# LAB 3. CÁC VIEW CƠ BẢN: KHAI BÁO, ÁNH XẠ VÀ ỨNG DỤNG

## A. MỤC TIÊU

Sau khi học xong bài này, sinh viên cần nắm rõ cách khai báo, ánh xạ và ứng dụng được các loại view cơ bản như: Textview, EditText, Listview, Gridview , Scrollview, Spinner, Button, Toast….Sinh viên có thể sử dụng thành thạo các view này trong lập trình Android trên công cụ Android Studio.

## B. NỘI DUNG

- Khai báo, ánh xạ các loại view cơ bản như: Textview, EditText, Listview, Gridview , Scrollview, Spinner, Button, Toast….

- Xử lý các sự kiện đơn giản tương ứng với các đối tượng này.

## C. YÊU CẦU PHẦN CỨNG, PHẦN MỀM

- Hệ điều hành: Windows 7 32-64 bit trở lên./ MAC OS X 10.8.5 trở lên.

- Bộ nhớ trong: Tối thiểu 4GB RAM, yêu cầu 8GB RAM.

- Ổ cứng: Tối thiểu 2GB trống cho ổ cứng, khuyến nghị 4GB (500MB cho IDE và 1.5GB cho Android SDK và image giả lập hệ thống).

- Tỉ lệ màn hình**:** tối thiểu 1280x800.

## D. KẾT QUẢ SAU KHI HOÀN THÀNH

- Sinh viên có thể tạo được một Android project và tự thực hành được các loại view cơ bản vào trong ứng dụng của mình**.**

**-** Kéo thả một số controls tùy ý

**-** Chuyển đổi qua lại giữa các loại layout và quan sát cách bố trí controls

**-** Chạy ứng dụng trên emulator

## E. HƯỚNG DẪN CHI TIẾT

### 1. Listview

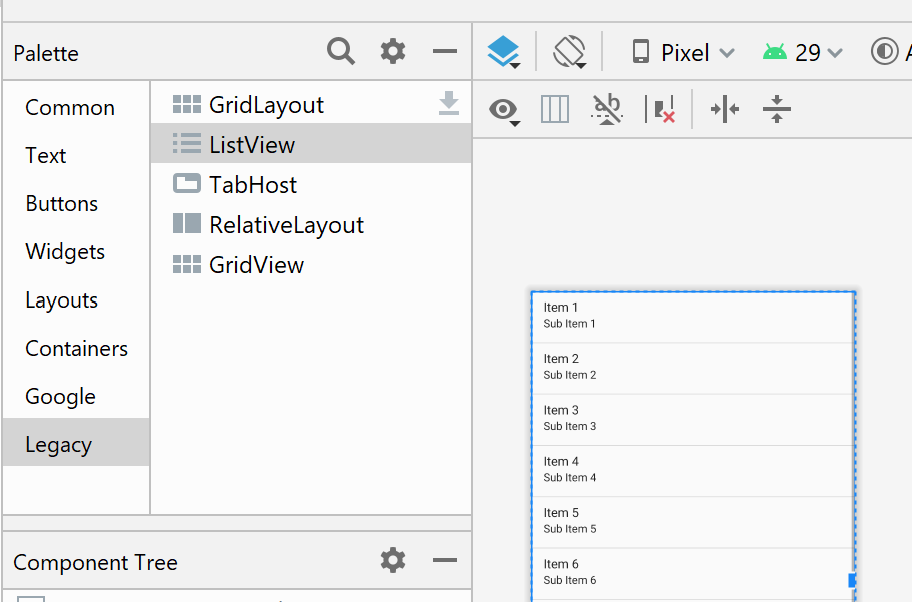
ListView là một danh sách các item trong đó dữ liệu được hiển thị dưới dạng một danh sách có thể cuộn được. Sau đó, người dùng có thể chọn bất kỳ mục danh sách nào bằng cách nhấp vào item đó.

ListView được sử dụng rất rộng rãi trong các ứng dụng Android. Một ví dụ rất phổ biến về ListView là danh bạ điện thoại, nơi bạn có danh sách các liên hệ của mình được hiển thị trong ListView và nếu bạn nhấp vào nó thì thông tin người dùng được hiển thị.

Adapter: Để đổ dữ liệu vào trong Listview ta sử dụng các adapter. Danh sách các item sẽ được thêm một cách tự động vào ListView thông qua adapter từ một nguồn xác định ví dụ như một mảng String hoặc một database chứa dữ liệu.

ListView có thể được đưa vào màn hình ứng dụng thông qua 2 cách:

**Cách 1:** Ở màn hình Design, kéo thả đối tượng ListView nằm trong mục Legacy vào màn hình ứng dụng.



Hình 2.1. Thiết kế ListView trên màn hình Design

**Cách 2:** Sử dụng code XML

Ta khai báo thẻ ListView nằm trong cặp thẻ của Layout cha

<**ListView  
 android:id="@+id/listview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />

Các thuộc tính của ListView:

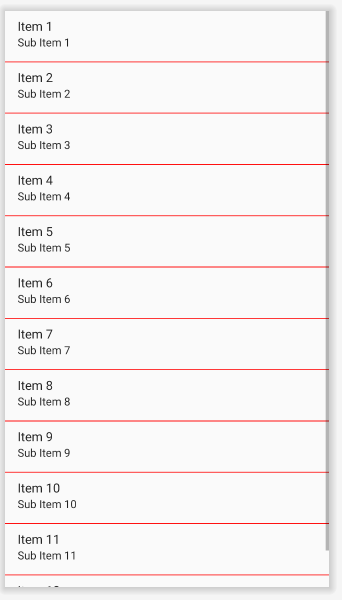
- id : id được sử dụng như một mã định danh duy nhất của ListView.

