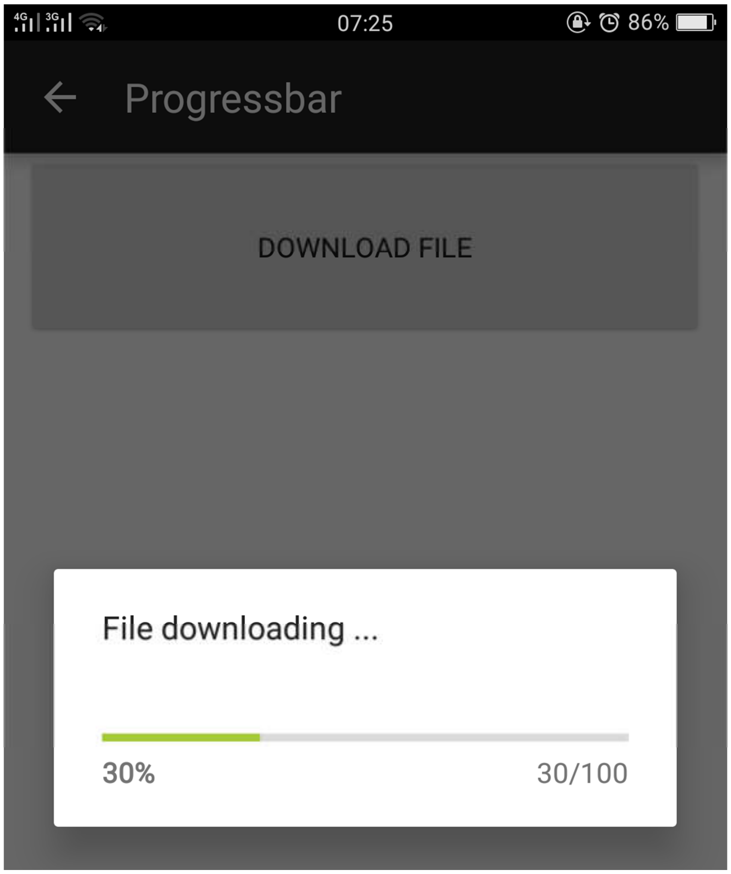
Lab 04: làm việc View (tiếp theo)

1. Progress bar



File downloading ...

30%

30/100

activity_main
LOCATION: res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http:// schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical" >

<Button
android:id="@+id/btnStartProgress"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Download File" />

</LinearLayout>
```

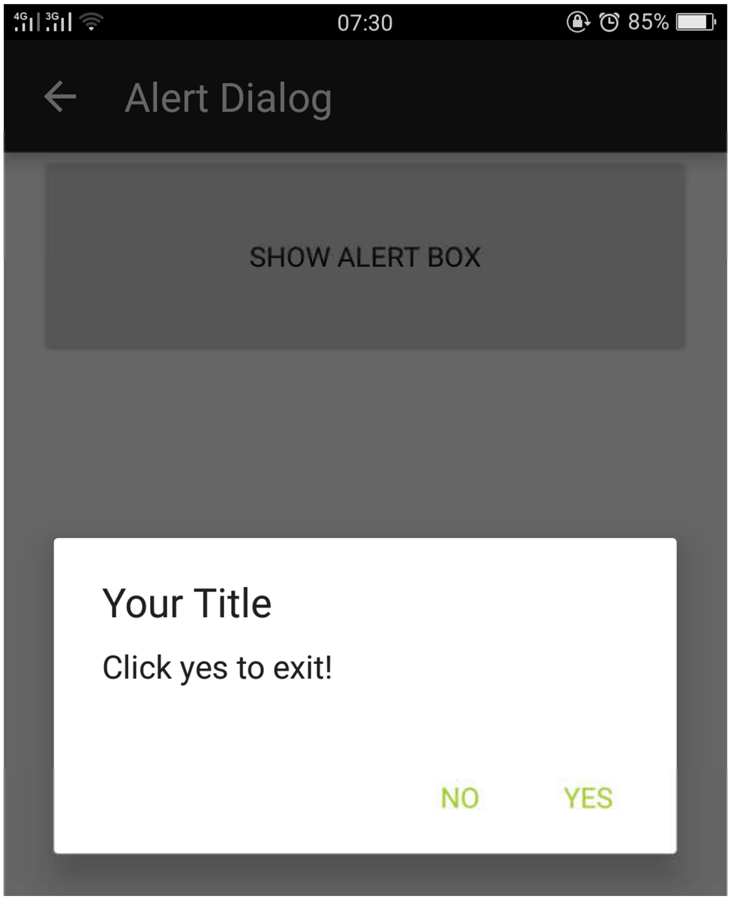
MainActivity
LOCATION: java/MainActivity.java

