**Android Menu :-**

In android application Menu is one of the important user interface entity which provides some action options for a particular View(eg. activity). We should use Menu (android.view.Menu interface) to present user action and other option in the menu.

* android.view.Menu use for managing the items in the Menu.
* There are different Menu type supported by android OS.

1. Option Menu.
2. Context Menu.
3. POP Up Menu.

***Note-*** For all Menu type android provides a standard XML format to define Menu item, instead of building a Menu in a Activity class, we should define a Menu and all its Menu items in a XML Menu resource (like menu\_main.xml). By this we can inflate the Menu resource as a Menu object in Activity class.

**Explanation of Menu Resource (main.xml) :-**

This has a <menu> as root element that contains multiple <item> elements.

**<menu> -**

It is a container for menu items that can contain one or more <item> elements.

**<item>-**

It is represented a menu item.it has following attributes :-

android:id

A resource ID that's unique to the item, which allows the application can recognize the item when the user selects it.

android:icon

A reference to a drawable to use as the item's icon.

android:title

A reference to a string to use as the item's title.

android:showAsAction

Specifies when and how this item should appear as an action item in the [action bar](http://developer.android.com/guide/topics/ui/actionbar.html).

***Note-*** You can add a submenu to an item in any menu (except a submenu) by adding a <menu> element as the child of an <item>.

Ex:- <?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
    <item android:id="@+id/file"  
          android:title="@string/file" >  
        <!-- "file" submenu -->  
        <menu>  
            <item android:id="@+id/create\_new"  
                  android:title="@string/create\_new" />  
            <item android:id="@+id/open"  
                  android:title="@string/open" />  
        </menu>  
    </item>  
</menu>

**How to inflate XML menu resource?**

To use the Menu in your Activity class you need to inflate the Menu resource using following code.

MenuInflater inflater=getMenuInfalter();

inflater.inflate(R.menu.menu\_main,menuObject.);

***Note-*** Each Menu item represents different functionality/action.

1. **Option Menu :-**

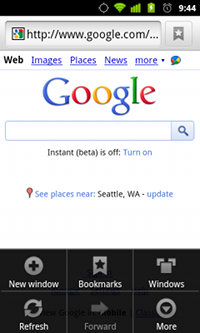
The [options menu](http://developer.android.com/guide/topics/ui/menus.html#options-menu) is the primary collection of menu items for an activity. It's where you should place actions that have a global impact on the app, such as "Search, and "Settings."

OR

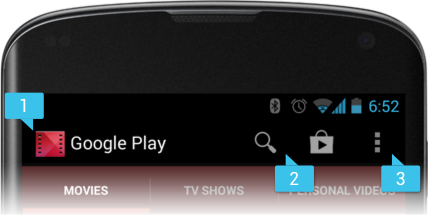
The options menu is a place where you should include actions and other options that are relevant to the current activity context, such as "Search," "Compose email," and "Settings."

Where the items in your options menu appear on the screen depends on the version for which you've developed your application:

1:- If you've developed your application for **Android 2.3.x (API level 10) or lower**, the contents of your options menu appear at the bottom of the screen when the user presses the Menu button, as shown in figure which holds up to six menu items. If your menu includes more than six items, Android places the sixth item and the rest into the overflow menu, which the user can open by selecting More.



2:- If you've developed your application for **Android 3.0 (API level 11) and higher**, items from the options menu are available in the [action bar](http://developer.android.com/guide/topics/ui/actionbar.html). By default, the system places all items in the action overflow, which the user can reveal with the action overflow icon on the right side of the action bar (or by pressing the device Menu button, if available). To enable quick access to important actions, you can promote a few items to appear in the action bar by adding android:showAsAction="ifRoom" to the corresponding <item> elements (see figure )



**Steps to create OptionMenu:-**

**Step-1:** Define menu\_main.xml in res/menu directory.

**Step-2:** Inflate xml menu resource by-

@override onCreateOptionMenu(Menu menu) of Activity

class in user defined Activity class and in this method we will write

code for inflating menu resource into Menu object. e.g.-

***@override***

*public boolean onCreateOptionMenu(Menu menu)*

*{*

*MenuInflater inflater=getMenuInfalter();*

*inflater.inflate(R.menu.menu\_main,menu);*

*return true;*

*}*

***Note-*** Always return true otherwise option menu will not be displayed by android OS this method invoke by android OS when the user open the menu for the first time.

