**一、 使用xml定义Menu**

菜单资源文件必须放在res/menu目录中。菜单资源文件必须使用<menu>标签作为根节点。除了<menu>标签外，还有另外两个标签用于设置菜单项和分组，这两个标签是<item>和<group>。

<menu>标签没有任何属性，但可以嵌套在<item>标签中，表示子菜单的形式。不过<item>标签中不能再嵌入<item>标签。

1.<item>标签的属性含义如下:

Id：表示菜单项的资源ID

menuCategory：同种菜单项的种类。该属性可取4个值：**[Container](http://lib.csdn.net/base/docker" \o "Docker知识库" \t "_blank)**、system、secondary和alternative。通过menuCategroy属性可以控制菜单项的位置。例如将属性设为system，表示该菜单项是系统菜单，应放在其他种类菜单项的后面。

orderInCategor:同种类菜单的排列顺序。该属性需要设置一个整数值。例如menuCategory属性值都为system的3个菜单项(item1、item2和item3)。将这3个菜单项的orderInCategory属性值设为3、2、1，那么item3会显示在最前面，而item1会显示在最后面。

title：菜单项标题(菜单项显示的文本)

titleCondensed：菜单项的短标题。当菜单项标题太长时会显示该属性值

icon：菜单项图标资源ID

alphabeticShortcut:菜单项的字母快捷键

numericShortcut：菜单项的数字快捷键

checkable：表示菜单项是否带复选框。该属性可设计为true或false

checked：如果菜单项带复选框(checkable属性为true)，该属性表示复选框默认状态是否被选中。可设置的值为true或false

visible：菜单项默认状态是否可视

enable：菜单项默认状态是否被激活

2.<group>标签的属性含义如下：

id:表示菜单组的ID

menuCategory：与<item>标签的同名属性含义相同。只是作用域为菜单组

orderInCategory：与<item>标签的同名属性含义相同。只是作用域为菜单组

checkableBehavior：设置该组所有菜单项上显示的选择组件(CheckBox或Radio Button)。如果将该属性值设为all，显示CheckBox组件；如果设为single，显示Radio Button组件；如果设为none，显示正常的菜单项(不显示任何选择组件)。要注意的是，**[Android](http://lib.csdn.net/base/android" \o "Android知识库" \t "_blank)** SDK官方文档在解释该属性时有一个笔误，原文是：

Whether the items are checkable. Valid values: none, all(exclusive/radiobuttons), single(non-exclusive/checkboxes).

相反了，正确应该是

all(non-exclusive/checkboxes),single(exclusive/radiobuttons).

visible：表示当前组中所有菜单项是否显示。该属性可设置的值是true或false

enable：表示当前组中所有菜单项是否被激活。该属性可设置的值是true或false

3.具体使用

在代码中使用

public boolenonCreateOptionsMenu(Menu menu){

MenuInflatemenuInflate = getMenuInflate();

menuInflate.inflate(R.menu.option\_menu,menu);

returntrue;

}

**二、使用代码定义Menu**

public boolean onCreateOptionsMenu(Menu menu){

// 方法一，用代码构建

 menu.add(Menu.NONE, Menu.NONE, 1, "菜单1");

 menu.add(Menu.NONE, Menu.NONE, 2, "菜单2");

 menu.add(Menu.NONE, Menu.NONE, 3, "菜单3");

 menu.add(Menu.NONE, Menu.NONE, 4, "菜单4");

menu.add(Menu.NONE, Menu.NONE, 5, "菜单5");

menu.add(Menu.NONE, Menu.NONE, 6, "菜单6");

return true;

}

**三、菜单响应**

重写onOptionsItemSelected(MenuItem item)这个方法就可以做响应的操作了。

**switch**(item.getItemId()){}

**return super**.onOptionsItemSelected(item);

**submenu**

**public boolean onCreateOptionsMenu(Menu menu) {  
 SubMenu M1 = menu.addSubMenu("Menu1");  
 SubMenu M2 = menu.addSubMenu("Menu2");  
 M1.add(0, *menu3*, 0, "Menu3");  
 M1.add(0, *menu4*, 0, "Menu4");  
 return true;  
}  
  
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
 switch (item.getItemId()) {  
 case *menu3*:  
 Toast.*makeText*(this, "你点击的是Menu3", Toast.*LENGTH\_LONG*).show();  
 break;  
 case *menu4*:  
 Toast.*makeText*(this, "你点击的是Menu4", Toast.*LENGTH\_LONG*).show();  
 break;  
  
 }  
 return true;  
}**

add(groupId, itemId, order, titleRes)

三:ContexgMenu（上下文菜单）

当我们长时间按住屏幕的时候，会弹出菜单，那个就是ContextMenu（）;

看先文档中对创建ContextMenu（）的方法:

【注】 长按按钮，来显示上下文菜单我们必须要去调用  registerForContextMenu(View) ，onCreateContextMenu(ContextMenu, View, ContextMenu.ContextMenuInfo).

1. **public** **class** Context\_Menu\_Activity **extends** Activity {
2. **private** **static** **final** **int** menu1 = 1;
3. **private** **static** **final** **int** menu2 = 2;
4. **private** **static** **final** **int** menu3 = 3;
6. **private** TextView context\_demo;
8. @Override
9. **protected** **void** onCreate(Bundle savedInstanceState) {
10. // TODO Auto-generated method stub
11. **super**.onCreate(savedInstanceState);
12. setContentView(R.layout.context);
13. context\_demo = (TextView) **this**.findViewById(R.id.context\_demo);
14. registerForContextMenu(context\_demo);
15. }
17. @Override
18. **public** **void** onCreateContextMenu(ContextMenu menu, View v,
19. ContextMenuInfo menuInfo) {
21. **super**.onCreateContextMenu(menu, v, menuInfo);
22. menu.add(0, menu1, 0, "Menu1");
23. menu.add(0, menu2, 0, "Menu2");
24. menu.add(0, menu3, 0, "Menu3");
25. }
27. @Override
28. **public** **boolean** onContextItemSelected(MenuItem item) {
29. **switch** (item.getItemId()) {
30. **case** menu1:
31. Toast.makeText(**this**, "你 点击的是Menu1", Toast.LENGTH\_LONG).show();
32. context\_demo.setText("你点击的是Menu1");
33. **break**;
35. **case** menu2:
36. Toast.makeText(**this**, "你点击的是Menu2", Toast.LENGTH\_LONG).show();
37. context\_demo.setText("你点击的是Menu2");
38. **break**;
39. **case** menu3:
40. Toast.makeText(**this**, "你点击的是Menu3", Toast.LENGTH\_LONG).show();
41. context\_demo.setText("你点击的是Menu3");
42. **break**;
43. }
44. **return** **true**;
45. }
46. }