**Option Menu Example**

In android, **Options Menu** is a primary collection of menu items for an [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) and it is useful to implement actions that have a global impact on the app, such as Settings, Search, etc.

By using Options Menu, we can combine multiple actions and other options that are relevant to our current [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle). We can define items for the options menu from either our [Activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) or [Fragment](https://www.tutlane.com/tutorial/android/android-fragments-with-examples) class.

In case, if we define items for the options menu in both activity and fragment, then those items will be combined and display in UI.

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android" >  
    <item android:id="@+id/search\_item"  
        android:title="Search" />  
    <item android:id="@+id/upload\_item"  
        android:title="Upload" />  
    <item android:id="@+id/copy\_item"  
        android:title="Copy" />  
    <item android:id="@+id/print\_item"  
        android:title="Print" />  
    <item android:id="@+id/share\_item"  
        android:title="Share" />  
    <item android:id="@+id/bookmark\_item"  
        android:title="BookMark" />  
</menu>

Once we are done with creation of menu, we need to load this menu XML resource from our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) using **onCreateOptionsMenu()** callback method, for that open main [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) file **MainActivity.java** from **\java\com.tutlane.optionsmenu** path and write the code like as shown below.

**MainActivity.java**

package com.example.optionsmenu;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.widget.Toast;

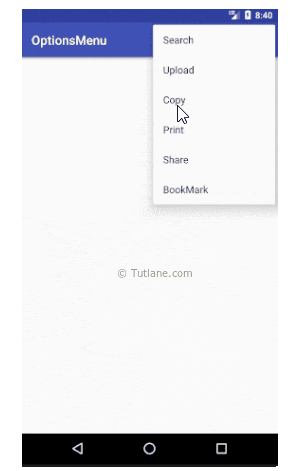
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity\_main);  
    }  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        getMenuInflater().inflate(R.menu.options\_menu, menu);  
        return true;  
    }  
    @Override  
    public boolean onOptionsItemSelected(MenuItem item) {  
        Toast.makeText(this, "Selected Item: " +item.getTitle(), Toast.LENGTH\_SHORT).show();  
        switch (item.getItemId()) {  
            case R.id.search\_item:  
               // do your code  
                return true;  
            case R.id.upload\_item:  
                // do your code  
                return true;  
            case R.id.copy\_item:  
                // do your code  
                return true;  
            case R.id.print\_item:  
                // do your code  
                return true;  
            case R.id.share\_item:  
                // do your code  
                return true;  
            case R.id.bookmark\_item:  
                // do your code  
                return true;  
            default:  
                return super.onOptionsItemSelected(item);  
        }  
    }  
}

If you observe above code we are overriding **onCreateOptionsMenu()** method in activity to create options menu and loaded defined menu resource using **MenuInflater.inflate()**.

Generally, during the launch of our activity, **onCreate()** callback method will be called by the android framework to get the required layout for an activity.

Output of Android Options Menu Example

When we run above example using an android virtual device (AVD) we will get a result like as shown below.



**Context menu**

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"  
    android:orientation="vertical" >  
    <Button  
        android:id="@+id/btnShow"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:text="Long press me"  
        android:layout\_marginTop="200dp" android:layout\_marginLeft="100dp"/>  
</LinearLayout>

If you observe above code we created a one [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) control in XML Layout file to show the context menu when we do long press on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples).

Once we are done with the creation of layout with required control, we need to load the XML layout resource from our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) **onCreate()** callback method.

