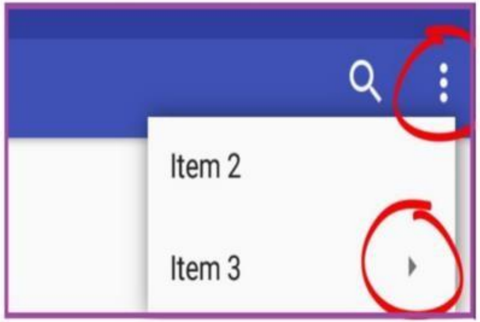
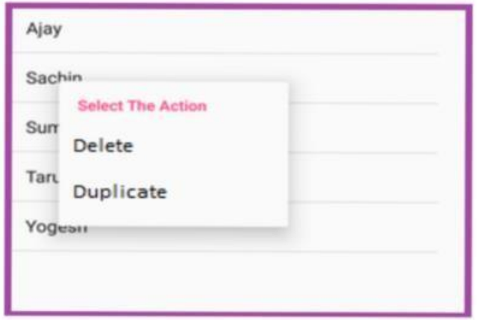
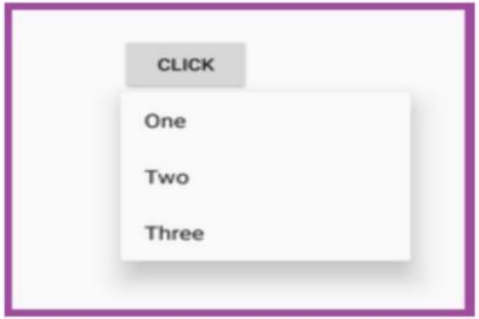
Options Menu



Contextual Menu



Popup Menu



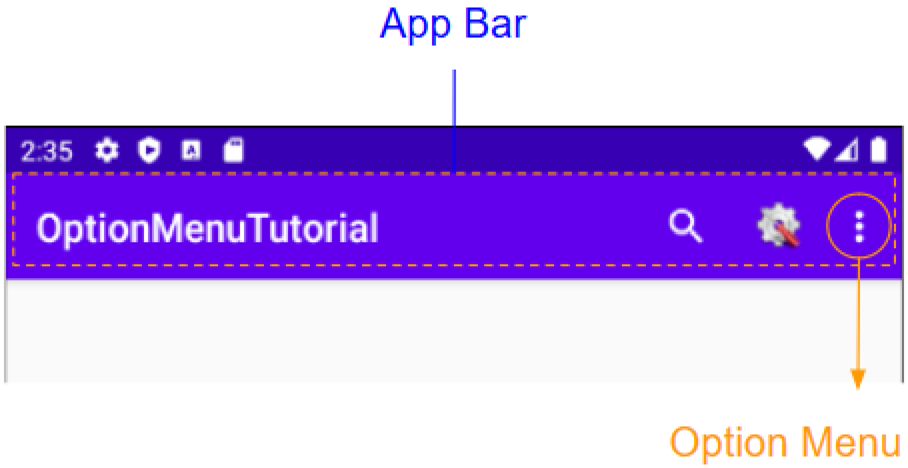
Menus In Android

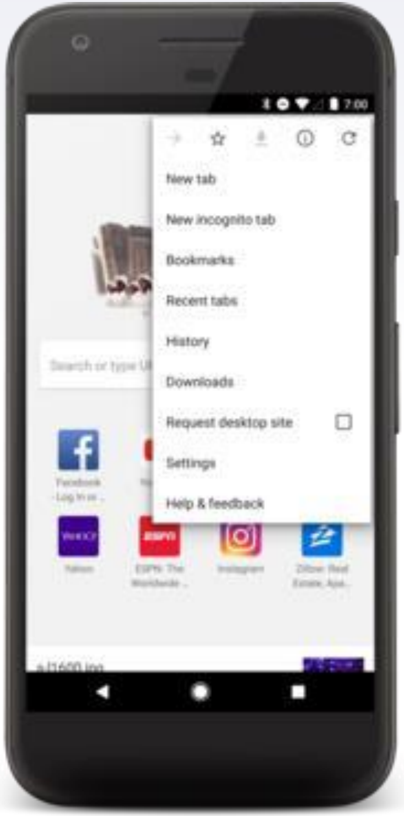
- In android, Menu is an important part of the UI component which is used to provide some common functionality around the application.
- With the help of menu, users can experience a smooth and consistent experience throughout the application.
- In order to use menu, we should define it in a separate XML file and use that file in our application based on our requirements.
- Also, we can use menu APIs to represent user actions and other options in our android application activities.
- There are 3 types of menus in Android:
    - Option Menu
    - Context Menu
    - Pop-up Menu

1- Android Option Menu

In Android, an Option Menu is a set of primary options of an application which users can select one of the options to perform an action. The Option Menu appears on the right side of the App Bar.

The options menu is the primary collection of menu items for an activity. It's where you should place actions that have a overall impact on the app, such as Search, Compose Email and Settings.

