

# LẬP TRÌNH ĐỘNG

---

Các view dạng list

# Nội dung

---

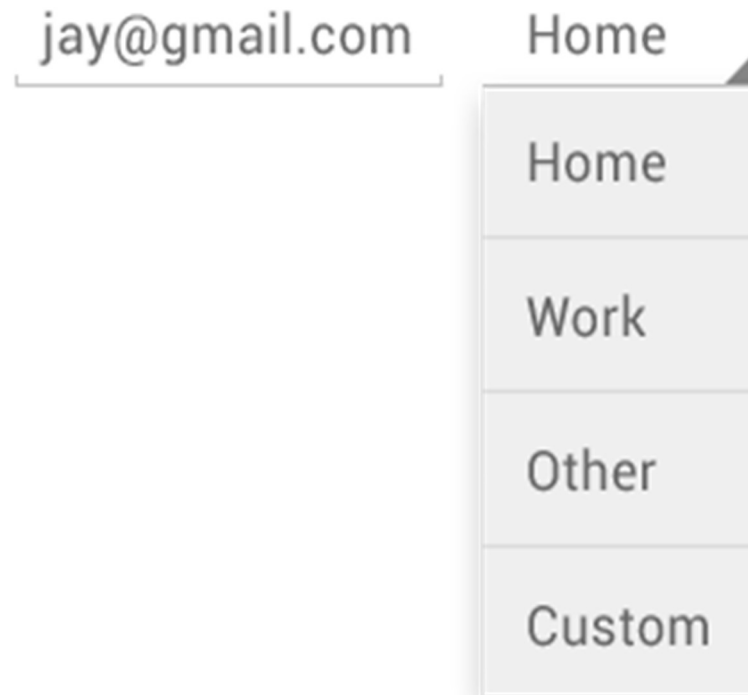


- Spinner
- ListView
- GridView

# Spinner



- Cho phép người dùng chọn một giá trị từ một tập các tùy chọn
- Spinner hiển thị các tùy chọn ở dạng một danh sách thả dọc, người dùng phải nhấn vào để thấy tất cả các giá trị



# Spinner



- Tạo Spinner
  - Tạo phần tử <Spinner> trong file XML layout

```
<Spinner  
    android:id="@+id/planets_spinner"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content" />
```

# Spinner

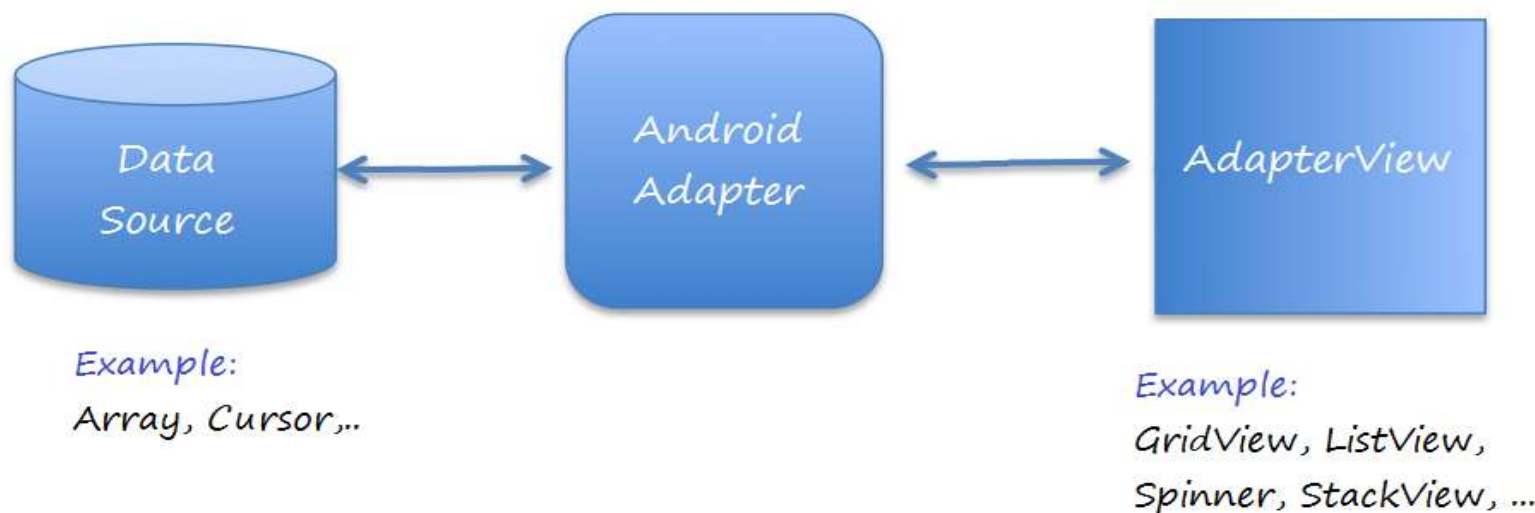


- **SpinnerAdapter**: đưa dữ liệu vào Spinner
- Các giá trị tùy chọn cho Spinner có thể được lấy từ nhiều nguồn khác nhau, có thể từ một mảng, hoặc từ cơ sở dữ liệu.
- **ArrayAdapter** và **CursorAdapter** là các lớp trừu tượng kế thừa từ **SpinnerAdapter**.
  - Nếu lấy dữ liệu từ mảng: dùng ArrayAdapter
  - Nếu lấy dữ liệu từ cơ sở dữ liệu: dùng CursorAdapter

