

The background features a dark gray upper half and a white lower half, separated by a horizontal orange line. Concentric circles in shades of gray are centered on the left side, spanning both the dark and light areas.

USER INTERFACE- VIEW

LAYOUT

- Các loại layout cơ bản và các thuộc tính
- Làm quen với Linear Layout – Relative Layout
- Sự khác biệt giữa hai loại layout.
- Các thông số trong linear layout
- Các thông số trong relative layout
- List View (Mặc định và cách customize, một số điểm chú ý)

USER INTERFACE

- Trong các ứng dụng android, giao diện người dùng (UI) được xây dựng từ các View và ViewGroups
- View là lớp cơ bản nhất về giao diện, còn được gọi là “widget”. Các View cơ bản như TextView, Button, CheckBox ...
- ViewGroup là một tập hợp các View như các layout (trong đó các LinearLayout và RelativeLayout là hai layout thường được dùng)
- Các thuộc tính bắt buộc phải khai báo trong View hoặc ViewGroup

android:id="@+id/androidId"

android:layout_width="fill_parent / wrap_content / match_parent(>=2.2)"

android:layout_height="fill_parent / wrap_content / match_parent(>=2.2)"

LINEARLAYOUT

- LinearLayout là một ViewGroup hiển thị các View liên tục theo một hướng xác định trước (ngang hoặc dọc)

<LinearLayout

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

android:orientation="vertical"

android:layout_width="fill_parent"

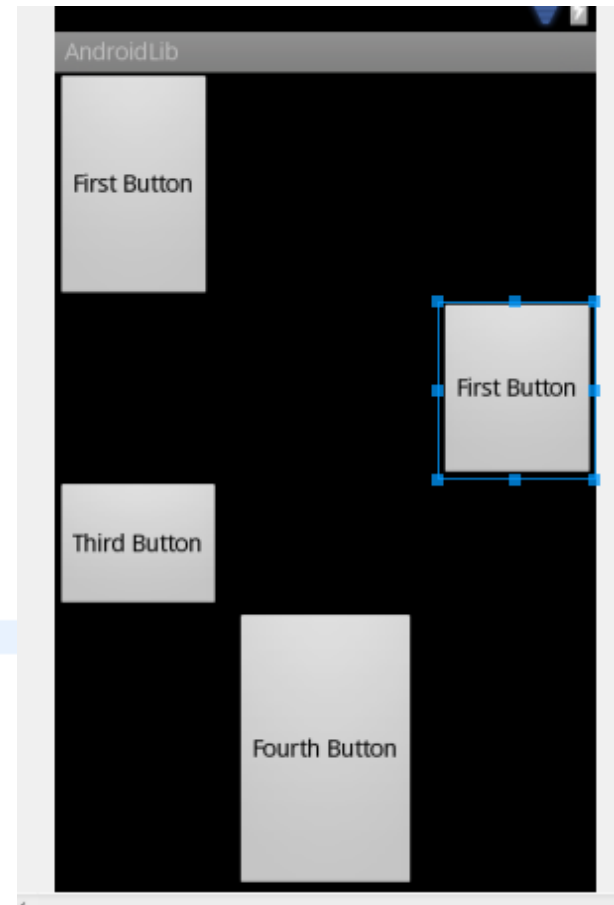
android:layout_height="fill_parent">

- Một LinearLayout nếu không xét hướng hiển thị, các View sẽ được mặc định hiển thị theo chiều ngang
- Trong LinearLayout ta sử dụng `layout_weight`, `layout_gravity` cho các View để xác định vị trí hoặc phân bố kích thước cho View

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:id="@+id/linearLayout1"
    android:orientation="vertical"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <Button
        android:id="@+id/btnFirst"
        android:layout_weight="3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="First Button"/>
    <Button
        android:id="@+id/btnSecond"
        android:layout_weight="2"
        android:layout_gravity="right"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="First Button"/>
    <Button
        android:id="@+id/btnThird"
        android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Third Button"/>
    <Button
        android:id="@+id/btnFourth"
        android:layout_weight="4"
        android:layout_gravity="center"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fourth Button"/>
</LinearLayout>

```



RELATIVELAYOUT

- RelativeLayout hiển thị các View một cách tương đối
- Các View trong RelativeLayout sử dụng `layout_alignParentLeft`, `layout_alignParentRight`, `layout_below`, `layout_abow` ... để xác định vị trí hiển thị

<TextView

