

# Giao diện người dùng và xử lý sự kiện

---

GV: TRƯƠNG BÁ THÁI

Email: [truongbathai@tdc.edu.vn](mailto:truongbathai@tdc.edu.vn)

ĐT: 0932.577.765

# I MỤC TIÊU THỰC HIỆN

- Xây dựng được giao diện cho ứng dụng
- Xử lý được các sự kiện cho ứng dụng
- Sử dụng thành thạo IDE Android Studio để viết chương trình Android
- Hình thành thói quen thiết kế chương trình theo tiếp cận Top-Down



# Custom ListView

# | Custom ListView

- Trong thực tế, các ứng dụng Android có liên quan đến ListView thì đa phần chúng ta phải điều chỉnh lại layout cho đúng với yêu cầu của khách hàng. Và cách điều chỉnh lại layout là tạo Custom Layout cho ListView
- Sự khác nhau giữa ListView thông thường và ListView có layout được điều chỉnh lại là ở việc khởi tạo Adapter.

**Sri Lanka**

**Tokelau**

**United Kingdom**

**Uruguay**

**Vatican**

**VietNam**



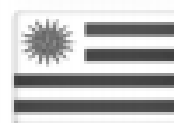
Sri Lanka



Tokelau



United Kingdom



Uruguay



Vatican



VietNam

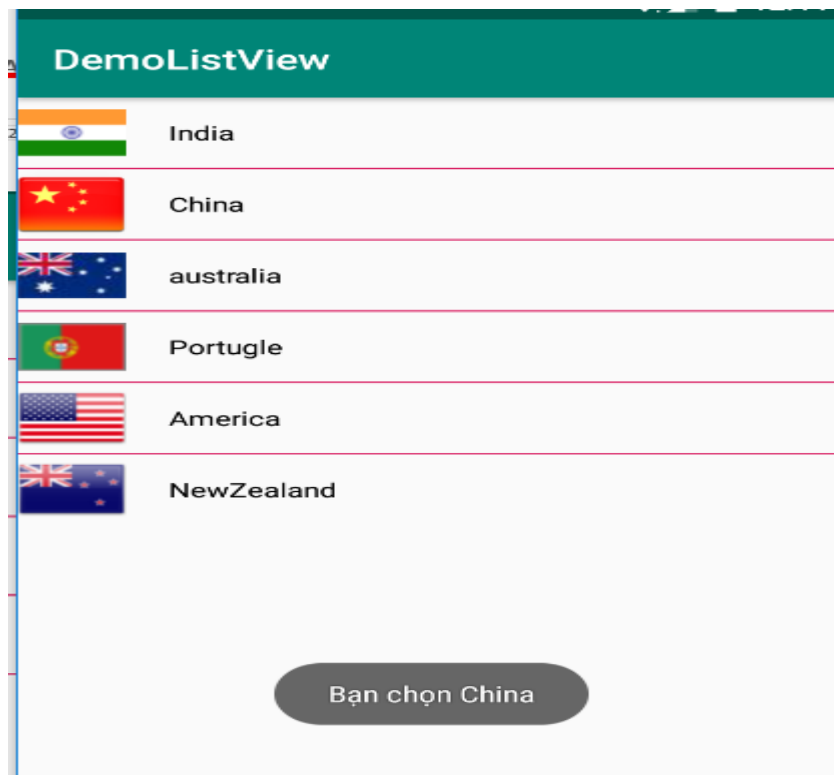
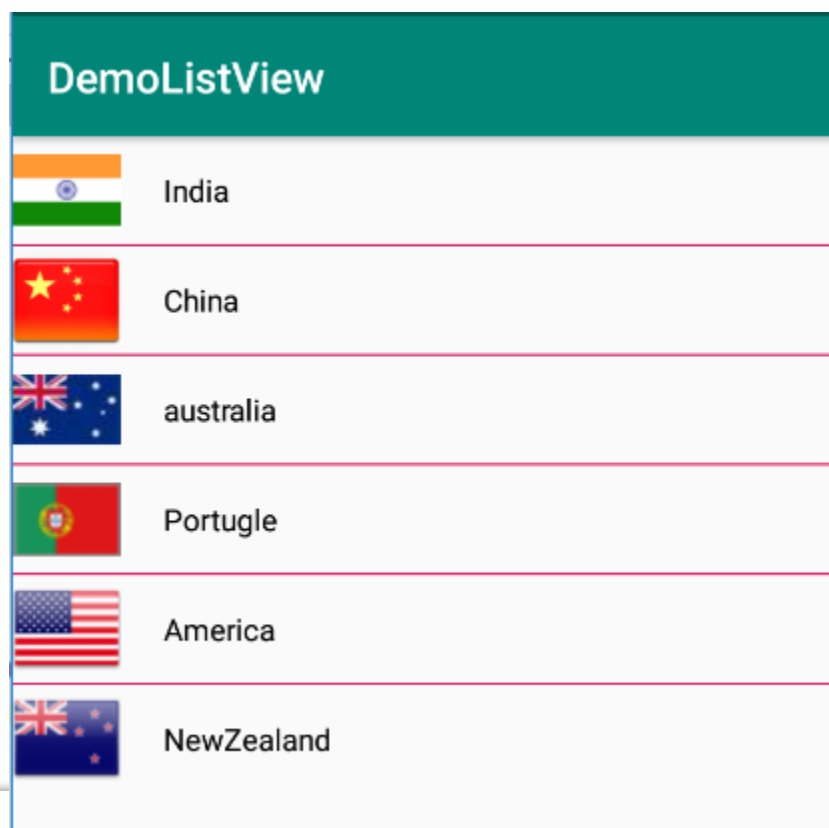


