượng bố cục (layout) có cấu trúc và các điều khiển UI cho phép xây dựng UI đồ họa cho ứng dụng.
- ❖ Android cũng cung cấp nhiều module UI khác cho các giao diện đặc biệt như dialogs, notifications, và menus.

- Layout xác định cấu trúc hiển thị cho UI (trong Activity). Tất cả các thành phần trong layout được xây dựng kế thừa các đối tượng View và ViewGroup.
- View thường đưa ra một số cái người dùng có thể thấy và tương tác. Trong khi đó, ViewGroup là một nơi chứa không nhìn thấy xác định cấu trúc layout cho View và các đối tượng ViewGroup khác.



➤ Có 2 cách khai báo Layout:

- Khai báo các phần tử UI trong XML.
- Khởi tạo các phần tử layout vào thời điểm chạy.

➤ XML file.

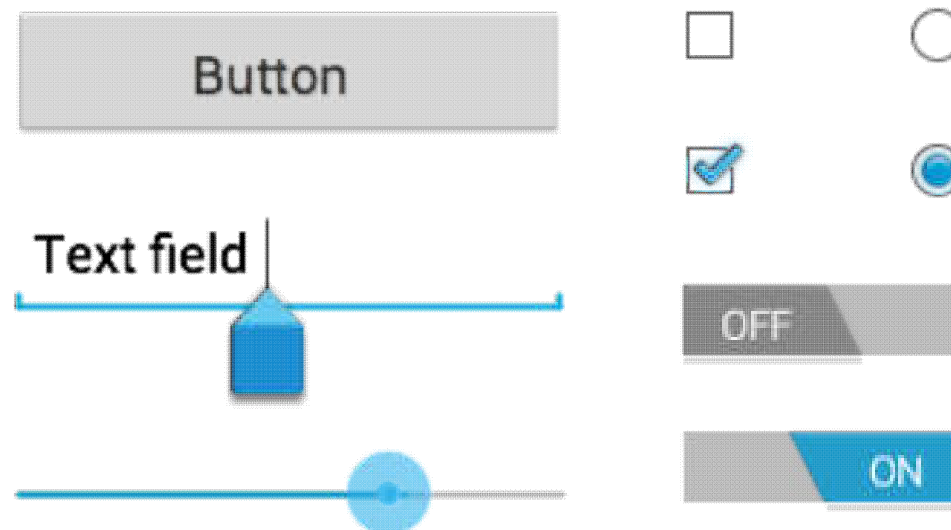
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```



Các kiểu Layout

- **LinearLayout** là một ViewGroup, sắp xếp các đối tượng bên trong theo một hướng duy nhất, chiều dọc hoặc ngang.
- **RelativeLayout** là một ViewGroup, canh các đối tượng theo các vị trí liên quan.
- **TableLayout** là một View, nhóm các View thành dòng và cột.
- **AbsoluteLayout** cho phép chỉ định chính xác của đối tượng.
- **ListView** là một ViewGroup hiển thị danh sách các mục..
- **GridView** là một ViewGroup hiển thị các mục trong một lưới 2 chiều.

- Controls đầu vào là các thành phần tương tác trong giao diện người dùng. Android cung cấp nhiều loại control khác nhau như: ***button***, ***text fields***, ***seek bars***, ***check box***, ***zoom buttons***, ***toggle button***, và nhiều hơn.





Event Handling

- Sự kiện (events) là một cách hữu hiệu để thu thập dữ liệu về việc tương tác của người dùng với các thành phần tương tác của ứng dụng. Như bấm nút hoặc chạm vào màn hình.
- Có 3 khái niệm liên quan đến quản lý sự kiện Android:
 - Event Listeners.
 - Event Listeners Registration.
 - Event Handlers



Event Handling

- **Event Listeners:** là một giao diện trong lớp View, chứa một phương thức gọi lại duy nhất. Các phương thức sẽ được gọi bởi khung Android khi View listener đã đăng ký được kích hoạt bởi người dùng tương tác trên UI.
- **Event Listeners Registration:** Đăng ký Event là quá trình xử lý bởi trình xử lý sự kiện đăng ký với Event Listener để trình xử lý được gọi khi Event Listener kích hoạt sự kiện.



