**·**在代码中通过**R.string.app\_name**可以获得该字符串的引用

**·**在XML中通过**@string/app\_name**可以获得改字符串的引用

**·**App目录下的build.gradle中的**minifyEnabled**用于指定是否对项目的代码进行混淆

**·**所有活动必须在**AndroidManifest.xml**中注册才能生效

**·Log:**

Log.v() 那些最为琐碎的、意义最小的日志信息。对应级别verbose。

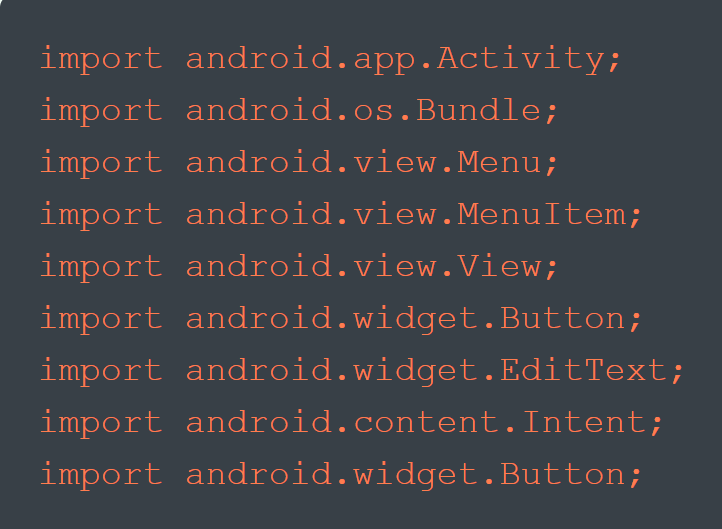
Log.d() 一些调试信息，对调试程序和分析问题应该是最有帮助的。级别debug。

Log.i() 一些比较重要的数据，这些数据可以帮你分析用户行为数据，对应级别info。

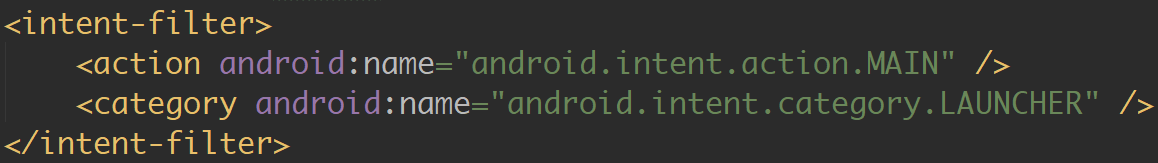
Log.w() 警告信息，程序在这个地方可能会有潜在风险，最好修复一下出现的警告。warn。

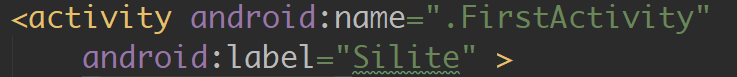
Log.e() 进入catch语句中，当有错误信息，一般出现了严重问题。对应级别error。

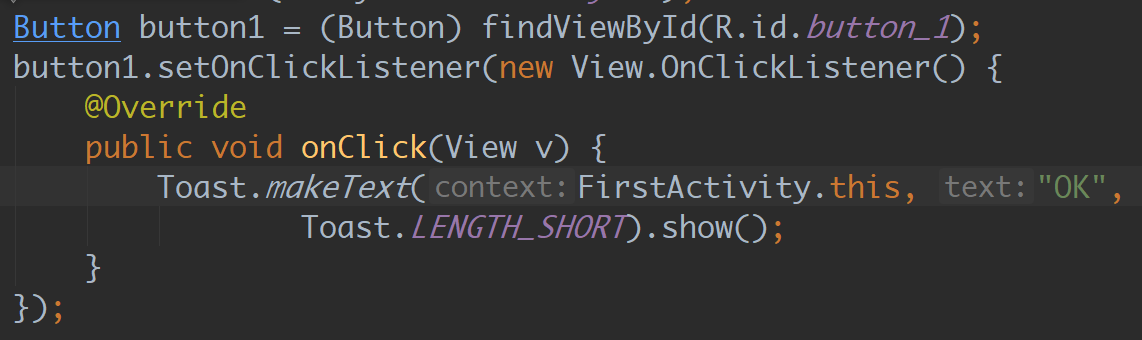
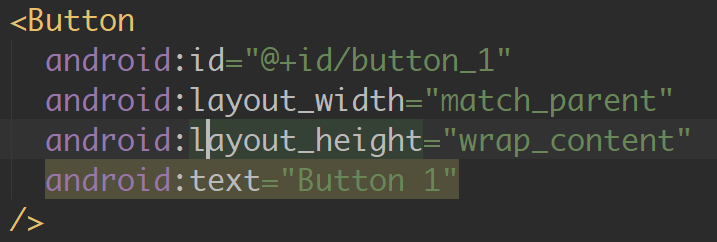
**·**在**onCreate()**方法外面输入logt然后按Tab 自动生成一个TAG常量

