

Android Option Menu

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.  import android.os.Bundle;
2.  import android.support.v7.app.AppCompatActivity;
3.  import android.support.v7.widget.Toolbar;
4.  import android.view.Menu;
5.  import android.view.MenuItem;
6.  import android.widget.Toast;
7.
8.  public class MainActivity extends AppCompatActivity {
9.
10.     @Override
11.     protected void onCreate(Bundle savedInstanceState) {
12.         super.onCreate(savedInstanceState);
13.         setContentView(R.layout.activity_main);
14.         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
15.         setSupportActionBar(toolbar);
16.     }
17.
18.     @Override
19.     public boolean onCreateOptionsMenu(Menu menu) {

```

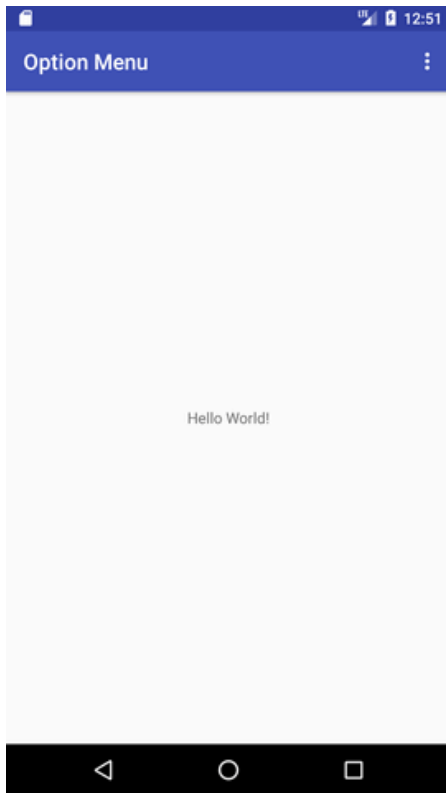
```

20. // Inflate the menu; this adds items to the action bar if it is present.
21. getMenuInflater().inflate(R.menu.menu_main, menu);
22. return true;
23. }
24.
25. @Override
26. public boolean onOptionsItemSelected(MenuItem item) {
27.     int id = item.getItemId();
28.     switch (id){
29.         case R.id.item1:
30.             Toast.makeText(getApplicationContext(),"Item 1 Selected",Toast.LENGTH_LONG).show();
31.             return true;
32.         case R.id.item2:
33.             Toast.makeText(getApplicationContext(),"Item 2 Selected",Toast.LENGTH_LONG).show();
34.             return true;
35.         case R.id.item3:
36.             Toast.makeText(getApplicationContext(),"Item 3 Selected",Toast.LENGTH_LONG).show();
37.             return true;
38.         default:
39.             return super.onOptionsItemSelected(item);
40.     }
41. }
42. }

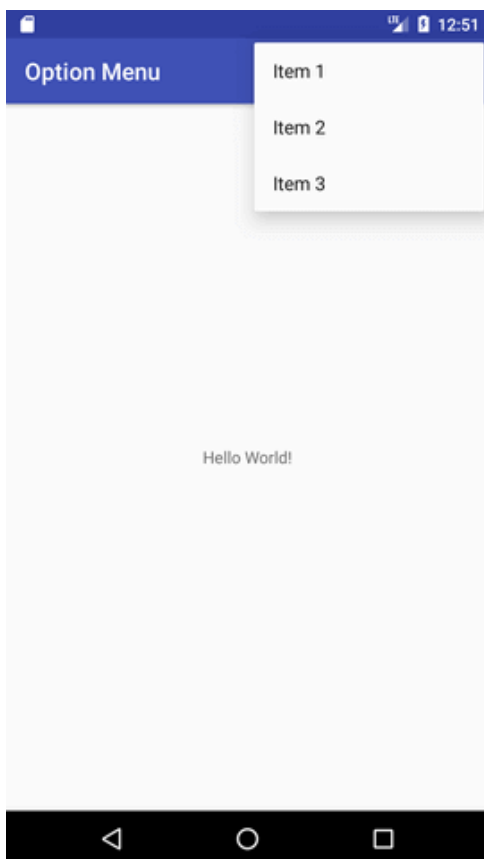
```

Output:

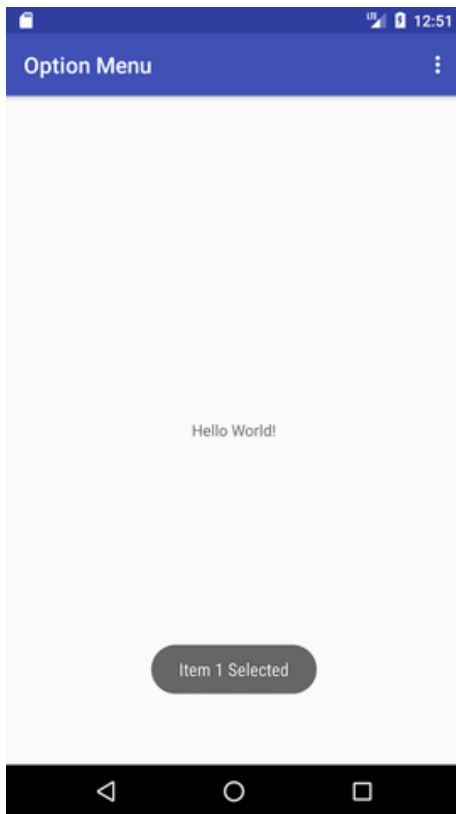
Output without clicking on the menu button.



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>`

