

Android Option Menu Example

Android Option Menus are the primary menus of android. They can be used for settings, search, delete item etc.

Here, we are going to see two examples of option menus. First, the simple option menus and second, options menus with images.

Here, we are inflating the menu by calling the **inflate()** method of **MenuInflater** class. To perform event handling on menu items, you need to override **onOptionsItemSelected()** method of Activity class.

Android Option Menu Example

Let's see how to create menu in android. Let's see the simple option menu example that contains three menu items.

activity_main.xml

We have only one textview in this file.

File: activity_main.xml

1. `<?xml version="1.0" encoding="utf-8"?>`
2. `<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"`
3. `xmlns:app="http://schemas.android.com/apk/res-auto"`
4. `xmlns:tools="http://schemas.android.com/tools"`
5. `android:layout_width="match_parent"`
6. `android:layout_height="match_parent"`
7. `tools:context="example.javatpoint.com.optionmenu.MainActivity">`
- 8.
9. `<android.support.design.widget.AppBarLayout`
10. `android:layout_width="match_parent"`
11. `android:layout_height="wrap_content"`
12. `android:theme="@style/AppTheme.AppBarOverlay">`
- 13.
14. `<android.support.v7.widget.Toolbar`
15. `android:id="@+id/toolbar"`
16. `android:layout_width="match_parent"`
17. `android:layout_height="?attr/actionBarSize"`
18. `android:background="?attr/colorPrimary"`
19. `app:popupTheme="@style/AppTheme.PopupOverlay" />`
- 20.
21. `</android.support.design.widget.AppBarLayout>`
- 22.
23. `<include layout="@layout/content_main" />`

24.

25. **</android.support.design.widget.CoordinatorLayout>**

File: context_main.xml

1. **<?xml** version="1.0" encoding="utf-8"?>
2. **<android.support.constraint.ConstraintLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. xmlns:app="http://schemas.android.com/apk/res-auto"
4. xmlns:tools="http://schemas.android.com/tools"
5. android:layout_width="match_parent"
6. android:layout_height="match_parent"
7. app:layout_behavior="@string/appbar_scrolling_view_behavior"
8. tools:context="example.javatpoint.com.optionmenu.MainActivity"
9. tools:showIn="@layout/activity_main">
- 10.
11. **<TextView**
12. android:layout_width="wrap_content"
13. android:layout_height="wrap_content"
14. android:text="Hello World!"
15. app:layout_constraintBottom_toBottomOf="parent"
16. app:layout_constraintLeft_toLeftOf="parent"
17. app:layout_constraintRight_toRightOf="parent"
18. app:layout_constraintTop_toTopOf="parent" />
- 19.
20. **</android.support.constraint.ConstraintLayout>**

menu_main.xml

It contains three items as show below. It is created automatically inside the res/menu directory.

File: menu_main.xml

1. **<menu** xmlns:android="http://schemas.android.com/apk/res/android"
2. xmlns:app="http://schemas.android.com/apk/res-auto"
3. xmlns:tools="http://schemas.android.com/tools"
4. tools:context="example.javatpoint.com.optionmenu.MainActivity">
- 5.
6. **<item** android:id="@+id/item1"
7. android:title="Item 1"/>
8. **<item** android:id="@+id/item2"
9. android:title="Item 2"/>
10. **<item** android:id="@+id/item3"
11. android:title="Item 3"
12. app:showAsAction="withText"/>
13. **</menu>**

Activity class

This class displays the content of menu.xml file and performs event handling on clicking the menu items.