- divider: là một hình ảnh hoặc đường kẻ màu nhằm phân tách giữa các item trong ListView. Bên dưới là ví dụ khi cho divider màu đỏ xen kẽ các item.

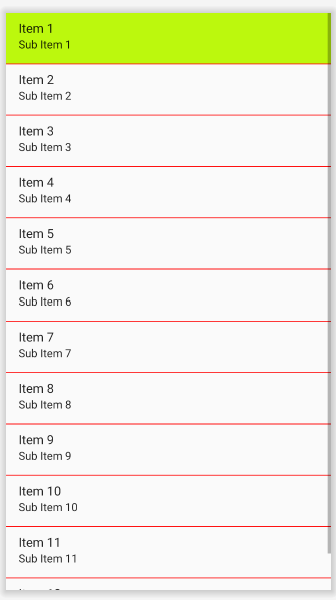
<**ListView  
 android:id="@+id/listview"  
 android:divider="@color/colorAccent"  
 android:dividerHeight="1dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />

- dividerHeight: quy định độ dày của divider,như trong ví dụ trên là 1dp.Độ dày có thể nhận giá trị là dp(density pixel),sp(scale independent pixel) hoặc px(pixel).

- listSelector : quy định màu của item mỗi khi nó được chọn,thông thường sẽ là màu cam hoặc xanh dương nhưng có thể tùy chỉnh theo ý thích.



**android:listSelector="@color/colorPrimary"**



**Sử dụng Adapter trong ListView:**

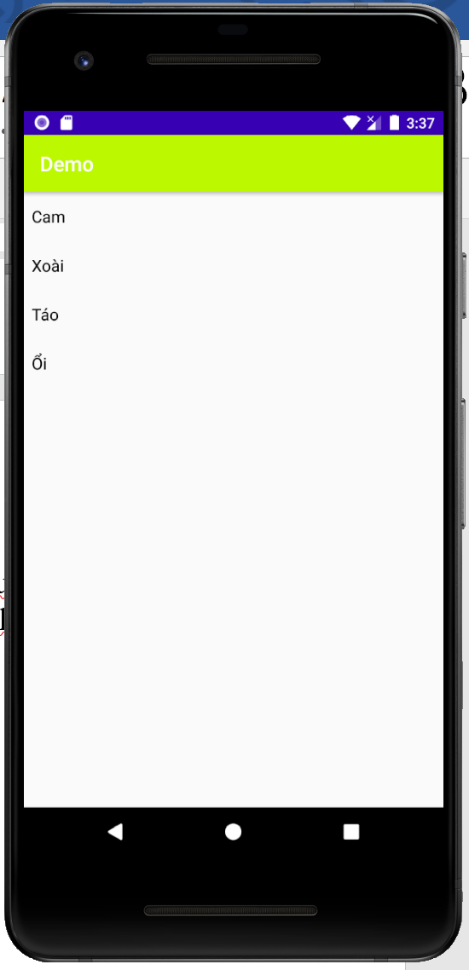
Adapter là cầu nối dữ liệu giữa các thành phần giao diện với nguồn dữ liệu.Nó giữ data và đẩy lên adapter view,sau đó các view có thể lấy những data này từ adapter view và hiển thị data thông qua ListView,GridView,…

Trong Android,những adapter được sử dụng phổ biến là: Array Adapter,Base Adapter (Custom Adapter).

**Array Adapter**

Khi một danh sách các item được lưu trữ dưới dạng 1 mảng, ta có thể sử dụng Array Adapter để hiển thị, tuy nhiên, mặc định Array Adapter chỉ hỗ trợ hiển thị đơn dòng dữ liệu, vì thế nếu cần hiển thị những giao diện phức tạp cho item thì nên custom adapter.

**Ví dụ 1**. Xây dựng ListView sau:



***Bước 1***: Trong file xml thêm đối tượng ListView theo 1 trong 2 cách đã hướng dẫn bên trên.

***Bước 2***: Trong file java khai báo một đối tượng ListView và ánh xạ nó tới ListView đã tạo bên file xml thông qua id.

ListView list;*// khai báo đối tượng*@Override  
protected void onCreate(Bundle savedInstanceState){  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 list=(ListView)findViewById(R.id.listview);  
 *//ánh xạ sang ListView bên XML* list.setDivider(null)

***Bước 3:*** Tạo một list các phần tử kiểu String và add các phần tử vào để làm nguồn cấp dữ liệu.

ArrayList<String> arr=**new** ArrayList<>();

arr.add(**"Cam"**);  
 arr.add(**"Xoài"**);  
 arr.add(**"Táo"**);  
 arr.add(**"Ổi"**);

***Bước 4:*** Khai báo một đối tượng ArrayAdapter với các tham số truyền vào là tên Activity hiện tại,layout của các item và cuối cùng là tên nguồn cấp dữ liệu.

Sau đó setAdapter cho ListView.

ArrayAdapter adapter=**new** ArrayAdapter(**this**,R.layout. ***support\_simple\_spinner\_dropdown\_item***,arr);  
**list**.setAdapter(adapter);

**Base Adapter (Custom Adapter)**

Ta dùng Base Adapter trong trường hợp muốn tùy chỉnh các kiểu hiển thị phức tạp hơn cho ListView (hoặc GridView,….). ArrayAdapter thực chất cũng là sự kế thừa từ Base Adapter.