Event Handling

- **Event Handlers:** khi một sự kiện xảy ra và chúng được đăng ký việc lắng nghe sự kiện cho một sự kiện, việc lắng nghe sự kiện gọi là Event Handlers, nó là phương thức xử lý thực sự sự kiện.



Event Handling Examples

```
hesoa = (EditText)findViewById(R.id.editText_hesoA);  
hesob = (EditText)findViewById(R.id.editText_hesoB);  
bt_tong = (Button)findViewById(R.id.button_Tong);  
bt_thuong = (Button)findViewById(R.id.button_Thuong);  
ketqua = (TextView)findViewById(R.id.textView_ketqua);
```

```
a = Integer.parseInt(hesoa.getText().toString());  
b = Integer.parseInt(hesob.getText().toString());
```

```
bt_tong.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        ketqua.setText("Kết quả: " + (a + b));  
    }  
});  
bt_thuong.setOnClickListener((view) → {  
    if(b != 0)  
        ketqua.setText("Kết quả: " + a/b*1.0f);  
    else  
        ketqua.setText("Mẫu số bằng 0, nên không chia được.");  
});
```



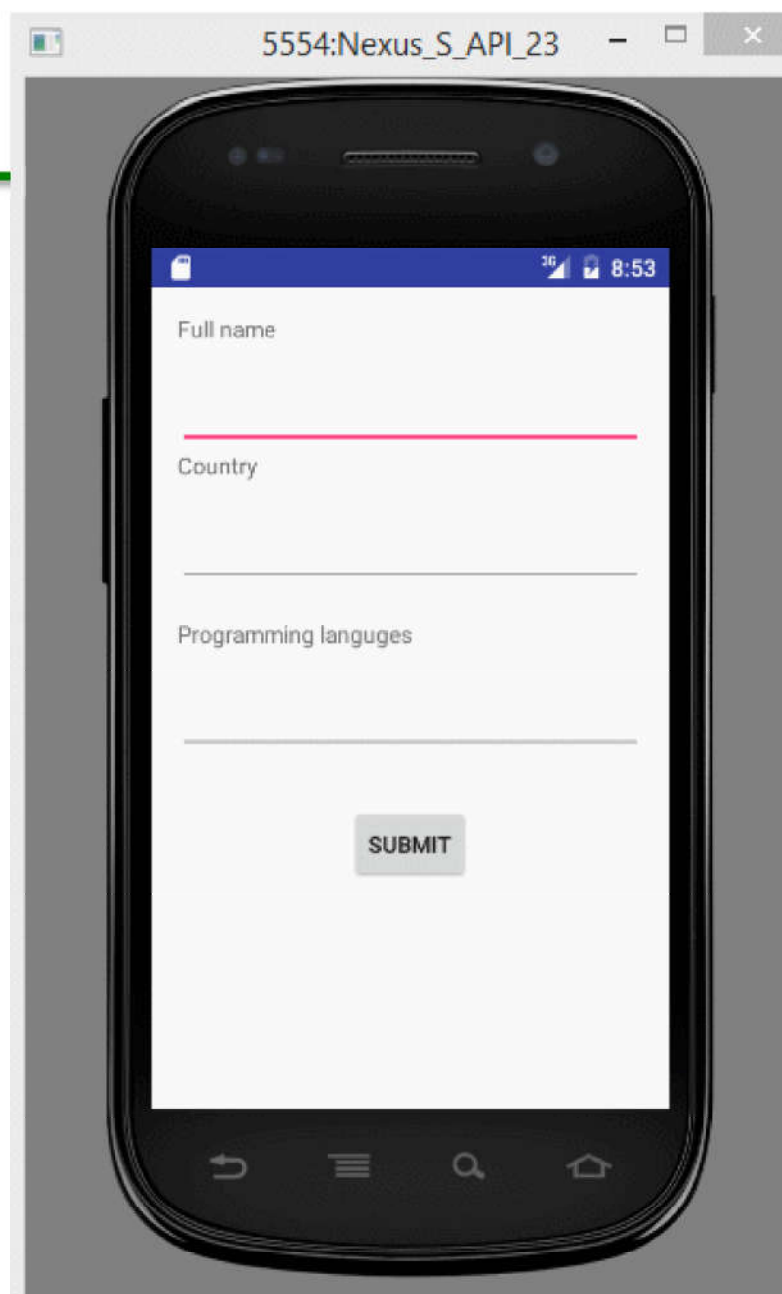
Basic Controls

- TextView, EditText, AutoCompleteTextView, ImageView
- Button, ImageButton, ToggleButton
- CheckBox, RadioButton, RadioGroup
- ProgressBar, Spinner
- TimePicker, DatePicker