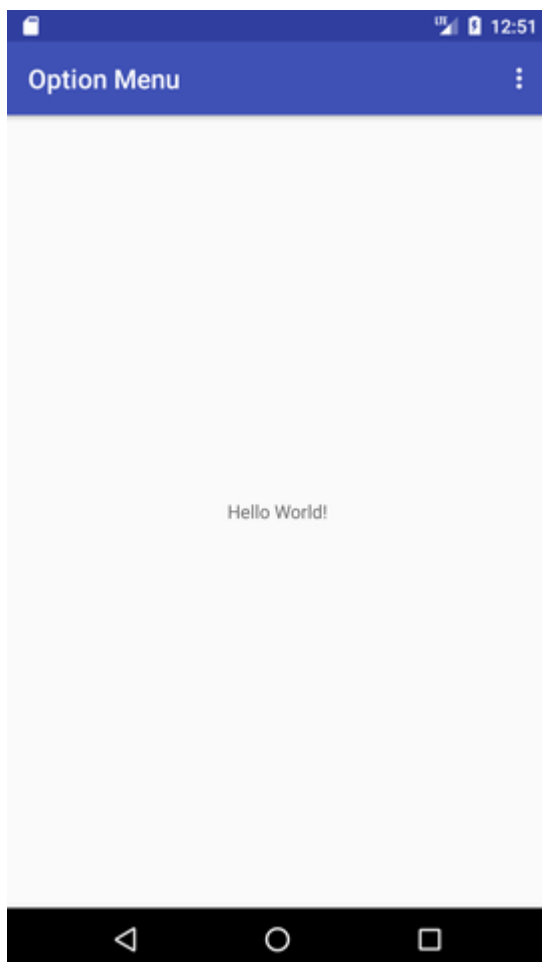
File: MainActivity.java

```
1. package example.javatpoint.com.optionmenu;
2.
3. import android.os.Bundle;
4. import android.support.v7.app.AppCompatActivity;
5. import android.support.v7.widget.Toolbar;
6. import android.view.Menu;
7. import android.view.MenuItem;
8. import android.widget.Toast;
9.
10. public class MainActivity extends AppCompatActivity {
11.
12.     @Override
13.     protected void onCreate(Bundle savedInstanceState) {
14.         super.onCreate(savedInstanceState);
15.         setContentView(R.layout.activity_main);
16.         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
17.         setSupportActionBar(toolbar);
18.     }
19.
20.     @Override
21.     public boolean onCreateOptionsMenu(Menu menu) {
22.         // Inflate the menu; this adds items to the action bar if it is present.
23.         getMenuInflater().inflate(R.menu.menu_main, menu);
24.         return true;
25.     }
26.
27.     @Override
28.     public boolean onOptionsItemSelected(MenuItem item) {
29.         int id = item.getItemId();
30.         switch (id){
31.             case R.id.item1:
32.                 Toast.makeText(getApplicationContext(),"Item 1 Selected",Toast.LENGTH_LONG).show();
33.                 return true;
34.             case R.id.item2:
35.                 Toast.makeText(getApplicationContext(),"Item 2 Selected",Toast.LENGTH_LONG).show();
```

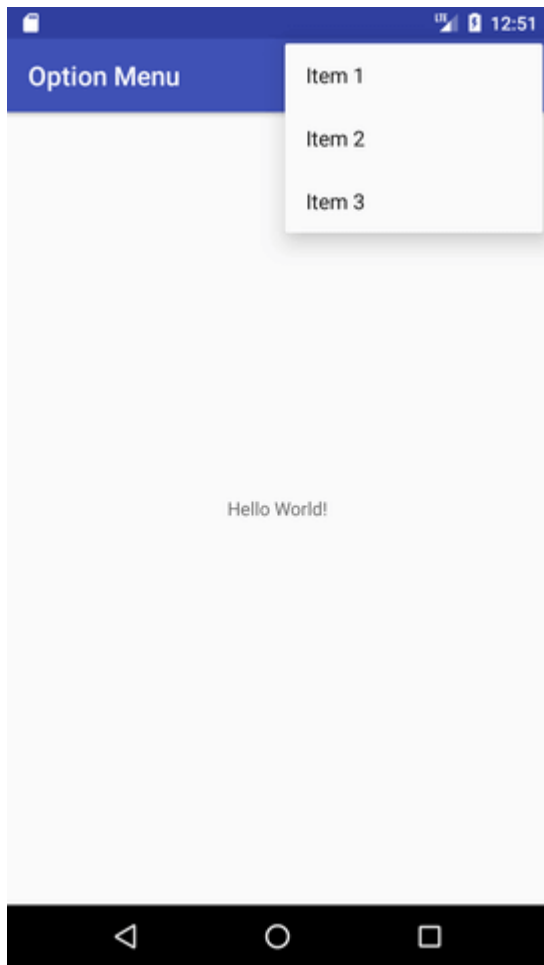
```
36.         return true;
37.     case R.id.item3:
38.         Toast.makeText(getApplicationContext(), "Item 3 Selected", Toast.LENGTH_LO
    NG).show();
39.         return true;
40.     default:
41.         return super.onOptionsItemSelected(item);
42.     }
43. }
44. }
```

