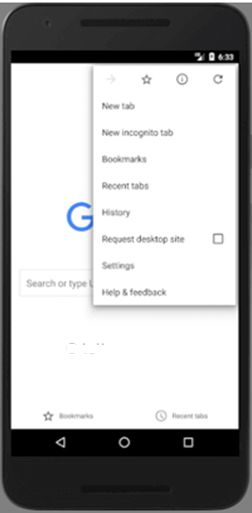
1. **Android Menus (Options, Context, Popup)**

In android, **Menu** is a part of user interface (UI) component which is used to handle some common functionality around the application. By using Menus in our applications, we can provide better and consistent user experience throughout the application.

We can use Menu APIs to represent user actions and other options in our android application activities.

Following is the pictorial representation of using menus in android application.



In android, we can define a Menu in separate XML file and use that file in our [activities](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) or [fragments](https://www.tutlane.com/tutorial/android/android-fragments-with-examples) based on our requirements.

**Define an Android Menu in XML File**

For all menu types, Android provides a standard XML format to define menu items. Instead of building a menu in our [activity's](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) code, we should define a menu and all its items in an XML menu resource and load menu resource as a Menu object in our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) or [fragment](https://www.tutlane.com/tutorial/android/android-fragments-with-examples).

In android, to define menu, we need to create a new folder **menu** inside of our project resource directory (**res/menu/**) and add a new XML file to build the menu with the following elements.

| **Element** | **Description** |
| --- | --- |
| <menu> | It’s a root element to define a Menu in XML file and it will hold one or more and elements. |
| <item> | It is used to create a menu item and it represent a single item in menu. This element may contain a nested <menu> element in order to create a submenu. |
| <group> | It’s an optional and invisible for <item> elements. It is used to categorize the menu items so they share properties such as active state and visibility. |

Following is the example of defining a menu in XML file (**menu\_example.xml**).

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <menu xmlns:android="http://schemas.android.com/apk/res/android">     <item android:id="@+id/mail"         android:icon="@drawable/ic\_mail"         android:title="@string/mail" />     <item android:id="@+id/upload"         android:icon="@drawable/ic\_upload"         android:title="@string/upload"         android:showAsAction="ifRoom" />     <item android:id="@+id/share"         android:icon="@drawable/ic\_share"         android:title="@string/share" /> </menu> |

The **<item>** element in **menu** supports different type of attributes to define item’s behaviour and appearance. Following are the some of commonly used **<item>** attributes in android applications.

| **Attribute** | **Description** |
| --- | --- |
| android:id | It is used to uniquely identify element in application. |
| android:icon | It is used to set the item's icon from drawable folder. |
| android:title | It is used to set the item's title |
| android:showAsAction | It is used to specify how the item should appear as an action item in the app bar. |

In case if we want to add **submenu** in **menu** item, then we need to add a **<menu>** element as the child of an **<item>**. Following is the example of defining a submenu in menu item.

|  |
| --- |
| <?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> |

**Load Android Menu from an Activity**

Once we are done with creation of menu, we need to load the menu resource from our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) using **MenuInflater.inflate()** like as shown below.

|  |
| --- |
| @Override public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo) {     super.onCreateContextMenu(menu, v, menuInfo);     MenuInflater inflater = getMenuInflater();     inflater.inflate(R.menu.menu\_example, menu); } |

If we observe above code we are calling our menu using MenuInflater.inflate() method in the form of **R.menu.menu\_file\_name**. Here our xml file name is **menu\_example.xml** so we used file name **menu\_example**.

**Handle Android Menu Click Events**

In android, we can handle a menu item click events using **ItemSelected()** event based on the menu type. Following is the example of handling a context menu item click event using **onContextItemSelected()**.

|  |
| --- |
| @Override public boolean onContextItemSelected(MenuItem item) {     switch (item.getItemId()) {         case R.id.mail:             // do something             return true;         case R.id.share:             // do something             return true;         default:             return super.onContextItemSelected(item);     } } |

If we observe above code, the **getItemId()** method will get the id of selected menu item based on that we can perform our actions.

**Android Different Types of Menus**

In android, we have a three fundamental type of Menus available to define a set of options and actions in our android applications.

Following are the commonly used Menus in android applications.

* [Options Menu](https://www.tutlane.com/tutorial/android/android-options-menu-with-examples)
* [Context Menu](https://www.tutlane.com/tutorial/android/android-context-menu-with-examples)
* [Popup Menu](https://www.tutlane.com/tutorial/android/android-popup-menu-with-examples)

**Android Options Menu**

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.

**Android Context Menu**

In android, **Context Menu** is a floating menu that appears when the user performs a long click on an element and it is useful to implement an actions that effect the selected content or context frame.

**Android Popup Menu**

In android, **Popup Menu** displays a list of items in a vertical list that’s anchored to the view that invoked the menu and it’s useful for providing an overflow of actions that related to specific content.

**Questions:**

1. How to create “**menu**” in Android, what are the different items of “**menu**”?
2. What are the commonly used **<item>** attributes in android applications for “**menu**”?
3. How to call the method “MenuInflater.inflate()” for “**menu**”? Write the code.
4. What is the method to handle a menu item click events? Write the code.
5. Why to use onContextItemSelected()?