

Android Studio使用基础

北京理工大学计算机学院
金旭亮

概述



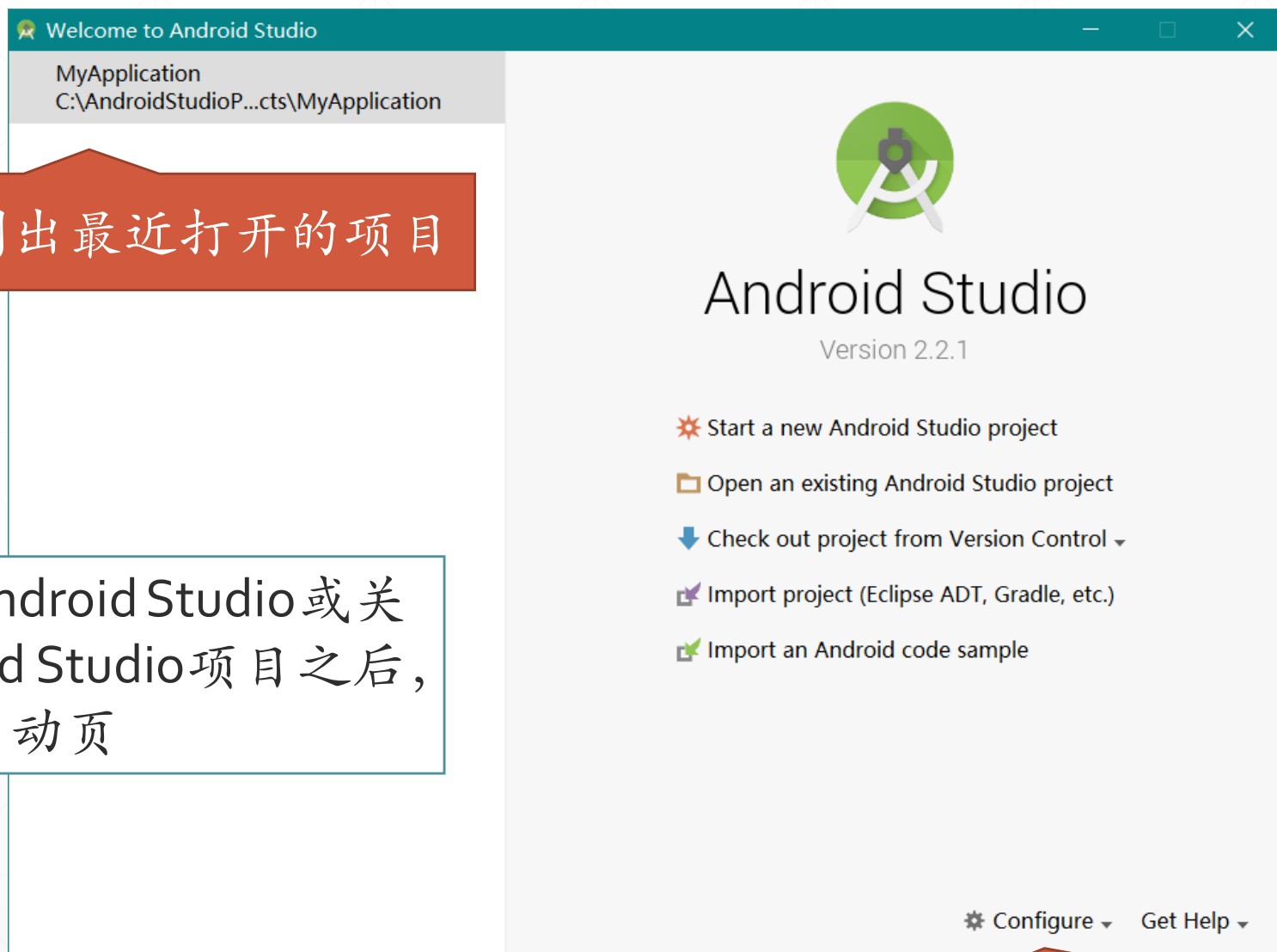
Android Studio是Google官方指定的Android开发环境，是开发Android本地应用（Native APP）的主要工具