Output:

Output without clicking on the menu button.



Output after clicking on the menu button.



Output after clicking on the second menu item .



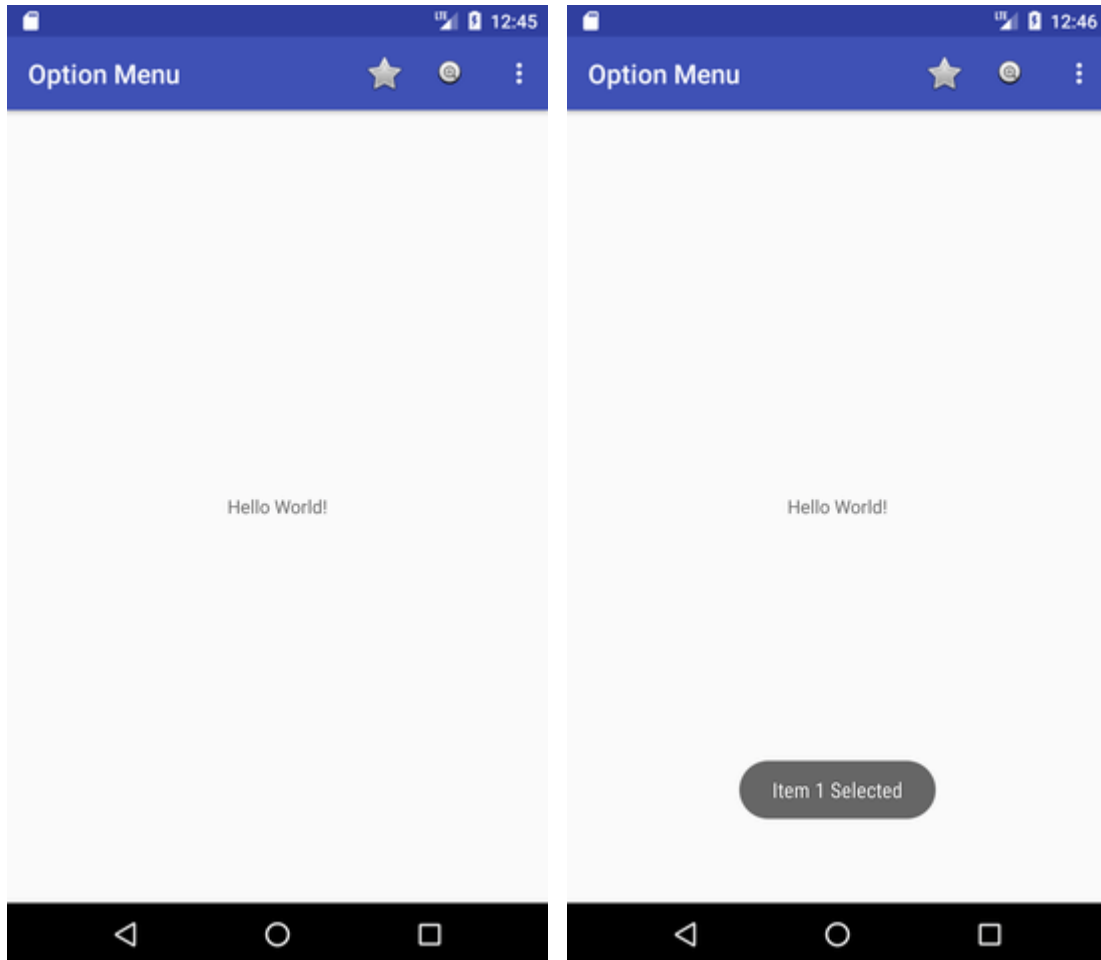
Option Menu with Icon

You need to have icon images inside the res/drawable directory. The `android:icon` element is used to display the icon on the option menu. You can write the string information in the strings.xml file. But we have written it inside the menu_main.xml file.

File: menu_main.xml

1. `<menu xmlns:android="http://schemas.android.com/apk/res/android"`
2. `xmlns:app="http://schemas.android.com/apk/res-auto"`
3. `xmlns:tools="http://schemas.android.com/tools"`
4. `tools:context="example.javatpoint.com.optionmenu.MainActivity">`
- 5.
6. `<item android:id="@+id/item1"`
7. `android:title="Item 1"`
8. `app:showAsAction="always"`
9. `android:icon="@android:drawable/btn_star" />`
10. `<item android:id="@+id/item2"`
11. `android:title="Item 2"`
12. `app:showAsAction="ifRoom"`

13. `android:icon="@android:drawable/btn_plus"/>`
14. `<item android:id="@+id/item3"`
15. `android:title="Item 3"`
16. `app:showAsAction="withText"`
17. `android:icon="@android:drawable/btn_plus"/>`
18. `</menu>`



Android Context Menu Example

Android context menu appears when user press long click on the element. It is also known as floating menu.

It affects the selected content while doing action on it.

It doesn't support item shortcuts and icons.

Android Context Menu Example

