

Event Handling in Android

Events are a useful way to collect data about a user's interaction with interactive components of Applications. Like button presses or screen touch etc. The Android framework maintains an event queue as first-in, first-out (FIFO) basis. You can capture these events in your program and take appropriate action as per requirements. There are following three concepts related to Android Event Management –

Event Listeners –

An event listener is an interface in the View class that contains a single callback method. These methods will be called by the Android framework when the View to which the listener has been registered is triggered by user interaction with the item in the UI.

Event Listeners Registration –

Event Registration is the process by which an Event Handler gets registered with an Event Listener so that the handler is called when the Event Listener fires the event.

Event Handlers –

When an event happens and we have registered an event listener for the event, the event listener calls the Event Handlers, which is the method that actually handles the event.

Event Listeners & Event Handlers

Event Handler	Event Listener & Description
onClick()	OnClickListener() This is called when the user either clicks or touches or focuses upon any widget like button, text, image etc. You will use onClick() event handler to handle such event.
onLongClick()	OnLongClickListener() This is called when the user either clicks or touches or focuses upon any widget like button, text, image etc. for one or more seconds. You will use onLongClick() event

	handler to handle such event.
onFocusChange()	<p>OnFocusChangeListener()</p> <p>This is called when the widget loses its focus ie. user goes away from the view item. You will use onFocusChange() event handler to handle such event.</p>
onKey()	<p>OnFocusChangeListener()</p> <p>This is called when the user is focused on the item and presses or releases a hardware key on the device. You will use onKey() event handler to handle such event.</p>
onTouch()	<p>OnTouchListener()</p> <p>This is called when the user presses the key, releases the key, or any movement gesture on the screen. You will use onTouch() event handler to handle such event.</p>
onMenuItemClick()	<p>OnMenuItemClickListener()</p> <p>This is called when the user selects a menu item. You will use onMenuItemClick() event handler to handle such event.</p>
onCreateContextMenu()	<p>onCreateContextMenuListener()</p> <p>This is called when the context menu is being built(as the result of a sustained "long click)</p>

There are many more event listeners available as a part of **View** class like OnHoverListener, OnDragListener etc which may be needed for your application. So I

recommend to refer official documentation for Android application development in case you are going to develop a sophisticated apps.

