

# التفاعل مع المستخدم 2

الوحدة التاسعة

# ماهي قائمة الاختيار Menu؟

- في نظام اندرويد تعد القائمة جزءًا من مكون واجهة المستخدم الذي يستخدم للتعامل مع بعض الوظائف الشائعة حول التطبيق.
- باستخدام القوائم في تطبيقاتنا يمكننا توفير تجربة مستخدم أفضل ومتسقة في جميع أنحاء التطبيق.

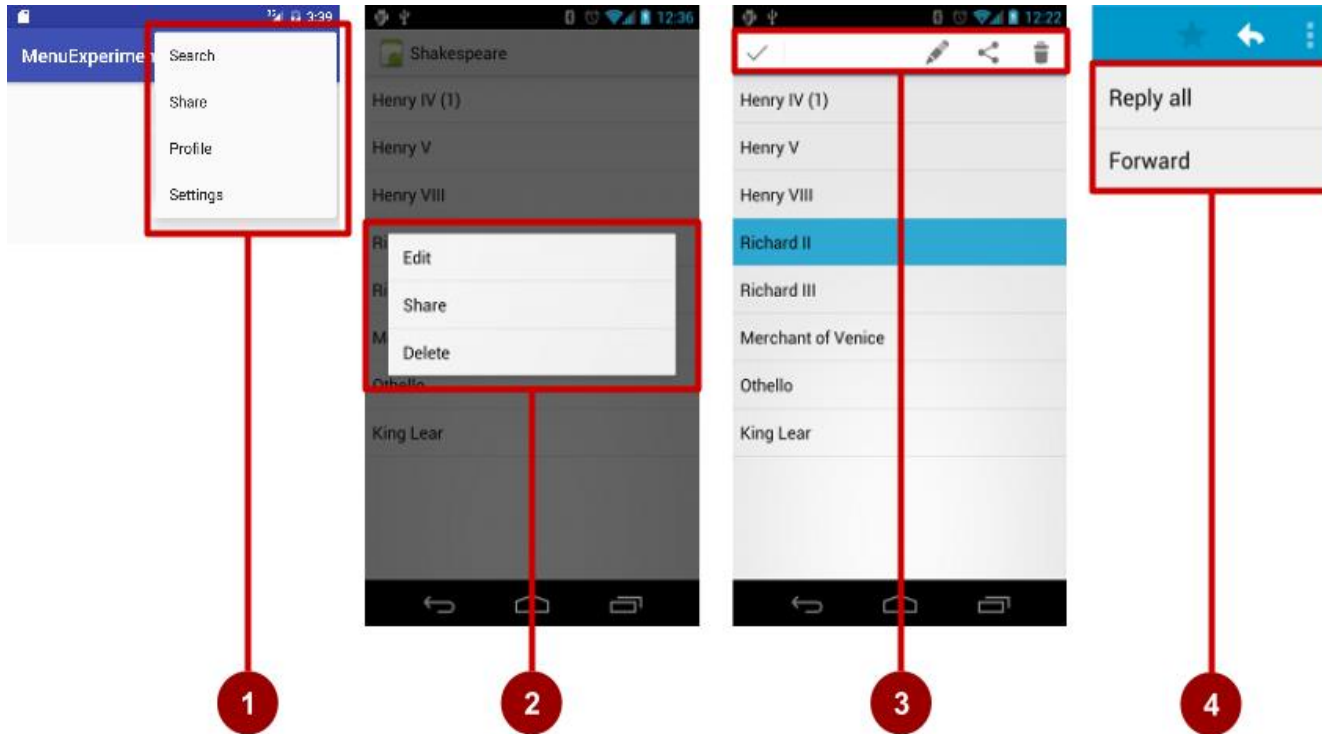
# أنواع القوائم

- شريط التطبيق مع قائمة الخيارات (optional menu)

- قائمة السياق (context menu)

- شريط الإجراءات السياقية

- قائمة منبثقة (popup menu)