```
android:id="@+id/label"  
android:layout_width="fill_parent"  
android:layout_height="wrap_content"  
android:text="Text"/>
```

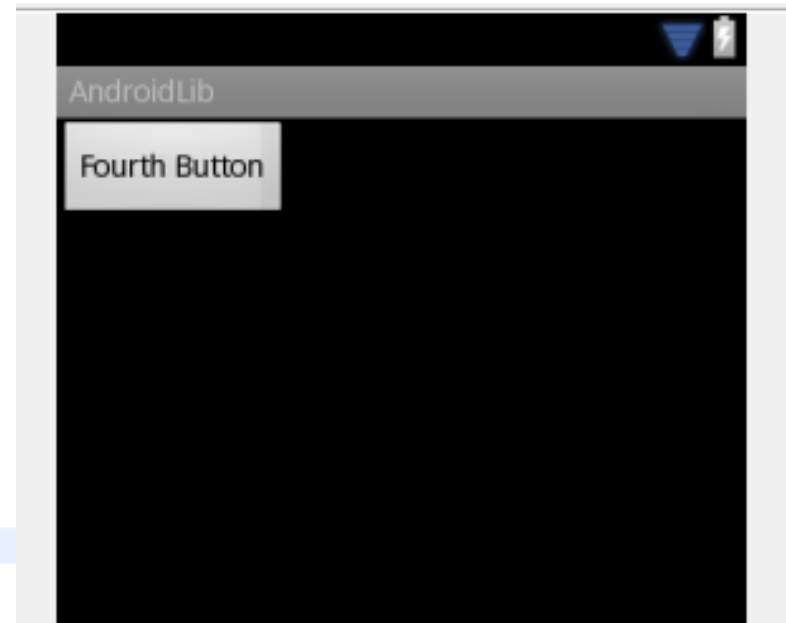
<EditText

```
android:id="@+id/entry"  
android:layout_width="fill_parent"  
android:layout_height="wrap_content"  
android:layout_below="@id/label"/>
```

RELATIVELAYOUT

- Nếu các View không được xác định vị trí hiển thị, chúng sẽ được sắp xếp ở góc trên bên trái và xếp chồng lên nhau

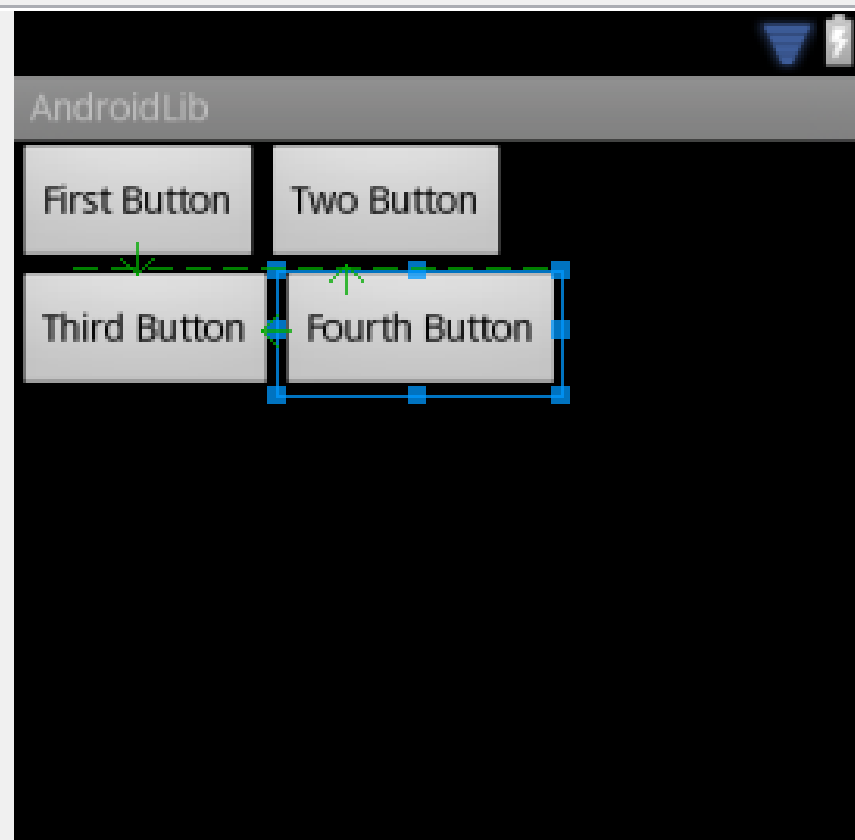
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout android:id="@+id/relativeLayout1"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <Button
        android:id="@+id/btnFirst"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="First Button"/>
    <Button
        android:id="@+id/btnSecond"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Two Button"/>
    <Button
        android:id="@+id/btnThird"
        android:layout_height="wrap_content"
        android:text="Third Button"/>
    <Button
        android:id="@+id/btnFourth"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fourth Button"/>
</RelativeLayout>
```



```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout android:id="@+id/relativeLayout1"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <Button
        android:id="@+id/btnFirst"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="First Button"/>
    <Button
        android:id="@+id/btnSecond"
        android:layout_toRightOf="@+id/btnFirst"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Two Button"/>
    <Button
        android:id="@+id/btnThird"
        android:layout_below="@+id/btnFirst"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Third Button"/>
    <Button
        android:id="@+id/btnFourth"
        android:layout_toRightOf="@+id/btnThird"
        android:layout_below="@+id/btnFirst"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fourth Button"/>
</RelativeLayout>