```
import android.app.Activity;
import android.app.ProgressDialog;
import android.os.Bundle;
import android.os.Handler;
import android.widget.Button;
import android.view.View;
import android.view.View.OnClickListener;

public class MyAndroidAppActivity extends Activity {

Button btnStartProgress;
ProgressDialog progressBar;
private int progressBarStatus = 0;
private Handler progressBarHandler = new Handler();

private long fileSize = 0;

@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);

addListenerOnButton();

}
public void addListenerOnButton() {
btnStartProgress = (Button) findViewById(R.id.btnStartProgress);
btnStartProgress.setOnClickListener(
new OnClickListener() {
@Override public void onClick(View v) {

// prepare for a progress bar dialog
progressBar = new ProgressDialog(v.getContext());
progressBar.setCancelable(true);
progressBar.setMessage("File downloading ...");
progressBar.setProgressStyle (ProgressDialog.STYLE_HORIZONTAL);
progressBar.setProgress(0);
progressBar.setMax(100);
progressBar.show();

//reset progress bar status
progressBarStatus = 0;

//reset filesize
fileSize = 0;

new Thread(new Runnable() {
```

```
public void run() {
while (progressBarStatus < 100) {

// process some tasks
progressBarStatus = doSomeTasks();

// your computer is too fast, sleep 1 second
try {
Thread.sleep(1000);
} catch (InterruptedException e) {
e.printStackTrace();
}

// Update the progress bar
progressBarHandler.post(new Runnable() {
public void run() {
progressBar.setProgress(progressBarStatus);
}
});
}

// ok, file is downloaded,
if (progressBarStatus >= 100) {

// sleep 2 seconds, so that you can see the 100%
try {
Thread.sleep(2000);
} catch (InterruptedException e) {
e.printStackTrace();
}

// close the progress bar dialog
progressBar.dismiss();
}
}
}).start();
}
});
}
// file download simulator... a really simple
public int doSomeTasks() {
while (fileSize <= 1000000) {
fileSize++;
if (fileSize == 100000) {
return 10;
} else if (fileSize == 200000) {
return 20;
} else if (fileSize == 300000) {
```

```
return 30;
}
// ...add your own
}
return 100;
}
}
```

2. Albert dialog



Your Title

Click yes to exit!

activity_main
LOCATION: res/layout/activity_main.xml

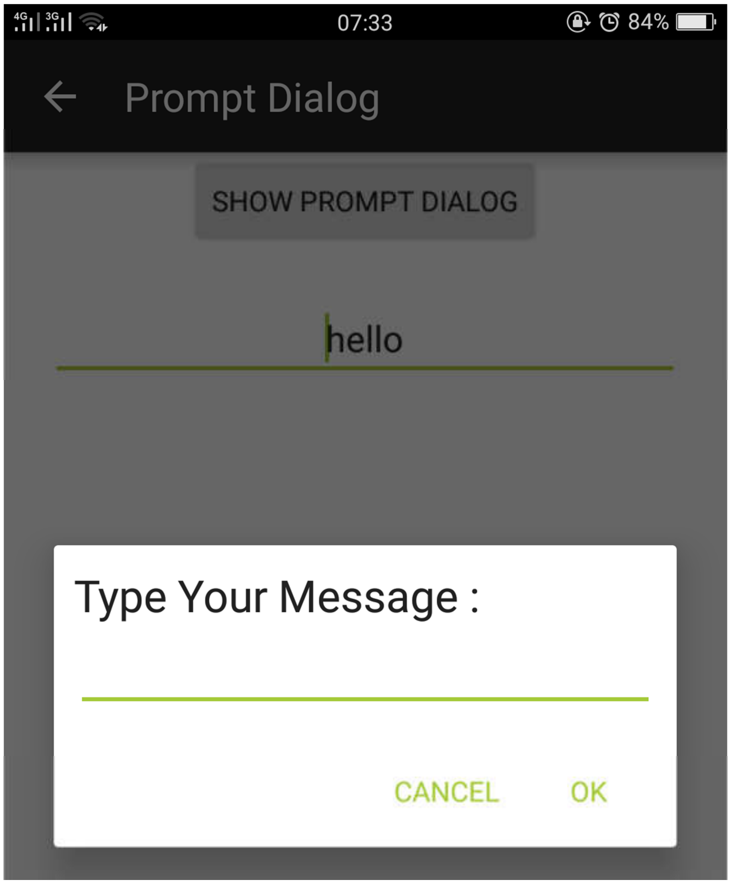
```
<LinearLayout xmlns:android="http:// schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical" >