# Android Adapter



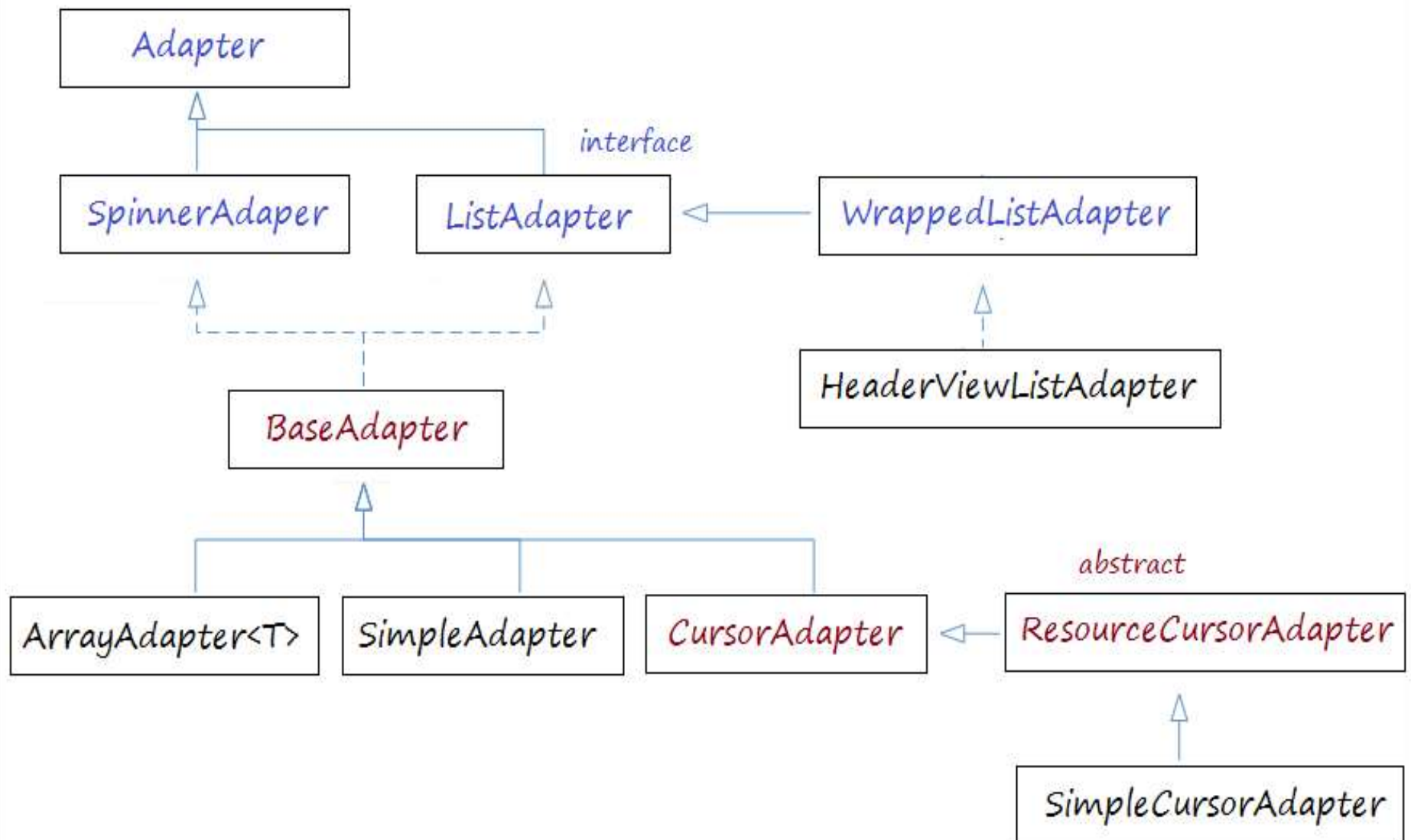
- Adapter (Bộ tiếp nối)
  - **Android Adapter** là cầu nối giữa các View (Spinner, ListView,...) và nguồn dữ liệu.
  - Adapter quản lý dữ liệu và ghép nối với các dòng riêng lẻ (Item) của view.
  - Ràng buộc Adapter với View: dùng phương thức ***setAdapter***.



# Android Adapter



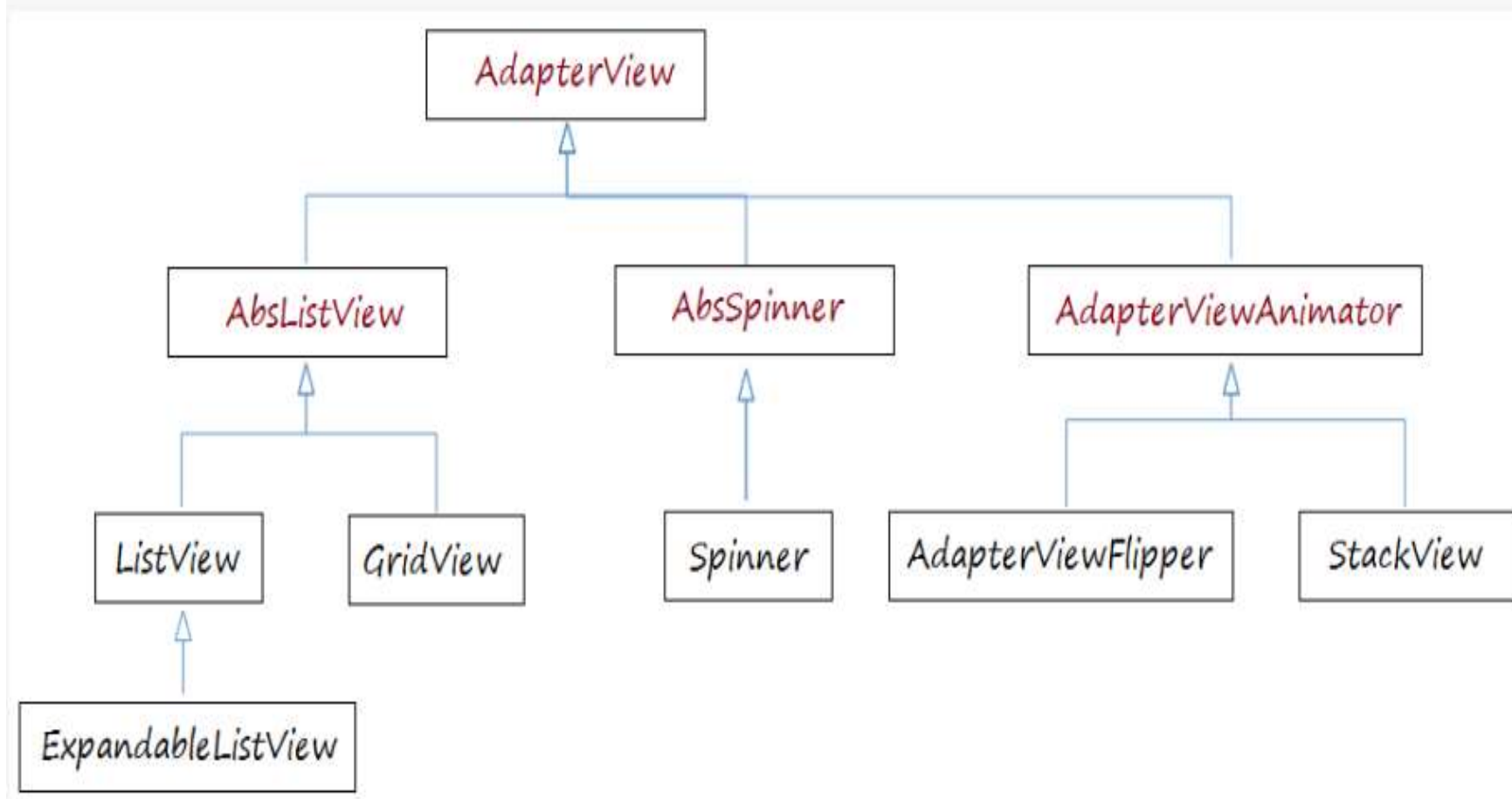
## Android Adapter



# AdapterView



- Có nhiều **View** cần tới **Android Adapter** để quản lý dữ liệu hiển thị, các View này là con của lớp **AdapterView**