至2016年10月为止，Android Studio的最新版本是2.2.1，本PPT介绍它的基本使用方法。

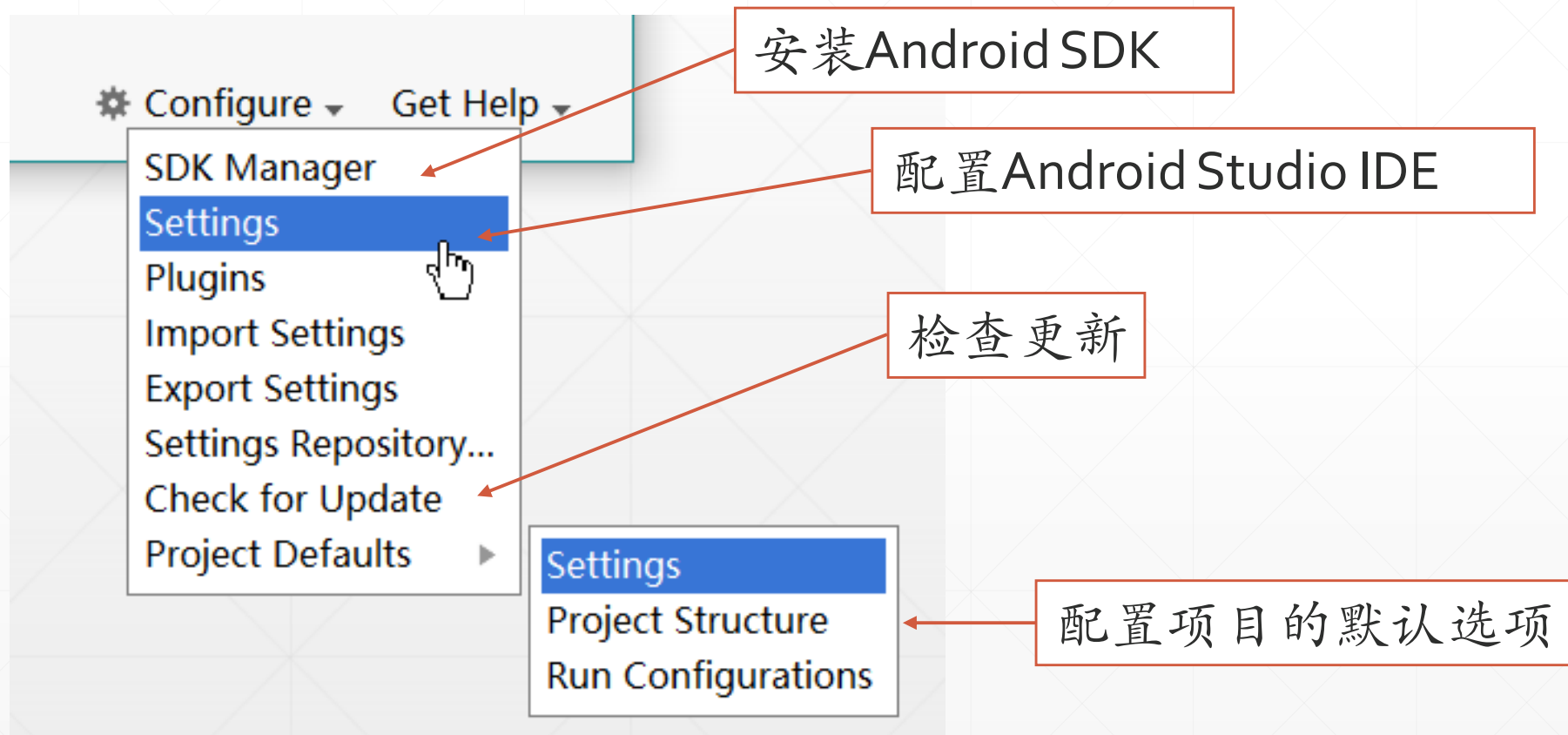
列出最近打开的项目

第一次打开Android Studio或关闭一个Android Studio项目之后，