```



LISTVIEW

- Là một ViewGroup chứa danh sách nhiều item. Giao diện và dữ liệu của mỗi item được thiết lập thông qua ListAdapter
- Có thể tạo một ListView trong ListActivity (chỉ có thể chứa ListView) hoặc Activity (ngoài ListView có thể thêm các View hoặc ViewGroup khác)
- Android cung cấp một số simple style cho các item của ListView

android.R.layout.simple_list_item_1

android.R.layout.simple_list_item_2

android.R.layout.simple_list_item_checked

android.R.layout.simple_list_item_multiple_choice

android.R.layout.simple_list_item_single_choice

- Để hiển thị dữ liệu lên ListView ta cần gán Adapter cho ListView
- Adapter quyết định giao diện cũng như dữ liệu sẽ được hiển thị trên ListView
- Ví dụ :

```
ArrayList<String> testData = new ArrayList<String>();
```

```
list.setAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, testData));
```

- Các Adapter cơ bản của Android bao gồm : ArrayAdapter<T>, BaseAdapter, CursorAdapter, HeaderViewListAdapter, ResourceCursorAdapter, SimpleCursorAdapter, WrapperListAdapter

- Để có thể có các giao diện mới ngoài các mẫu được cung cấp sẵn thì cần phải xây dựng một adapter mới và kế thừa từ lớp BaseAdapter

```
public class MyAdapter extends BaseAdapter{
```

```
...
```

```
}
```

- Các hàm cơ bản thực thi trong class này bao gồm :

getCount() trả về số lượng item sẽ được hiển thị

getItem(int position) trả về đối tượng nằm ở vị trí xác định trong ListView

getItemId(int position) trả về Id của item

getView(int position, View convertView, ViewGroup parent) hiển thị giao diện và dữ liệu của item

GETVIEW – FAST WAY TO BUILD!

- Khai báo một static class chứa các đối tượng giao diện sẽ sử dụng để hiển thị

```
static class ViewHolder {  
    TextView text;  
    ImageView icon;  
}
```

GETVIEW – FAST WAY TO BUILD!

```
public View getView(int position, View convertView, ViewGroup parent) {  
    ViewHolder holder;  
    /* Load giao diện từ file list_item_icon_text.xml và khởi tạo các đối tượng */  
    if (convertView == null) {  
        convertView = inflater.inflate(R.layout.list_item_icon_text, parent, false);  
        holder = new ViewHolder();  
        holder.text = (TextView) convertView.findViewById(R.id.text);  
        holder.icon = (ImageView) convertView.findViewById(R.id.icon);  
        convertView.setTag(holder);  
    } else {  
        holder = (ViewHolder) convertView.getTag();  
    }  
}
```

GETVIEW – FAST WAY TO BUILD!

```
/* Hiển thị dữ liệu */
```

```
holder.text.setText(DATA[position]);
```

```
holder.icon.setImageBitmap((position & 1) == 1 ? mlcon1 : mlcon2);
```

```
return convertView;
```

```
}
```

LISTVIEW WARNING!

- Không xác định chiều cao của ListView là wrap_content
- Không để ListView trong ScrollView
- Không sử dụng ListView với các dữ liệu vô hạn (unbounded data)
- Không sử dụng ListView trong các trường hợp có thể sử dụng LinearLayout kết hợp ScrollLayout

LISTVIEW TIP AND TRICK

- Thêm header và footer (nếu cần thiết) trước khi setAdapter();
- Trường hợp bị nền đen khi scroll ListView. Thêm thuộc tính vào file xml `android:cacheColorHint="#0000"`
- Khi có thay đổi dữ liệu và cần update lại ListView, thực thi hàm `notifyDataSetChanged()` đối với Adapter và chỉ được gọi trong UI Thread
- Luôn bắt đầu ListView ở item dưới cùng. Thêm thuộc tính `android:stackFromBottom="true"`
- Khi có nhiều View có ActionListener trong một itemView, thuộc tính `focusable` của các View phải là `false`