**Ví dụ 2**: Hiển thị một danh sách tên quốc gia cùng quốc kỳ ra ListView. Ta sử dụng custom adapter sau đây:



***Bước 1:*** Tạo file giao diện chính có chứa 1 ListView.

<ListView  
 android:id="@+id/listview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />

***Bước 2:*** Tạo file giao diện cho mỗi item của ListView.

<ImageView  
 android:id="@+id/imgflag"  
 android:layout\_width="100dp"  
 android:layout\_height="70dp"/>  
<TextView  
android:id="@+id/name"  
android:gravity="center"  
android:textSize="28sp"  
android:textStyle="bold"  
android:layout\_width="300dp"  
android:layout\_height="70dp"/>

***Bước 3:*** Tạo custom adapter.

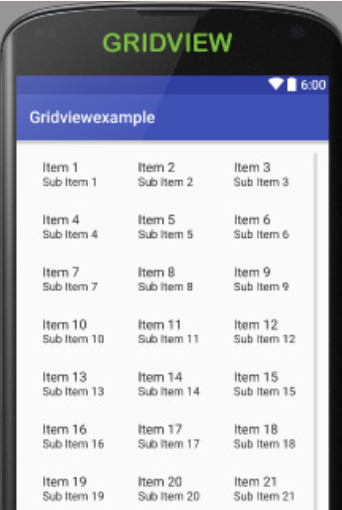
public class CustomAdapter extends BaseAdapter {  
 Context context;  
 String[] list;  
 int[] flags;  
 LayoutInflater inflater;  
  
 public CustomAdapter(Context context, String[] country, int[] flag) {  
 this.context = context;  
 this.list = country;  
 this.flags = flag;  
 inflater = LayoutInflater.from(context);  
 }  
 @Override  
 public int getCount() {  
 return list.length;  
 }  
 @Override  
 public Object getItem(int position) {  
 return null;  
 }  
 @Override  
 public long getItemId(int position) {  
 return 0;  
 }  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 convertView = inflater.inflate(R.layout.list\_item, parent, false);  
 TextView country = (TextView) convertView.findViewById(R.id.name);  
 ImageView flag = (ImageView) convertView.findViewById(R.id.imgflag);  
 country.setText(list[position]);  
 flag.setImageResource(flags[position]);  
 return convertView;  
 }  
}

***Bước 4:*** Gọi vào main và thực thi.

public class MainActivity extends AppCompatActivity {  
 ListView list;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 list=(ListView) findViewById(R.id.listview);  
 list.setDivider(null);  
 String[] country={"Vietnam","England","USA","Swizerland"};  
 int[] flag={R.drawable.vietnam,R.drawable.uk,R.drawable.us , R.drawable.switzerland};  
 CustomAdapter adapter=new CustomAdapter(this,country,flag);  
 list.setAdapter(adapter);  
 }  
}

### 2. GridView

GridView là view dùng để hiển thị danh sách tương tự với ListView, tuy nhiên nó sẽ có thuộc tính cột để chia item ra thành dạng lưới:



Hình 2.2. Màn hình GridView

Khai báo GridView trong file layout tương tự với ListView,và thêm thuộc tính numColums quy định số cột, horizontalSpacing quy định khoảng cách giữa 2 item theo chiều ngang và verticalSpacing là theo chiều dọc.

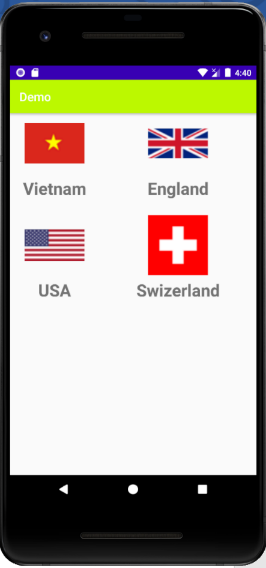
<GridView  
 android:id="@+id/listview"  
 android:numColumns="2"  
 android:verticalSpacing="20dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />

**Ví dụ 3: Xây dựng một GridView**

Kế thừa ví dụ của ListView, chỉ thay đổi trong file giao diện, còn file main và file custom adapter được giữ nguyên. Ngoài thay đổi ở file layout chính, ta cần thay đổi cả ở file layout của item trong list thành dạng vuông vì list giờ sẽ hiển thị theo dạng lưới.

<LinearLayout android:layout\_height="150dp"  
 android:layout\_width="150dp"  
 android:orientation="vertical"  
 xmlns:android="http://schemas.android.com/apk/res/android">  
 <ImageView  
 android:id="@+id/imgflag"  
 android:layout\_gravity="center"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"/>  
 <TextView  
 android:id="@+id/name"  
 android:gravity="center"  
 android:textSize="28sp"  
 android:textStyle="bold"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"/>  
</LinearLayout>

Và kết quả khi chạy chương trình:



### 3. TextView

TextView là một view hiển thị text cho người dùng và có thể cho phép họ chỉnh sửa nó. Thường thì TextView không cho phép chỉnh sửa mà chỉ để hiển thị, tuy nhiên,ta hoàn toàn cho phép nó có thể chỉnh sửa được bằng lập trình.