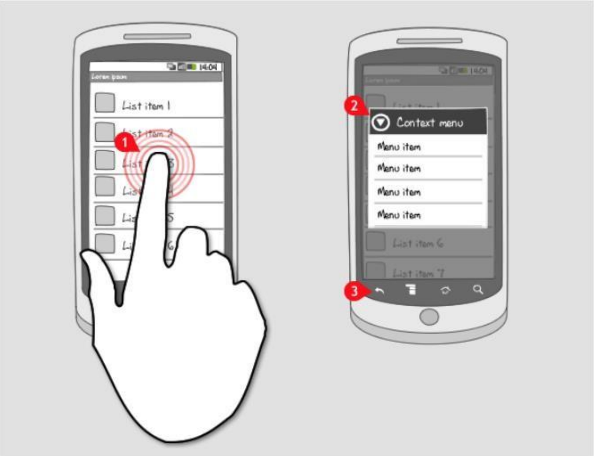


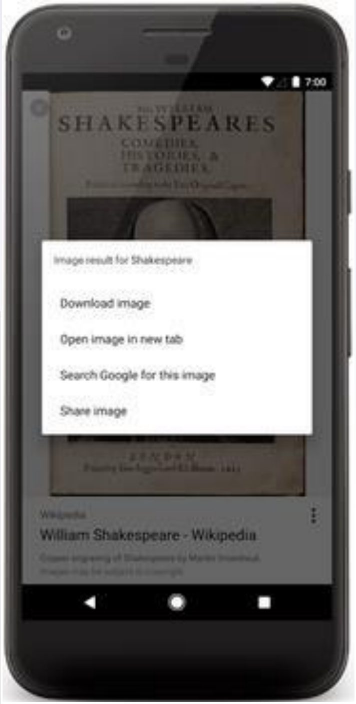


2 – Android Context Menu

A context menu is a floating menu that appears when the user performs a long-click on an element.

A contextual menu offers actions that affect a specific item or context frame in the UI.




7:00
SHAKESPEARES
COMEDIES, HISTORIES, & TRAGEDIES
Image result for Shakespeare
Download image
Open image in new tab
Search Google for this image
Share image
Wikipedia
William Shakespeare - Wikipedia

There are two ways to provide contextual actions:

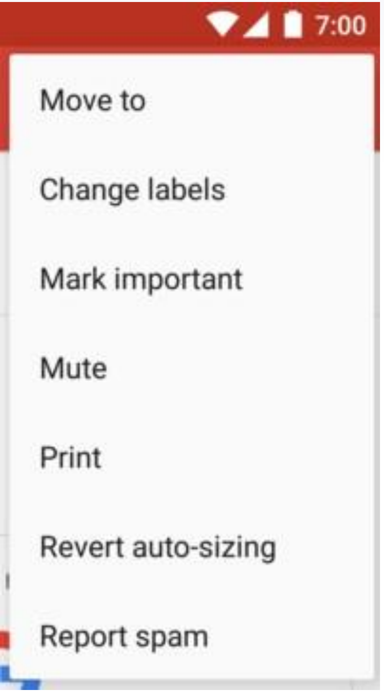
In a floating context menu.

A menu appears as a floating list of menu items (similar to a dialog) when the user performs a long-click (press and hold) on a view that declares support for a context menu. Users can perform a contextual action on one item at a time.

In the contextual action mode

This mode is a system implementation of ActionMode that displays a contextual action bar at the top of the screen with action items that affect the selected item(s). When this mode is active, users can perform an action on multiple items at once (if your app allows it).

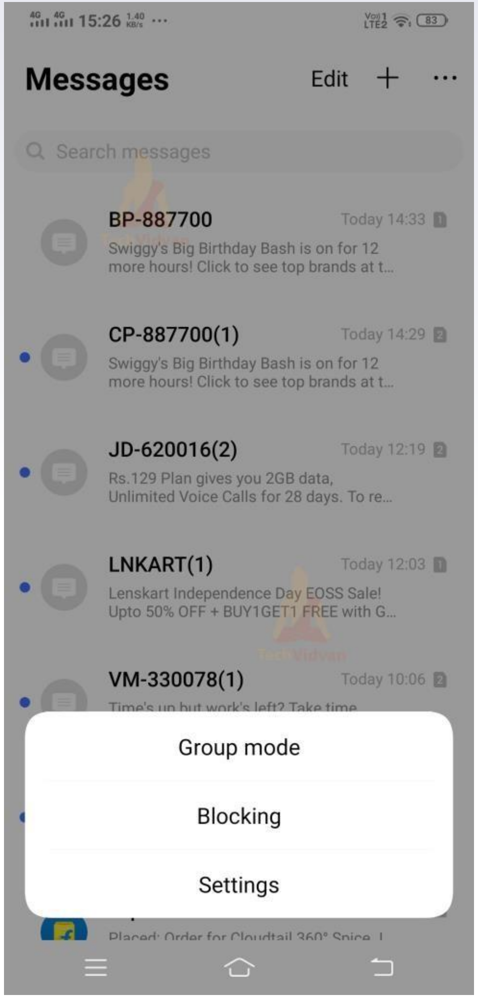
3 – Android PopUp Menu



A popup menu displays a list of items in a vertical list that is anchored(sticked) to the view that invoked the menu.

It's good for providing an overflow of actions that relate to specific content or to provide options for a second part of a command.

Note: Actions in a popup menu should not directly affect the corresponding content—that's what contextual actions are for. Rather, the popup menu is for extended actions that relate to regions of content in your activity.



BP-887700 Today 14:33
Swiggy's Big Birthday Bash is on for 12 more hours! Click to see top brands at t...

CP-887700(1) Today 14:29
Swiggy's Big Birthday Bash is on for 12 more hours! Click to see top brands at t...