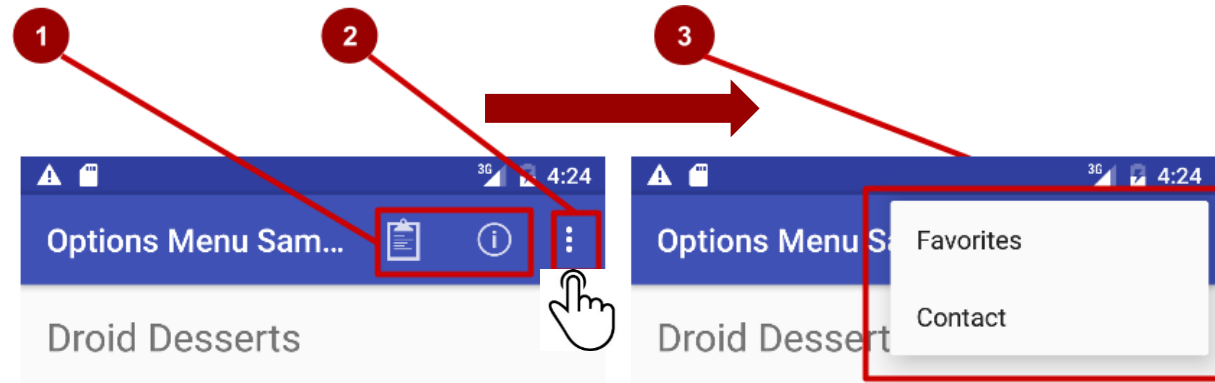
# كيف نعرف menu في ملف XML

- في نظام اندرويد لتعريف قائمة نحتاج إلى إنشاء مجلد جديد داخل دليل موارد المشروع باسم menu ( res / menu / ) وإضافة ملف XML جديد لإنشاء القائمة بالعناصر التالية.

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 represents a single item on the 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.

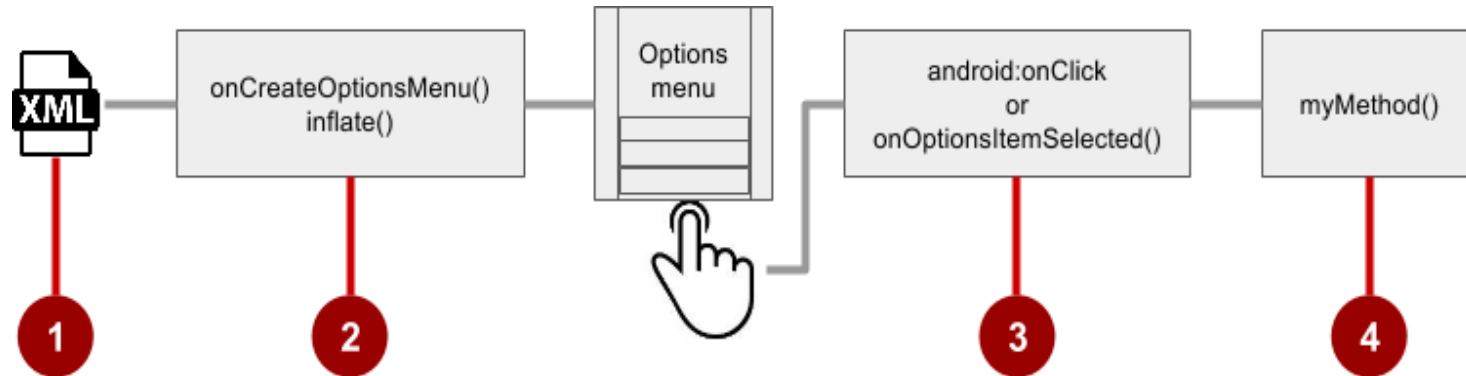
# قائمة الخيارات options menu

- في نظام اندرويد تعد قائمة الخيارات مجموعة أساسية من عناصر القائمة لنشاط ما ، ومن المفيد تنفيذ الإجراءات التي لها تأثير عام على التطبيق ، مثل الإعدادات والبحث.



# خطوات لإنشاء قائمة خيارات

1. XML menu resource (menu\_main.xml)
2. onCreateOptionsMenu() to inflate the menu
3. onClick attribute or onOptionsItemSelected()
4. Method to handle item click



# انشاء مصدر القوائم

1. Create menu resource directory
2. Create XML menu resource (menu\_main.xml)
3. Add entry for each menu item (**Settings** and **Favorites**):

```
<item android:id="@+id/option_settings"  
      android:title="Settings" />
```

```
<item android:id="@+id/option_favorites"  
      android:title="Favorites" />
```