TextView khi khai báo trong file XML:

<TextView  
 android:id="@+id/textview"  
 android:text="Xin chào các bạn"  
 android:gravity="center"  
 android:textSize="28sp"  
 android:textStyle="bold"  
 android:layout\_width="250dp"  
 android:layout\_height="50dp"/>

Các thuộc tính của TextView:

- id : mã định danh duy nhất của textview,tương tự như các view khác.

- gravity: vị trí của text nằm trong textview,có thể nhận giá trị center,left,right,..

- text: thuộc tính bắt buộc của textview,một view để hiển thị text chắc chắn phải có text.

- textColor: màu chữ.

- textSize: cỡ chữ.

- textStyle: kiểu chữ đậm,nghiêng,gạch dưới.

- padding: khoảng cách của chữ trong TextView so với phần khung của TextView.

- margin: khoảng cách của TextView so với layout phía ngoài hoặc với các view khác.

**Ví dụ 4: Tạo một TextView:**



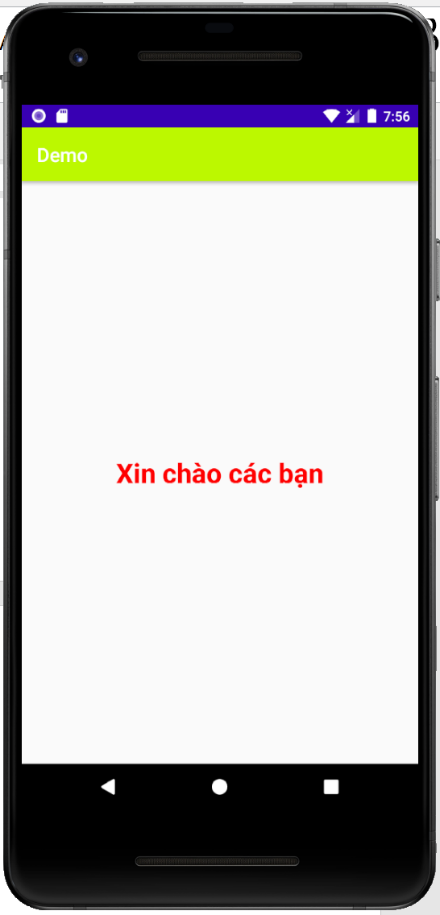
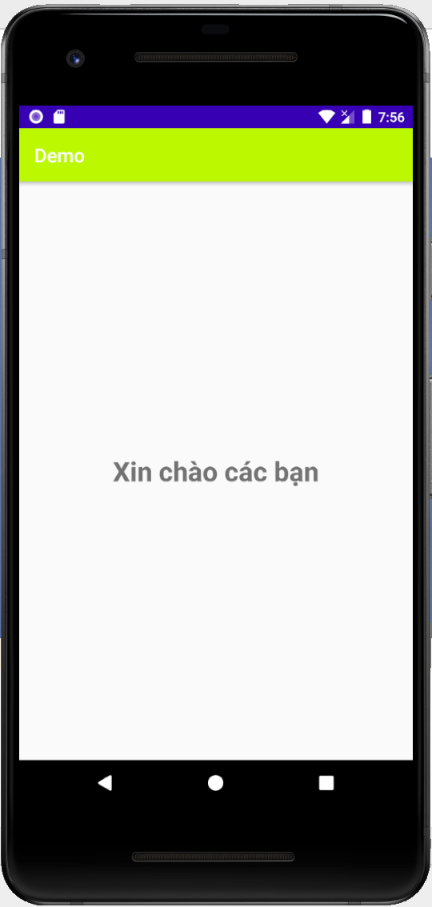
Giờ ta sẽ viết code cho một ví dụ nhỏ với TextView, khi người dùng click vào thì text ở TextView sẽ đổi sang màu đỏ.

***Bước 1:*** Tạo file layout tương tự như trên.

***Bước 2:*** Viết code java xử lý sự kiện onClick

public class MainActivity extends AppCompatActivity {  
 TextView tv;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 tv=(TextView) findViewById(R.id.textview);  
 tv.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 tv.setTextColor(Color.RED);  
 }  
 });  
 }  
}

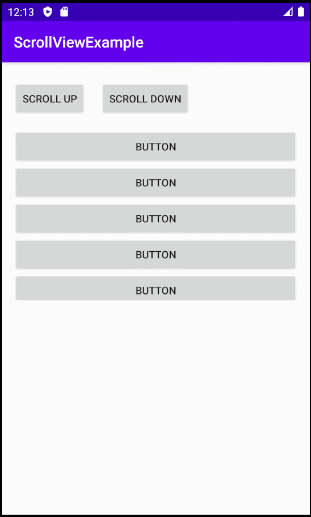
Chạy chương trình:



Trước khi nhấn Sau khi nhấn

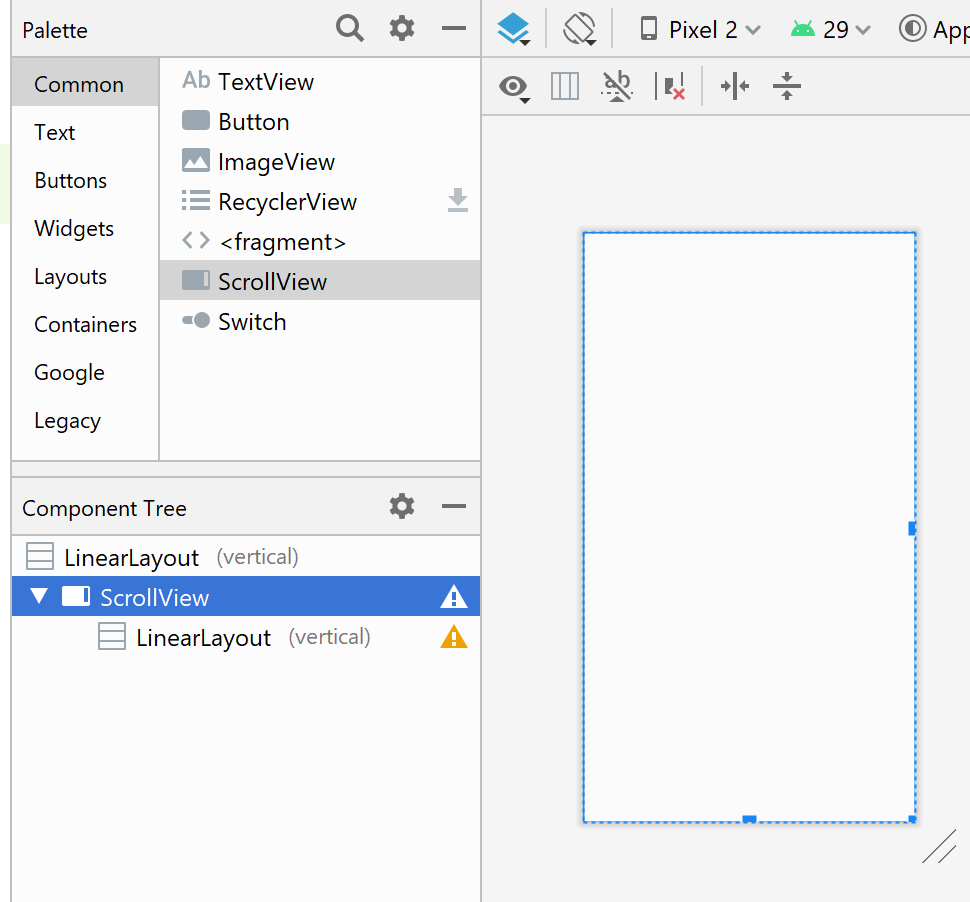
### 4. ScrollView

Trong Android, ScrollView là một kiểu Layout, nó là một bộ chứa (container) hình chữ nhật có thanh cuộn thẳng đứng và có thể chứa một thành phần khác có kích thước lớn hơn nó. Tương tự với ScrollView, HorizontalScrollView là một bộ chứa với thanh cuộn nằm ngang.



ScrollView và HorizontalScrollView chỉ có thể chứa một thành phần con trực tiếp, vì vậy thành phần con của nó thường là một bộ chứa (container) (ví dụ LinearLayout) khác để có thể chứa một hoặc nhiều thành phần con.

Trên Android Studio, khi kéo thả một ScrollView (hoặc HorizontalScrollView) vào giao diện nó sẽ tự động được thêm vào một thành phần con là LinearLayout, ta có thể xóa thành phần con này để sử dụng một thành phần con khác.



Hình 2.3. Màn hình thiết kế ScrollView

Không nên dùng kết hợp ScrollView với ListView hoặc GridView vì 2 view kia vốn dĩ đã có cơ chế scroll của nó rồi.

Các thuộc tính của ScrollView: Tương tự các view khác, ScrollView cũng có những thuộc tính đặc trưng sau:

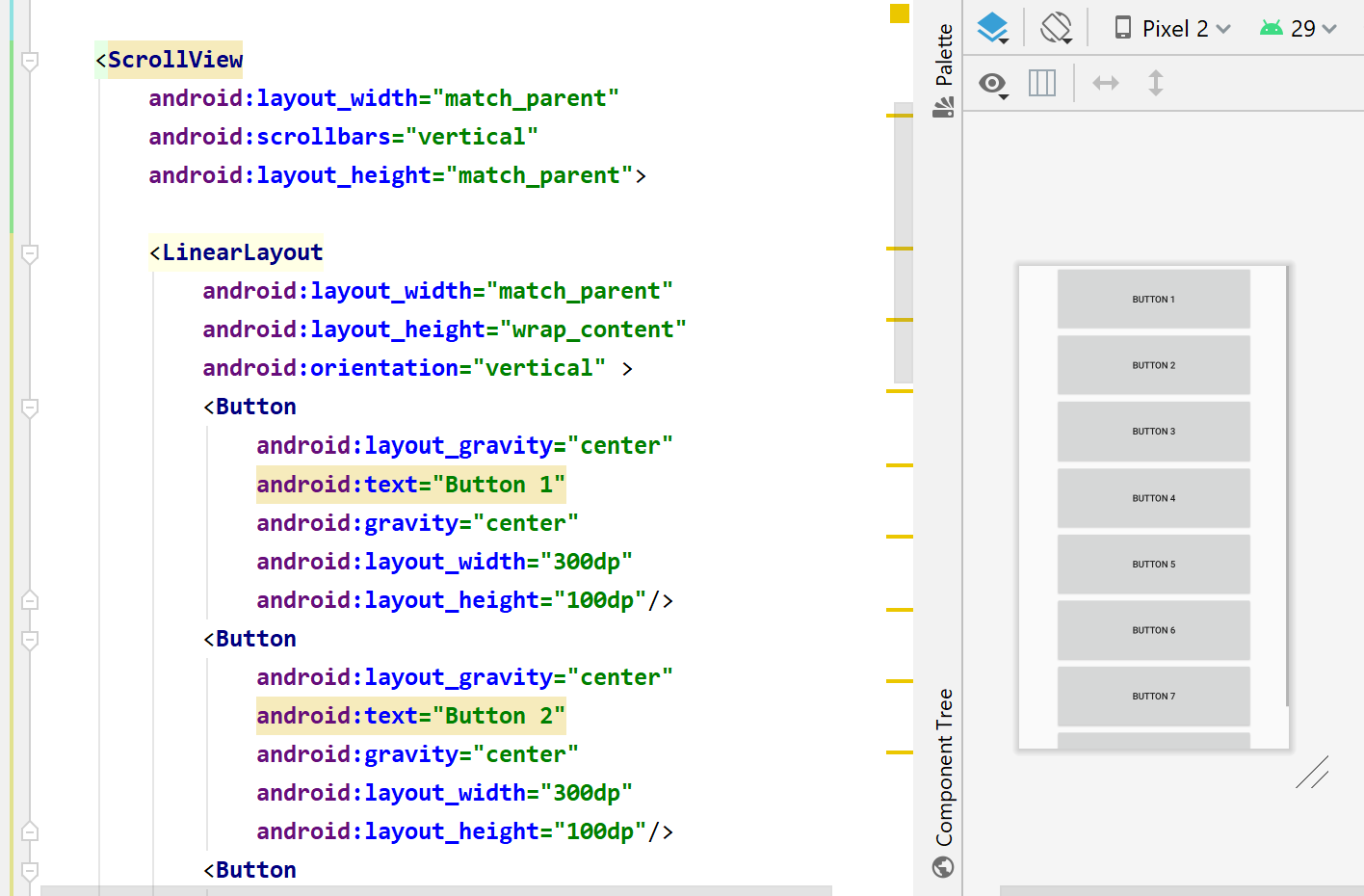
- id: tên định danh của ScrollView.