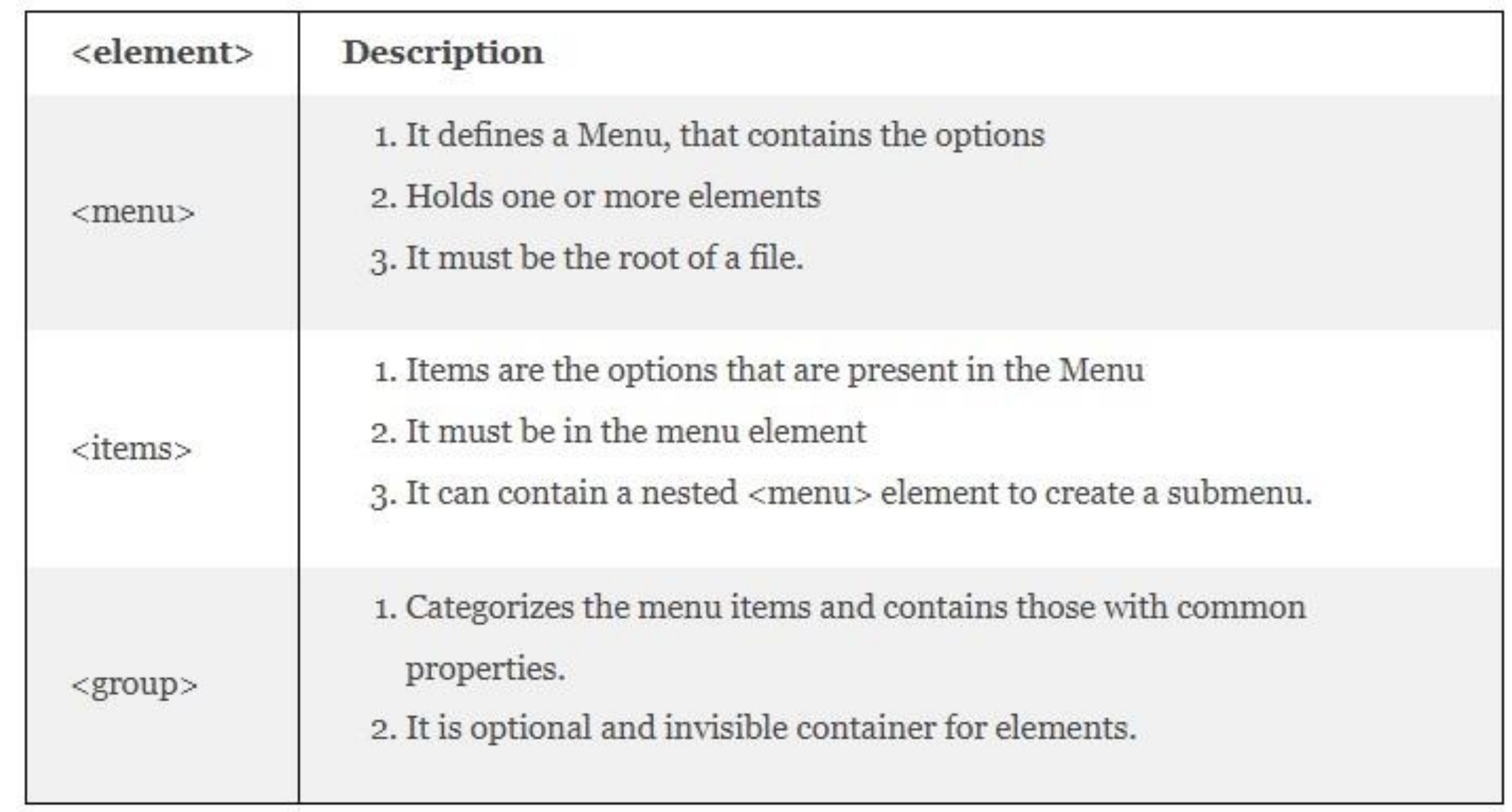
JD-620016(2) Today 12:19
Rs.129 Plan gives you 2GB data, Unlimited Voice Calls for 28 days. To re...

LNKART(1) Today 12:03
Lenskart Independence Day EOSS Sale! Upto 50% OFF + BUY1GET1 FREE with G...

VM-330078(1) Today 10:06
Time's up but work's left? Take time

Placed: Order for Cloudtail 360° Spice, I

Following are important elements of a menu:



<element>	Description
<menu>	1. It defines a Menu, that contains the options 2. Holds one or more elements 3. It must be the root of a file.
<items>	1. Items are the options that are present in the Menu 2. It must be in the menu element 3. It can contain a nested <menu> element to create a submenu.
<group>	1. Categorizes the menu items and contains those with common properties. 2. It is optional and invisible container for elements.

Like any other UI component, even Android menus can be customized with the help of attributes like-

android: id - It uniquely identifies the item of the menu.

android: icon - It sets an icon to represent the item.

android: title - It sets the title of the item.

android:showAsAction - It specifies when and how this item should appear as an action item in the app bar.

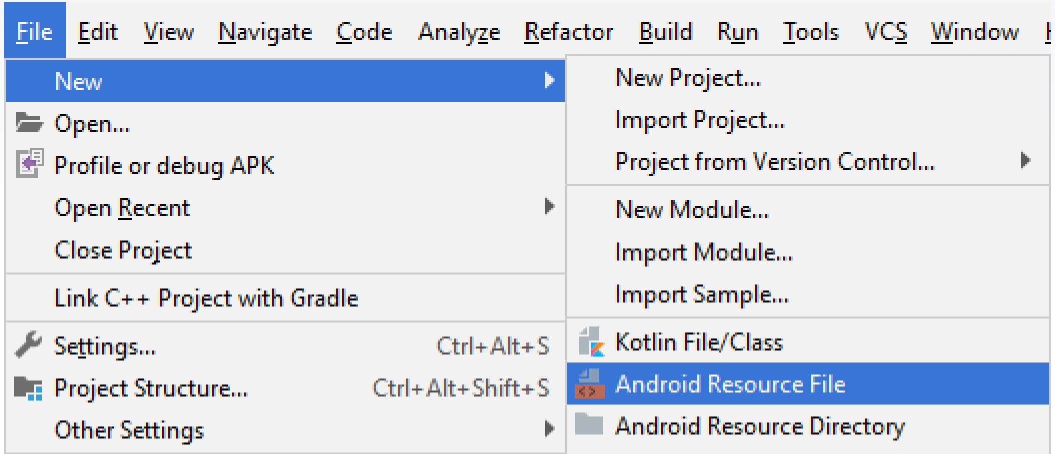
The value can be *ifRoom, never, withText, always, collapseActionView*.