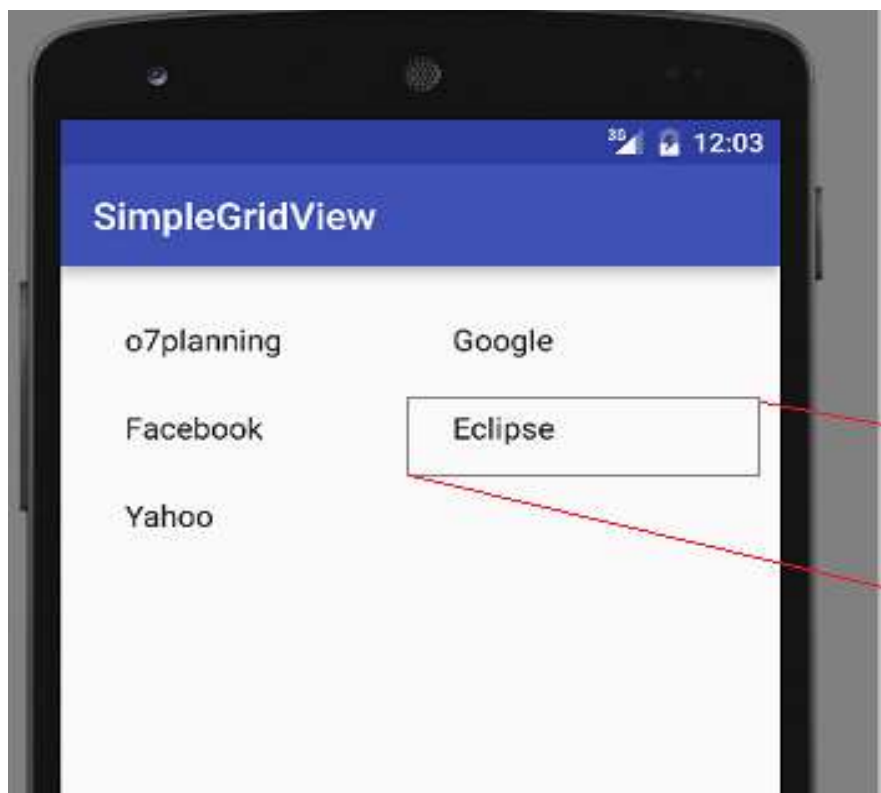




# AdapterView



- ArrayAdapter: thường được dùng để đưa dữ liệu lên các **View**, các Item trong View có thể làm từ duy nhất một TextView, CheckedTextView, EditText,...



*ArrayAdapter sử dụng để tạo các GridView, ListView,..., mà các items làm từ duy nhất một TextView*



*TextView*

# Spinner



- VD1: sử dụng mảng để lưu các giá trị và tải lên cho Spinner thông qua ArrayAdapter:

- Khai báo một mảng như sau

```
String numbers[] = { "ONE", "TWO",  
                      "THREE", "FOUR", "FIVE" };
```

- Đưa dữ liệu vào Spinner

```
sp = (Spinner) findViewById(R.id.spin);  
ArrayAdapter<String> adapter1;  
adapter1 = new ArrayAdapter<String>(  
    this,  
    android.R.layout.simple_spinner_item,  
    numbers);  
sp.setAdapter(adapter1);
```

# Spinner



- VD2: thêm vào file strings.xml trong thư mục res/values đoạn code sau và được tải lên cho Spinner thông qua ArrayAdapter:

```
<string-array name="planets_array">
    <item>Mercury</item>
    <item>Venus</item>
    <item>Earth</item>
    <item>Mars</item>
    <item>Jupiter</item>
    <item>Saturn</item>
    <item>Uranus</item>
    <item>Neptune</item>
</string-array>
```

# Spinner



- Trong Activity, sử dụng đoạn mã sau để tải các tùy chọn cho Spinner:

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);  
// Create an ArrayAdapter using the string array and a default spinner layout  
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,  
    R.array.planets_array, android.R.layout.simple_spinner_item);  
// Apply the adapter to the spinner  
spinner.setAdapter(adapter);
```

# Spinner



- Phương thức **createFromResource**: tạo một ArrayAdapter từ một mảng các giá trị được tải lên từ file resource
- Tham số thứ 3 trong phương thức **createFromResource**: cho biết layout nào được sử dụng để hiển thị nội dung của Spinner. Mặc định là *android.R.layout.simple\_spinner\_item*.

# Spinner



## ■ Xử lý sự kiện

- Khi người dùng chọn một mục từ danh sách sổ xuống, Spinner sẽ nhận sự kiện **on-item-selected**.
- Để định nghĩa phương thức xử lý sự kiện chọn trên Spinner thực hiện như sau:
  - Tạo một đối tượng **adapterView.OnItemSelectedListener**
  - Gán cho Spinner thông qua phương thức **setOnItemSelectedListener**.

# Spinner



## ■ Xử lý sự kiện

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);  
spinner.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int  
position, long id) {  
        String itemText = (String) parent.getItemAtPosition(position);  
        Toast.makeText(MainActivity.this, itemText, Toast.LENGTH_SHORT).show();  
    }  
    @Override  
    public void onNothingSelected(AdapterView<?> parent) {}  
});
```

# ListView



- **ListView** là một view group, hiển thị các thành phần (elements) theo một danh sách, có thể cuộn được theo chiều thẳng đứng.
- Ví dụ sổ liên lạc, có thể sử dụng **ListView** để hiển thị thông tin của một liên lạc gồm tên, số điện thoại, địa chỉ...