# | Các bước thực hiện

- Tạo một lớp dùng để quản lý dữ liệu.
- Tạo thêm một layout cho một item của ListView.
- Tạo lớp Custom Adapter kế thừa từ lớp ArrayAdapter.
- Hiển thị dữ liệu lên ListView

# Ví dụ:

- Xây dựng ứng dụng gồm có một **ListView** để hiển thị tên các nước cùng với lá cờ nước đó.



# | Bước 1:

- Tạo một project tên là ListViewCustom: **File → New → Android Application Project** điền các thông tin **→ Next → Finish.**



## Bước 2:

- Mở **res** → **layout** → **xml** (hoặc)  
**activity\_main.xml**

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <ListView
        android:id="@+id/simpleListView"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:divider="@color/colorAccent"
        android:dividerHeight="1dp"
        android:footerDividersEnabled="false" />
</LinearLayout>
```

## | Bước 3:

- Tạo mới một `activity_listview.xml` vào trong thư mục layout và thêm code sau

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <ImageView
        android:id="@+id/icon"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:src="@drawable/america" />
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="America"
        android:layout_marginLeft="20dp"
        android:textColor="#000" />
</LinearLayout>
```

## | Bước 4:

- Tạo mới một lớp CustomAdapter.java bên trong package và thêm code sau

```
public class CustomAdapter extends ArrayAdapter {  
    Context context;  
    String countryList[];  
    int flags[];  
    int layoutID;  
    public CustomAdapter(Context context, int layoutID,String[] countryList,int  
flags[]) {  
        super(context,layoutID);  
        this.context = context;  
        this.countryList = countryList;  
        this.flags = flags;  
        this.layoutID = layoutID;  
    }  
}
```

### **@Override**

```
public int getCount() {  
    // TODO Auto-generated method stub  
    return countryList.length;  
}
```

### **@Override**

```
public View getView(int position, View convertView, ViewGroup parent) {  
    // TODO Auto-generated method stub  
    LayoutInflater inflater = (LayoutInflater.from(context));  
    convertView = inflater.inflate(layoutID, null);  
    TextView country = (TextView) convertView.findViewById(R.id.textView);  
    ImageView icon = (ImageView) convertView.findViewById(R.id.icon);  
    country.setText(countryList[position]);  
    icon.setImageResource(flags[position]);  
    return convertView;  
}  
}
```

## | Bước 5:

- Mở **app** → **src** → **MainActivity.java** và thêm code Trong bước này chúng ta khởi tạo **ListView**. Tiếp theo, chúng ta tạo 1 mảng cho image, 1 mảng cho tên image. Lưu các hình ảnh vào thư mục drawable.

```

public class MainActivity extends AppCompatActivity {
    //Khai báo các widget
    ListView simpleList;
    String countryList[] = {"India", "China", "australia", "Portugle", "America",
        "NewZealand"};
    int flags[] = {R.drawable.india, R.drawable.china, R.drawable.australia,
        R.drawable.portugle, R.drawable.america, R.drawable.new_zealand};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        setControl();
        setEvent();
    }
    private void setControl() {
        simpleList = (ListView) findViewById(R.id.simpleListView);
    }
}