<Button
android:id="@+id/buttonAlert"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

```
android:text="Show Alert Box" />

</LinearLayout>
```

MainActivity
LOCATION: java/MainActivity.java

```
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class MainActivity extends Activity {

final Context context = this;
private Button button;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);
setContentView(R.layout.main);

button = (Button) findViewById(R.id.buttonAlert);

// add button listener
button.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View arg0) {

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(context);

// set title
alertDialogBuilder.setTitle("Your Title");

// set dialog message
alertDialogBuilder
.setMessage("Click yes to exit!")
.setCancelable(false)
.setPositiveButton("Yes",new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog,int id) {

// if this button is clicked, close
// current activity
MainActivity.this.finish();
}
})
.setNegativeButton("No",new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog,int id){
// if this button is clicked, just close
// the dialog box and do nothing
dialog.cancel();
}
});
// create alert dialog
AlertDialog alertDialog = alertDialogBuilder.create();

// show it
alertDialog.show();
}
});
}
}
```

3. Frompt dialog



hello

Type Your Message :

activity_main
LOCATION: res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical" >

<Button
android:id="@+id/buttonPrompt"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Show Prompt Dialog" />

<EditText
android:id="@+id/editTextResult"
android:layout_width="match_parent"
android:layout_height="wrap_content"/>

</LinearLayout>
```

prompts
LOCATION: res/layout/prompts.xml

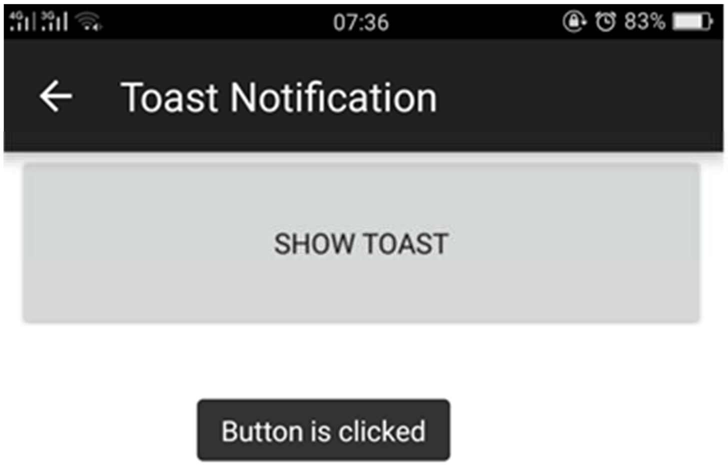
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http:// schemas.android.com/apk/res/android"
android:id="@+id/layout_root"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical"
android:padding="10dp" >

<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Type Your Message : "
android:textAppearance="?android:attr/textAppearanceLarge" />

<EditText
android:id="@+id/editTextDialogUserInput"
android:layout_width="match_parent"
android:layout_height="wrap_content" >
<requestFocus />
</EditText>
```

```
</LinearLayout>
```

MainActivity
LOCATION: java/MainActivity.java

```
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity {

final Context context = this;
private Button button;
private EditText result;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);
setContentView(R.layout.main);

// components from main.xml
button = (Button) findViewById(R.id.buttonPrompt);
result = (EditText) findViewById(R.id.editTextResult);

// add button listener
button.setOnClickListener(new OnClickListener() {


@Override
public void onClick(View arg0) {

// get prompts.xml view
LayoutInflater li = LayoutInflater.from(context);
View promptsView = li.inflate(R.layout.prompts, null);

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(context);

// set prompts.xml to alertdialog builder
alertDialogBuilder.setView(promptsView);
final EditText userInput = (EditText) promptsView.findViewById(R.id.
editTextDialogUserInput);

// set dialog message
alertDialogBuilder
.setCancelable(false)
.setPositiveButton("OK",
new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog,int id) {

// get user input and set it to result
// edit text
result.setText(userInput.getText());
}
}
).setNegativeButton("Cancel",
new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog,int id) {
dialog.cancel();
}
});

// create alert dialog
AlertDialog alertDialog = alertDialogBuilder.create();

// show it
alertDialog.show();
}
});
}
}
```