**MainActivity.java**

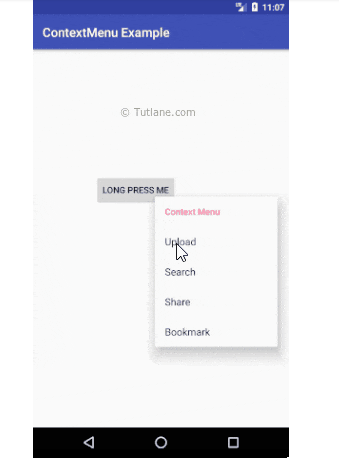
package com.tutlane.contextmenuexample;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.ContextMenu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity\_main);  
        Button btn = (Button) findViewById(R.id.btnShow);  
        registerForContextMenu(btn);  
    }  
    @Override  
    public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {  
        super.onCreateContextMenu(menu, v, menuInfo);  
        menu.setHeaderTitle("Context Menu");  
        menu.add(0, v.getId(), 0, "Upload");  
        menu.add(0, v.getId(), 0, "Search");  
        menu.add(0, v.getId(), 0, "Share");  
        menu.add(0, v.getId(), 0, "Bookmark");  
    }  
    @Override  
    public boolean onContextItemSelected(MenuItem item) {  
        Toast.makeText(this, "Selected Item: " +item.getTitle(), Toast.LENGTH\_SHORT).show();  
        return true;  
    }  
}

If you observe above code we are overriding **onCreateContextMenu()** method in activity to create context menu and registered view for context menu using **registerForContextMenu()**.

Generally, during the launch of our activity, **onCreate()** callback method will be called by android framework to get the required layout for an activity.

## Output of Android Context Menu Example

When we run above example using an android virtual device (AVD) we will get a result like as shown below.

****

## Popup Menu

## activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"  
    android:orientation="vertical" >  
  
    <Button  
        android:id="@+id/btnShow"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:text="Show Popup Menu"  
        android:layout\_marginTop="200dp" android:layout\_marginLeft="100dp"/>  
</LinearLayout>

## If you observe above code we created a one Button control in XML Layout file to show the popup menu when we click on Button.

## popup\_menu.xml

## <?xml version="1.0" encoding="utf-8"?> <menu xmlns:android="http://schemas.android.com/apk/res/android" >     <item android:id="@+id/search\_item"         android:title="Search" />     <item android:id="@+id/upload\_item"         android:title="Upload" />     <item android:id="@+id/copy\_item"         android:title="Copy" />     <item android:id="@+id/print\_item"         android:title="Print" />     <item android:id="@+id/share\_item"         android:title="Share" />     <item android:id="@+id/bookmark\_item"         android:title="BookMark" /> </menu>

Once we are done with the creation of the **menu**, we need to load this menu XML resource from our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) by instantiating a **Popup** constructor.

## MainActivity.java

## package com.tutlane.popupmenuexample; import android.support.v7.app.AppCompatActivity; import android.os.Bundle; import android.view.MenuItem; import android.view.View; import android.widget.Button; import android.widget.PopupMenu; import android.widget.Toast; public class MainActivity extends AppCompatActivity implements PopupMenu.OnMenuItemClickListener {     @Override     protected void onCreate(Bundle savedInstanceState) {         super.onCreate(savedInstanceState);         setContentView(R.layout.activity\_main);         Button btn = (Button) findViewById(R.id.btnShow);         btn.setOnClickListener(new View.OnClickListener() {             @Override             public void onClick(View v) {                 PopupMenu popup = new PopupMenu(MainActivity.this, v);                 popup.setOnMenuItemClickListener(MainActivity.this);                 popup.inflate(R.menu.popup\_menu);                 popup.show();             }         });     }     @Override     public boolean onMenuItemClick(MenuItem item) {         Toast.makeText(this, "Selected Item: " +item.getTitle(), Toast.LENGTH\_SHORT).show();         switch (item.getItemId()) {             case R.id.search\_item:                 // do your code                 return true;             case R.id.upload\_item:                 // do your code                 return true;             case R.id.copy\_item:                 // do your code                 return true;             case R.id.print\_item:                 // do your code                 return true;             case R.id.share\_item:                 // do your code                 return true;             case R.id.bookmark\_item:                 // do your code                 return true;             default:                 return false;         }     } }

If you observe above code we are trying to show popup menu on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) click, loaded defined menu resource using **Popup.inflate()** and implement popup menu items click event.

Generally, during the launch of our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle), the **onCreate()** callback method will be called by the android framework to get the required layout for an [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle).

## Output of Android Popup Menu Example

When we run above example using an android virtual device (AVD) we will get a result like as shown below.