Basic Controls

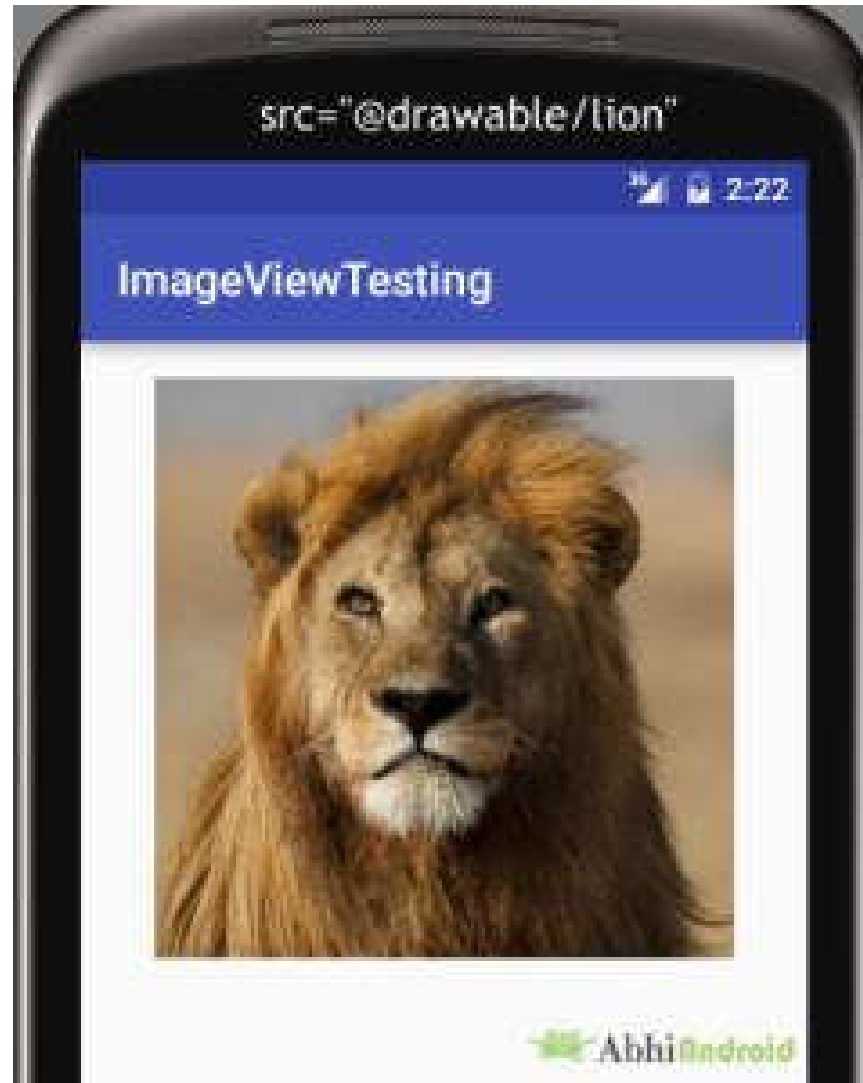
➤ AutoCompleteTextView





Basic Controls

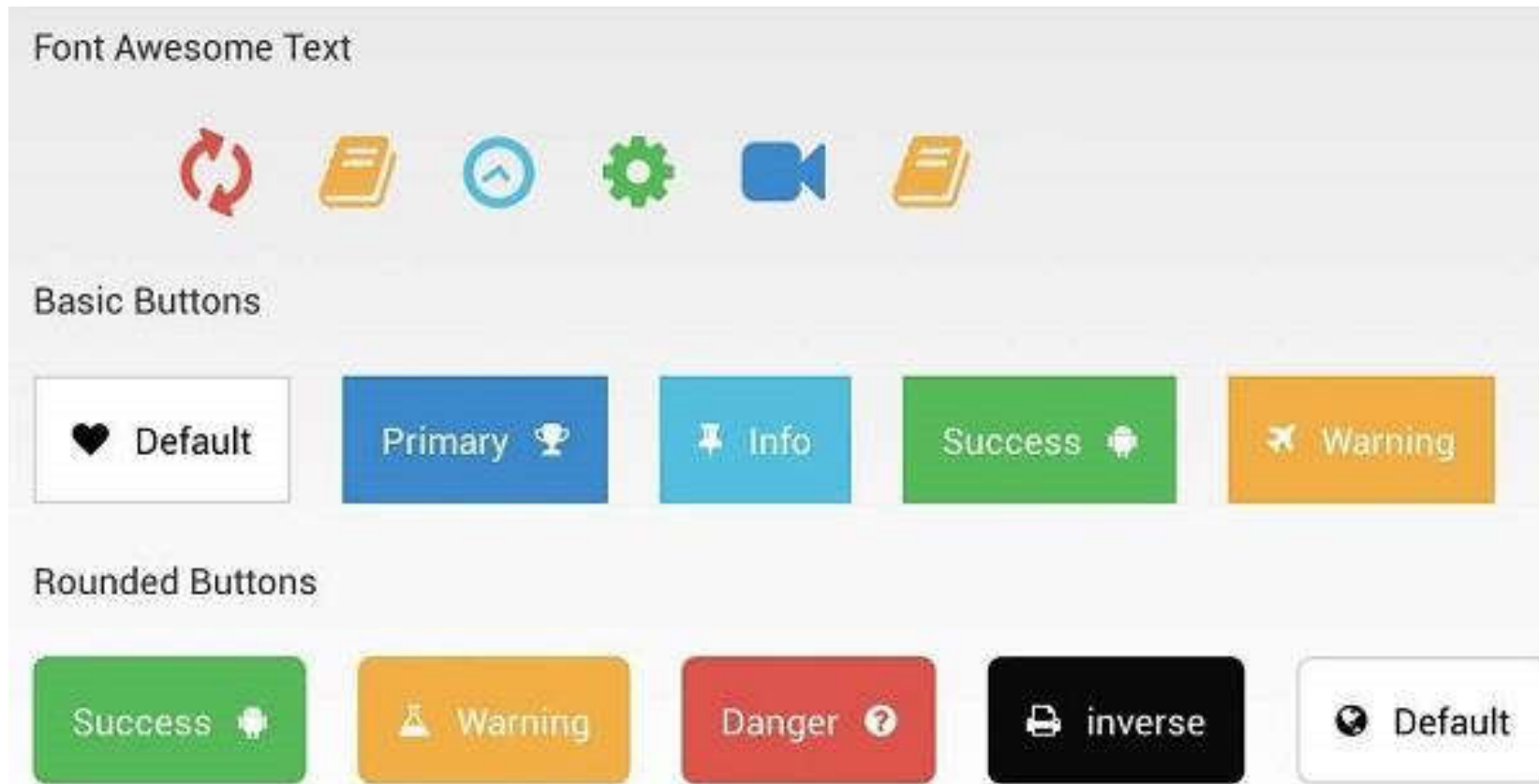
➤ ImageView





Basic Controls

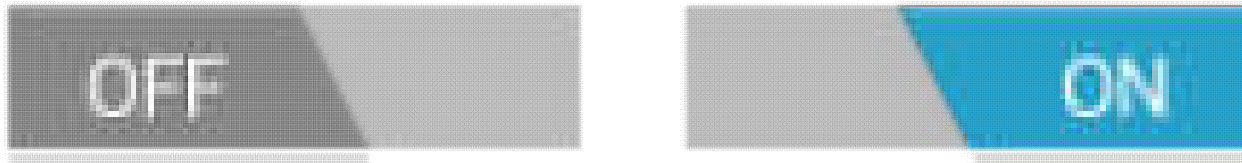
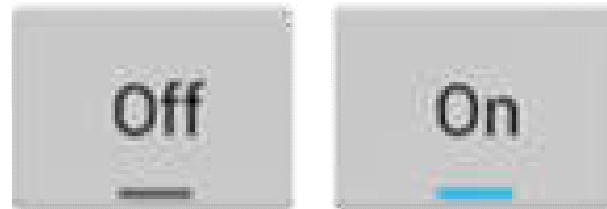
➤ ImageButton