Let's see the simple example of context menu in android.

activity_main.xml

Drag one listview from the pallette, now the xml file will look like this:

File: activity_main.xml

1. `<?xml version="1.0" encoding="utf-8"?>`
 2. `<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"`
 3. `xmlns:app="http://schemas.android.com/apk/res-auto"`
 4. `xmlns:tools="http://schemas.android.com/tools"`
 5. `android:layout_width="match_parent"`
 6. `android:layout_height="match_parent"`
 7. `tools:context="example.javatpoint.com.contextmenu.MainActivity">`
 - 8.
 9. `<ListView`
 10. `android:layout_width="368dp"`
 11. `android:layout_height="495dp"`
 12. `android:id="@+id/listView"`
 13. `android:layout_marginEnd="8dp"`
 14. `android:layout_marginStart="8dp"`
 15. `android:layout_marginTop="8dp"`
 16. `app:layout_constraintEnd_toEndOf="parent"`
 17. `app:layout_constraintHorizontal_bias="0.0"`
 18. `app:layout_constraintStart_toStartOf="parent"`
 19. `app:layout_constraintTop_toTopOf="parent" />`
 20. `</android.support.constraint.ConstraintLayout>`
-

main_menu.xml

Create a separate menu_main.xml file in menu directory for menu items.

1. `<?xml version="1.0" encoding="utf-8"?>`
2. `<menu xmlns:android="http://schemas.android.com/apk/res/android">`
3. `<item android:id="@+id/call"`
4. `android:title="Call" />`
5. `<item android:id="@+id/sms"`
6. `android:title="SMS" />`
7. `</menu>`

Activity class

Let's write the code to display the context menu on press of the listview.

