Android UI Control [Button]

Android – UI Control

Button

- **Button** is a user interface control that is used to perform an action whenever the user clicks or tap on it.

- Buttons in android will contain a text or an icon or both and perform an action when the user touches it.

- Following is the pictorial representation of using **Buttons** in android applications. 

- We have a different type of buttons available to use based on our requirements, those are **ImageButton**, **ToggleButton**, **RadioButton**.

- In android, we can create **Button** control in two ways either in XML layout file or create it in Activity file programmatically.

Android – UI Control

Create a Button in Layout File

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical" android:layout_width="match_parent"
        android:layout_height="match_parent">
        <Button
           android:id="@+id/addBtn"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:text="Add" />
    </LinearLayout>
```


Android – UI Control

Create Button Control in Activity File

```
LinearLayout layout = (LinearLayout)findViewById(R.id.l_layout);
    Button btn = new Button(this);
    btn.setText("Test");
    layout.addView(btn);
```

Android – UI Control

Android Handle Button Click Events

- Whenever the user clicks on a **Button**, the **Button** object will receives an on-click event.

- In android, we can define a button click event in two ways either in the XML layout file or create it in the activity file programmatically.

- **Define Button Click Event in XML Layout File**
  - We can define click event handler for button by adding **android:onClick** attribute to the **<Button>** element in our XML layout file.
  - The value of **android:onClick** attribute must be the name of the method which we need to call in response to a click event and the activity file which hosting XML layout must implement the corresponding method.

Android – UI Control

Android Handle Button Click Events

- Following is the example of defining a button click event using **android:onClick** attribute in an XML layout file.

```
<?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">
      <Button
      android:id="@+id/addBtn"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Add"
      android:onClick="addOperation"/>
</LinearLayout>
```

Android – UI Control

Android Handle Button Click Events

- In Activity that hosts our XML layout file, we need to implement click event method like as shown below

```
/** Called when the user touches the button */
public void addOperation(View view) {
   // Do something in response to the button click
}
```


Android – UI Control

Define Button Click Event in Activity File

- We can define the button click event programmatically in the activity file rather than XML layout file.

- To define button click programmatically, create **View.OnClickListener** object and assign it to the button by calling **setOnClickListener(View.OnClickListener)** like as shown below.

```
Button btnAdd = (Button)findViewById(R.id.addBtn);
btnAdd.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
// Do something in response to button click
}
});
}
```

Android – UI Control

Android Button Control Attributes

Attribute	Description
android:id	It is used to uniquely identify the control
android:gravity	It is used to specify how to align the text like left, right, center, top, etc.
android:text	It is used to set the text.
android:textColor	It is used to change the color of the text.
android:textSize	It is used to specify the size of the text.
android:textStyle	It is used to change the style (bold, italic, bolditalic) of text.
android:background	It is used to set the background color for button control.

Android – UI Control

Android Button Control Attributes

Attribute	Description
android:padding	It is used to set the padding from left, right, top and bottom.
android:drawableBottom	It's drawable to be drawn to the below of text.
android:drawableRight	It's drawable to be drawn to the right of text.
android:drawableLeft	It's drawable to be drawn to the left of the text.

Android – UI Control

Android Button control Example

- In this example we define a **Button** and two **EditText** controls in **LinearLayout** to get the data of **EditText** controls when click on **Button** in android application.

- Create a new android application using android studio and give names as **ButtonExample**.

- Now open an **activity_main.xml** file from **\res\layout** path and write the code like as shown below

Android – UI Control

Android Button Control Example

activity_main.xml

```
<?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">
   <TextView
      android:id="@+id/fstTxt"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="100dp"
      android:layout_marginTop="150dp"
      android:text="First Number" />
   <EditText
      android:id="@+id/firstNum"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="100dp"
      android:ems="10" />
   TextView
      android:id="@+id/secTxt"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Second Number"
      android:layout_marginLeft="100dp" />
   <EditText
      android:id="@+id/secondNum"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="100dp"
      android:ems="10" />
   <Button
      android:id="@+id/addBtn"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="100dp"
      android:text="Add" />
</LinearLayout>
```

Android – UI Control

Android Button control Example

MainActivity.java

```
package com.tutlane.buttonexample;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      final EditText firstNum = (EditText)findViewById(R.id.firstNum);
      final EditText secNum = (EditText)findViewById(R.id.secondNum);
      Button btnAdd = (Button)findViewById(R.id.addBtn);
```

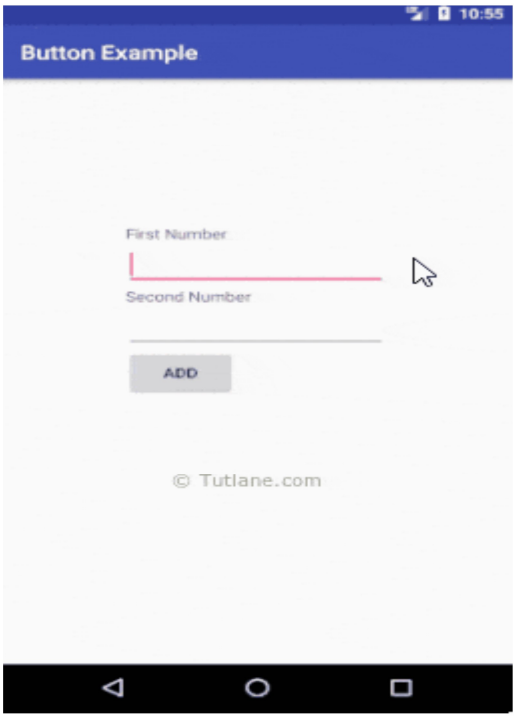

Android – UI Control

Android Button control Example

```
btnAdd.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            if(firstNum.getText().toString().isEmpty() ||
                secNum.getText().toString().isEmpty())
            {
               Toast.makeText(getApplicationContext(), "Please fill all the fields",
                Toast.LENGTH_SHORT).show();
            }
            else {
               int num1 = Integer.parseInt(firstNum.getText().toString());
               int num2 = Integer.parseInt(secNum.getText().toString());
               Toast.makeText(getApplicationContext(), "SUM = " + (num1 + num2),
                Toast.LENGTH_SHORT).show();
            }
         }
      });
   }
}
```

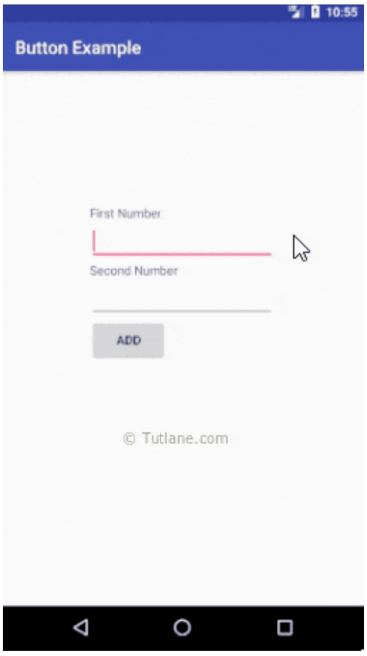
Android – UI Control

Output of Android Button Example



Android – UI Control

Output of Android Button Example (Once we enter details in all fields and click on Button we will get a result like as shown below.)




THANKS