4. Toast notification



Button is clicked

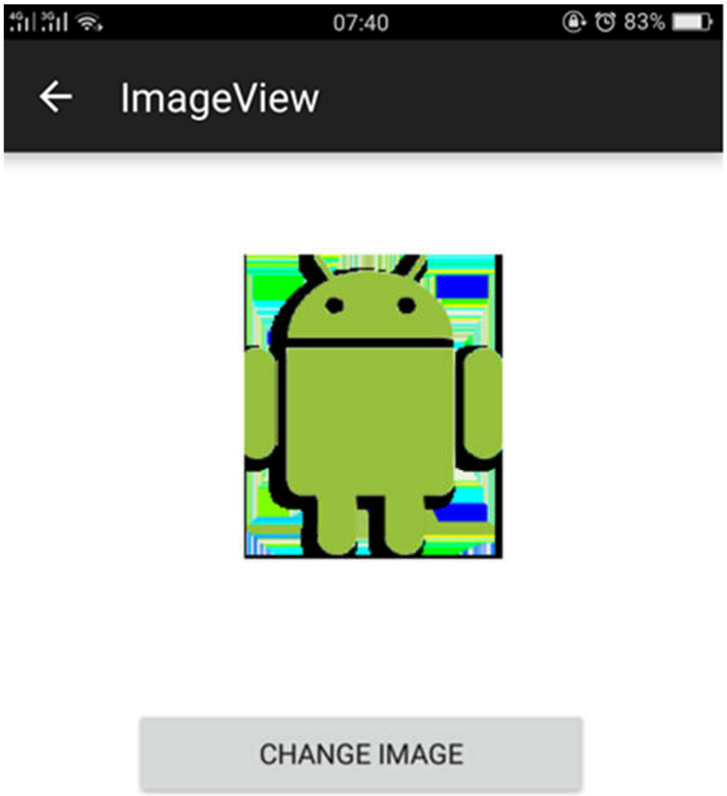
activity_main
LOCATION: res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http:// schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical" >

<Button
android:id="@+id/buttonToast"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Show Toast" />

</LinearLayout>
```

MainActivity
LOCATION: java/MainActivity.java

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends Activity {

private Button button;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);
setContentView(R.layout.main);

button = (Button) findViewById(R.id.buttonToast);

button.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View arg0) {

Toast.makeText(getApplicationContext(), "Button is clicked",
Toast.LENGTH_LONG).show();

}
});
}
}
```

5. ImageView



activity_main
LOCATION: res/layout/activity_main.xml

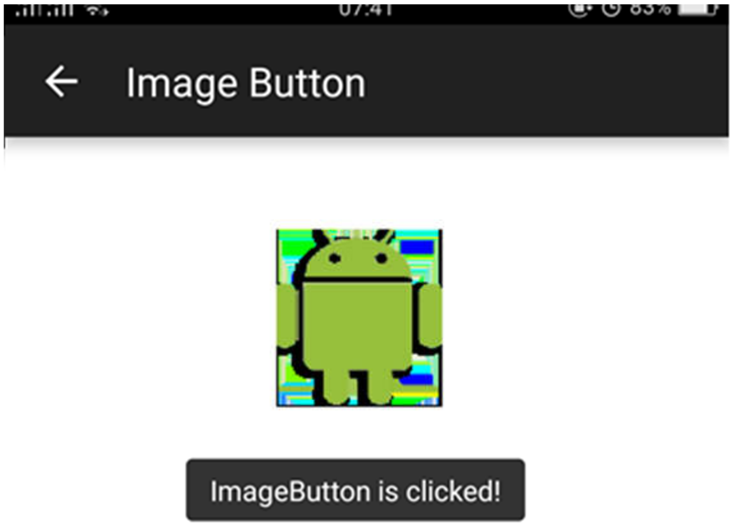
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http:// schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical" >

<ImageView
android:id="@+id/imageView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:src="@drawable/image_one" />

<Button
android:id="@+id/btnChangeImage"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Change Image" />
```

```
</LinearLayout>
```

MainActivity
LOCATION: java/MainActivity.java

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.Button;
import android.widget.ImageView;
import android.view.View;
import android.view.View.OnClickListener;

public class MyAndroidAppActivity extends Activity {

Button button;
ImageView image;

@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);

addListenerOnButton();

}

public void addListenerOnButton() {

image = (ImageView) findViewById(R.id.imageView1);
button = (Button) findViewById(R.id.btnChangeImage);
button.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View arg0) {
image.setImageResource(R.drawable. image_two);
}
});
}
}
```

6. Image button



ImageButton is clicked!