**Step-3:** **Handling Click event:-**

@override

onOptionItemSelected(MenuItem item) method of Activity class in user defined Activity class because when the user select a menu item from option menu then the OS calls this method and passes selected Item as an object of implementing class of MenuItem interface.

* We can identify the item by calling getItemId() method.
* getItemId() method of MenuItem interface which returns an unique ID of selected MenuItem.

**@override**

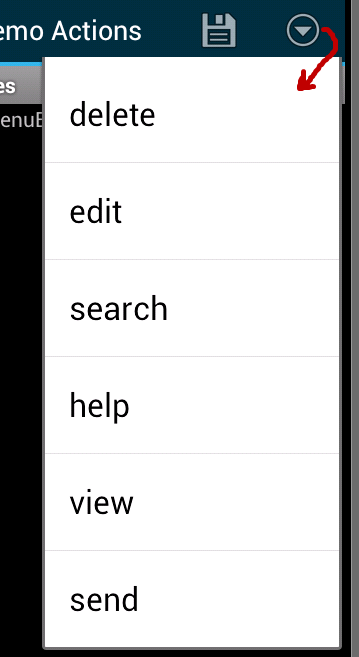
public boolean onOptionItemSelected(MenuItem item)

{

switch(item.getItemId())

{

case R.id.item1:

-  -

return true; case R.id.item2:

-

-

return true;

case R.id.item3:

-

-

return true;

default:

super.onCreateItemSelected(item);

or

return false;

}

}

Note:- the default implementation returns false.

**2. Context Menu :-**

* It is a sub Interface of Menu interface defined in android.view package and object of its implementing class of this interface represents functionality of a Menu that appear when a user long presses any view.
* We can provide a context menu for any view but they are mostly used for items in a List view or Grid View , In which the user can perform direct action on each item.
* There are two ways to provide Context Menu-----

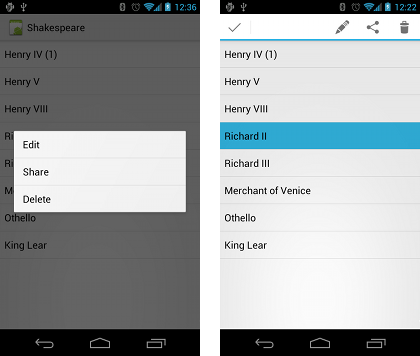
**1-Floating Context Menu:-**

A Menu is appeared as a floating List of MenuItems when user perform long press click on any view.

**2-Contextual Action Mode:-**

This mode is system implementation of action mode that displays a contextual action bar at the Top/Bottom of screen with action items.

Figure:-  Screenshots of a floating context menu (left) and the contextual action bar (right).



***Note-***

* The contextual action mode is available on Android3.1 and Higher version and it preferred technique for displaying contextual action item.
* If your apps supported version lower than 3.0 then your app will have context menu as Floating context Menu only.

**Steps to create Floating contextual Menu:-**

**Step-1:** Define main.xml in res/menu directory.

**Step-2:** Register the View(here listview) to which context menu should be associated for this we

have to call *registerForContextMenu(View v)* and pass it the view.

**Step-3:** @override *onCreateContextMenu()* of Activity class in user defined Activity

class because when the register view receive a long click event the system calls your

*onCreateContextMenu()* method.

In this method we have to define inflating code:

*public void onCreateContextMenu(ContextMenu menu,View v,ContextMenuInfo info)*

*{*

*MenuInflater inflater=getMenuInfalter();*

*inflater.inflate(R.menu.main,menu);*

*}*

**Parameters:-**

ContextMenu---------------------------> Object of implementing class of ContextMenu interface.