```



```

private void setEvent() {
    CustomAdapter adapter = new CustomAdapter(this,R.layout.activity_listview,
countryList, flags);
    simpleList.setAdapter(adapter);

    simpleList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
            Toast.makeText(MainActivity.this, "Bạn chọn " + countryList[position],
Toast.LENGTH_SHORT).show();
        }
    });
}
}

```



# TimePickerDialog

# | TimePickerDialog

- Time Picker trong Android cho phép lựa chọn thời gian của ngày trong chế độ hoặc 24 h hoặc AM/PM. Thời gian bao gồm các định dạng hour, minute, và clock. Android cung cấp tính năng này thông qua lớp TimePicker.

# Các phương thức thường dùng của TimePicker

- **setCurrentHour(Integer currentHour):** Phương thức này dùng để thiết lập ngày hiện tại cho một **TimePicker**
- **setCurrentMinute(Integer currentMinute):** Phương thức này dùng để thiết lập ngày hiện tại cho một **TimePicker**
- **getCurrentHour():** Phương thức này lấy giờ hiện tại của **TimePicker**
- **getCurrentMinute():** Phương thức này lấy phút hiện tại của **TimePicker**

- **setIs24HourView(Boolean is24HourView):** Phương thức dùng thiết lập chế độ giờ hiển thị là 24h hay là hiển thị dạng **AM/PM**. Trong phương thức này thiết lập chế độ **true/false**. Nếu giá trị **true** chế độ hiển thị theo 24h, nếu **false** hiển thị theo chế độ **AM/PM**
- **is24HourView():** Phương này kiểm tra xem chế độ hiện tại là **24h** hay **AM/PM**. Phương thức này trả về **true** nếu nó đang ở chế độ **24h** ngược lại **false** đang ở chế độ **AM/PM**
- **setOnTimeChangeListener(TimePicker.OnTimeChangeListener onTimeChangeListener):** Phương thức này thiết lập hàm callback mà chỉ rằng thời gian đã được chỉnh sửa bởi người dùng

# Một số thuộc tính thường dùng của TimePicker

- **android:id:** Là thuộc tính duy nhất của **TimePicker**
- **android:timePickerMode:** thuộc tính này thường được sử dụng để thiết lập chế độ hiển thị dạng đồng hồ là **spinner** hay là **clock**.
- **android:background:** Thuộc tính này thiết lập màu nền hoặc image trong thư mục drawable cho **TimePicker**.
- **android:padding:** Thuộc tính này xác định khoảng cách từ đường viền của **TimePicker** với nội dung nó chứa: **left, right, top or bottom**.

*<TimePicker*

*android:id="@+id/simpleTimePicker"*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content"*

*android:timePickerMode="spinner"*

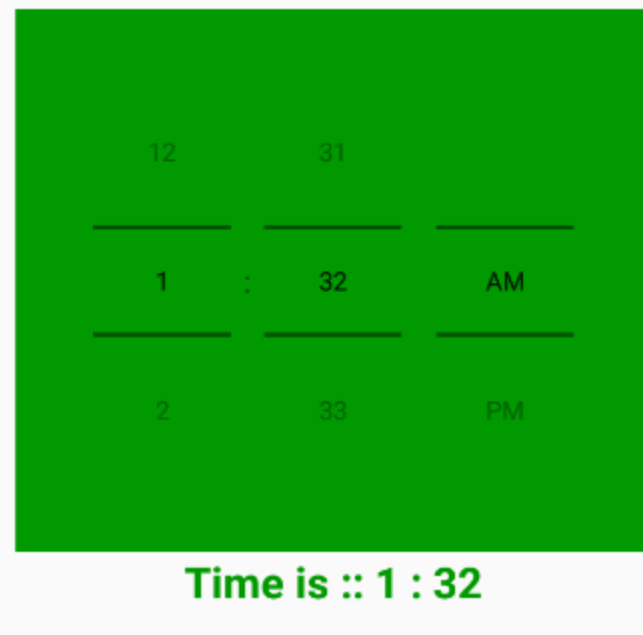
*android:layout\_centerHorizontal="true"*

*android:layout\_marginTop="50dp"*

*android:padding="50dp"/>*

# Ví dụ:

- Xây dựng ứng dụng gồm có một **TimePicker** và một **TextView**. Khi người sử dụng tùy chỉnh giờ trên TimePicker chúng ta sẽ lấy giá trị của nó rồi hiển thị trên **TextView** và **Toast**.