Basic Controls

➤ ToggleButton





Basic Controls

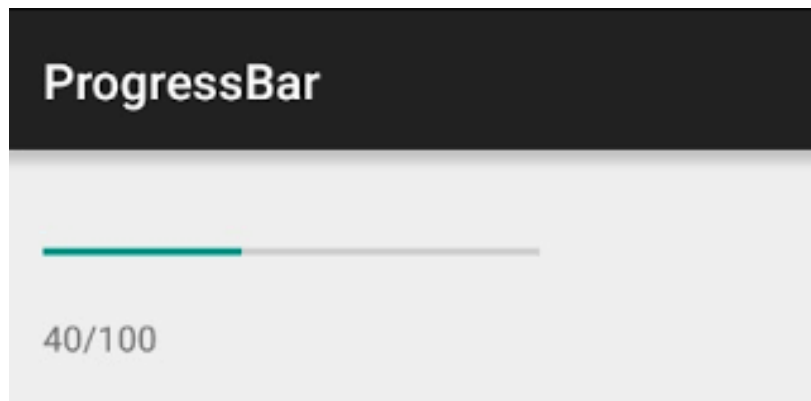
➤ RadioButton

The screenshot shows an Android application interface with a blue header bar labeled 'AndroidTuts'. Below the header, there are two sections of radio buttons. The first section is labeled 'Gender' and contains two options: 'Male' and 'Female'. The 'Female' option is selected, indicated by a red dot in the center of the radio button. The second section is labeled 'Pet' and contains four options: 'Dog', 'Cat', 'Bird', and 'Fish'. The 'Cat' option is selected, also indicated by a red dot in the center of the radio button. At the bottom of the screen, there is a dark gray rounded rectangle containing the text 'Pet: [meow] Cat'.