- scrollbars: chỉ định hướng của thanh trượt là vertical hay horizontal,mặc định là vertical với scrollview và horizontal với horizontalscrollview.

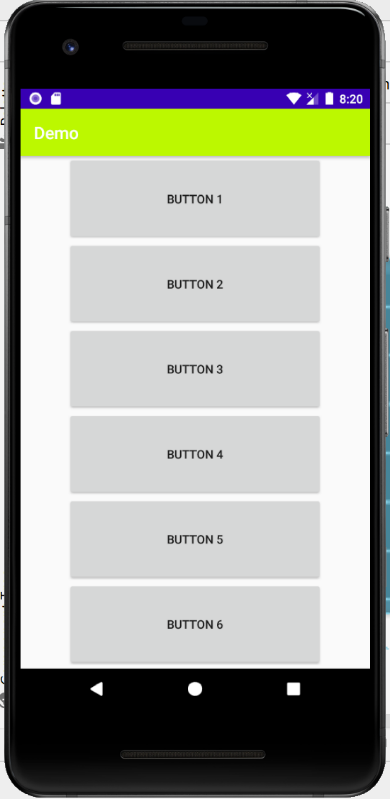
**android:scrollbars="vertical"**

**Ví dụ 5. Tạo một ScrollView**: ví dụ sau sẽ sử dụng 8 nút bấm và dùng ScrollView để cuộn chúng theo chiều thẳng đứng:

***Bước 1***: Tạo file layout với ScrollView và 1 LinearLayout bên trong, trong Linear khai báo 10 thẻ Button.



***Bước 2:*** Chạy chương trình:



### 5. Spinner

Trong Android, spinner là một list các item có thể xổ xuống, tương tự combobox đã từng gặp ở các ngôn ngữ khác. Ở trạng thái mặc định, spinner sẽ hiển thị giá trị của item nó đang chọn.

Spinner là lớp con của lớp AsbSpinner.

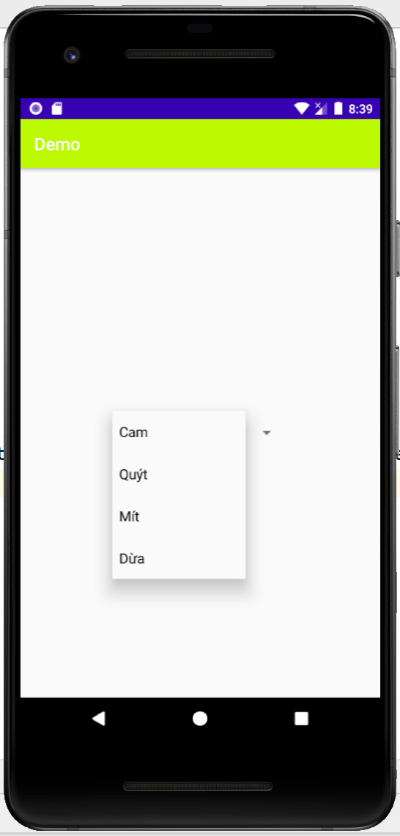
Tương tự ListView, spinner dùng adapter để đổ data vào, mặc định là Array Adapter với kiểu dữ liệu String. Nếu muốn dùng những kiểu hiển thị phức tạp hơn như đã làm với ListView, ta dùng Base Adapter.