# مثال

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/new_game"
          android:icon="@drawable/ic_new_game"
          android:title="@string/new_game"
          android:showAsAction="ifRoom" />
    <item android:id="@+id/help"
          android:icon="@drawable/ic_help"
          android:title="@string/help" />
</menu>
```

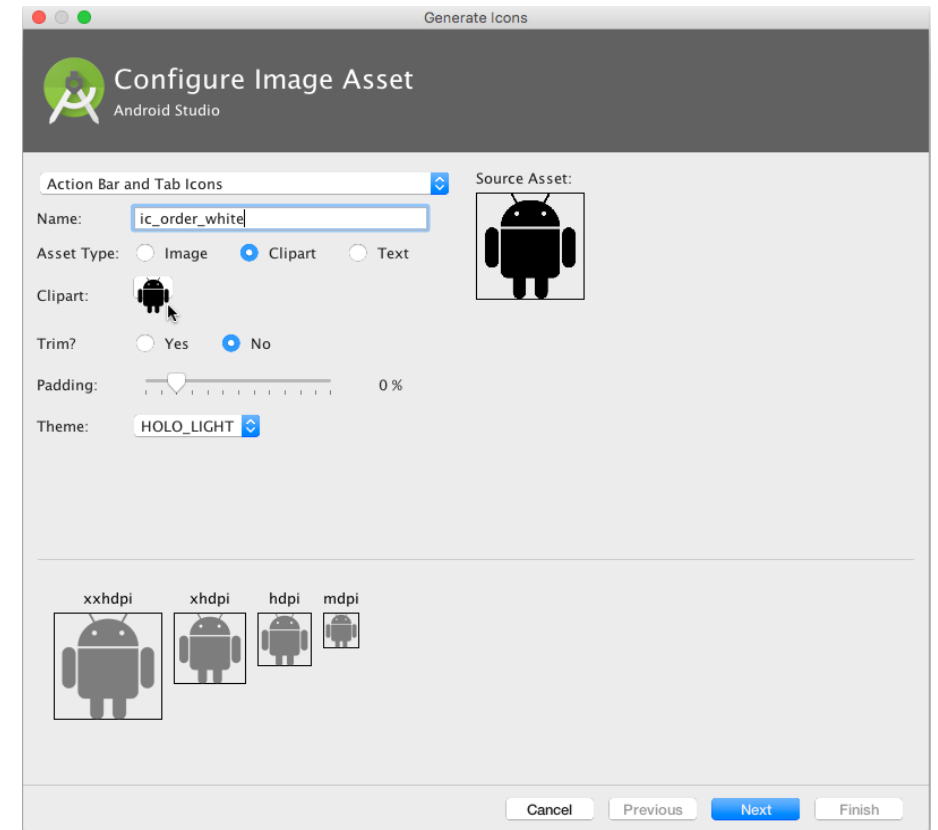


# عملية الربط مع القائمة في مجلد الموارد

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.game_menu, menu);
    return true;
}
```

# إضافة ايقونات لعناصر اختيارات القائمة

1. Right-click **drawable**
2. Choose **New > Image Asset**
3. Choose **Action Bar and Tab Items**
4. Edit the icon name
5. Click clipart image, and click icon
6. Click **Next**, then **Finish**



# التعامل مع كود الخاص بالعناصر items

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {
        case R.id.new_game:
            newGame();
            return true;
        case R.id.help:
            showHelp();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

# قائمة السياق context menu

في نظام اندرويد تشبه قائمة السياق قائمة عائمة وتظهر عندما يقوم المستخدم بضغط طويلة أو ينقر على عنصر ، ومن المفيد تنفيذ الإجراءات التي تؤثر على المحتوى المحدد أو إطار السياق.

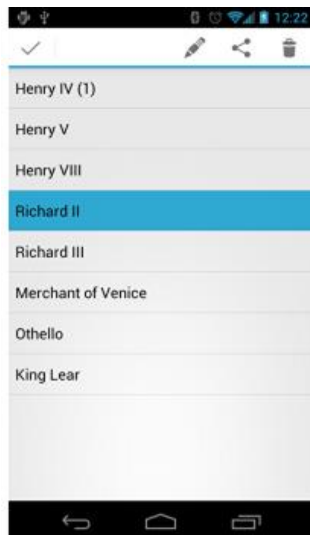
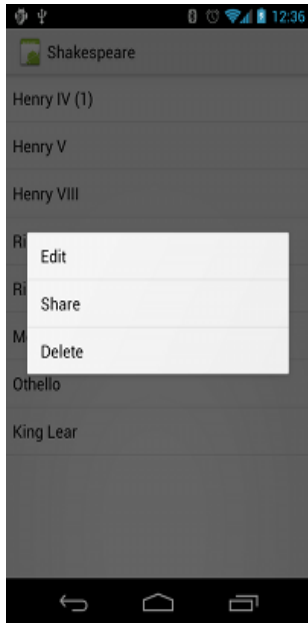
تشبه قائمة سياق اندرويد القائمة التي يتم عرضها عند النقر بزر الماوس الأيمن في نظام