会打开这个启动页



点击此可从下拉菜单中选择命令
配置Android Studio

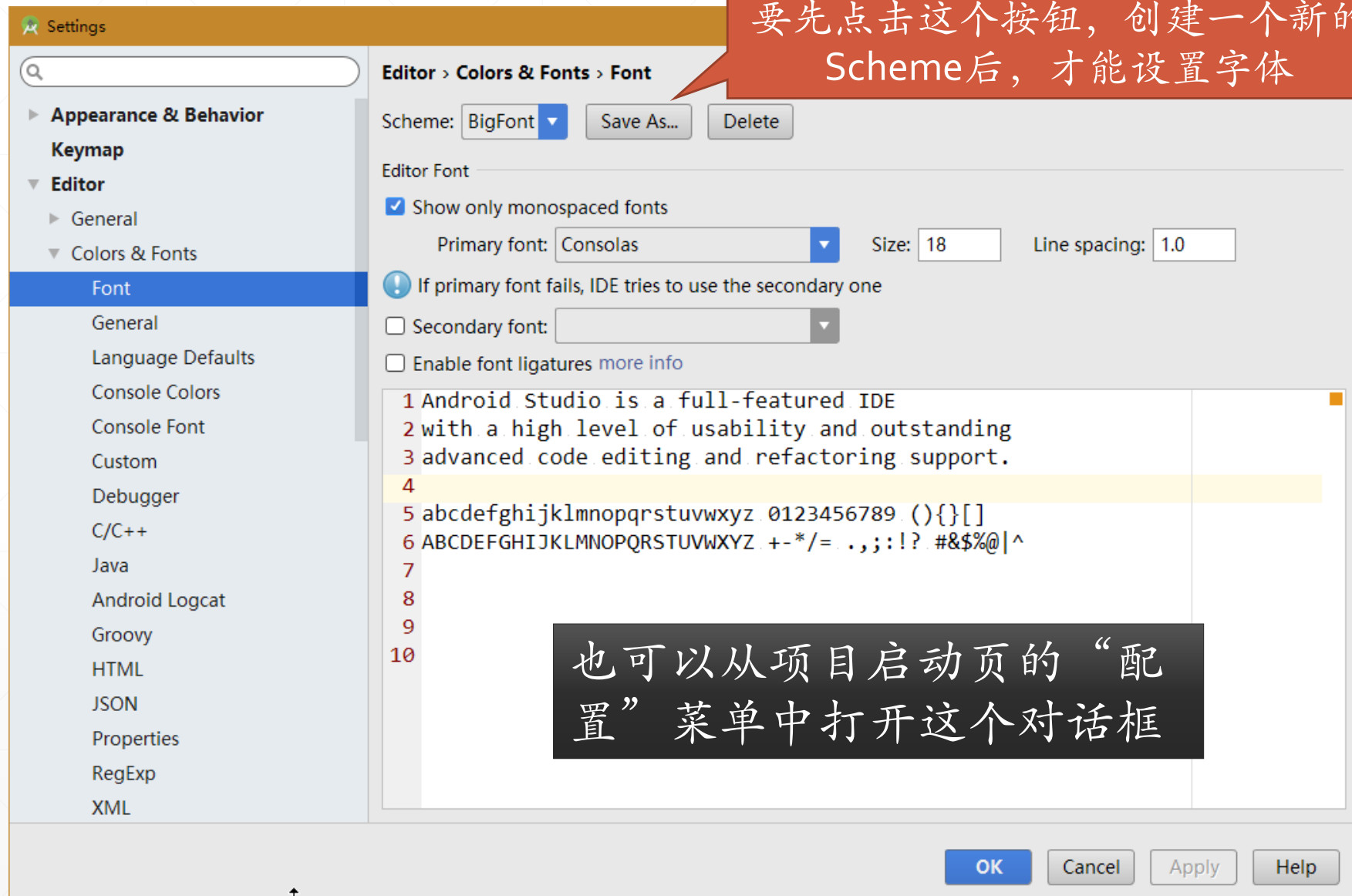
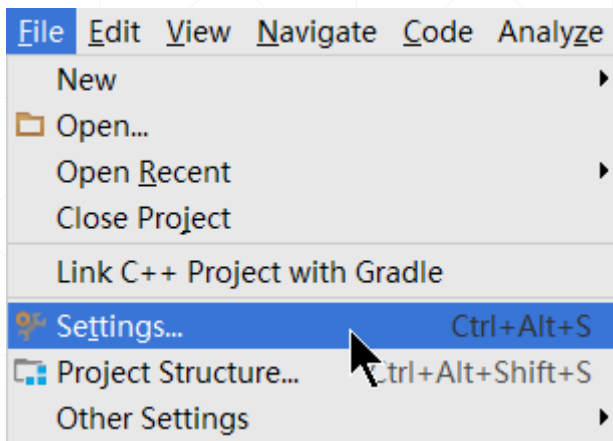
在启动页右下角可以进行的配置功能