File: MainActivity.java

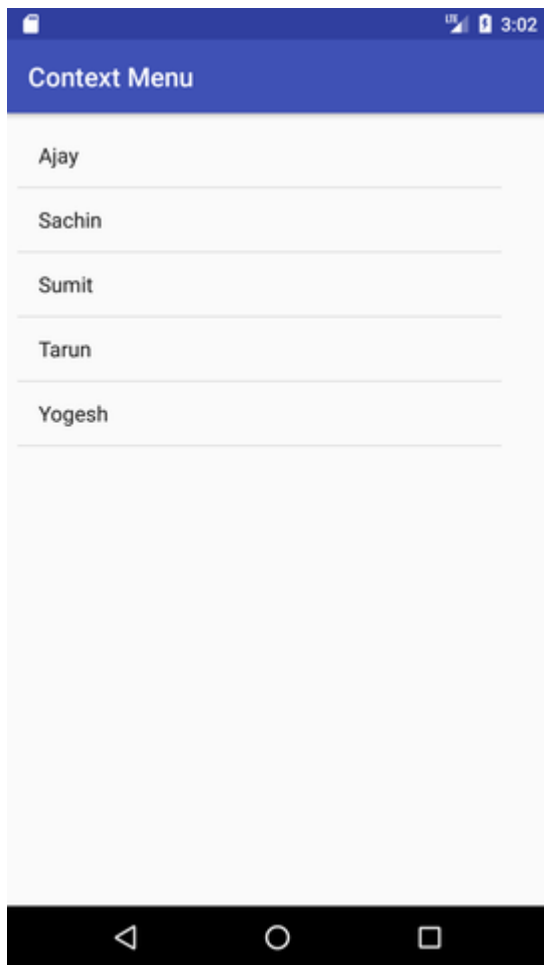

```

1. package example.javatpoint.com.contextmenu;
2.
3. import android.support.v7.app.AppCompatActivity;
4. import android.os.Bundle;
5. import android.view.ContextMenu;
6. import android.view.MenuInflater;
7. import android.view.MenuItem;
8. import android.view.View;
9. import android.widget.AdapterView;
10. import android.widget.ListView;
11. import android.widget.Toast;
12.
13. public class MainActivity extends AppCompatActivity {
14.     ListView listView;
15.     String contacts[]={"Ajay","Sachin","Sumit","Tarun","Yogesh"};
16.     @Override
17.     protected void onCreate(Bundle savedInstanceState) {
18.         super.onCreate(savedInstanceState);
19.         setContentView(R.layout.activity_main);
20.         listView=(ListView)findViewById(R.id.listView);
21.         ArrayAdapter<String> adapter=new ArrayAdapter<String>(this,android.R.layout.
            simple_list_item_1,contacts);
22.         listView.setAdapter(adapter);
23.         // Register the ListView for Context menu
24.         registerForContextMenu(listView);
25.     }
26.     @Override
27.     public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.Context
        MenuInfo menuInfo)
28.     {
29.         super.onCreateContextMenu(menu, v, menuInfo);
30.         MenuInflater inflater = getMenuInflater();
31.         inflater.inflate(R.menu.menu_main, menu);
32.         menu.setHeaderTitle("Select The Action");
33.     }
34.     @Override
35.     public boolean onContextItemSelected(MenuItem item){
36.         if(item.getItemId()==R.id.call){
37.             Toast.makeText(getApplicationContext(),"calling code",Toast.LENGTH_LONG).sh
                ow();
38.         }
39.         else if(item.getItemId()==R.id.sms){
40.             Toast.makeText(getApplicationContext(),"sending sms code",Toast.LENGTH_LO
                NG).show();

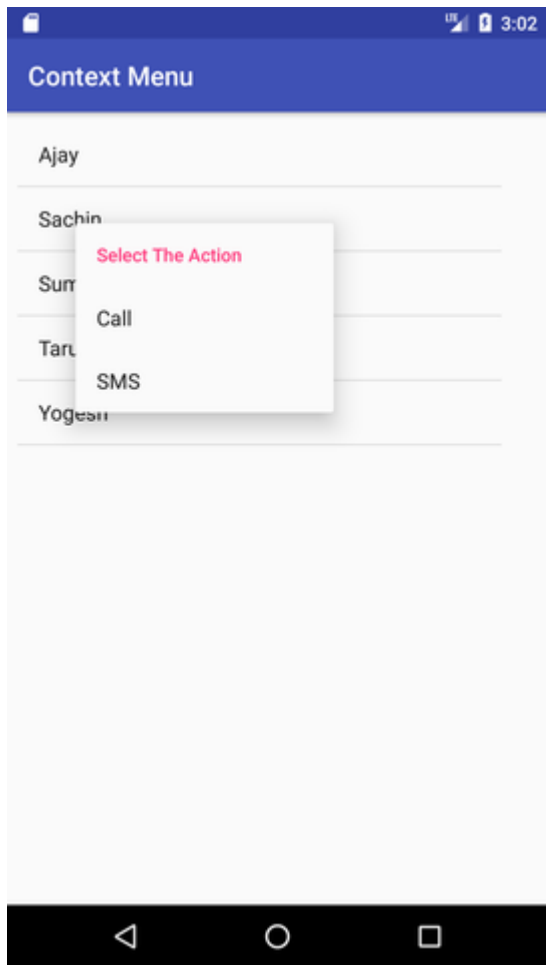
```

```
41.     }else{
42.         return false;
43.     }
44.     return true;
45. }
46. }
```