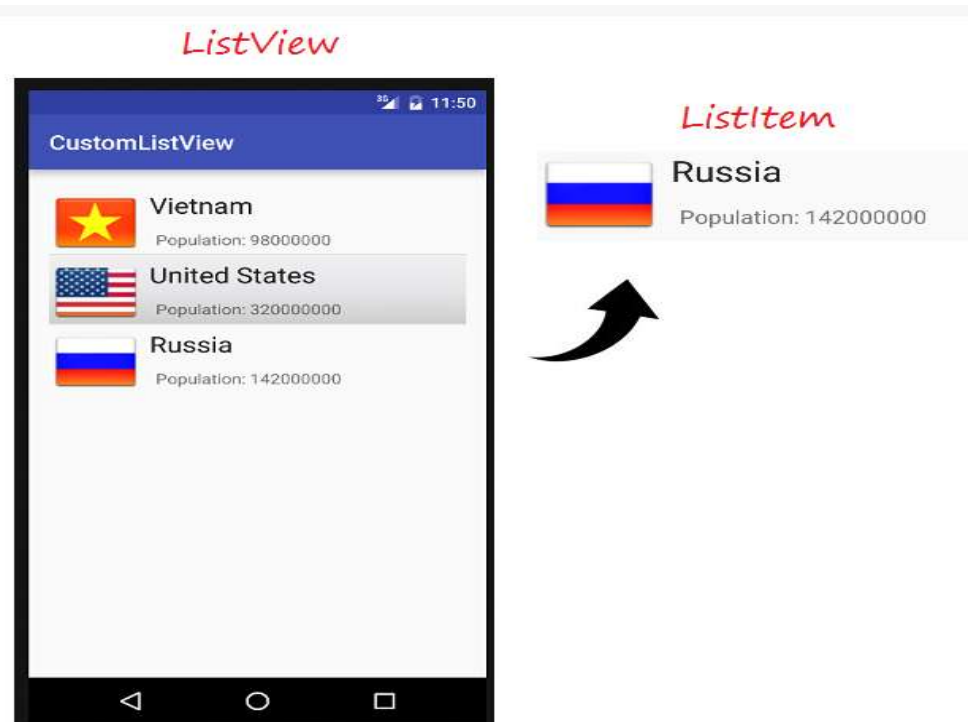


# ListView



## ■ ListItem

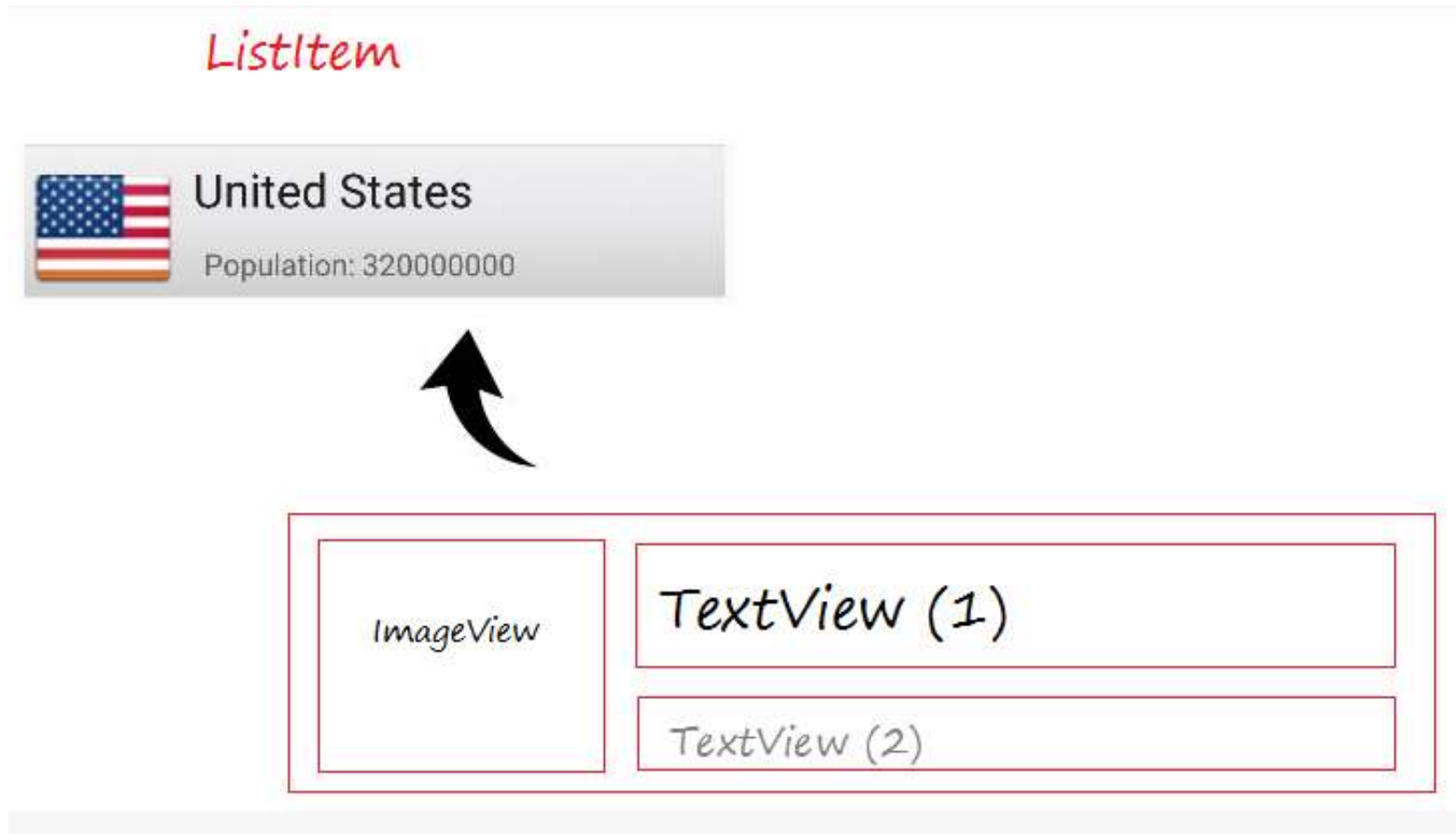
- Một ListView được tạo từ một danh sách các ListItem.
- ListItem là một dòng (row) riêng lẻ trong Listview nơi mà dữ liệu sẽ được hiển thị.



# ListView



- ListItem: là một mảnh giao diện, nó có thể được làm bởi một số View.



# ListView



## ■ Các thuộc tính trình bày dữ liệu trên ListView

Thuộc tính	Mô tả
<b>android:id</b>	ID định danh cho ListView
<b>android:divider</b>	Nhận giá trị là một drawable hoặc một màu, dùng để vẽ đường ngăn giữa các mục.
<b>android:dividerHeight</b>	Chỉ định chiều cao của đường ngăn cách.
<b>android:entries</b>	Chỉ định tham chiếu đến mảng dữ liệu cho ListView.
<b>android:footerDividersEnabled</b>	Nếu giá trị là “false”, sẽ không vẽ đường ngăn trước footer view. Giá trị mặc định là “true”.
<b>android:headerDividersEnabled</b>	Nếu giá trị là “false”, sẽ không vẽ đường ngăn sau header view. Giá trị mặc định là “true”.

# ListView



- Trình bày dữ liệu trên ListView: Sử dụng thuộc tính **android:entries**
- Ví dụ: Tạo mảng các quốc gia và hiển thị mảng này lên ListView.
  - Định nghĩa mảng các quốc gia trong resource (res/values/strings.xml)
  - Đưa dữ liệu này lên ListView thông qua thuộc tính **android:entries**

# ListView



## ■ Ví dụ...

```
<resources>
<string name="app_name">DemoListView</string>
<string-array name="countries">
    <item>VietNam</item>
    <item>Laos</item>
    <item>Cambodia</item>
    <item>Indonesia</item>
    <item>Thailand</item>
    <item>Malaysia</item>
    <item>Philippines</item>
    <item>Singapore</item>
    <item>Myanmar</item>
    <item>Brunei</item>
    <item>Timor-Leste</item>
</string-array>
</resources>
```

# ListView



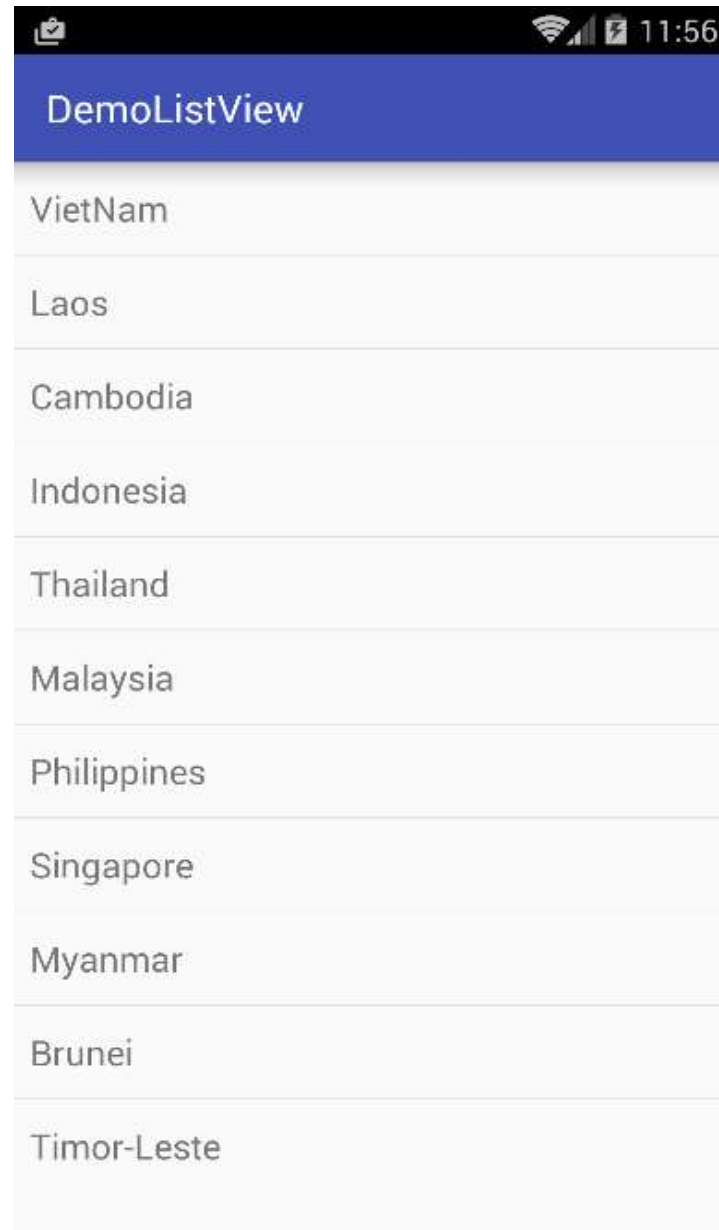
- Ví dụ...
  - Trong file XML layout, tạo một ListView, gán cho thuộc tính entries mảng countries đã tạo ở trên.

```
<?xml version="1.0" encoding="utf-8"?>
<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" tools:context=".MainActivity">
    <ListView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/listView"
        android:entries="@array/countries" />
</LinearLayout>
```