Implementation of Android Option Menu

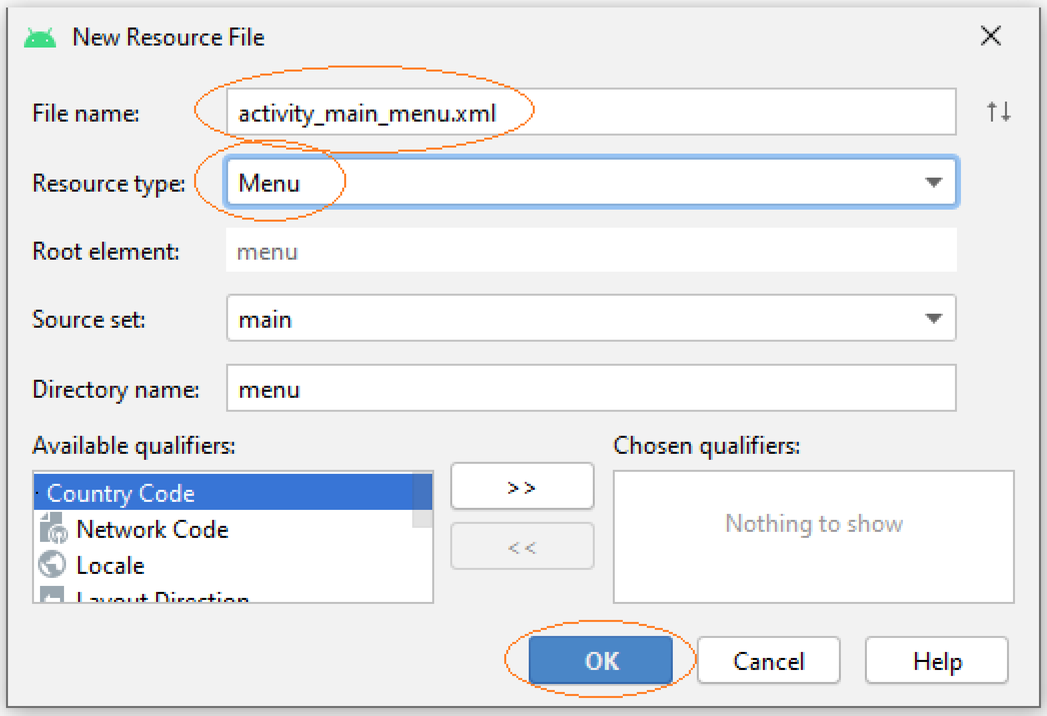
First, create a new project and name it.

On Android Studio, select:

File > New > Android Resource File



File name :activity_main_menu.xml
Resource Type: Menu



New Resource File

File name: activity_main_menu.xml

Resource type: Menu

Root element: menu

Source set: main

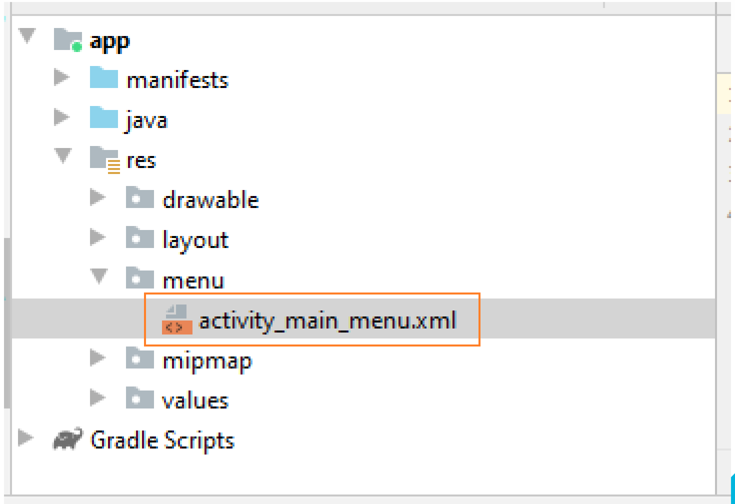
Directory name: menu

Available qualifiers:

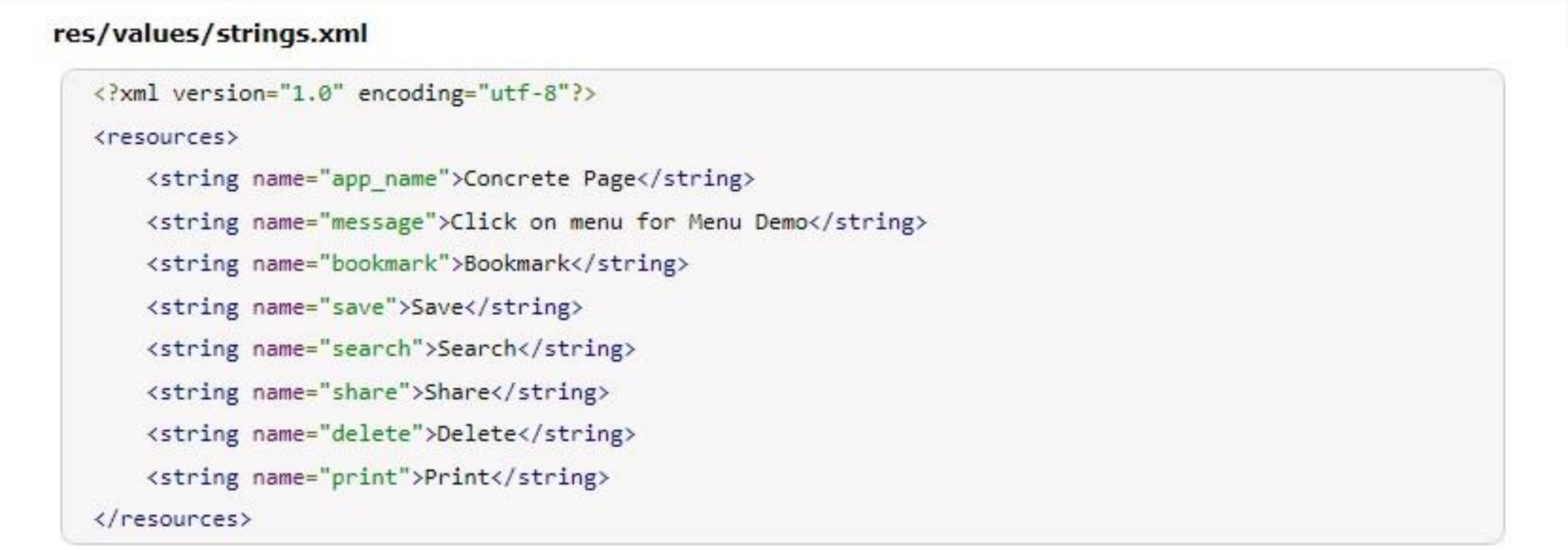
- Country Code
- Network Code
- Locale
- Layout Direction

Chosen qualifiers:

Nothing to show

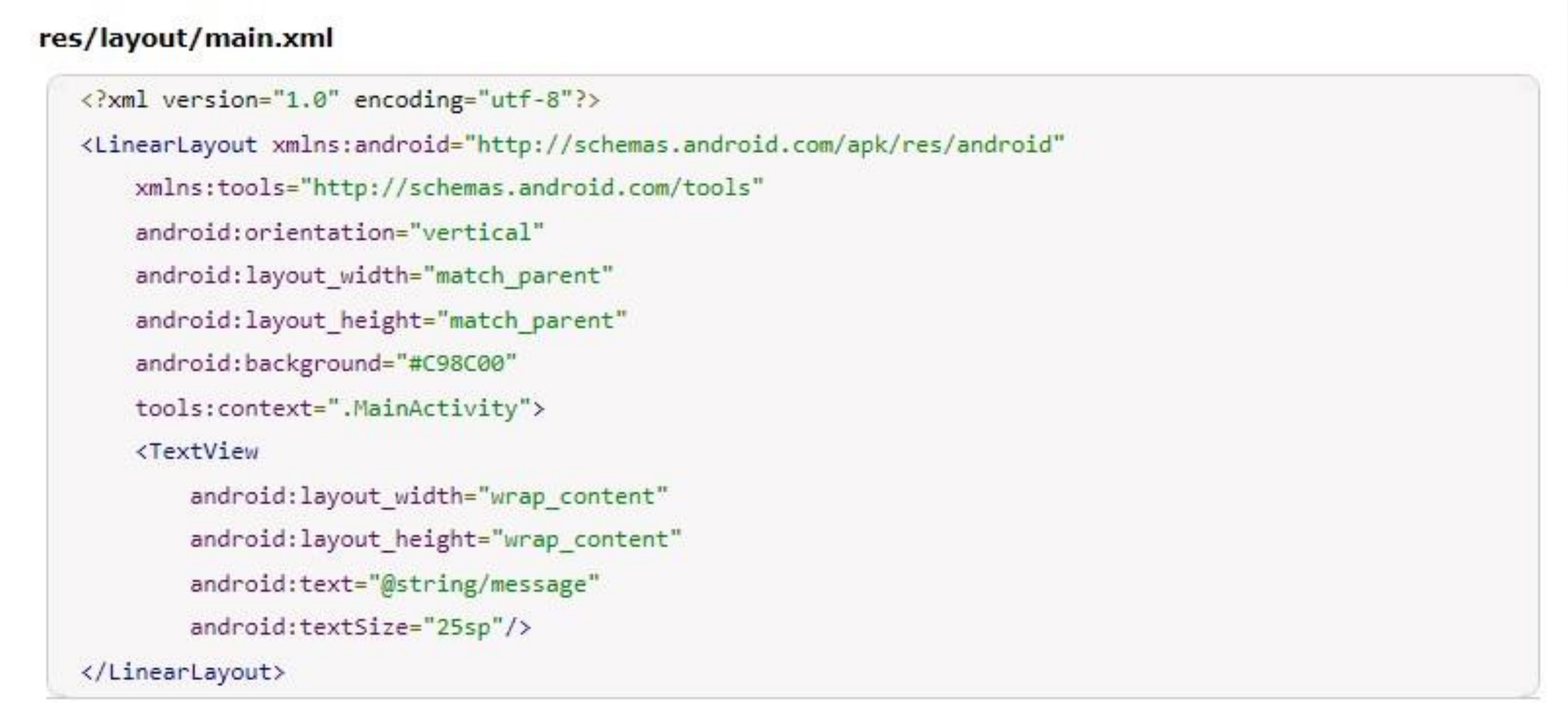


```
app
  manifests
  java
  res
    drawable
    layout
    menu
      activity_main_menu.xml
    mipmap
    values
Gradle Scripts
```