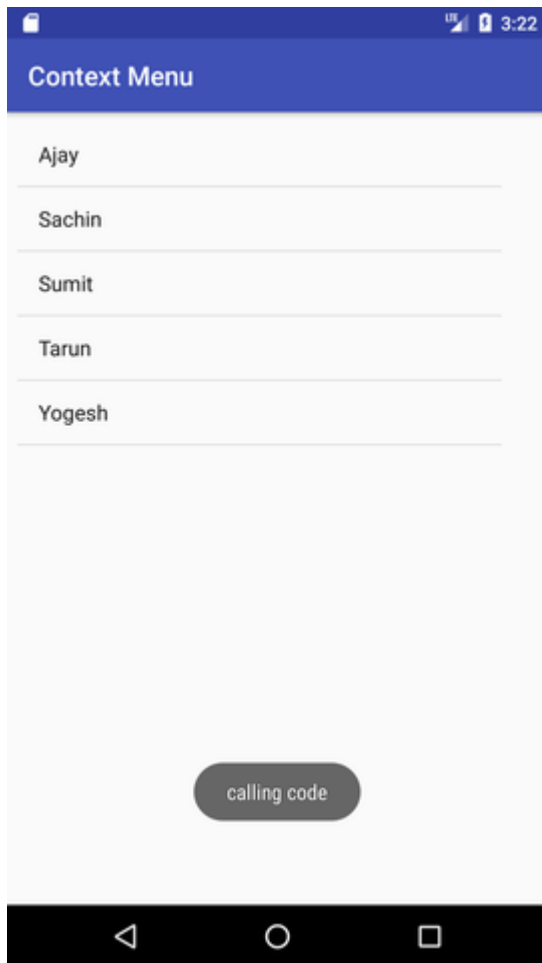
Output:



Output after long press on the listview.



Output after clicking on the context menu.



Android Popup Menu Example

Android Popup Menu displays the menu below the anchor text if space is available otherwise above the anchor text. It disappears if you click outside the popup menu.

The `android.widget.PopupMenu` is the direct subclass of `java.lang.Object` class.

Android Popup Menu Example

Let's see how to create popup menu in android.

`activity_main.xml`

It contains only one button.

File: activity_main.xml

1. `<?xml version="1.0" encoding="utf-8"?>`
2. `<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"`
3. `xmlns:app="http://schemas.android.com/apk/res-auto"`

```

4.   xmlns:tools="http://schemas.android.com/tools"
5.   android:layout_width="match_parent"
6.   android:layout_height="match_parent"
7.   tools:context="example.javatpoint.com.popupmenu.MainActivity">
8.
9.   <Button
10.    android:layout_width="wrap_content"
11.    android:layout_height="wrap_content"
12.    android:id="@+id/button"
13.    android:text="Click"
14.    app:layout_constraintBottom_toBottomOf="parent"
15.    app:layout_constraintLeft_toLeftOf="parent"
16.    app:layout_constraintRight_toRightOf="parent"
17.    app:layout_constraintTop_toTopOf="parent" />
18.
19. </android.support.constraint.ConstraintLayout>

```

popup_menu.xml

It contains three items as show below. It is created inside the res/menu directory.