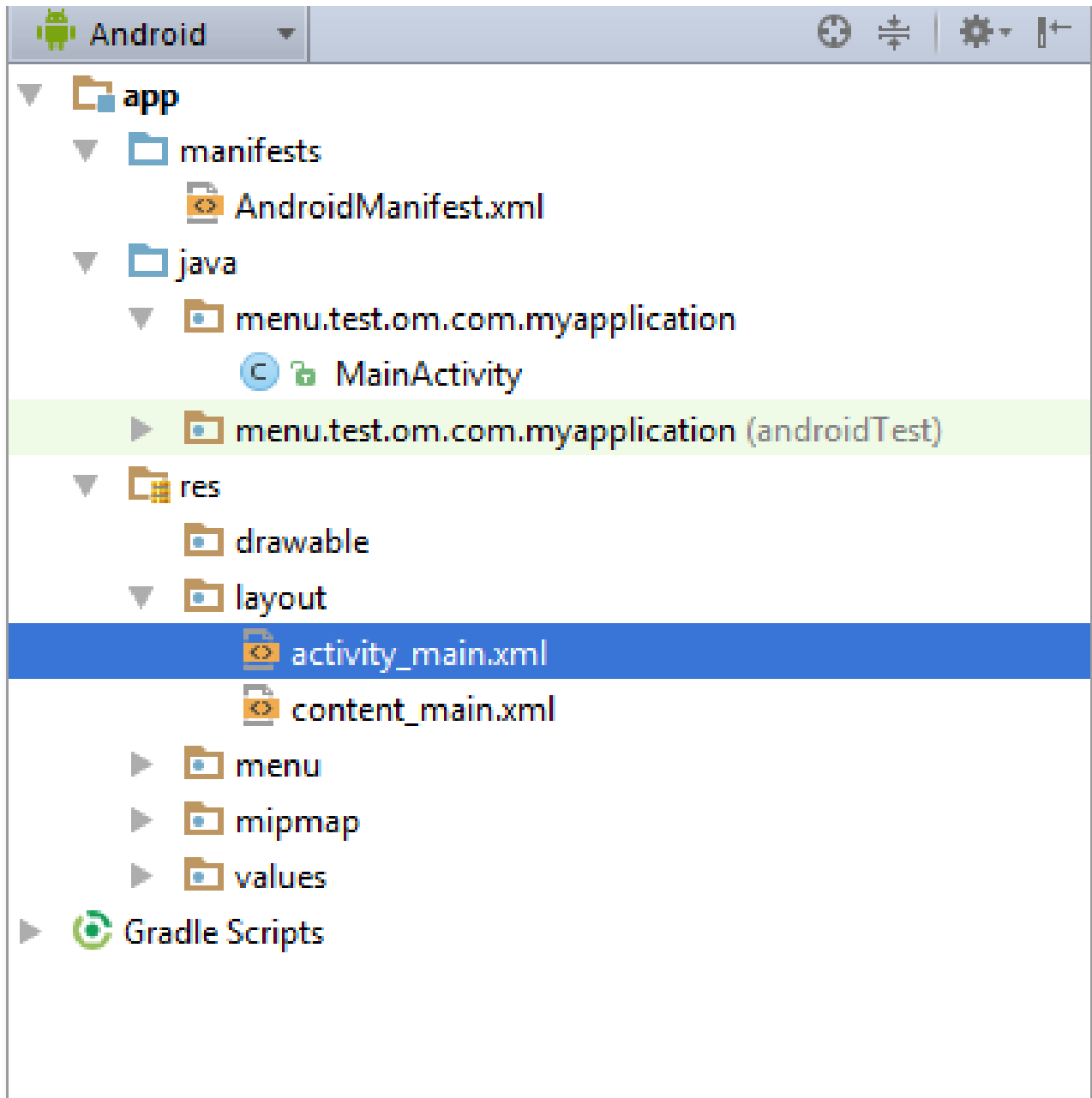
There are three ways to handle event in android programming just see below examples which shows you three types:

Type 1: Activity class implements the Listener interface

Type 2: Using Layout file of "content_main.xml" to specify event handler directly

Type 3: Using an Anonymous Inner Class

Type 1: Interface implementing example for onclick event:



(1)step 1: write code for MainActivity.java:

```
package menu.test.om.com.myapplication;

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

public class MainActivity extends Activity implements OnClickListener
```

```

{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button b=(Button) findViewById(R.id.button);
        b.setOnClickListener(this);
    }

    public void onClick(View v)
    {
        EditText e1=(EditText) findViewById(R.id.text1);
        EditText e2=(EditText) findViewById(R.id.text2);
        e2.setText(e1.getText().toString());
    }
}

```

(2)write code for activity_main.xml:

```

<?xml version="1.0" encoding="utf-8" ?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:fitsSystemWindows="true"
    tools:context=".MainActivity">

    <android.support.design.widget.AppBarLayout android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main" />

```

```
<android.support.design.widget.FloatingActionButton android:id="@+id/fab"
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:layout_gravity="bottom|end" android:layout_margin="@dimen/fab_margin"
    android:src="@android:drawable/ic_dialog_email" />
```

```
</android.support.design.widget.CoordinatorLayout>
```

(3)now write code for content_main.xml :

```
<?xml version="1.0" encoding="utf-8"?>
```

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

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:paddingRight="@dimen/activity_horizontal_margin"
```

```
    android:paddingTop="@dimen/activity_vertical_margin"
```

```
    android:paddingBottom="@dimen/activity_vertical_margin"
```

```
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
```

```
    tools:showIn="@layout/activity_main" tools:context=".MainActivity">
```

```
<Button
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="New Button"
```