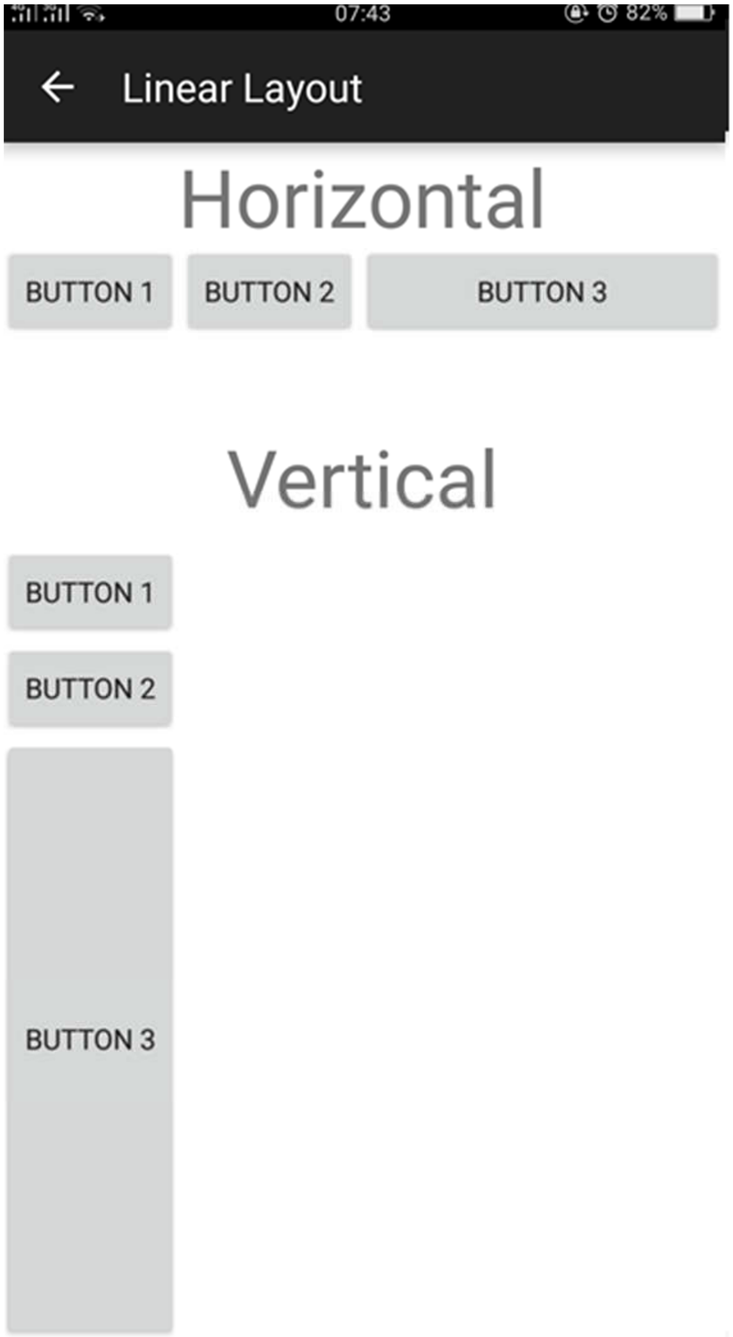
activity_main
LOCATION: res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http:// schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical" >

<ImageButton
android:id="@+id/imageButton1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:src="@drawable/imagebutton" />

</LinearLayout>
```

MainActivity
LOCATION: java/MainActivity.java

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.ImageButton;
import android.widget.Toast;
import android.view.View;
import android.view.View.OnClickListener;

public class MyAndroidAppActivity extends Activity {

ImageButton imageButton;

@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);

addListenerOnButton();

} public void addListenerOnButton() {
imageButton = (ImageButton) findViewById(R.id.imageButton1);

imageButton.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View arg0) {

Toast.makeText(MyAndroidAppActivity.this,
"ImageButton is clicked!", Toast.LENGTH_SHORT).show();
}
});
}
}
```

7. Linear layout



activity_main
LOCATION: res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http:// schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="horizontal" >