res/values/strings.xml

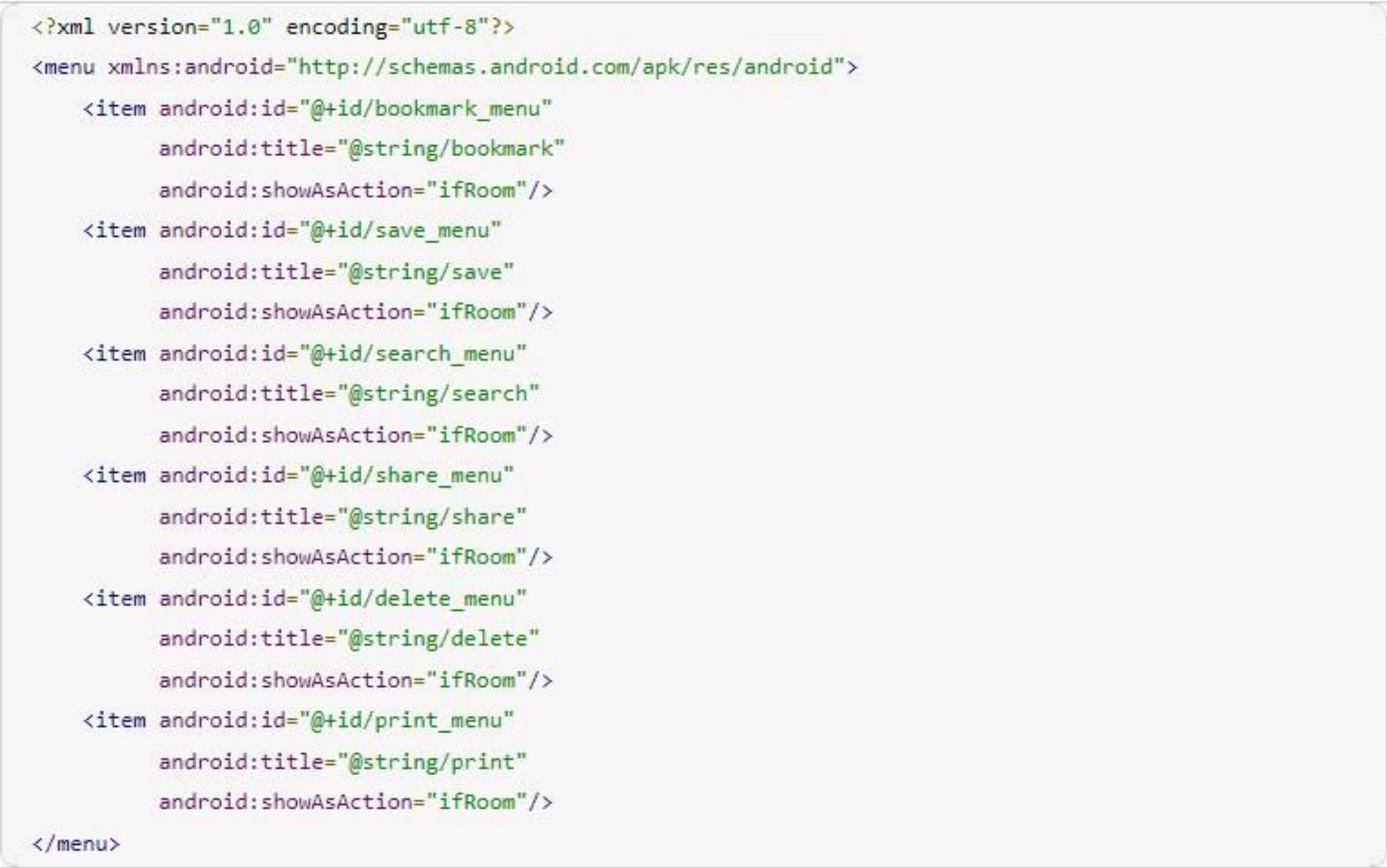
```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Concrete Page</string>
    <string name="message">Click on menu for Menu Demo</string>
    <string name="bookmark">Bookmark</string>
    <string name="save">Save</string>
    <string name="search">Search</string>
    <string name="share">Share</string>
    <string name="delete">Delete</string>
    <string name="print">Print</string>
</resources>
```