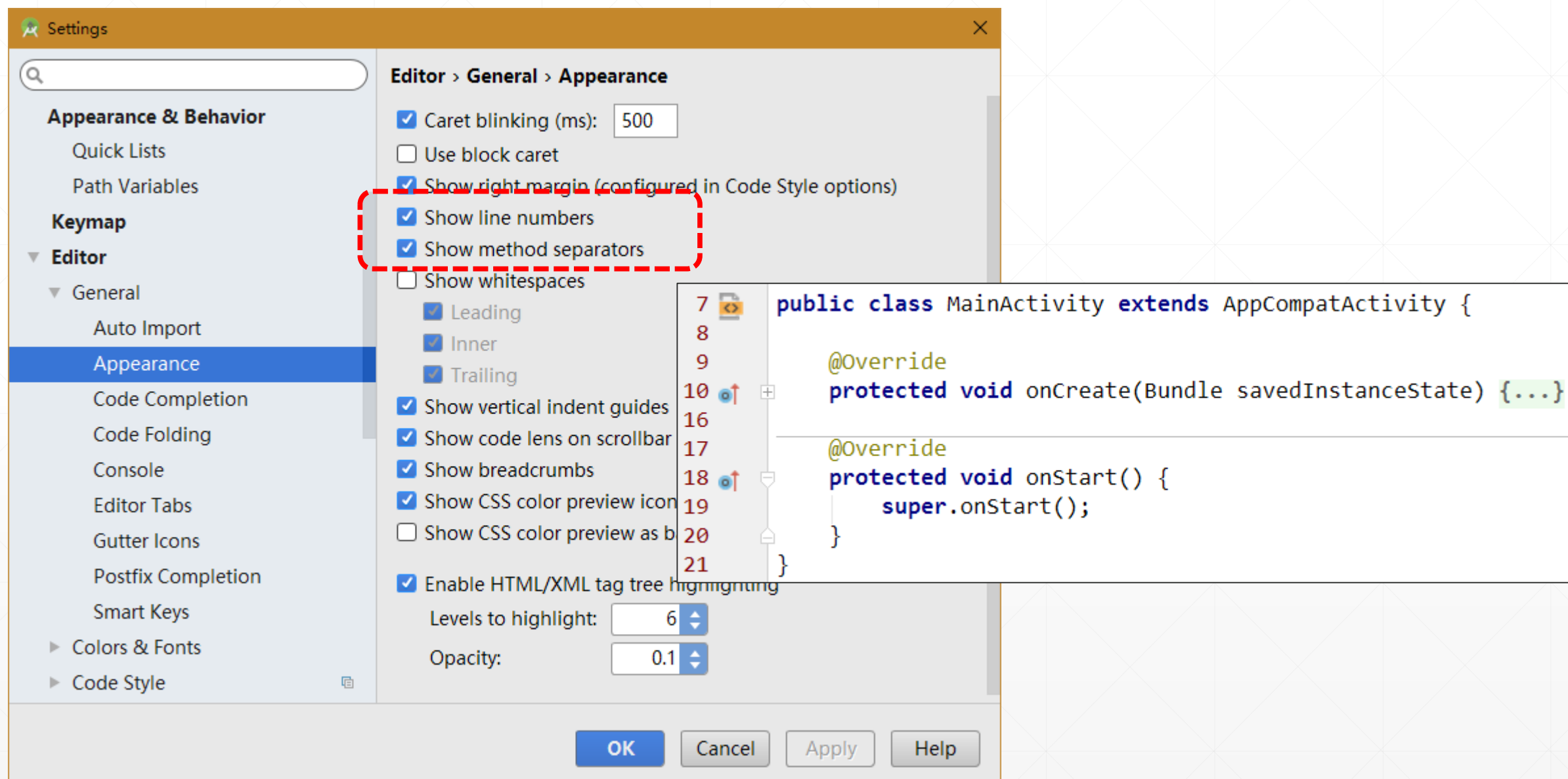
基本配置

设置字体

从File菜单打开“配置”对话框

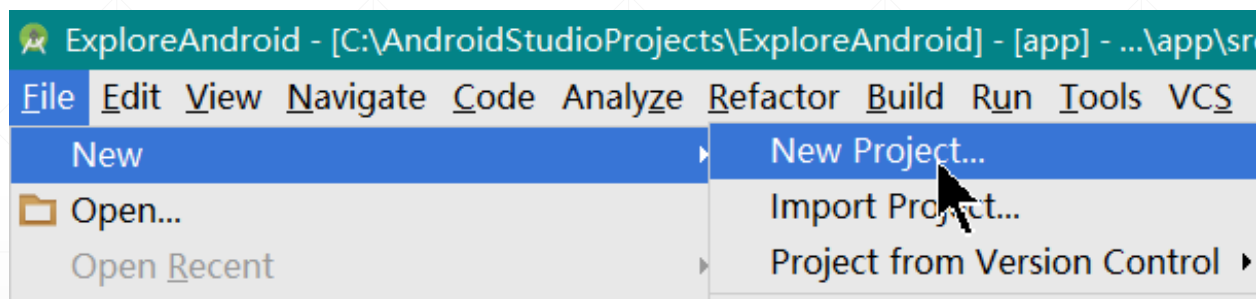
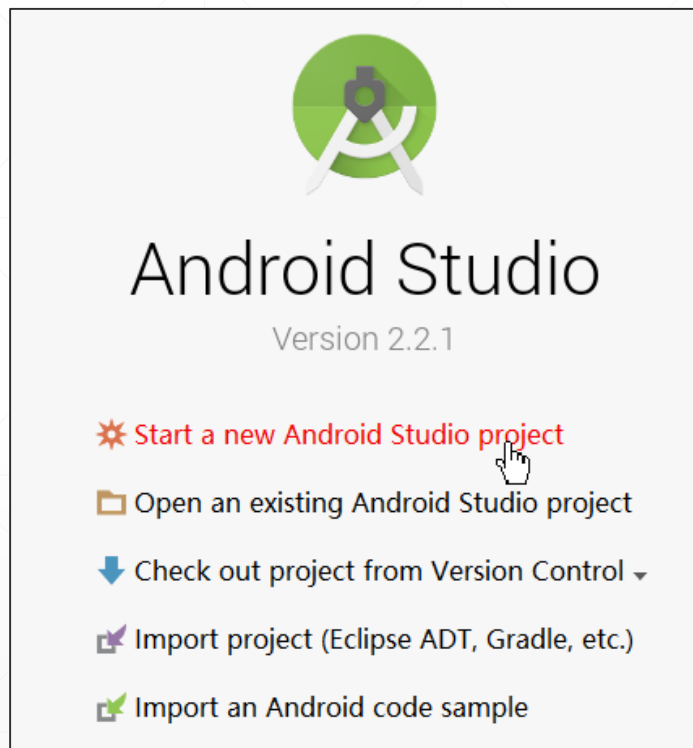


显示行号和方法分隔线



Android Studio应用开发流程

新建项目



设置项目名称与位置

Create New Project

New Project
Android Studio

Configure your new project

包名 = 公司域名 + 应用名, 点 “Edit”
可以手工指定包名

Application name: ExploreAndroid

Company Domain: jinxuliang.com

Package name: com.jinxuliang.exploreandroid


☐ Include C++ Support

指定项目保存位置, 右边按钮可以修
改项目保存位置

Project location: C:\AndroidStudioProjects\ExploreAndroid

Previous Next Cancel Finish

Create New Project

 Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☒ Phone and Tablet

Minimum SDK

API 19: Android 4.4 (KitKat)

Lower API levels target more devices, but have fewer features available.
By targeting API 19 and later, your app will run on approximately **73.9%** of the devices that are active on the Google Play Store.
[Help me choose](#)