التشغيل Windows أو Linux

# انواعه

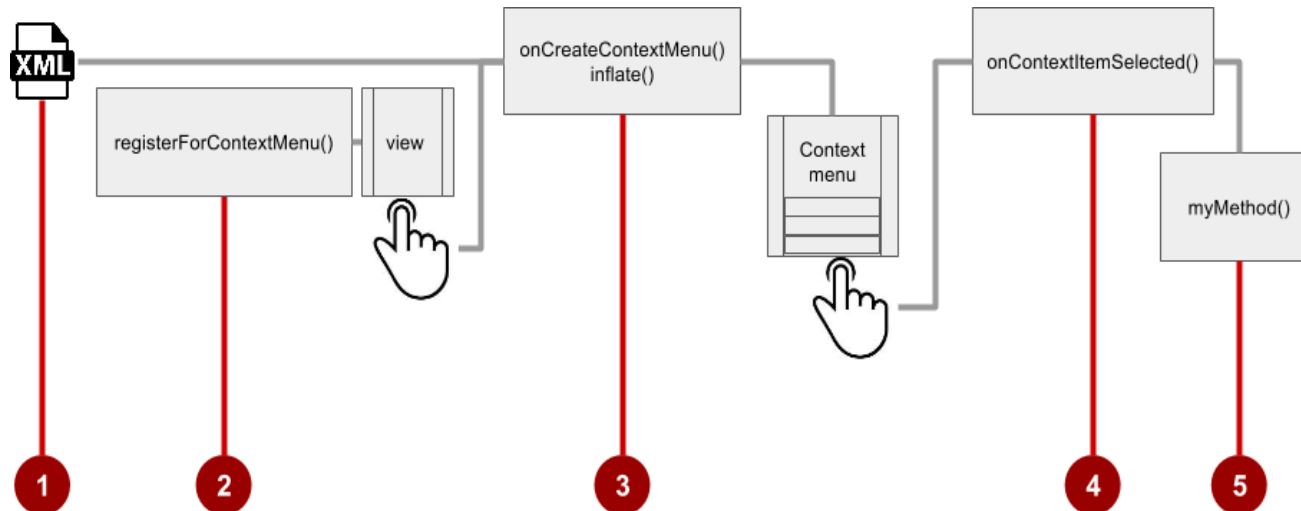
- قائمة السياق العائمة - الضغط لفترة طويلة على زر يمكن للمستخدم تعديل العرض أو استخدامه بطريقة ما يقوم المستخدم بإجراء على عرض واحد في كل مرة

- وضع الإجراء السياقي - شريط إجراءات مؤقت بدلاً من شريط التطبيق أو أسفله تؤثر عناصر الإجراء على عنصر (عناصر) العرض المحدد يمكن للمستخدم تنفيذ إجراء على عناصر عرض متعددة في وقت واحد



# خطوات انشاء القائمة

1. Create XML menu resource file and assign appearance and position attributes
2. Register View using `registerForContextMenu()`
3. Implement `onCreateContextMenu()` in Activity to inflate menu
4. Implement `onOptionsItemSelected()` to handle menu item clicks
5. Create method to perform action for each context menu item



## انشاء مصدر القائمة

- Create XML menu resource (menu\_context.xml)

```
<item
    android:id="@+id/context_edit"
    android:title="Edit"
    android:orderInCategory="10"/>

<item
    android:id="@+id/context_share"
    android:title="Share"
    android:orderInCategory="20"/>
```

## مثال

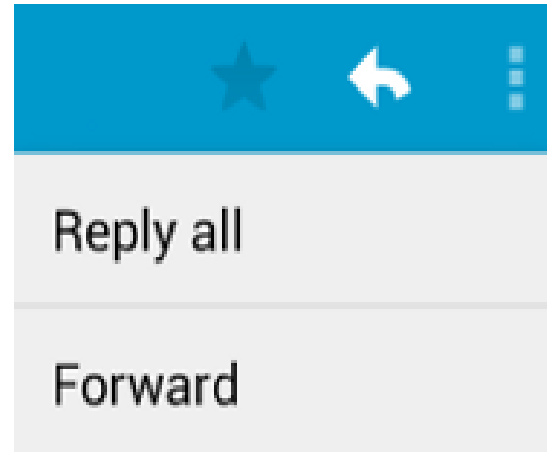
```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                               ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.context_menu, menu);
}
```

```
@Override
public boolean onContextItemSelected(MenuItem item) {
    AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();
    switch (item.getItemId()) {
        case R.id.edit:
            editNote(info.id);
            return true;
        case R.id.delete:
            deleteNote(info.id);
            return true;
        default:
            return super.onContextItemSelected(item);
    }
}
```