<Button
```

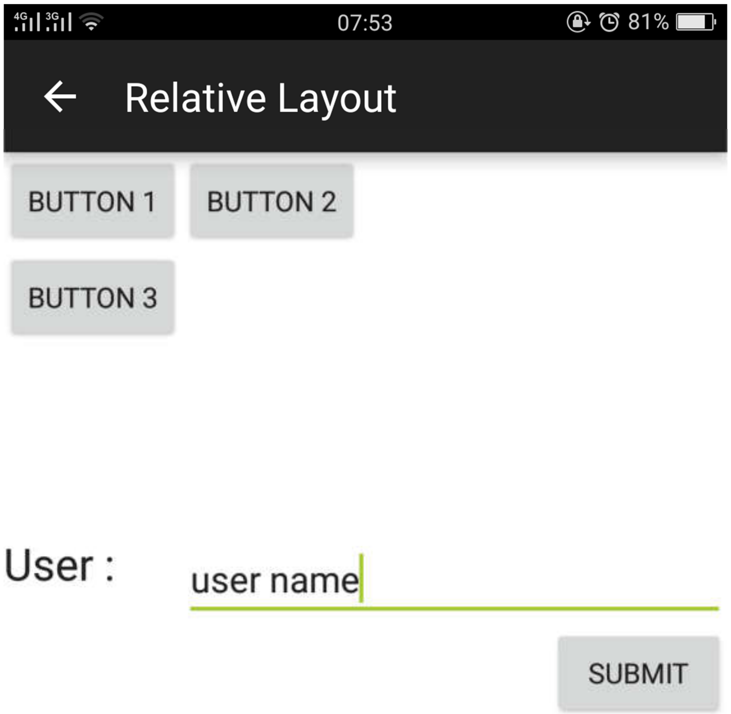
```
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Button 1" />

<Button
android:id="@+id/button2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Button 2" />

<Button
android:id="@+id/button3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Button 3"
android:layout_weight="1"/>

</LinearLayout>
```

activity_main
LOCATION: res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http:// schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical" >

<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Button 1" />

<Button
android:id="@+id/button2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Button 2" />

<Button
android:id="@+id/button3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Button 3"
android:layout_weight="1"/>
```

```
</LinearLayout>
```

8. Relative layout



Note
In Android, RelativeLayout let you position your component base on the nearby (relative or sibling) component's position. It's the most flexible layout, that allow you to position your component to display in anywhere you want (if you know how to "relative" it).

In RelativeLayout, you can use "above, below, left and right" to arrange the component position, for example, display a "button1" below "button2", or display "button3" on right of the "button1".

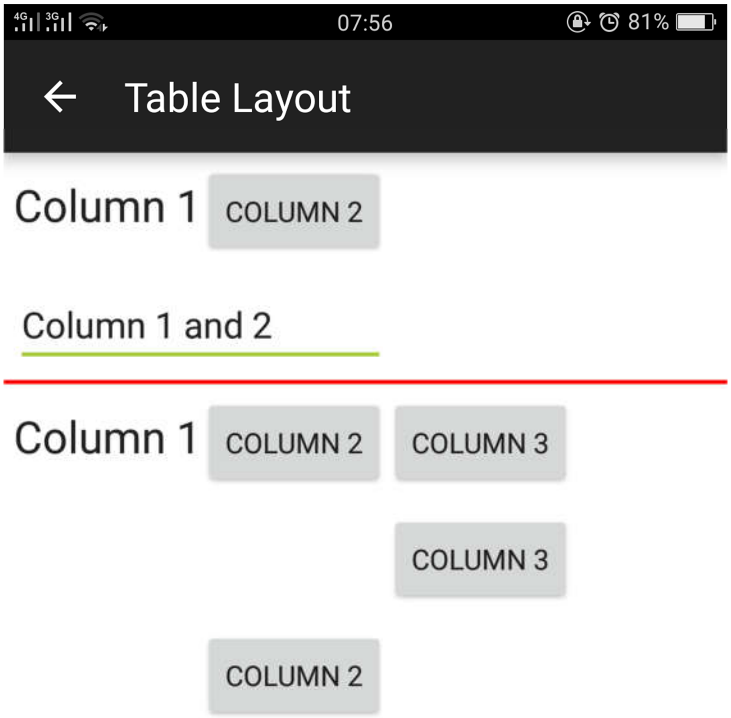
In this tutorial, I show you how to arrange / position button, textview and editbox via "RelativeLayout".

activity_main
LOCATION: res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http:// schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent" >
```

```
<Button
android:id="@+id/btnButton1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Button 1"/>

<Button
android:id="@+id/btnButton2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Button 2"
android:layout_toRightOf="@+id/btnButton1"/>

<Button
android:id="@+id/btnButton3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Button 3"
android:layout_below="@+id/btnButton1"/>

<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/btnButton3"
android:layout_marginTop="94dp"
android:text="User :"
android:textAppearance="?android:attr/textAppearanceLarge" />

<EditText
android:id="@+id/editText1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentRight="true"
android:layout_alignTop="@+id/textView1"
android:layout_toRightOf="@+id/btnButton3" />