View -----------------------------------> Object of Long Pressed View(eg- ListView or GridView).

ContextMenu. ContextMenuInfo-> Object of ContextMenuInfo nested interface that provide

additional info about Long Pressed item.

**Step-4:** **Handling Click event:-**

@override

onContextItemSelected(MenuItem item) method of Activity class in user defined Activity class because when the user select a menu item of Context menu then the OS calls this method. So We can perform appropriate action by defining code in this method.

* We can identify the item by calling getItemId() method.
* getItemId() method of MenuItem interface which return unique ID of selected MenuItem.

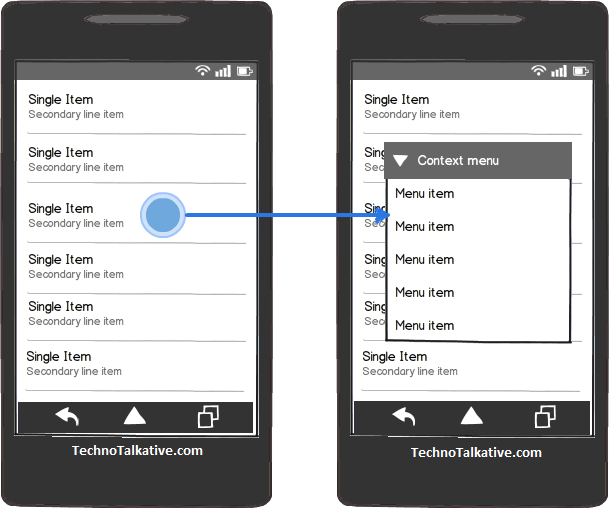
**@override**

public boolean onContextItemSelected(MenuItem item)

{

switch(item.getItemId())

{

case R.id.item1: - -

return true;

case R.id.item2:

-

-

return true;

case R.id.item3:

-

-

return true;

default:

return false;

}

}

3. **POP Up Menu :-**

It is class defined in android.widget package an object of this class displays a menu in a modal Pop Up window anchored to a view. The Pop Up will appear bellow to anchored view when clicked if there is space available OR above to anchor view, if there is no space bellow to anchor view.

***Note-*** Anchor view can be any view object Like—Button object, TextView object shown.

**Steps to create Pop Up Menu:-**

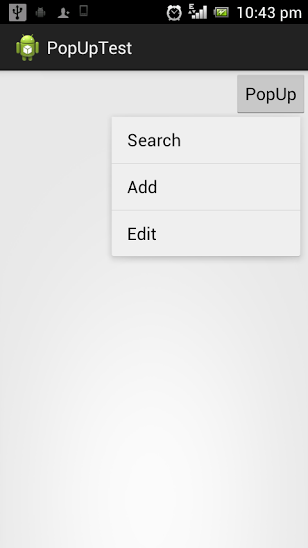
**Step-1:** Minimum API (must be) required API-11(SDK-11).

**Step-2:** Define main.xml in res/menu subdirectory.

**Step-3:**Define anchor view(here button) in activity\_main.xml which will anchored by PopUp Menu.

**Step-4:** Define event handler call back (here button) for anchor view.

**Step-5:** Create object of PopUp Menu class and inflate xml menu resource into PopUp menu Object. All these steps define in event handler (call back) method.

void showPopUp(view v)

*{*

*PopUpMenu popup=new PopUpMenu(this,v);*

*MenuInflater inflater=popup.getMenuInfalter();*

*inflater.inflate(R.menu.main,popup.getMenu());*

*popup.show();*

popup.setOnMenuItemClickListener(**this**);

*}*

**Step-6:** **Handling Click event:-**

For this need to implements onMenuItemClickListioner interface and provide implementation of onMenuItemClick() to write appropriate code for an action to be performed on Menu item cilck.

**public** **boolean** onMenuItemClick(MenuItem item)

{

}

Note:- We have to register onMenuItemClickListioner with pop by…..

popup.setOnMenuItemClickListener(**this**);