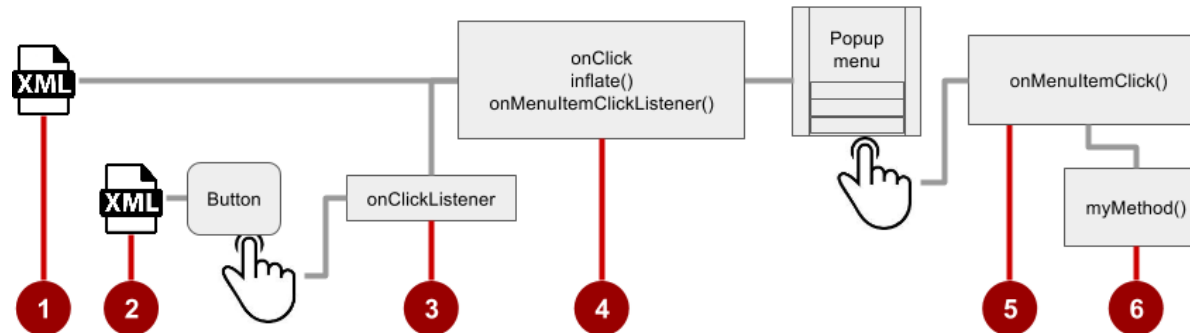
# القائمة المنبثقة Popup menu

في نظام اندرويد تعرض القائمة المنبثقة قائمة بالعناصر في نافذة منبثقة مشروطة مثبتة في العرض. توفر قائمة اندرويد المنبثقة قائمة نمط تجاوز للإجراءات المتعلقة بمحتوى معين



# خطوات انشاءه

1. Create XML menu resource file and assign appearance and position attributes
2. Add ImageButton for the popup menu icon in the XML activity layout file
3. Assign onClickListener to ImageButton
4. Override onClick() to inflate the popup and register it with onOptionsItemSelected()
5. Implement onOptionsItemSelected()
6. Create a method to perform an action for each popup menu item



## إضافة زر صورة

```
<ImageButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/button_popup"  
  
    android:src="@drawable/ic_action_popup"/>
```

# مثال

```
public void showPopup(View v) {  
    PopupMenu popup = new PopupMenu(this, v);  
    MenuInflater inflater = popup.getMenuInflater();  
    inflater.inflate(R.menu.actions, popup.getMenu());  
    popup.show();  
}
```

```
public void showMenu(View v) {  
    PopupMenu popup = new PopupMenu(this, v);  
  
    // This activity implements OnMenuItemClickListener  
    popup.setOnMenuItemClickListener(this);  
    popup.inflate(R.menu.actions);  
    popup.show();  
}
```

@Override

```
public boolean onMenuItemClick(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.archive:  
            archive(item);  
            return true;  
        case R.id.delete:  
            delete(item);  
            return true;  
        default:  
            return false;  
    }  
}
```

# ماهو مربع الحوار Dialog؟

في اندرويد يعد مربع الحوار نافذة صغيرة تطلب من المستخدم برسائل لتحديد تفاصيل إضافية أو القيام بإدخال إجراءات معينة على التطبيق من خلال مربع الحوار .

بشكل عام ، يتم استخدام مربعات الحوار مع حدث شكلي وهذه مفيدة لحث المستخدمين على تنفيذ إجراء معين للمضي قدمًا في التطبيق.