<Button
android:id="@+id/btnSubmit"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentRight="true"
android:layout_below="@+id/editText1"
android:text="Submit" />

</RelativeLayout>
```

9. Table layout



Note
In Android, TableLayout let you arranges components in rows and columns, just like the standard table layout in HTML, <tr> and <td>.

In this tutorial, we show you how to use TableLayout to arrange button, textview and edittext in rows and columns format, and also demonstrates the use of "android:layout_span" to span view in 2 cells, and "android:layout_column" to display the view in specified column.

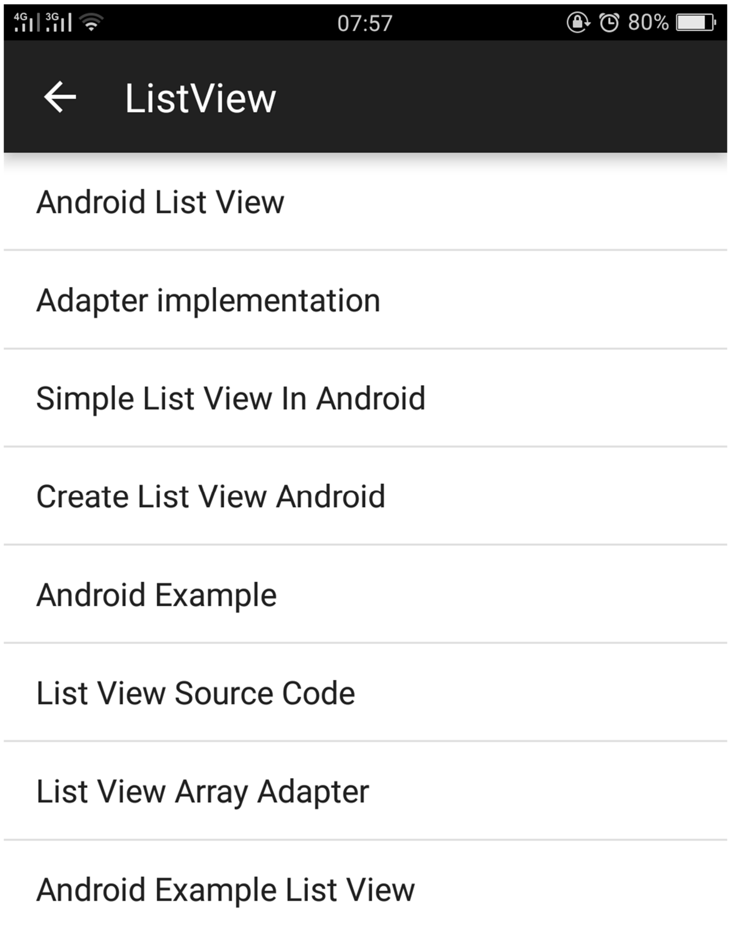
activity_main
LOCATION: res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http:// schemas.android.com/apk/res/android"
android:id="@+id/tableLayout1"
android:layout_width="fill_parent"
android:layout_height="fill_parent" >


<TableRow
android:id="@+id/tableRow1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:padding="5dip" >
```

```
<TextView
android:id="@+id/textView1"
android:text="Column 1"
android:textAppearance="?android:attr/textAppearanceLarge" />

<Button
android:id="@+id/button1"
android:text="Column 2" />
</TableRow>


<TableRow
android:id="@+id/tableRow2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:padding="5dip" >

<EditText
android:id="@+id/editText1"
android:layout_span="2"
android:text="Column 1 & 2" />
</TableRow>


<View
android:layout_height="2dip"
android:background="#FF0000" />


<TableRow
android:id="@+id/tableRow3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:padding="5dip" >

<TextView
android:id="@+id/textView2"
android:text="Column 1"
android:textAppearance="?android:attr/textAppearanceLarge" />

<Button
android:id="@+id/button2"
android:text="Column 2" />

<Button
android:id="@+id/button3"
android:text="Column 3" />
</TableRow>
```

```
<TableRow
android:id="@+id/tableRow4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:padding="5dip" >

<Button
android:id="@+id/button4"
android:layout_column="2"
android:text="Column 3" />
</TableRow>


<TableRow
android:id="@+id/tableRow5"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:padding="5dip" >

<Button
android:id="@+id/button5"
android:layout_column="1"
android:text="Column 2" />
</TableRow>

</TableLayout>
```