```
    android:id="@+id/button"
```

```
    android:layout_alignParentTop="true"
```

```
    android:layout_centerHorizontal="true"
```

```
    android:layout_marginTop="204dp" />
```

```
<EditText
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

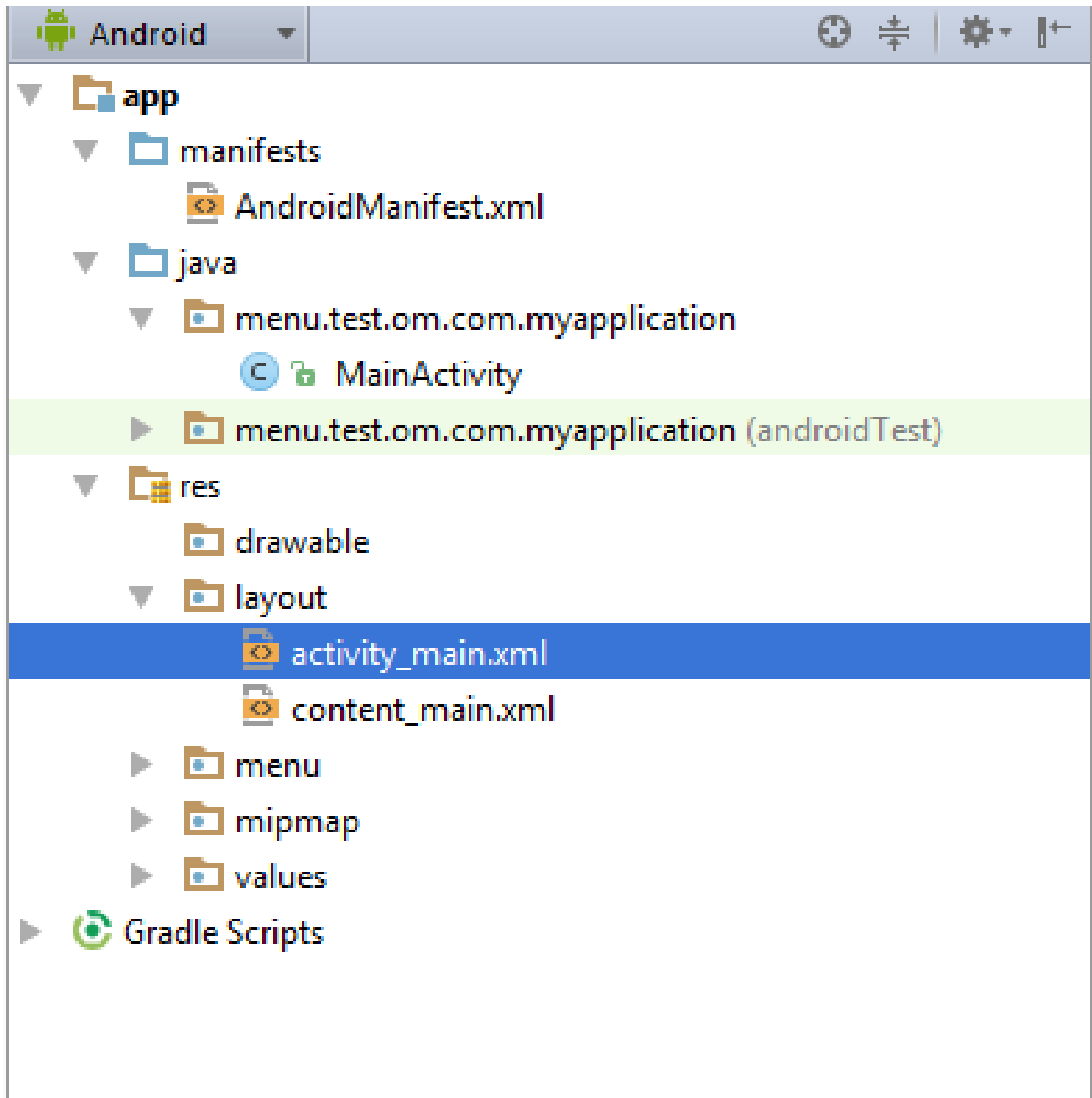
```
android:id="@+id/text1"
android:text="it is text1"
android:layout_alignParentTop="true"
android:layout_alignLeft="@+id/button"
android:layout_alignStart="@+id/button" />
```

```
<EditText
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/text2"
android:text="it is text2"
android:layout_below="@+id/text1"
android:layout_alignLeft="@+id/text1"
android:layout_alignStart="@+id/text1"
android:layout_marginTop="42dp" />
```

```
</RelativeLayout>
```

Type 2: Implementing onclick event using xml file



(1) write code for MainActivity.java:

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

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



```

    }

    public void onclick(View View )
    {
        EditText e1=(EditText) findViewById(R.id.text1);
        EditText e2=(EditText) findViewById(R.id.text2);
        e2.setText(e1.getText().toString());
    }
}

```

(2)write code for activity_main.xml:

```

<?xml version="1.0" encoding="utf-8" ?>

<android.support.design.widget.CoordinatorLayout

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

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

    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"

    android:layout_height="match_parent" android:fitsSystemWindows="true"

    tools:context=".MainActivity">

    <android.support.design.widget.AppBarLayout android:layout_height="wrap_content"

        android:layout_width="match_parent"

        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar android:id="@+id/toolbar"

            android:layout_width="match_parent"

            android:layout_height="?attr/actionBarSize"

            android:background="?attr/colorPrimary"

            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main" />

    <android.support.design.widget.FloatingActionButton android:id="@+id/fab"

        android:layout_width="wrap_content" android:layout_height="wrap_content"

        android:layout_gravity="bottom|end" android:layout_margin="@dimen/fab_margin"

        android:src="@android:drawable/ic_dialog_email" />

```

```
</android.support.design.widget.CoordinatorLayout>
```

(3) write code for content_main.xml:

```
<?xml version="1.0" encoding="utf-8" ?>
```

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

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:paddingRight="@dimen/activity_horizontal_margin"
```

```
    android:paddingTop="@dimen/activity_vertical_margin"
```

```
    android:paddingBottom="@dimen/activity_vertical_margin"
```

```
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
```

```
    tools:showIn="@layout/activity_main" tools:context=".MainActivity">
```

```
<EditText
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/text1"
```

```
    android:layout_alignParentTop="true"
```

```
    android:layout_centerHorizontal="true"
```

```
    android:layout_marginTop="67dp"
```

```
    android:text="it ise text1" />
```

```
<EditText
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/text2"
```

```
    android:layout_below="@+id/text1"
```

```
    android:layout_centerHorizontal="true"
```

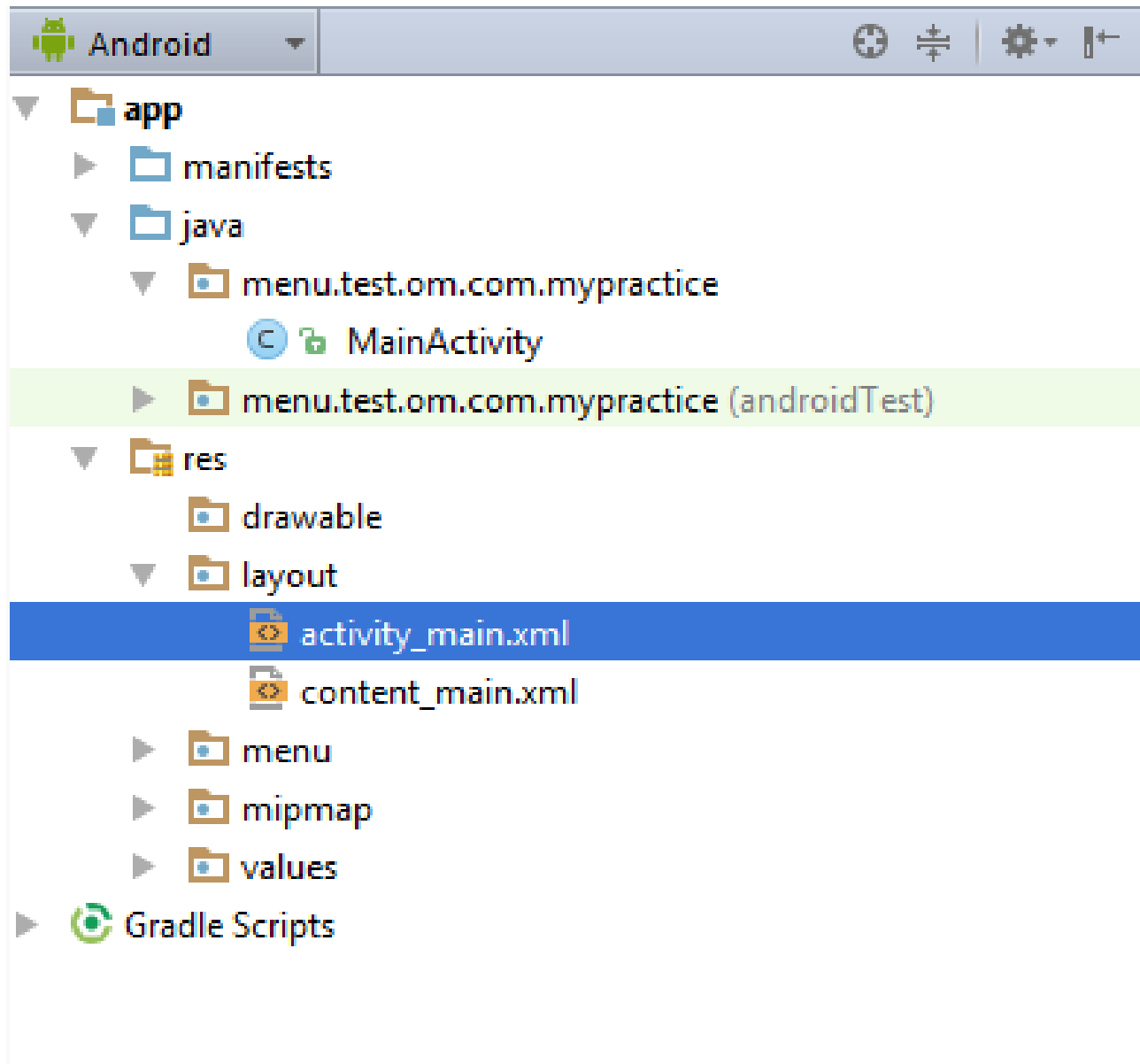
```
    android:layout_marginTop="39dp"
```