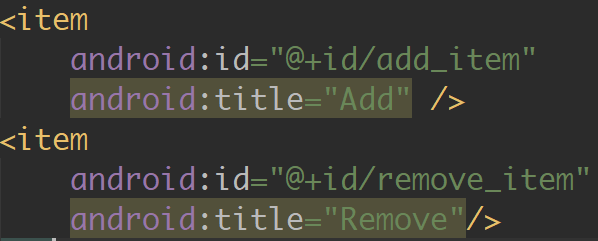
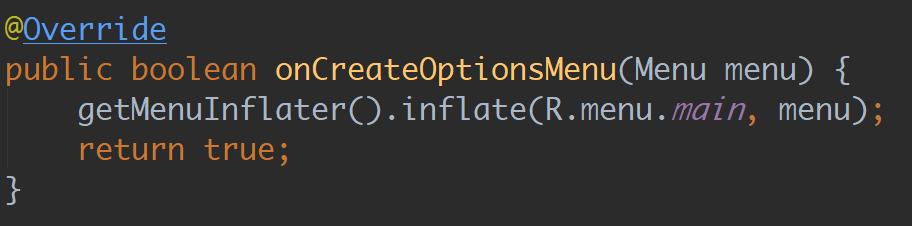
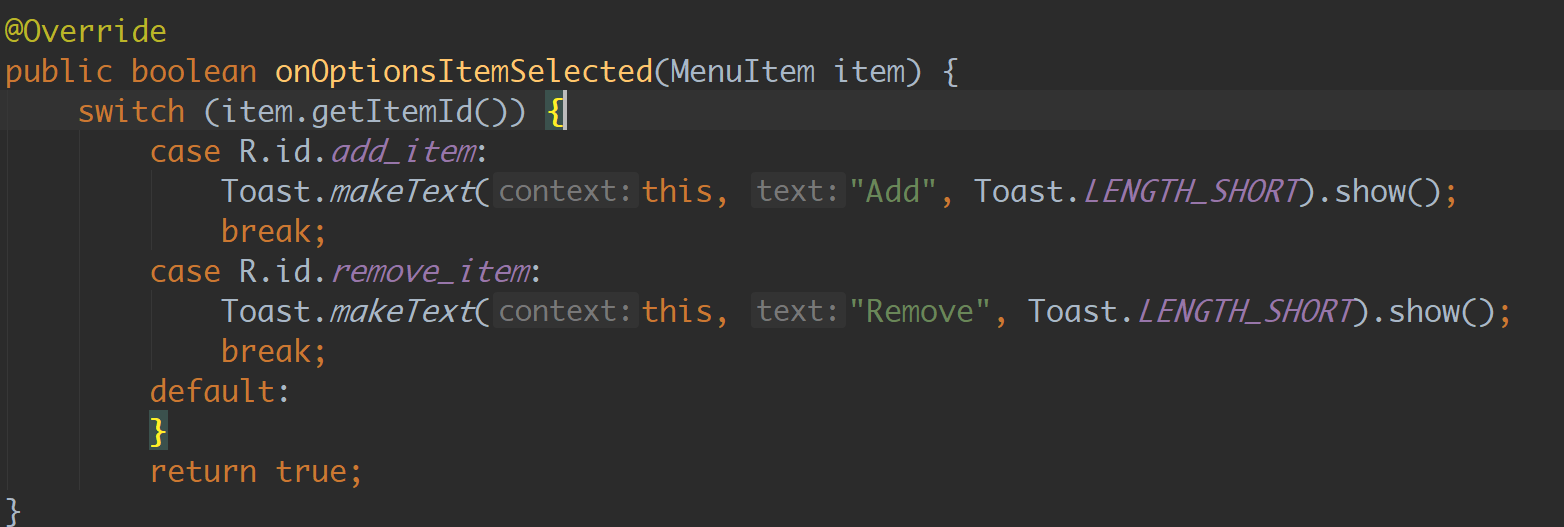
**·必要库**