# | Bước 1:

- Tạo một project tên là Demo**TimePicker**: **File** → **New** → **Android Application Project** điền các thông tin → **Next** → **Finish**.

## | Bước 2:

- Mở **res** → **layout** → **xml** (hoặc)  
**activity\_main.xml**

## Bước 3:

- Mở app → src -> **MainActivity.java** và thêm code. Trong bước này chúng ta khởi tạo **TimePicker** và **TextView**. **TextView** dùng để hiển thị thời gian của **TimePicker**. Trong bước này chúng ta thiết lập sự kiện cho **TimePicker** thông qua sự kiện **setOnTimeChangeListener()** chúng ta sẽ lấy giá trị thời gian hiện tại của **TimePicker** sau đó hiển thị ra **TextView** và **Toast**.

```
public class MainActivity extends AppCompatActivity {  
    //Khai báo các widget  
    TextView time;  
    TimePicker simpleTimePicker;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        setControl();  
        setEvent();  
    }  
    private void setControl() {  
        time = (TextView) findViewById(R.id.time);  
        simpleTimePicker = (TimePicker) findViewById(R.id.simpleTimePicker);  
    }
```

```

private void setEvent() {
    simpleTimePicker.setIs24HourView(false); // used to display AM/PM mode
    // perform set on time changed listener event
    simpleTimePicker.setOnTimeChangedListener(new
TimePicker.OnTimeChangedListener() {
        @Override
        public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {
            // display a toast with changed values of time picker
            Toast.makeText(getApplicationContext(), hourOfDay + " " + minute,
Toast.LENGTH_SHORT).show();
            time.setText("Time is :: " + hourOfDay + " : " + minute); // set the
current time in text view
        }
    });
}
}

```



# DatePicker Dialog

# | DatePicker Dialog

- Date Picker trong Android cho phép lựa chọn date bao gồm ngày, tháng, và năm trong Custom UI của. Với tính năng này, Android cung cấp các thành phần DatePicker và DatePickerDialog.

# Các phương thức thường dùng của DatePicker

- **setSpinnersShown(boolean shown):** Phương thức này thường sử dụng để hiển thị DatePicker ở dạng spinner hay không. Trong phương thức này chúng ta có thể thiết lập giá trị **true** hoặc **false**. Nếu giá trị là **false** thì spinner không hiển thị. Mặc định là **true**
- **getDayOfMonth():** Phương thức này được sử dụng để lấy ngày trong tháng từ **DatePicker**. Phương thức này trả về một số nguyên.



- **getMonth():** Phương thức này được sử dụng để lấy tháng từ **DatePicker**. Phương thức này trả về một số nguyên.
- **getYear():** Phương thức này được sử dụng để lấy năm từ **DatePicker**. Phương thức này trả về một số nguyên.
- **getFirstDayOfWeek():** Phương thức này lấy ngày đầu tiên trong tuần. Phương thức này trả về một số nguyên.

# Một số thuộc tính thường dùng của DatePicker

- **android:id:** Là thuộc tính duy nhất của **DatePicker**.
- **android:datePickerMode:** Thuộc tính này thường dùng để thiết lập **DatePicker** theo chế độ **spinner** hay **calendar**.
- **android:background:** Thuộc tính này thiết lập màu nền hoặc image trong thư mục drawable cho **DatePicker**.

- **android:padding:** Thuộc tính này xác định khoảng cách từ đường viền của **TimePicker** với nội dung nó chứa: **left, right, top or bottom.**

# | Ví dụ:

- Xây dựng ứng dụng gồm có một **DatePicker** và một **Button**. Khi người sử dụng chọn ngày trên **DatePicker** và click vào **Button "Submit"** chúng ta sẽ lấy giá trị của **DatePicker** rồi hiển thị lên **Toast**.

< May 2019 >

Apr11

May12

Jun13

SMTWTFSS

1234

567891011

12131415161718

19202122232425

262728293031

SUBMIT

# | Bước 1:

- Tạo một project tên là Demo**DatePicker**: **File** → **New** → **Android Application Project** điền các thông tin → **Next** → **Finish**.