```
    android:text="it is text2" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="click "
    android:id="@+id/button"
    android:onClick="onclick"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true" />
</RelativeLayout>
```

=====

Type 3:Anonyms class example for onclick event.



(1) Write code for MainActivity.java:

```
package menu.test.om.com.mypractice;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

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

```

Button b=(Button)findViewById(R.id.button);
b.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        EditText t1=(EditText)findViewById(R.id.editText1);
        EditText t2=(EditText)findViewById(R.id.editText2);
        t2.setText(t1.getText().toString());
    }
});
}
}

```

(2)write code for activity_main.xml :

```

<?xml version="1.0" encoding="utf-8" ?>

<android.support.design.widget.CoordinatorLayout

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

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

    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"

    android:layout_height="match_parent" android:fitsSystemWindows="true"

    tools:context=".MainActivity">

    <android.support.design.widget.AppBarLayout android:layout_height="wrap_content"

        android:layout_width="match_parent"

        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar android:id="@+id/toolbar"

            android:layout_width="match_parent"

            android:layout_height="?attr/actionBarSize"

            android:background="?attr/colorPrimary"

            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main" />

    <android.support.design.widget.FloatingActionButton android:id="@+id/fab"

        android:layout_width="wrap_content" android:layout_height="wrap_content"

        android:layout_gravity="bottom|end" android:layout_margin="@dimen/fab_margin"

```

```
    android:src="@android:drawable/ic_dialog_email" />
```

```
</android.support.design.widget.CoordinatorLayout>
```

(3)write code for content_main.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
```

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

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"  
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:paddingRight="@dimen/activity_horizontal_margin"
```

```
    android:paddingTop="@dimen/activity_vertical_margin"
```

```
    android:paddingBottom="@dimen/activity_vertical_margin"
```

```
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
```

```
    tools:showIn="@layout/activity_main" tools:context=".MainActivity">
```

```
<EditText
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/editText1"
```

```
    android:layout_marginTop="132dp"
```

```
    android:text="it is edit text1" />
```

```
<EditText
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/editText2"
```

```
    android:layout_centerVertical="true"
```

```
    android:layout_alignParentLeft="true"
```

```
    android:layout_alignParentStart="true"
```

```
    android:text="it is edit text2" />
```

```
<Button
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="click"  
    android:id="@+id/button"  
    android:layout_marginTop="36dp"  
    android:layout_below="@+id/editText2"  
    android:layout_alignRight="@+id/editText2"  
    android:layout_alignEnd="@+id/editText2" />
```

```
</RelativeLayout>
```