Khai báo spinner trong file layout:

<Spinner  
 android:id="@+id/spinner1"  
 android:layout\_width="200dp"  
 android:layout\_height="50dp"/>

**Ví dụ 6.** Xây dựng Spinner với Array Adapter.

Cũng lần lượt các bước như ta đã quen ở các view trước, đầu tiên tạo file layout có Spinner rồi viết code xử lý. Ở đây ta dùng 1 ArrayAdapter để đổ data vào.

public class MainActivity extends AppCompatActivity {  
 Spinner spinner;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 spinner = (Spinner) findViewById(R.id.spinner1);  
 String[] arr = {"Cam", "Quýt", "Mít", "Dừa"};  
 ArrayAdapter adapter = new  
 (ArrayAdapter(this, R.layout.support\_simple\_spinner\_dropdown\_item, arr);  
 spinner.setAdapter(adapter);  
 }  
}

Kết quả sau khi chạy chương trình:

**Ví dụ 7.** Xây dựng Spinner với Base Adapter.

Dựa vào ví dụ 6, ta sẽ hiển thị các item của spinner là tên nước và cờ tương ứng.

***Bước 1***. Khai báo Spinner trong file XML

<ListView  
 android:id="@+id/listview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"

- Tạo file layout cho item.

<ImageView  
android:id="@+id/imgflag"  
android:layout\_width="100dp"  
android:layout\_height="70dp"/>  
<TextView  
android:id="@+id/name"  
android:gravity="center"  
android:textSize="28sp"  
android:textStyle="bold"  
android:layout\_width="300dp"  
android:layout\_height="70dp"/>

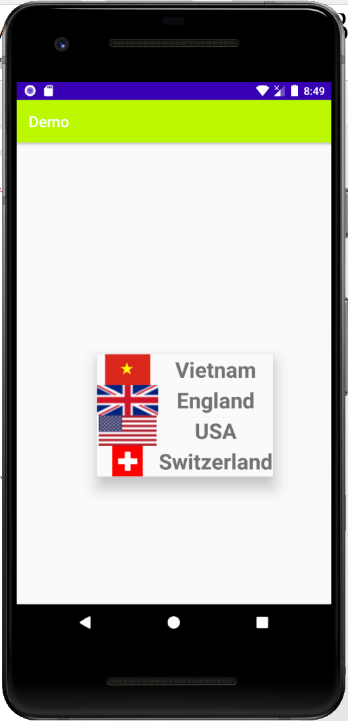
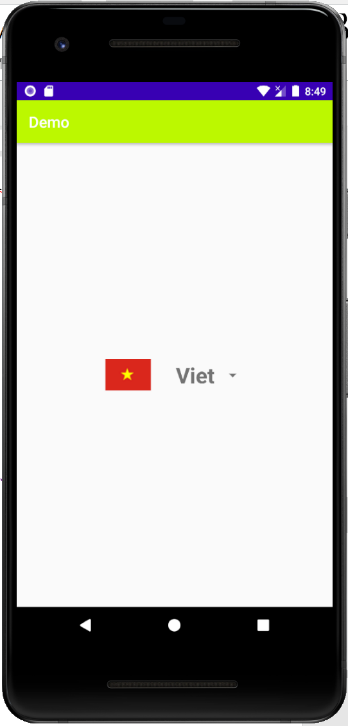
***Bước 2.*** Tạo custom adapter.

public class CustomAdapter extends BaseAdapter {  
 Context context;  
 String[] list;  
 int[] flags;  
 LayoutInflater inflater;  
 public CustomAdapter(Context context,String[] country,int[] flag)  
 {  
 this.context=context;  
 this.list=country;  
 this.flags=flag;  
 inflater=LayoutInflater.from(context);  
 }  
 @Override  
 public int getCount() {  
 return list.length;  
 }  
 @Override  
 public Object getItem(int position) {  
 return null;  
 }  
 @Override  
 public long getItemId(int position) {  
 return 0;  
 }  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent)  
 {  
 convertView=inflater.inflate(R.layout.list\_item,parent,false);  
 TextView country=(TextView) convertView.findViewById(R.id.name);  
 ImageView flag=(ImageView)convertView.findViewById(R.id.imgflag);  
 country.setText(list[position]);  
 flag.setImageResource(flags[position]);  
 return convertView;  
 }  
}

***Bước 3***. Gọi đến CustomAdapter trong main và truyền vào tham số, sau đó chạy chương trình xem kết quả:

public class MainActivity extends AppCompatActivity {  
 Spinner spinner;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 spinner = (Spinner) findViewById(R.id.spinner1);  
 String[] arr = {"Vietnam", "England", "USA", "Switzerland"};  
 int[] flag = {R.drawable.vietnam, R.drawable.uk, R.drawable.us, R.drawable.  
 switzerland};  
 CustomAdapter adapter = new CustomAdapter(this, arr, flag);  
 spinner.setAdapter(adapter);  
 }  
}

**Kết quả:**



### 6. EditText

EditText là một view rất hay dùng trong Android để nhập dữ liệu.Mặc dù không thường xuyên nhưng nó cũng có thể dùng để thay thế TextView hiển thị text.Ứng dụng thường thấy nhất là trong các màn hình đăng nhập,…

**Thuộc tính của EditText:**

- id: tên định danh của EditText.

- text: text sẵn có trong EditText.

- padding: độ co dãn của text nằm trong EditText.

- hint: gợi ý về nội dung nhập vào EditText.

- background: nền của EditText.

**Ví dụ 8. *Xây dựng EditText***: Tạo 2 EditText nhập tên và tuổi và một nút bấm, khi ấn nút sẽ Toast lên thông tin vừa nhập:



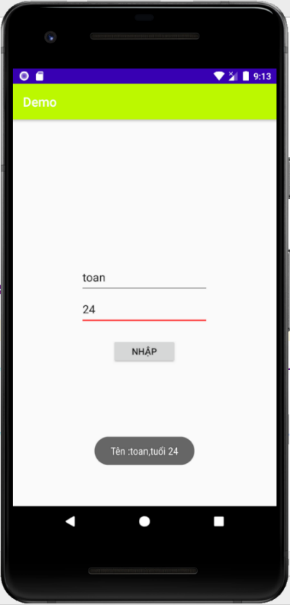
***Bước 1:*** Tạo file layout như đã quen thuộc ở các bài trước.