# ListView



- Kết quả



# ListView



- VD: sử dụng mảng để lưu các giá trị và tải lên cho ListView thông qua ArrayAdapter:

- Khai báo một mảng như sau

```
String numbers[] = { "ONE", "TWO",  
                      "THREE", "FOUR", "FIVE" };
```

- Đưa dữ liệu vào ListView

```
lv=(ListView) findViewById(R.id.lvNumber)  
ArrayAdapter<String> adapter=new  
    ArrayAdapter<String>(this,  
        android.R.layout.simple_list_item_1,  
        numbers);  
lv.setAdapter(adapter);
```



# ListView



- VD: sử dụng kiểu tập hợp để lưu các giá trị
  - Khai báo một tập hợp như sau
- Đưa dữ liệu lên ListView

```
listView = (ListView) findViewById(R.id.listview);  
list = new ArrayList<>();  
list.add("Lập Trình Android");  
list.add("Lập Trình Java");  
list.add("Lập Trình Web");  
ArrayAdapter adapter = new ArrayAdapter<>(this,  
    android.R.layout.simple_list_item_1,  
    list);  
listView.setAdapter(adapter);
```

# ListView

---

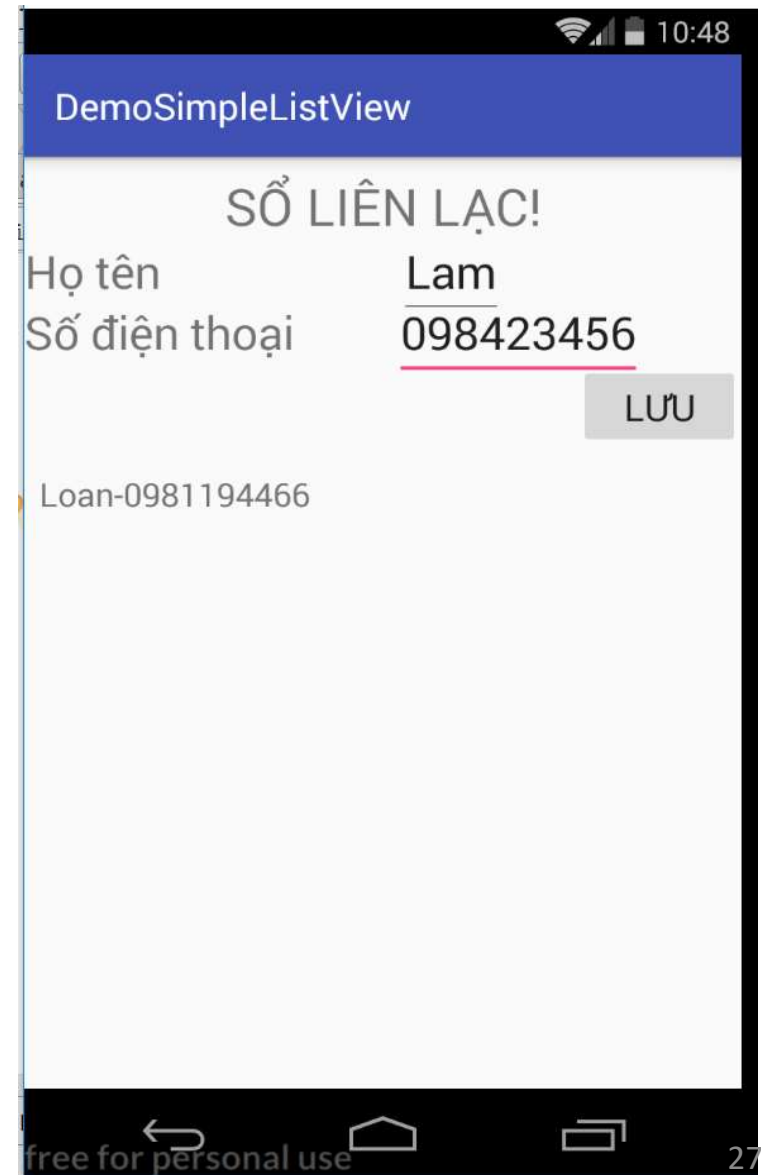


- Xử lý sự kiện chọn một item trong Listview ta dùng các phương thức sau:
  - `setOnItemClickListener`
  - `setOnLongClickListener`

# ListView



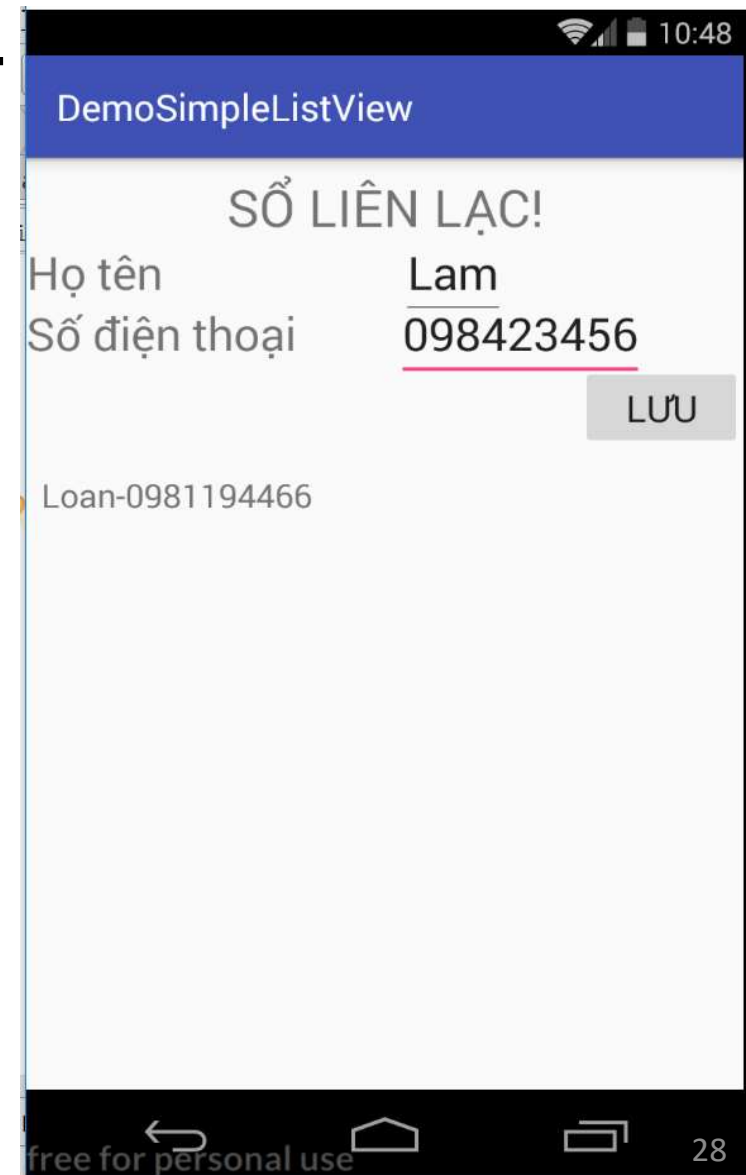
- Tạo ứng dụng đơn giản sau:
  - Yêu cầu:
    - Nhập tên, số điện thoại
    - Chọn nút lưu để hiển thị dữ liệu lên listview
    - Trước khi thêm vào danh sách liên lạc có kiểm tra trùng lặp