## | Bước 2:

- Mở **res** → **layout** → **xml** (hoặc)  
**activity\_main.xml**

## | Bước 3:

- Mở **app** → **src** -> **MainActivity.java** và thêm code. Trong bước này chúng ta khởi tạo **DatePicker** và **Button**. Sau đó, thiết lập sự kiện **onClickListener()** cho **Button**. Khi người sử dụng click vào **Button** này chúng ta lấy giá trị ngày, tháng, năm của **DatePicker** và hiển thị nó thông qua **Toast**.



```
public class MainActivity extends AppCompatActivity {
```

```
//Khai báo các widget
```

```
DatePicker simpleDatePicker;
```

```
Button btnSubmit;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    setControl();
```

```
    setEvent();
```

```
}
```

```
private void setControl() {
```

```
    simpleDatePicker = (DatePicker) findViewById(R.id.simpleDatePicker);
```

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

```
}
```

```
private void setEvent() {
```

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

```
        @Override
```

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

```
            // TODO Auto-generated method stub
```

```
            String day = "Ngày: " + simpleDatePicker.getDayOfMonth();
```

```
            String month = "\nTháng: " + simpleDatePicker.getMonth();
```

```
            String year = "\n Năm:" + simpleDatePicker.getYear();
```

```
            Toast.makeText(MainActivity.this, day + month + year,
```

```
            Toast.LENGTH_LONG).show();
```

```
        }
```

```
    });
```

```
}
```