File: poupup_menu.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <menu xmlns:android="http://schemas.android.com/apk/res/android">
3.   <item
4.     android:id="@+id/one"
5.     android:title="One" />
6.   <item
7.     android:id="@+id/two"
8.     android:title="Two" />
9.   <item
10.    android:id="@+id/three"
11.    android:title="Three" />
12. </menu>

```

Activity class

It displays the popup menu on button click.

File: MainActivity.java

```

1. package example.javatpoint.com.popupmenu;
2.
3. import android.support.v7.app.AppCompatActivity;
4. import android.os.Bundle;

```

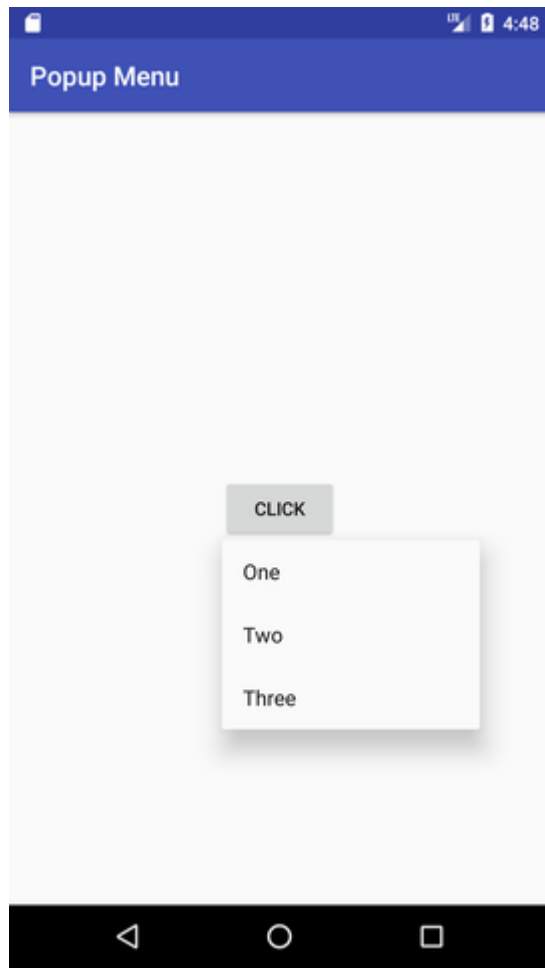
```

5. import android.view.MenuItem;
6. import android.view.View;
7. import android.widget.Button;
8. import android.widget.PopupMenu;
9. import android.widget.Toast;
10.
11. public class MainActivity extends AppCompatActivity {
12. Button button;
13.
14.     @Override
15.     protected void onCreate(Bundle savedInstanceState) {
16.         super.onCreate(savedInstanceState);
17.         setContentView(R.layout.activity_main);
18.
19.         button = (Button) findViewById(R.id.button);
20.         button.setOnClickListener(new View.OnClickListener() {
21.
22.             @Override
23.             public void onClick(View v) {
24.                 //Creating the instance of PopupMenu
25.                 PopupMenu popup = new PopupMenu(MainActivity.this, button);
26.                 //Inflating the Popup using xml file
27.                 popup.getMenuInflater().inflate(R.menu.popup_menu, popup.getMenu());
28.
29.                 //registering popup with OnMenuItemClickListener
30.                 popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListen
er() {
31.                     public boolean onMenuItemClick(MenuItem item) {
32.                         Toast.makeText(MainActivity.this, "You Clicked : " + item.getTitle(), Toa
st.LENGTH_SHORT).show();
33.                         return true;
34.                     }
35.                 });
36.
37.                 popup.show();//showing popup menu
38.             }
39.         });//closing the setOnClickListener method
40.     }
41. }

```

[download this android example](#)

Output:





4:48

Popup Menu

CLICK

You Clicked : Two