# ListView



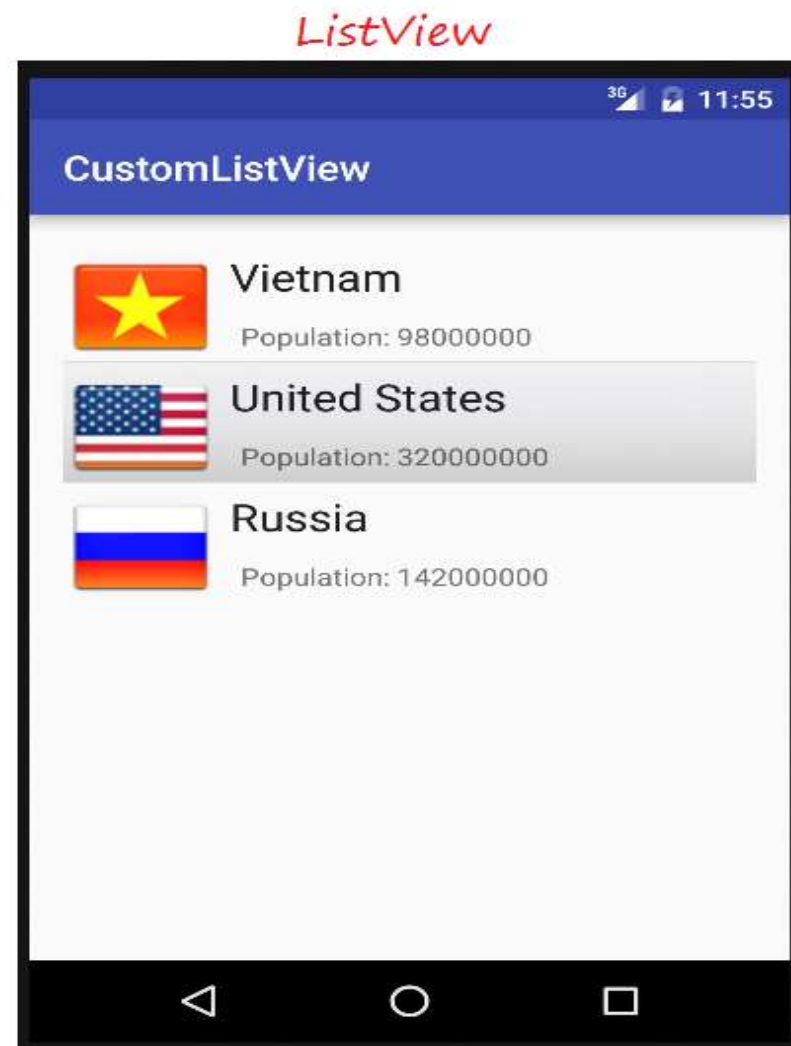
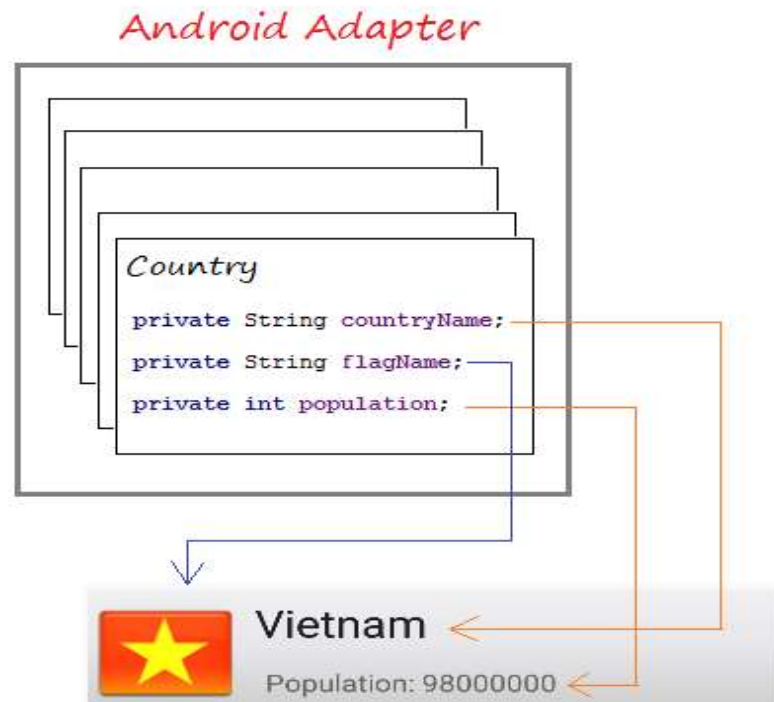
- Tạo ứng dụng đơn giản sau...
  - Thực hiện
    - Tạo layout như hình
    - Tạo một mảng lưu các contact
    - Tạo một adapter để chuyển tiếp dữ liệu
    - Đưa dữ liệu lên listview thông qua phương thức setAdapter



# ListView



- Custom Listview



# ListView



## ■ ListView Selector

- Để ListView hiển thị đẹp hơn, ta tùy biến các hiệu ứng, như đổi màu nền của ListItem khi con trỏ di chuyển trên nó hoặc đổi màu nền khi ListItem được chọn.



Vietnam

Population: 98000000

*Khi ListItem ở trạng thái thông thường*



United States

Population: 320000000

*Khi ListItem ở trạng thái được lựa chọn*



Russia

Population: 142000000

*Khi ListItem bị nhấn*

# Custom ListView

---



- Ví dụ: Hiển thị danh sách các quốc gia Đông Nam Á kèm theo cờ bên trái lên ListView.
- Các bước thực hiện:
  - Bước 1: Tạo nguồn dữ liệu (2 mảng: mảng tên quốc gia và mảng các hình lá cờ)
  - Bước 2: Tạo layout tương ứng với mỗi dòng trong ListView
  - Bước 3: Tạo Adapter mới, nhận vào là 2 mảng và layout, để đổ dữ liệu từ mảng vào ListView.
  - Bước 4: Trong Activity, tạo ListView và gán Adapter này cho ListView.

# Custom ListView



- **Bước 1:** tạo mảng tên các quốc gia và mảng lá cờ trong file countries.xml (ở thư mục: res/values/countries.xml) với nội dung sau:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="countryNames">
        <item>VietNam</item>
        <item>Laos</item>
        <item>Cambodia</item>
        <item>Indonesia</item>
        <item>Thailand</item>
        <item>Malaysia</item>
        <item>Philippines</item>
        <item>Singapore</item>
        <item>Myanmar</item>
        <item>Brunei</item>
        <item>Timor-Leste</item>
    </string-array>
```



# Custom ListView



- Lưu ý: Các hình ảnh lá cờ được lưu vào thư mục: res/drawable. Tên của các file ảnh chỉ cho phép các ký tự 0-9, a-z và dấu gạch dưới (không có ký tự in hoa).