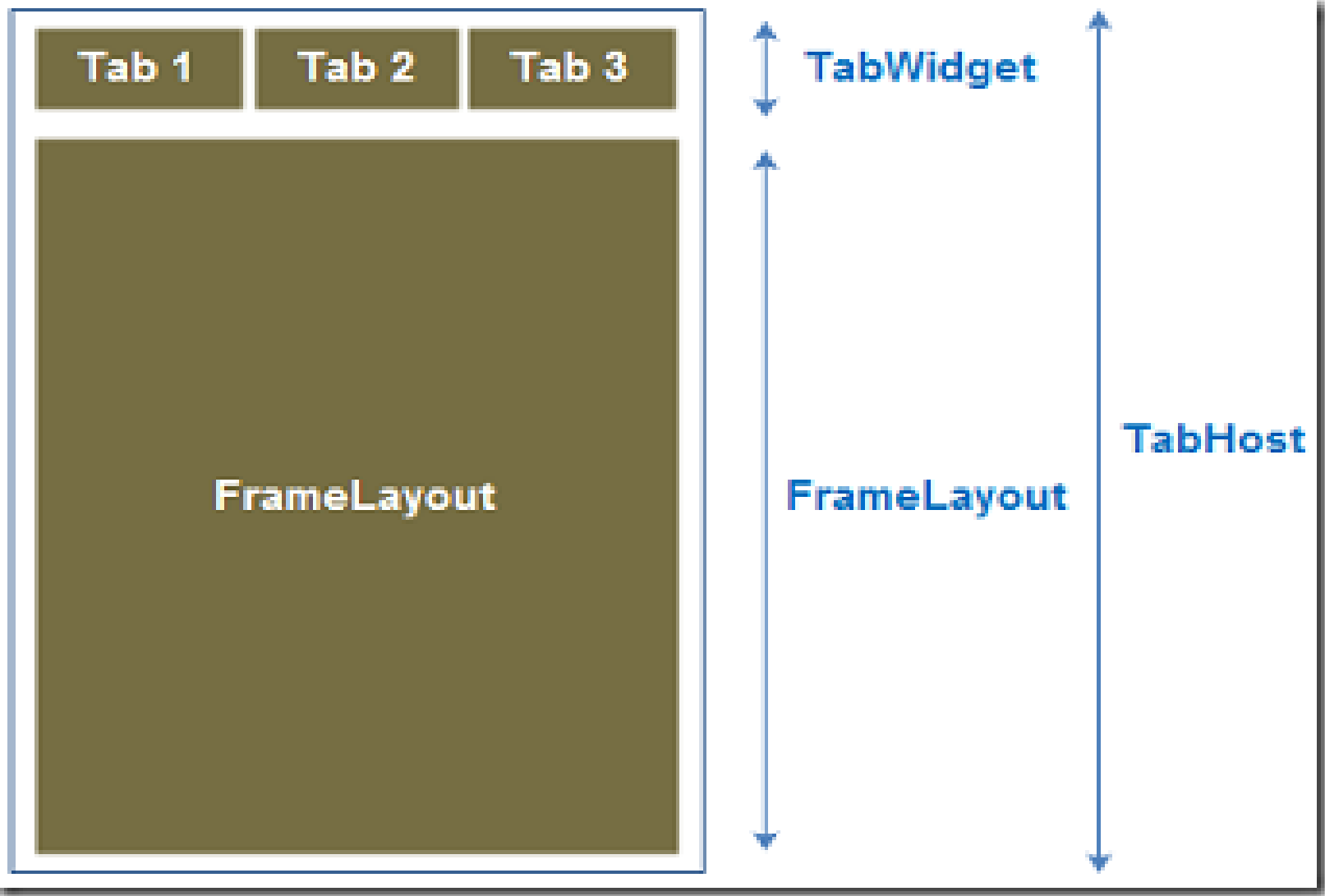
```
}
```



TabHost

# | TabHost

- Tabhost trong android là một dạng giao diện điều khiển bằng thẻ tab cho phép người dùng có thể chuyển đổi các khung hình khác nhau trong cùng một giao diện Activity.



- TabHost: Là container chính chứa các Tab buttons và Tab contents
- TabWidget : Để định dạng cho các Tab buttons : Nhãn, icon, ...
- FrameLayout : là container để chứa các layout cho Tab contents. Chỉ có FrameLayout là view group được dùng cho Tab contents, không thể dùng các loại Layout khác.

# Các phương thức quan trọng của TabSpec

- **setIndicator(CharSequence label):** Phương thức này được sử dụng để thiết lập chuỗi nhãn lên trên tab.
- ***setIndicator(CharSequence label, Drawable icon):*** Phương thức này được sử dụng để thiết lập chuỗi nhãn và một icon lên trên tab

# Một số phương thức quan trọng của Tabhost

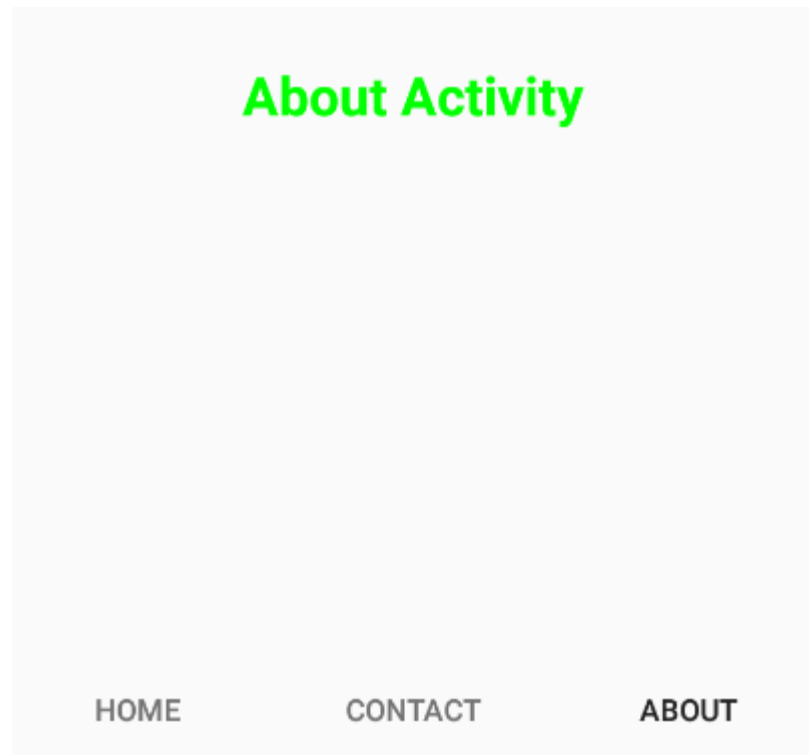
- ***addTab***(TabSpec tabSpec): Phương thức này được sử dụng để thêm một tab mới cho một tab widget.
- ***clearAllTabs***(): Phương thức này được sử dụng xóa tất cả các tab trên TabHost
- ***setCurrentTab***(int index): Phương thức này được sử dụng để thiết lập tab được chọn. Mặc định trong TabHost tab đầu tiên là tab hiện tại.