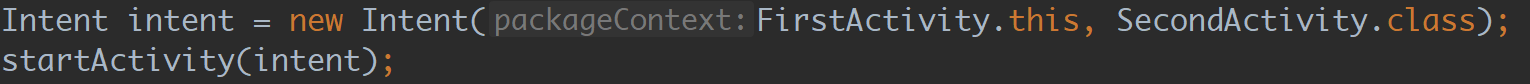
**·**当程序运行起来的时候，不知道首先要启用哪个活动。**在<activity>标签内部加入**

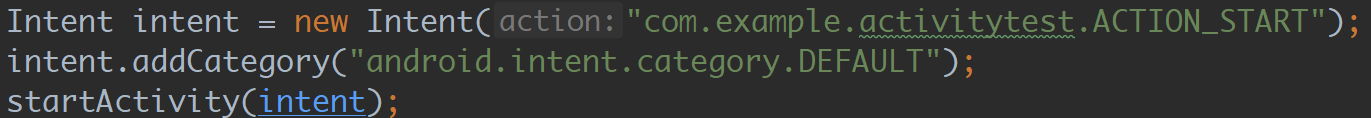
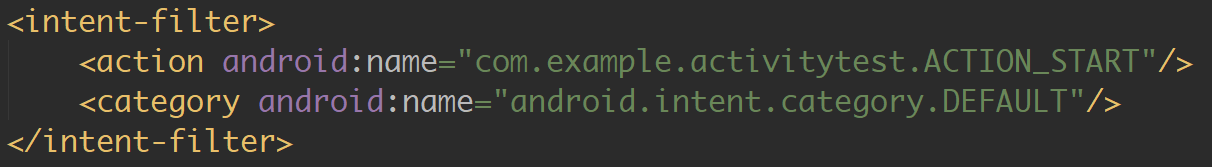


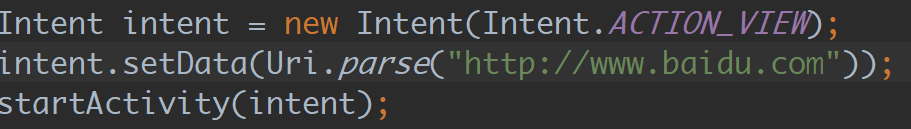
**·修改标题栏**

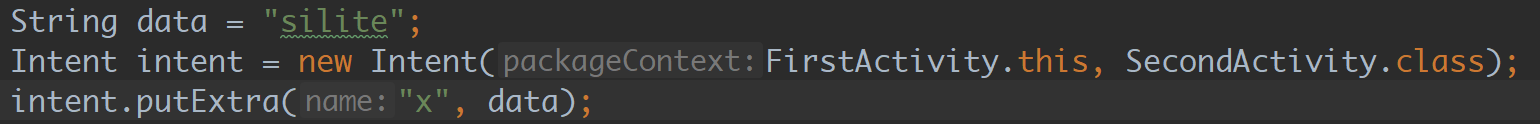
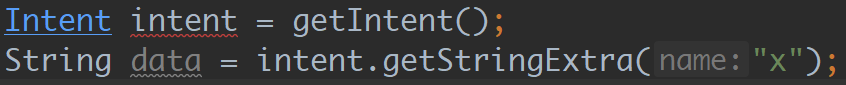
**·按钮**在layout里修改xml