# مربع التنبيه

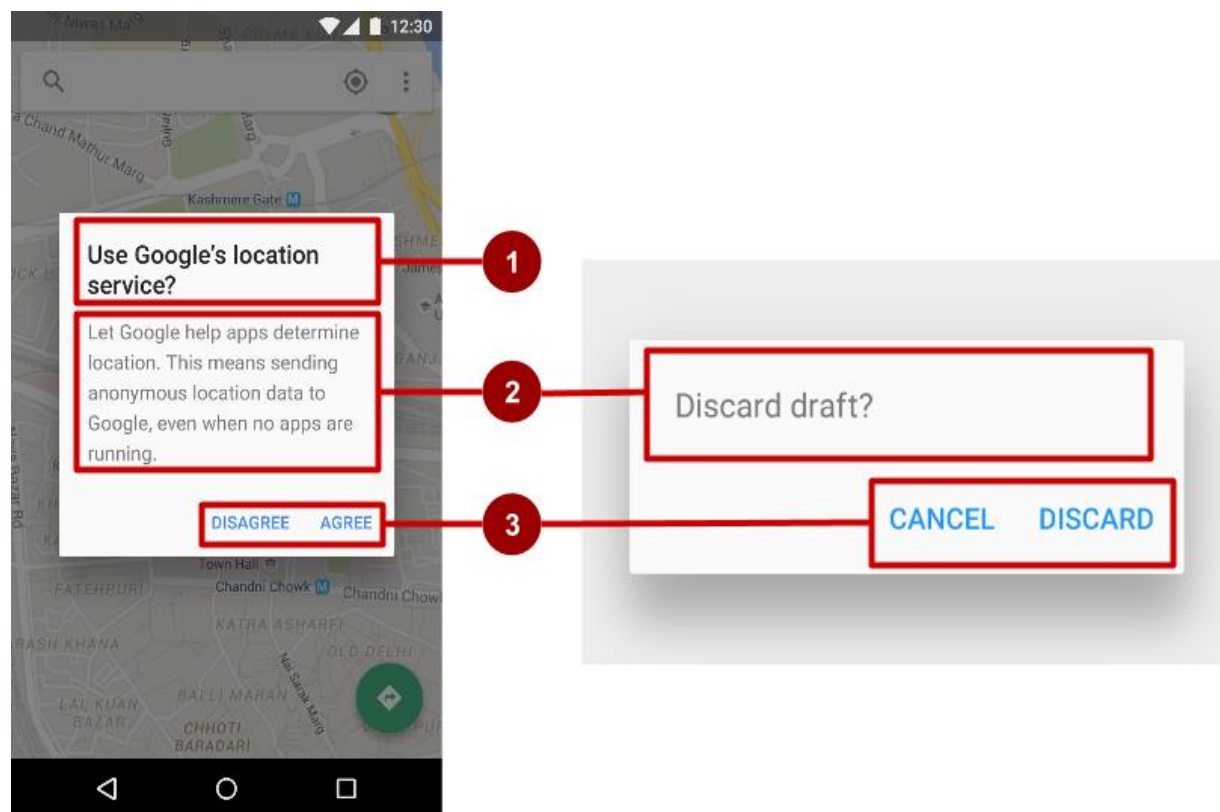
- يستخدم مربع التنبيه لمطالبة المستخدم بالرسائل والأزرار لتنفيذ إجراء معين داخل التطبيق.

- محتويات مربع التنبيه :

1- العنوان (اختياري)

2- المحتوى

3- أزرار الإجراء



# خصائص الميثود لإنشاء مربع التنبيه

Method	Description
setTitle()	It is used to set the title of alertdialog and its an optional component.
setIcon()	It is used to set the icon before the title
setMessage()	It is used to set the message required message to display in alertdialog.
setCancelable()	It is used to allow users to cancel alertdialog by clicking on outside of dialog area by setting true/false.
setPositiveButton()	It is used to set the positive button for alertdialog and we can implement click event of a positive button.
setNegativeButton()	It is used to set the negative button for alertdialog and we can implement click event of a negative button.
setNeutralButton()	It is used to set the neutral button for alertdialog and we can implement click event of a neutral button.

## انشاء مربع التنبيه

```
// 1. Instantiate an <a href="/reference/android/app/AlertDialog.Builder.html">AlertDialog.Builder</a> builder = new AlertDialog.Builder(getActivity());

// 2. Chain together various setter methods to set the dialog characteristics
builder.setMessage(R.string.dialog_message)
    .setTitle(R.string.dialog_title);

// 3. Get the <a href="/reference/android/app/AlertDialog.html">AlertDialog</a> dialog = builder.create();
```



## إضافة خصائص الميثود

```
AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
// Add the buttons
builder.setPositiveButton(R.string.ok, new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        // User clicked OK button
    }
});
builder.setNegativeButton(R.string.cancel, new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        // User cancelled the dialog
    }
});
// Set other dialog properties
...

// Create the AlertDialog
AlertDialog dialog = builder.create();
```