<EditText  
 android:id="@+id/edtname"  
 android:hint="Nhập tên"  
 android:layout\_width="200dp"  
 android:layout\_height="50dp" />  
<EditText  
android:id="@+id/edtage"  
android:hint="Nhập tuổi"  
android:layout\_width="200dp"  
android:layout\_height="50dp" />  
<Button  
android:id="@+id/btngo"  
android:text="Nhập"  
android:layout\_marginTop="20dp"  
android:layout\_width="100dp"  
android:layout\_height="40dp"/>

***Bước 2:*** Viết code xử lý cho nút bấm.

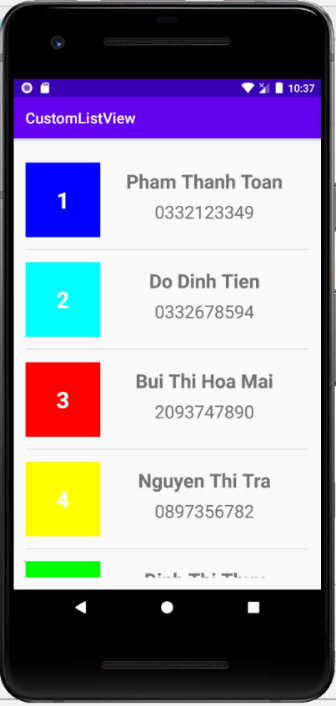
EditText edtname,edtage;  
 Button btngo;  
protected void onCreate(Bundle savedInstanceState){  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 edtname=(EditText)findViewById(R.id.edtname);  
 edtage=(EditText)findViewById(R.id.edtage);  
 btngo=(Button)findViewById(R.id.btngo);  
 btngo.setOnClickListener(new View.OnClickListener(){  
@Override  
public void onClick(View v){if(!edtname.getText().toString().equals("") &&!edtage.getText().toString().equals(""))  
 {  
 String a=edtname.getText().toString();  
 String b=edtage.getText().toString();  
 Toast.makeText(MainActivity.this,"Tên:"+a+",tuổi "+b,  
 Toast.LENGTH\_SHORT).show();  
 }  
 else{  
 Toast.makeText(MainActivity.this,"Không được để trống",  
 Toast.LENGTH\_SHORT).show();  
 }  
 }  
 });  
 }

Chạy chương trình:



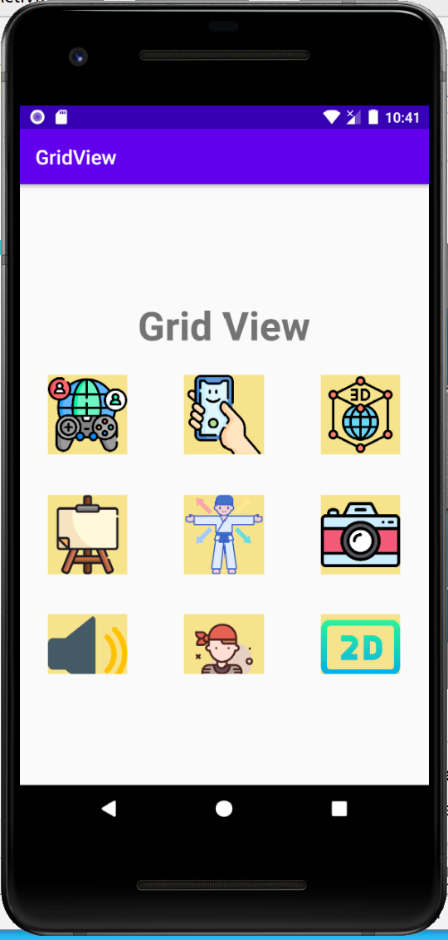
## F. BÀI TẬP ÔN TẬP

**Bài 1**: Tạo ListView với các item là thông tin sinh viên tùy ý bao gồm số thứ tự, họ tên và điện thoại như sau:



Gợi ý: dùng Base Adapter để tạo custom adapter cho dữ liệu,thiết kế layout item bằng linear layout.

**Bài 2**: Tạo GridView chứa các hình ảnh tùy ý như sau:



**Bài 3**: Thiết kế một giao diện có sử dụng ScrollView như hình sau:



Gợi ý: tên “Tin tức sáng” là TextView nằm ngoài ScrollView,trong ScrollView bao gồm một ImageView hiển thị hình ảnh bài viết và TextView chứa nội dung bài viết bất kỳ

**Bài 4**: Thiết kế giao diện có sử dụng Spinner như sau:



Spinner chứa tên 5 màu Đen,Trắng,Đỏ,Xanh lá cây,Xanh Dương

Khi chọn màu nào thì dòng text Xin chào chuyển sang màu đó.

**Bài 5**: Xây dựng ứng dụng xem ảnh sử dụng GridView, ảnh được lấy từ bộ nhớ trong của máy.

- Thiết kế file XML giao diện gồm 1 GridView

- Thiết kế file XML cho item của GridView là các hình vuông

- Tạo Custom Adapter đổ ảnh vào GridView

- Viết code file MainActivity