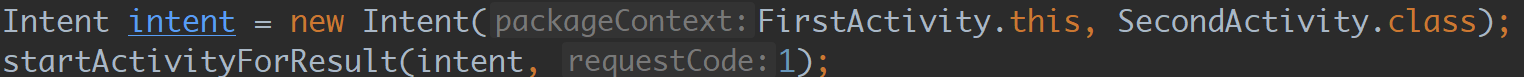
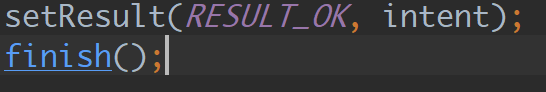
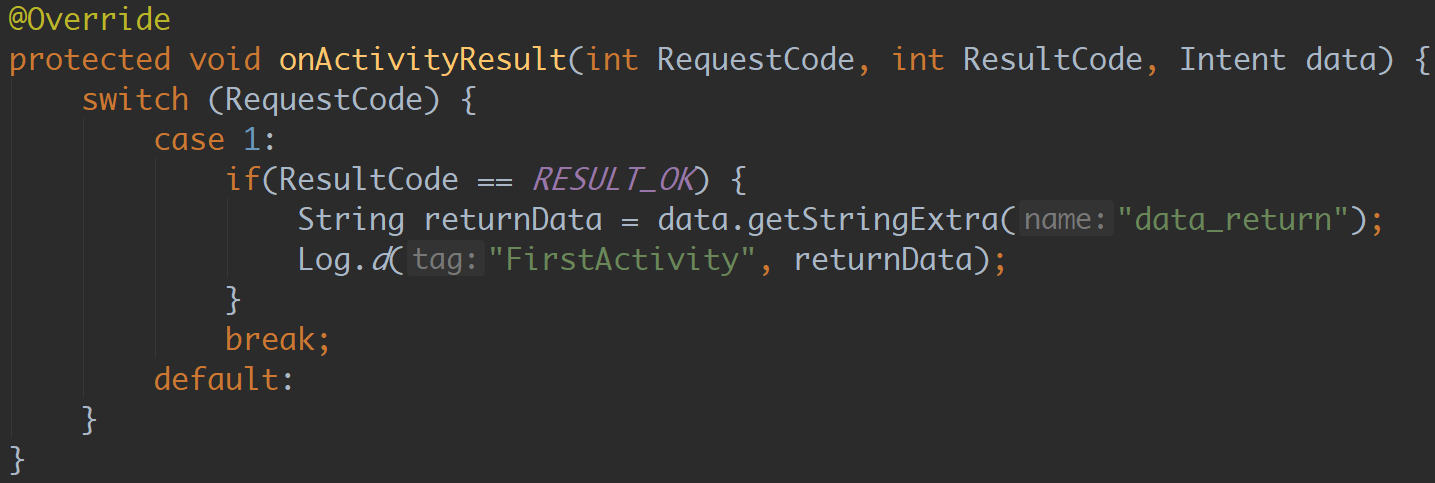
**·**菜单在res新建menu文件夹 **item是每一个小选项** 在java中显示菜单 绑定事件

**·显式Intent**

**·隐式Intent**并添加可以与上下文相应的二者都相符才可以相应

**·跳转网页**

**·向下一个活动中传入数据**在下一个活动中用对应键值取出字符串

**·返回数据给上一个活动**第二个参数是请求码，则是返回给上一个活动数据，由于使用的是startActivityForResult()来启动下一个活动的，在下一个活动销毁时会回调上一个活动的onActivityResult()方法，因此需要在上一个活动中重写。

**·Activity**类中定义了7个回调方法:

onCreate() 会在活动第一次创建时调用(加载布局，响应事件)

onStart() 会在活动由不可见变为可见时调用

onResume() 在活动准备好和用户进行交互时调用，此时活动位于栈顶

onPause() 会在系统准备启动或者恢复另一个活动时调用，释放一些消耗CPU的资源

onStop() 活动不可见时调用，如果新活动是对话框式的活动，onPause()会被执行

onDestroy() 在活动被销毁前调用

onRestart() 会在活动由停止变为运行状态之前调用，也就是活动被重新启用了

**·Activity**提供onSaveInstanceState()回调方法，这个方法可以保证活动在回收之前一定会被调用