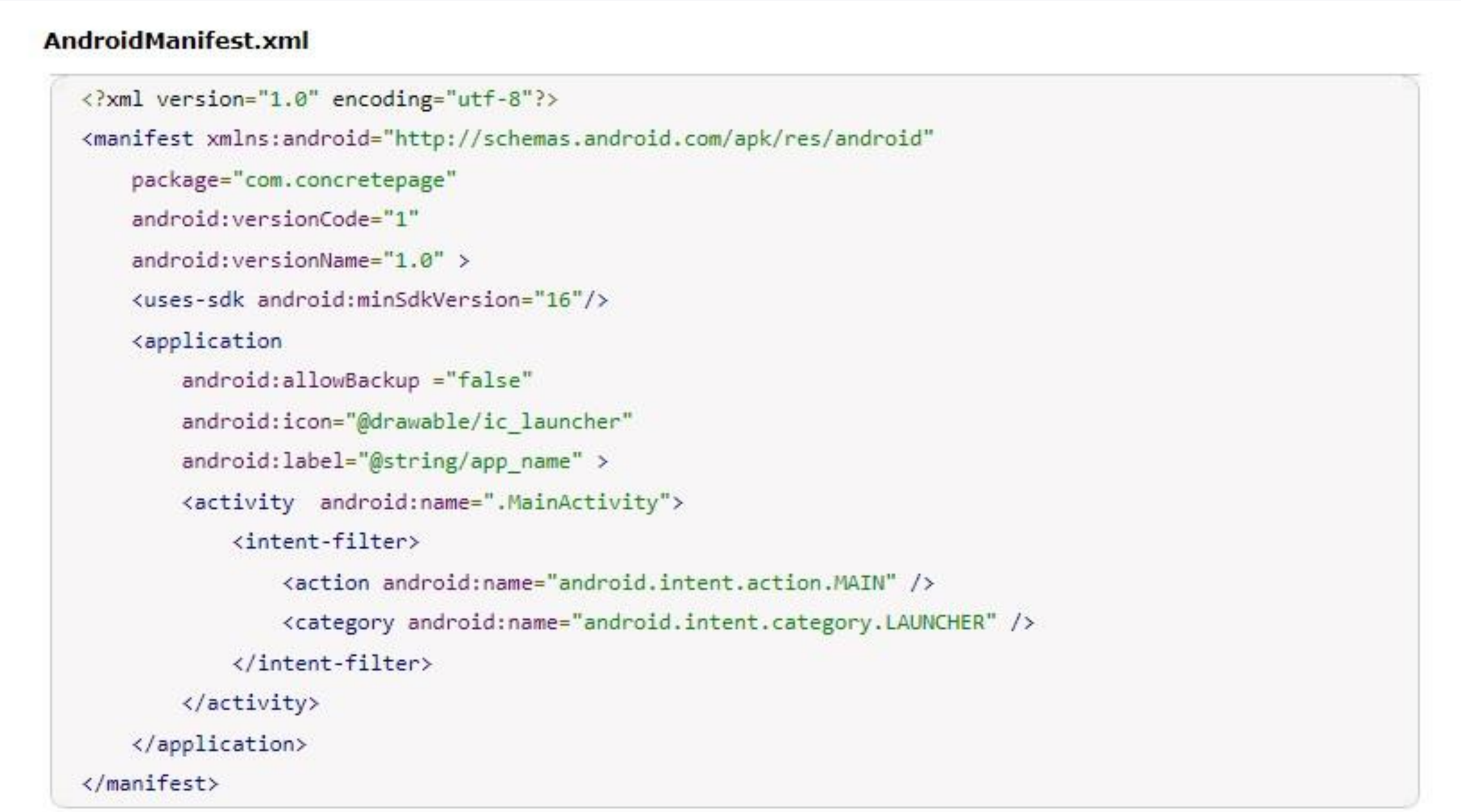
res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#C98C00"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/message"
        android:textSize="25sp"/>
</LinearLayout>
```

res/menu/main.xml



```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/bookmark_menu"
          android:title="@string/bookmark"
          android:showAsAction="ifRoom"/>
    <item android:id="@+id/save_menu"
          android:title="@string/save"
          android:showAsAction="ifRoom"/>
    <item android:id="@+id/search_menu"
          android:title="@string/search"
          android:showAsAction="ifRoom"/>
    <item android:id="@+id/share_menu"
          android:title="@string/share"
          android:showAsAction="ifRoom"/>
    <item android:id="@+id/delete_menu"
          android:title="@string/delete"
          android:showAsAction="ifRoom"/>
    <item android:id="@+id/print_menu"
          android:title="@string/print"
          android:showAsAction="ifRoom"/>
</menu>
```

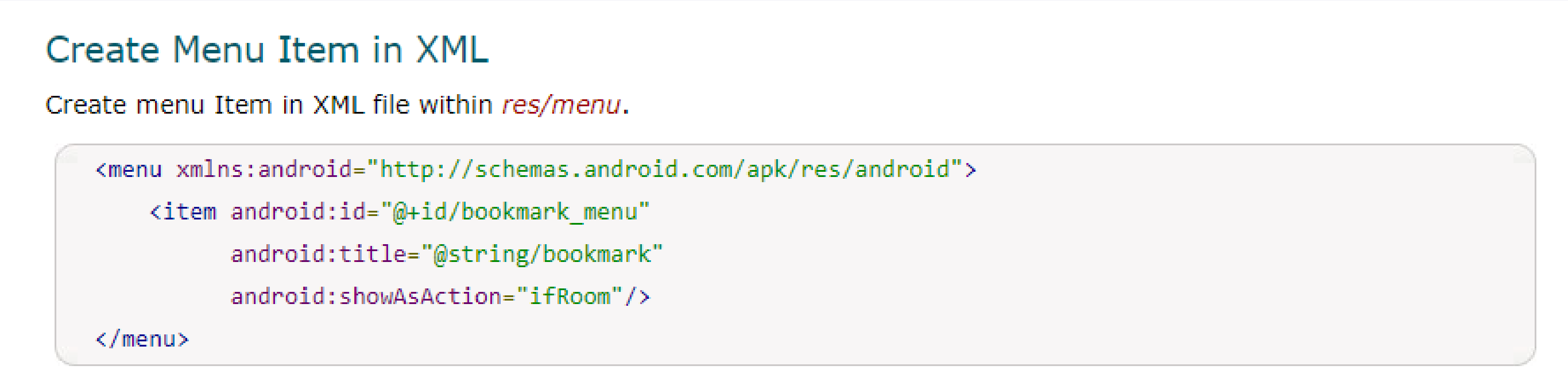


AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.concretepage"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="16"/>
    <application
        android:allowBackup ="false"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity  android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

To create menu we have to override **onCreateOptionsMenu**, in which we use **getMenuInflater().inflate** that inflates a menu hierarchy from XML resource.

To handle click event, override **onOptionsItemSelected** in Activity class.



Create Menu Item in XML

Create menu Item in XML file within *res/menu*.

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/bookmark_menu"
          android:title="@string/bookmark"
          android:showAsAction="ifRoom"/>
</menu>
```