```
<array name="countryIcons">
    <item>@drawable/vietnam</item>|
    <item>@drawable/laos</item>
    <item>@drawable/cambodia</item>
    <item>@drawable/indonesia</item>
    <item>@drawable/thailand</item>
    <item>@drawable/malaysia</item>
    <item>@drawable/philippines</item>
    <item>@drawable/singapore</item>
    <item>@drawable/myanmar</item>
    <item>@drawable/brunei</item>
    <item>@drawable/timor-leste</item>
</array>
</resources>
```

# Custom ListView



- **Bước 2:** Tạo một layout `list_item_country.xml` để hiển thị ListItem. Layout này sử dụng `ImageView` để hiển thị lá cờ (canh trái) và `TextView` để hiển thị tên quốc gia (canh phải)

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:layout_width="wrap_content"|
        android:layout_height="wrap_content"
        android:id="@+id/imgFlag"
        android:layout_alignParentLeft="true"
        android:layout_centerVertical="true"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Large Text"
        android:id="@+id/txtCountryName"
        android:layout_alignParentRight="true" />
</RelativeLayout>
```

# Custom ListView



- **Bước 3:** Tạo một Adapter, dựa trên ArrayAdapter để nhận vào hai mảng. Adapter mới này cũng có nhiệm vụ trình bày hai nội dung lên mỗi dòng của ListView nhờ vào hàm getView.

```
public class CountryAdapter extends ArrayAdapter<String> {  
    private int mLAYOUTId;  
    private TypedArray mImages;  
    private String[] mTexts;  
    public CountryAdapter(Context context, int resource, String[] texts,  
                           TypedArray images) {  
        super(context, resource, texts);  
        this.mLAYOUTId = resource;  
        this.mTexts = texts;  
        this.mImages = images;  
    }  
}
```



# Custom ListView



- Bước 3...

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    // Reference ArrayAdapter class for coding.
    View view;
    if(convertView == null)
    {
        LayoutInflater inflater = LayoutInflater.from(getContext());
        view = inflater.inflate(mLayoutId, null);
    }
    else {
        view = convertView;
    }
    ImageView imageView = (ImageView) view.findViewById(R.id.imgFlag);
    TextView textView = (TextView)
view.findViewById(R.id.txtCountryName);
    imageView.setImageDrawable(mImages.getDrawable(position));
    textView.setText(mTexts[position]);
    return view;
}
}
```

# Custom ListView



- **Bước 4:** trong Activity XML layout, tạo ListView

```
<?xml version="1.0" encoding="utf-8"?>
<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"
    tools:context=".MainActivity">
    <ListView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/listView" />
</LinearLayout>
```

# Custom ListView



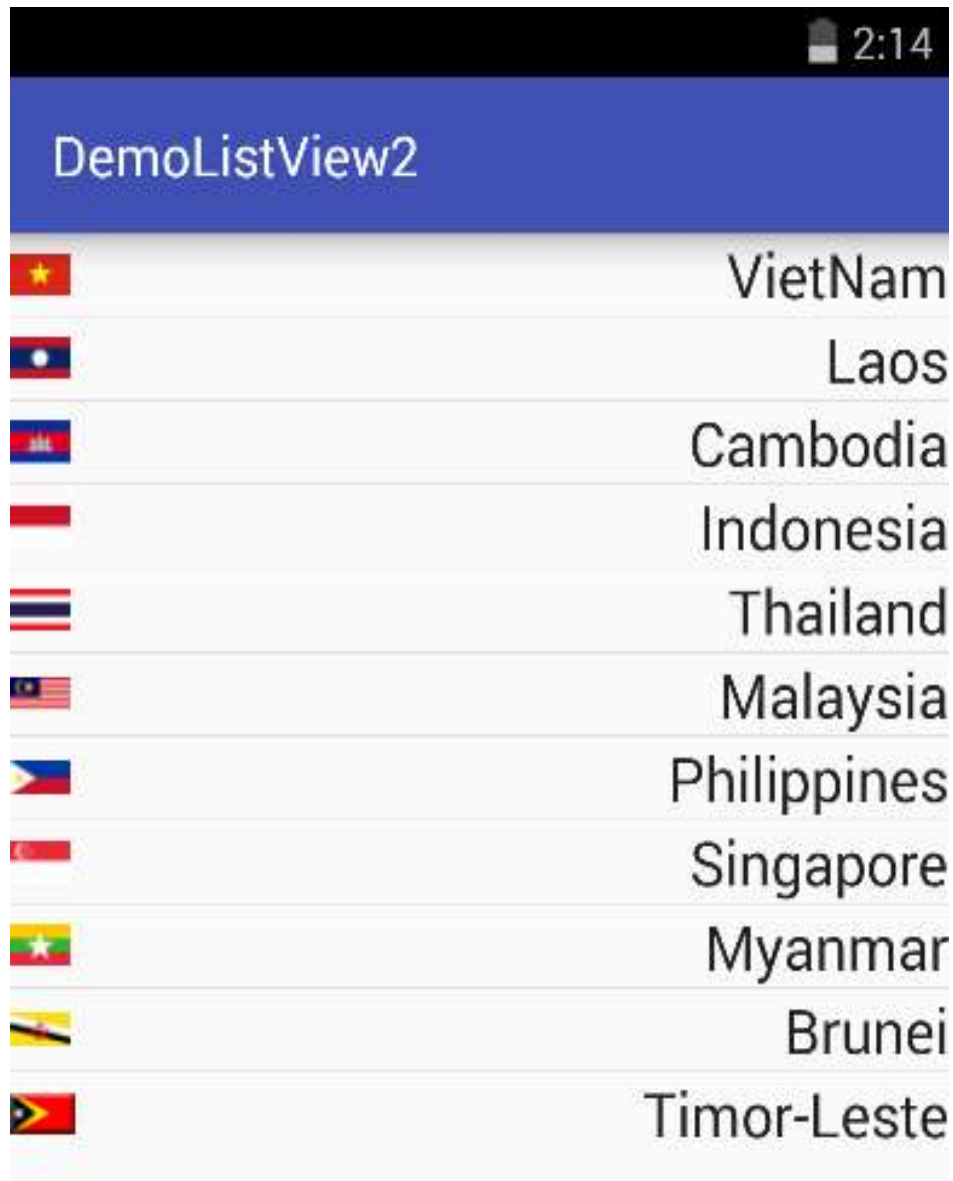
- Bước 4: Và trong mã java, khởi tạo CountryAdapter, truyền vào các tham số: ngữ cảnh hiện tại, layout id sử dụng để trình bày nội dung cho mỗi mục, mảng các quốc gia, mảng các lá cờ.

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        ListView listView = (ListView) findViewById(R.id.listView);  
        CountryAdapter adapter = new CountryAdapter(this,  
            R.layout.list_item_country,  
            getResources().getStringArray(R.array.countryNames),  
            getResources().obtainTypedArray(countryIcons));  
        listView.setAdapter(adapter);  
    }  
}
```