- **setOnTabChangeListener(OnTabChangeListener)**: Phương thức này được sử dụng khi một tab thay đổi

# | Ví dụ:

- xây dựng ứng dụng sử dụng tabhost dùng để hiển thị 3 tab : Home, Contact, About



# | Bước 1:

- Tạo một project tên là DemoTabHost: **File → New → Android Application Project** điền các thông tin → **Next → Finish.**

## | Bước 2:

- Mở **res** → **layout** → **xml** tạo **layout**

```
<TabHost xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:id="@android:id/tabhost"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent">
```

```
    <LinearLayout
```

```
        android:layout_width="fill_parent"
```

```
        android:layout_height="fill_parent"
```

```
        android:orientation="vertical">
```

```
        <FrameLayout
```

```
            android:id="@android:id/tabcontent"
```

```
            android:layout_width="fill_parent"
```

```
            android:layout_height="0dip"
```

```
            android:layout_weight="1" />
```

```
        <TabWidget
```

```
            android:id="@android:id/tabs"
```

```
            android:layout_width="fill_parent"
```

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

```
            android:layout_marginBottom="-4dp"
```

```
            android:layout_weight="0" />
```

```
    </LinearLayout>
```

```
</TabHost>
```

 activity\_main.xml

# | activity\_about.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:textSize="25sp"
        android:textColor="#0f0"
        android:textStyle="bold"
        android:text="About Activity" />
</RelativeLayout>
```

# | activity\_contact.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="Contact Activity"
        android:textColor="#00f"
        android:textSize="25sp"
        android:textStyle="bold" />
</RelativeLayout>
```

# | activity\_home.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="Home Activity"
        android:textColor="#f00"
        android:textSize="25sp"
        android:textStyle="bold" />
</RelativeLayout>
```



# Bước 3:

- Mở **app** → **src** và thêm code.

```
public class MainActivity extends TabActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        TabHost tabHost = (TabHost) findViewById(android.R.id.tabhost);  
        TabSpec spec;  
        Intent intent;  
        spec = tabHost.newTabSpec("home");  
        spec.setIndicator("HOME");  
        intent = new Intent(this, HomeActivity.class);  
        spec.setContent(intent);  
        tabHost.addTab(spec);  
        spec = tabHost.newTabSpec("Contact");  
        spec.setIndicator("CONTACT");
```

```

intent = new Intent(this, ContactActivity.class);
spec.setContent(intent);
tabHost.addTab(spec);
spec = tabHost.newTabSpec("About");
spec.setIndicator("ABOUT");
intent = new Intent(this, AboutActivity.class);
spec.setContent(intent);
tabHost.addTab(spec);
tabHost.setCurrentTab(1);
tabHost.setOnTabChangeListener(new TabHost.OnTabChangeListener() {
    @Override
    public void onTabChanged(String tabId) {
        // display the name of the tab whenever a tab is changed
        Toast.makeText(getApplicationContext(), tabId,
Toast.LENGTH_SHORT).show();
    }
});
}
}

```

# AboutActivity.java

```
public class AboutActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_about);  
    }  
}
```

# | ContactActivity.java