10. List view



Android List View

Adapter implementation

Simple List View In Android

Create List View Android

Android Example

List View Source Code

List View Array Adapter

Android Example List View

activity_main
LOCATION: res/layout/activity_main.xml

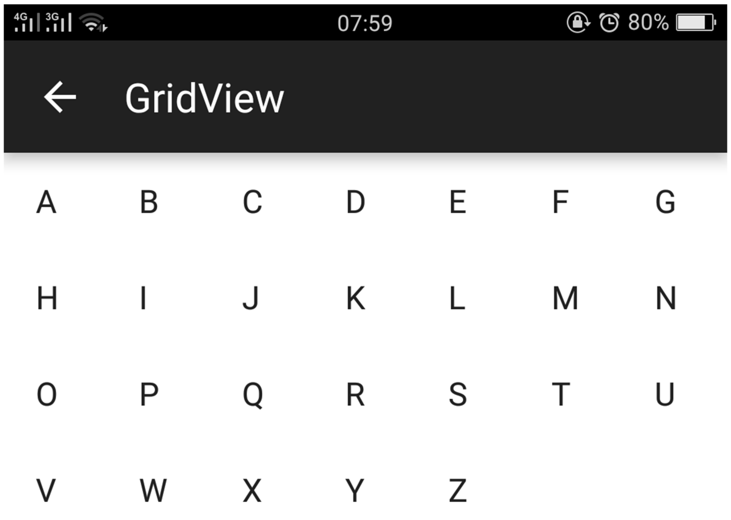
```
<LinearLayout xmlns:android="http:// schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical" >

<ListView
android:id="@+id/list"
android:layout_height="wrap_content"
android:layout_width="match_parent">
</ListView>

</LinearLayout>
```

MainActivity
LOCATION: java/MainActivity.java

```
import android.app.ListActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.AdapterView. OnItemClickListener;

public class MyAndroidAppActivity extends Activity {
ListView listView ;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);

// Get ListView object from xml
listView = (ListView) findViewById(R.id.list);

// Defined Array values to show in ListView
String[] values = new String[] { "Android List View",
"Adapter implementation",
"Simple List View In Android",
"Create List View Android",
"Android Example",
"List View Source Code",
"List View Array Adapter",
"Android Example List View"
};

// Define a new Adapter
// First parameter - Context
// Second parameter - Layout for the row
// Third parameter - ID of the TextView to which the data is written
// Forth - the Array of data

ArrayAdapter adapter = new ArrayAdapter(this,
android.R.layout.simple_list_item_1, android.R.id.text1, values);

// Assign adapter to ListView
listView.setAdapter(adapter);

// ListView Item Click Listener
listView.setOnItemClickListener(new OnItemClickListener() {

@Override
public void onItemClick(AdapterView parent, View view,int position, long id) {
```

```
// ListView Clicked item index
int itemPosition = position;
// ListView Clicked item value
String itemValue = (String) listView.getItemAtPosition(position);

// Show Alert
Toast.makeText(getApplicationContext(),
itemValue , Toast.LENGTH_LONG)
.show();

}
});
}
}
```

11. Grid view



A B C D E F G

H I J K L M N

O P Q R S T U

V W X Y Z

activity_main
LOCATION: res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<GridView xmlns:android="http:// schemas.android.com/apk/res/android"
android:id="@+id/gridView1"
android:numColumns="auto_fit"
android:gravity="center"
android:columnWidth="50dp"
android:stretchMode="columnWidth"
android:layout_width="fill_parent"
android:layout_height="fill_parent" >

</GridView>
```

MainActivity
LOCATION: java/MainActivity.java

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.GridView;
import android.widget.TextView;
import android.widget.Toast;
import android.view.View;
import android.widget.AdapterView. OnItemClickListener;

public class GridViewActivity extends Activity {

GridView gridView;

static final String[] numbers = new String[] {
"A", "B", "C", "D", "E",
"F", "G", "H", "I", "J",
"K", "L", "M", "N", "O",
"P", "Q", "R", "S", "T",
"U", "V", "W", "X", "Y", "Z"};

@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

setContentView(R.layout.main);

gridView = (GridView) findViewById(R.id.gridView1);

ArrayAdapter adapter = new ArrayAdapter(this,android.R.layout. simple_list_item_1,
numbers);

gridView.setAdapter(adapter);

gridView.setOnItemClickListener(new OnItemClickListener() {
public void onItemClick(AdapterView parent, View v,
int position, long id) {
Toast.makeText(getApplicationContext(),((TextView) v).getText(),
Toast.LENGTH_SHORT).show();
}
});
}
}
```