# Custom ListView



- Kết quả





# Ví dụ



- Một dạng view khác





# RecyclerView

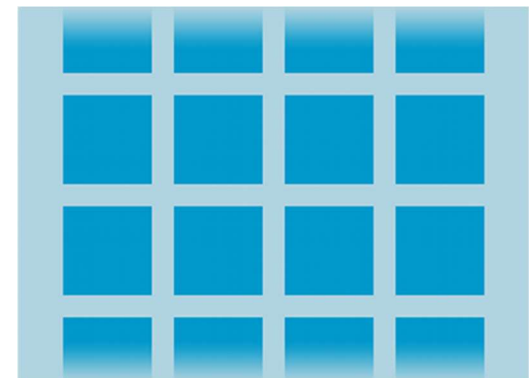


- RecyclerView là một ViewGroup được tích hợp trong Android Lollipop (Android API 21).
  - Những điểm mạnh vượt trội của RecyclerView
    - RecyclerView cung cấp **RecyclerView.LayoutManager** cho phép các item trong listView theo các kiểu khác nhau (ngang, dọc, dạng grid, dạng staggered grid).
    - Sử dụng **ItemAnimator** để add hay remove một item ra khỏi Recycler.
    - RecyclerView chỉ cung cấp một phương thức **onItemTouchListener** cho sự kiện chọn một item
- => Tìm hiểu thêm

# GridView



- Cho phép hiển thị các mục theo hai chiều (theo cột và dòng).
- Dữ liệu được đổ vào GridView thông qua **ListAdapter**. Có thể sử dụng ArrayAdapter hoặc tạo một lớp Adapter mới có cài đặt giao diện ListAdapter.
- Cho phép chọn từng mục trong GridView. Mặc định GridView hỗ trợ thanh trượt.
- GridView thường được sử dụng trong ứng dụng bộ sưu tập ảnh (Gallery).



# GridView



## ■ Các thuộc tính thường dùng

Thuộc tính	Mô tả
<b>android:id</b>	ID định danh cho GridView
<b>android:columnWidth</b>	Chỉ định chiều rộng cố định cho mỗi cột. Các đơn vị có thể là: px, dp, sp, in, mm
<b>android:gravity</b>	Chỉ định giá trị gravity bên trong mỗi ô. Các giá trị có thể là: top, bottom, left, right, center, center_vertical, center_horizontal ...

# GridView



## ■ Các thuộc tính thường dùng

Thuộc tính	Mô tả
<b>android: horizontalSpacing</b>	Chỉ định khoảng cách mặc định theo chiều ngang giữa các cột. Các đơn vị có thể là: px, dp, sp, in, or mm
<b>android: verticalSpacing</b>	Chỉ định khoảng cách mặc định theo chiều ngang giữa các dòng. Các đơn vị có thể là: px, dp, sp, in, or mm
<b>android: numColumns</b>	Định nghĩa số cột của gridview. Nó có thể là một số nguyên hoặc “auto_fit” (với auto_fit, nó sẽ hiển thị nhiều cột nhất có thể để tô đầy khoảng trống trên màn hình)

# GridView



## ■ Các thuộc tính thường dùng

Thuộc tính	Mô tả
<b>android: stretchMode</b>	<p>Định nghĩa cách tô đầy khoảng trống của gridview, có ba giá trị sau:</p> <p>None: Không giãn.</p> <p>columnWidth: Mỗi cột sẽ được giãn đều nhau (chiều rộng cột có thể sẽ thay đổi để tô đầy khoảng trống)</p> <p>spacingWidth: Các khoảng trống giữa các cột sẽ được giãn ra để tô đầy khoảng trống trên màn hình.</p> <p>spacingWidthUniform: Các khoảng trống giữa các cột sẽ được giãn ra một cách thống nhất để tô đầy khoảng trống trên màn hình.</p>

# GridView



## ■ Các thuộc tính thường dùng

Thuộc tính	Mô tả
<b>android: listSelector</b>	Chỉ định một Drawable được dùng khi một mục được chọn (Thuộc tính này kế thừa từ lớp trừu tượng <code>AbsListView</code> )
<b>android: drawSelectorOnTop</b>	Giá trị có thể là “true” hoặc “false”. Nếu là “true”, selector sẽ được vẽ lên trên mục được chọn, ngược lại, selector sẽ được vẽ phía dưới mục được chọn (Thuộc tính này kế thừa từ lớp trừu tượng <code>AbsListView</code> )

# GridView



- Ví dụ: Tạo một GridView trình bày các hình ảnh, mỗi ô có kích thước 90dp x 90dp.



# GridView



- Bước 1: Tạo đối tượng GridView.

```
<?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">
    <GridView
        android:id="@+id/gridview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/cyan"
        android:columnWidth="90dp"
        android:horizontalSpacing="4dp"
        android:verticalSpacing="4dp"
        android:numColumns="auto_fit"
        android:stretchMode="spacingWidthUniform"
        android:listSelector="@color/yellow"
        android:drawSelectorOnTop="true" />
</LinearLayout>
```



# GridView



- Bước 2: Tạo một layout để định nghĩa mỗi ô trong GridView. Tạo layout mới `grid_item.xml` với `ImageView` để hiển thị hình ảnh

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="@color/magenta">
  <ImageView
    android:layout_width="90dp"
    android:layout_height="90dp"
    android:id="@+id/imageView"/>
</LinearLayout>
```

# GridView



- Bước 3: Tạo mảng các hình ảnh để hiển thị. Trong Activity sử dụng layout, khởi tạo mảng các Drawable. Ở ví dụ này, các hình ảnh được download từ website [https://developer.android.com/shareables/sample\\_images.zip](https://developer.android.com/shareables/sample_images.zip)

```
private Integer[] mThumbIds = {  
    R.drawable.sample_2, R.drawable.sample_3,  
    R.drawable.sample_4, R.drawable.sample_5,  
    R.drawable.sample_6, R.drawable.sample_7,  
    R.drawable.sample_0, R.drawable.sample_1,  
    R.drawable.sample_2, R.drawable.sample_3,  
    R.drawable.sample_4, R.drawable.sample_5,  
    R.drawable.sample_6, R.drawable.sample_7,  
    R.drawable.sample_0, R.drawable.sample_1,  
    R.drawable.sample_2, R.drawable.sample_3,  
    R.drawable.sample_4, R.drawable.sample_5,  
    R.drawable.sample_6, R.drawable.sample_7  
};
```

# GridView



- Bước 4: Tạo Adapter để đổ dữ liệu vào GridView. Tạo lớp Java: GridViewAdapter.java, kế thừa ArrayAdapter với đoạn mã sau:

```
public class GridViewAdapter extends ArrayAdapter<Integer> {
    private int mResourceId;
    public GridViewAdapter(Context context, int resource, Integer[] objects)
    {
        super(context, resource, objects);
        mResourceId = resource;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        View view;
        if (convertView == null) {
            LayoutInflater inflater = LayoutInflater.from(getContext());
            view = inflater.inflate(mResourceId, null);
        } else { view = convertView; }
        ImageView imageView = (ImageView) view.findViewById(R.id.imageView);
        imageView.setImageResource(getItem(position));
        return view;
    }
}
```

# GridView



- Bước 5: Khởi tạo Adapter và gán nó cho GridView.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    GridView gridView = (GridView) findViewById(R.id.gridview);
    GridViewAdapter adapter = new GridViewAdapter(this, R.layout.grid_item,
mThumbIds);
    gridView.setAdapter(adapter);
    gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
            Toast.makeText(MainActivity.this,
                String.valueOf(position) + " : " +
                    String.valueOf(view.getWidth()) + " ; " +
                    String.valueOf(view.getHeight()),
                Toast.LENGTH_SHORT).show();
        }
    });
}
```