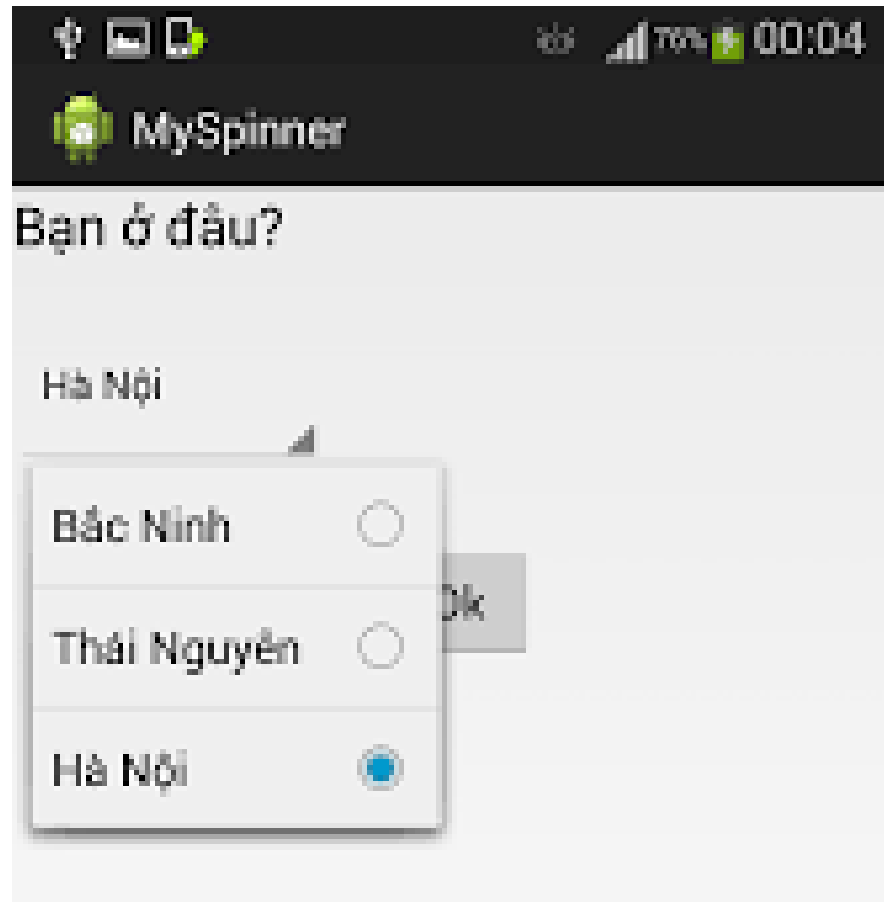


Basic Controls

➤ ProgressBar



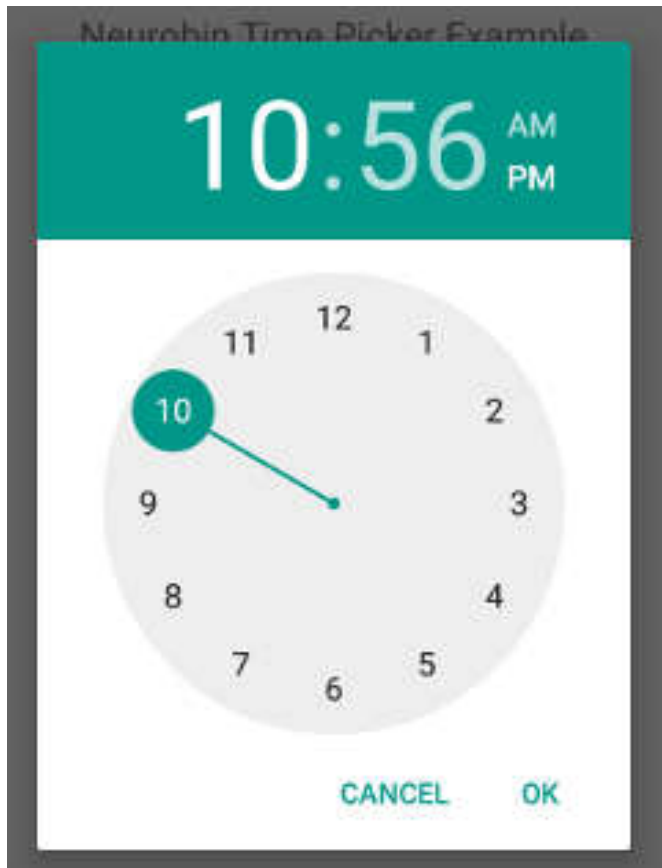
Spinner





Basic Controls

➤ TimePicker, DatePicker





Advence Controls

➤ ListView, GridView, Menu, Dialog

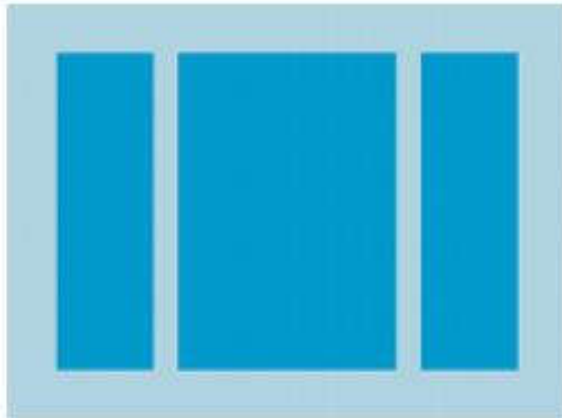


Notifications





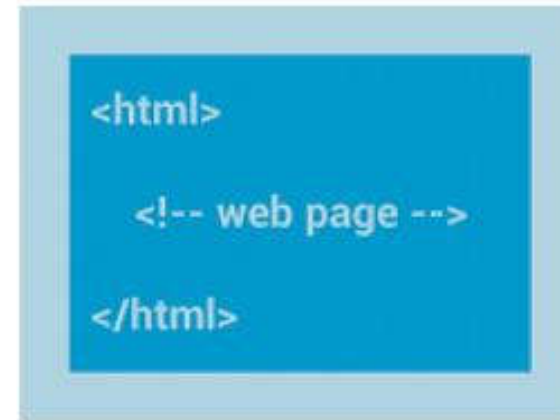
Linear Layout



Relative Layout



Web View



Bố trí các đối tượng thành hàng ngang hay thẳng đứng.

Cho phép chỉ định vị trí của đối tượng so với các đối tượng khác.