```
public class ContactActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_contact);  
    }  
}
```

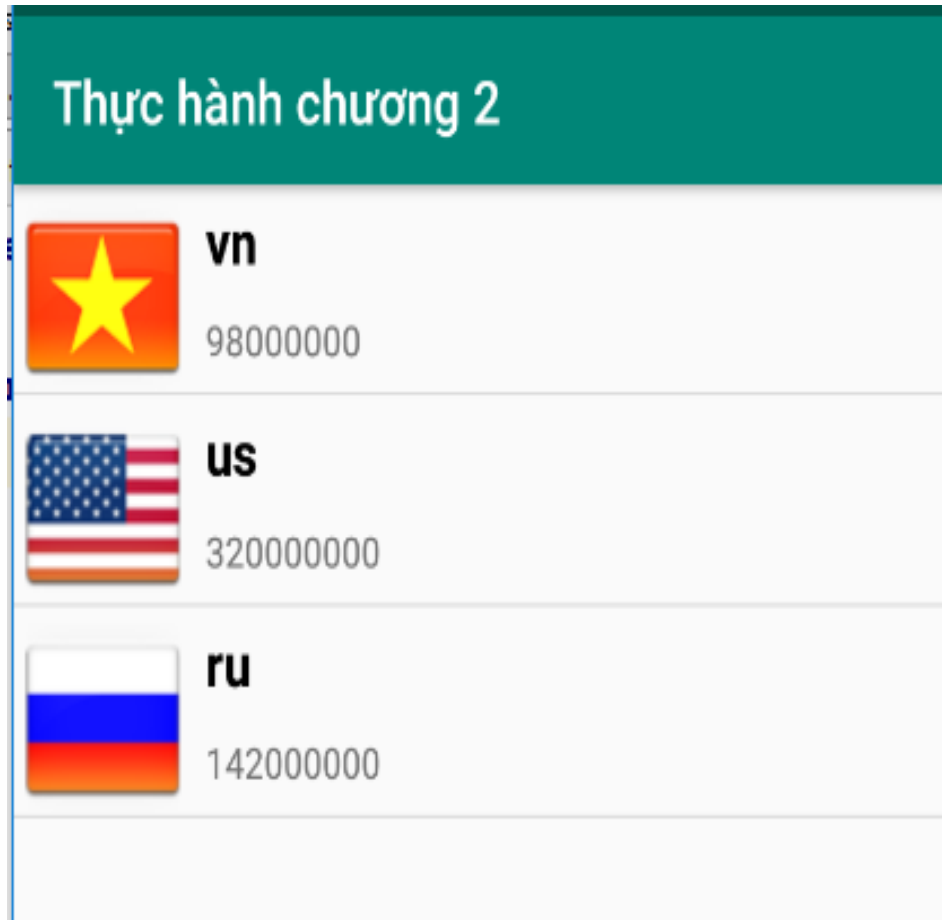
# | HomeActivity.java

```
public class HomeActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_home);  
    }  
}
```



Thực hành

# Bài 9: Xây dựng listview country



# Bài 10: Xây dựng chương trình đặt đồ ăn

## Thực hành chương 2

Đặt thức ăn và đồ uống

CHỌN THỨC ĂN

CHỌN THỨC UỐNG

THOÁT

Bún bò - Pepsi

## Thực hành chương 2



Bún bò



Cháo



Bánh canh



Phở

## Thực hành chương 2



Cô ca cô la



Pepsi











Nước lọc







Bia Heniken



# Bài 11: Xây dựng chương trình danh sách cầu thủ


Thực hành chương 2		
	<b>Xuan Truong</b> 28/4/1995 (23 age)	
	<b>David Beckham</b> 2/5/1975 (43 age)	
	<b>Cris Ronaldo</b> 5/2/1985 (34 age)	
	<b>Kylian Mbappe</b> 20/12/1998 (20 age)	

# Bài 12: Xây dựng chương trình hiển thị danh sách cầu thủ và cập nhật danh sách

Thực hành chương 2		
List Players		
	Xuân Trường	
	98	vn
	beckcam	
	320	fc
	cr7	
	142	db
	mbabe	
	14	db

Thực hành chương 2

Edit Players

 Xuân Trường

vn


98

OK

CANCEL


Thực hành chương 2


List Players


 Xuân Trường

98

vn

 beckcam



 10

14

db

**DELETE PLAYER**

Are you sure to delete this player?

NO

YES

# Bài 13: Hiện thị danh sách film và cập danh sách film

Thực hành chương 2

## List Films



### Chú chó nhỏ

(Bichon là chú chó thông minh, vui vẻ, đáng yêu, giàu tình cảm và chúng cũng rất cần được yêu thương. Chúng rất thích gần gũi



### Chú mèo to

(Sau khi chủ nhân ban đầu qua đời, Bronson bị bỏ rơi tại West Michigan Humane Society. Kích thước của chú mèo khiến mọi

Thực hành chương 2

## Edit Films

### Chú chó nhỏ

Bichon là chú chó thông minh, vui vẻ, đáng yêu, giàu tình cảm và chúng cũng rất cần được yêu thương. Chúng rất thích gần gũi, thân thiện

Score



OK

CANCEL

# Bài 14: Xây dựng chương trình quản lý danh bạ

**Thực hành chương 2**

**DANH BẠ**

Name: Trương Bá Thái

Phone: 0932577765

**SAVE**

Trương Bá Thái[0932577765]

**Thực hành chương 2**

**DANH BẠ**

Name: Trương Bá Thái

Phone: 0932577765

**Call- Sms**

Call to 0932577765

Send sms to 0932577765

Remove This Contact



**CẢM ƠN TẤT CẢ  
ĐÃ LẮNG NGHE**