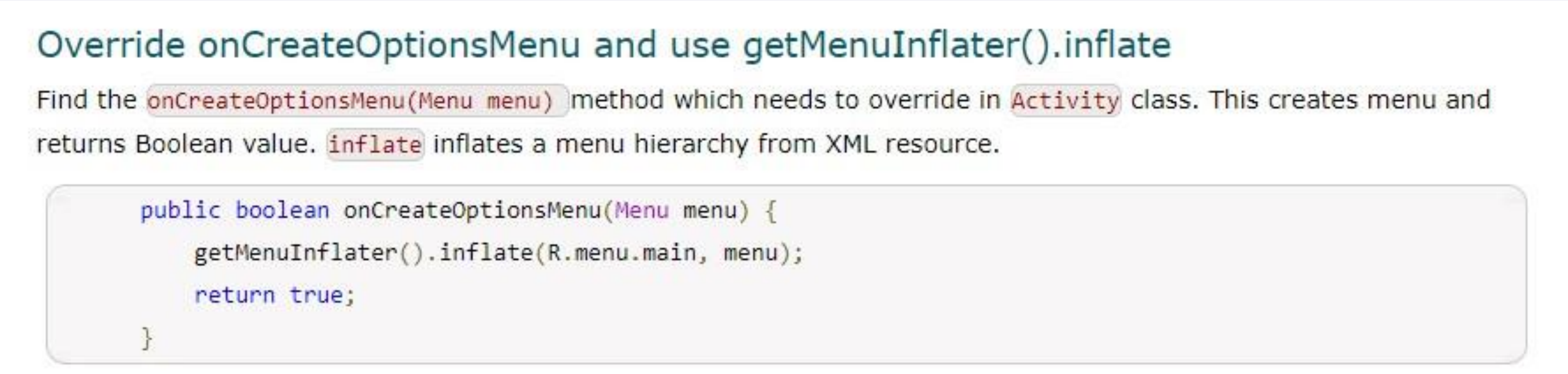
<menu>: Root node that contains one or more item.

<item>: It represents menu items.

android:id: Unique ID for the item.

android:title : Title for the item.

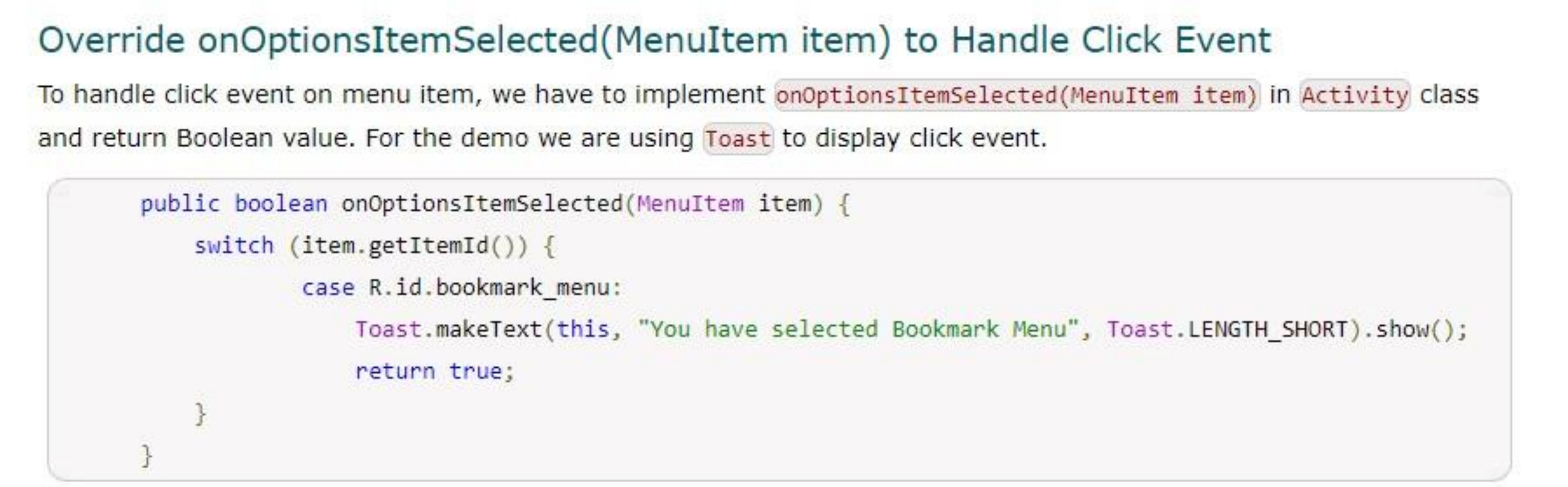
android:showAsAction : It specifies how an item to be shown as an action item. The value can be *ifRoom, never, withText, always, collapseActionView*.



Override onCreateOptionsMenu and use getMenuInflater().inflate

Find the `onCreateOptionsMenu(Menu menu)` method which needs to override in `Activity` class. This creates menu and returns Boolean value. `inflate` inflates a menu hierarchy from XML resource.

```
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
```



Override onOptionsItemSelected(MenuItem item) to Handle Click Event

To handle click event on menu item, we have to implement `onOptionsItemSelected(MenuItem item)` in `Activity` class and return Boolean value. For the demo we are using `Toast` to display click event.

```
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
            case R.id.bookmark_menu:
                Toast.makeText(this, "You have selected Bookmark Menu", Toast.LENGTH_SHORT).show();
                return true;
    }
}
```

After that write the following code in the MainActivity.java file:

```
package com.example.optionmenuapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.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 super.onCreateOptionsMenu(menu);
    }
```

```
    @Override
    public boolean onOptionsItemSelected(@NonNull 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);
        }
    }
}
```