☐ Wear

Minimum SDK

API 21: Android 5.0 (Lollipop)

☐ TV

Minimum SDK

API 21: Android 5.0 (Lollipop)

☐ Android Auto

☐ Glass

Minimum SDK

Glass Development Kit Preview (API 19)

Previous

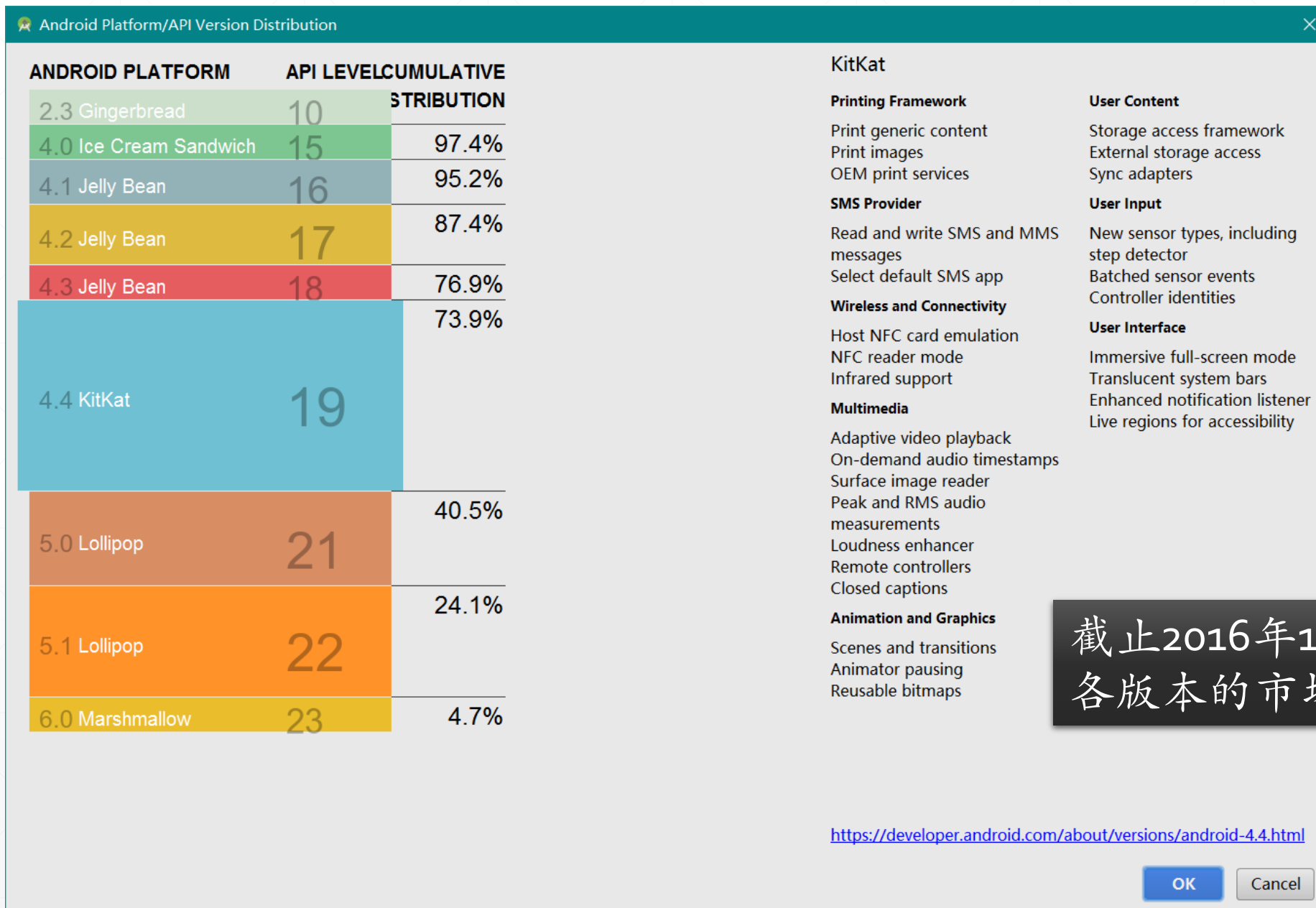
Next

Cancel

Finish

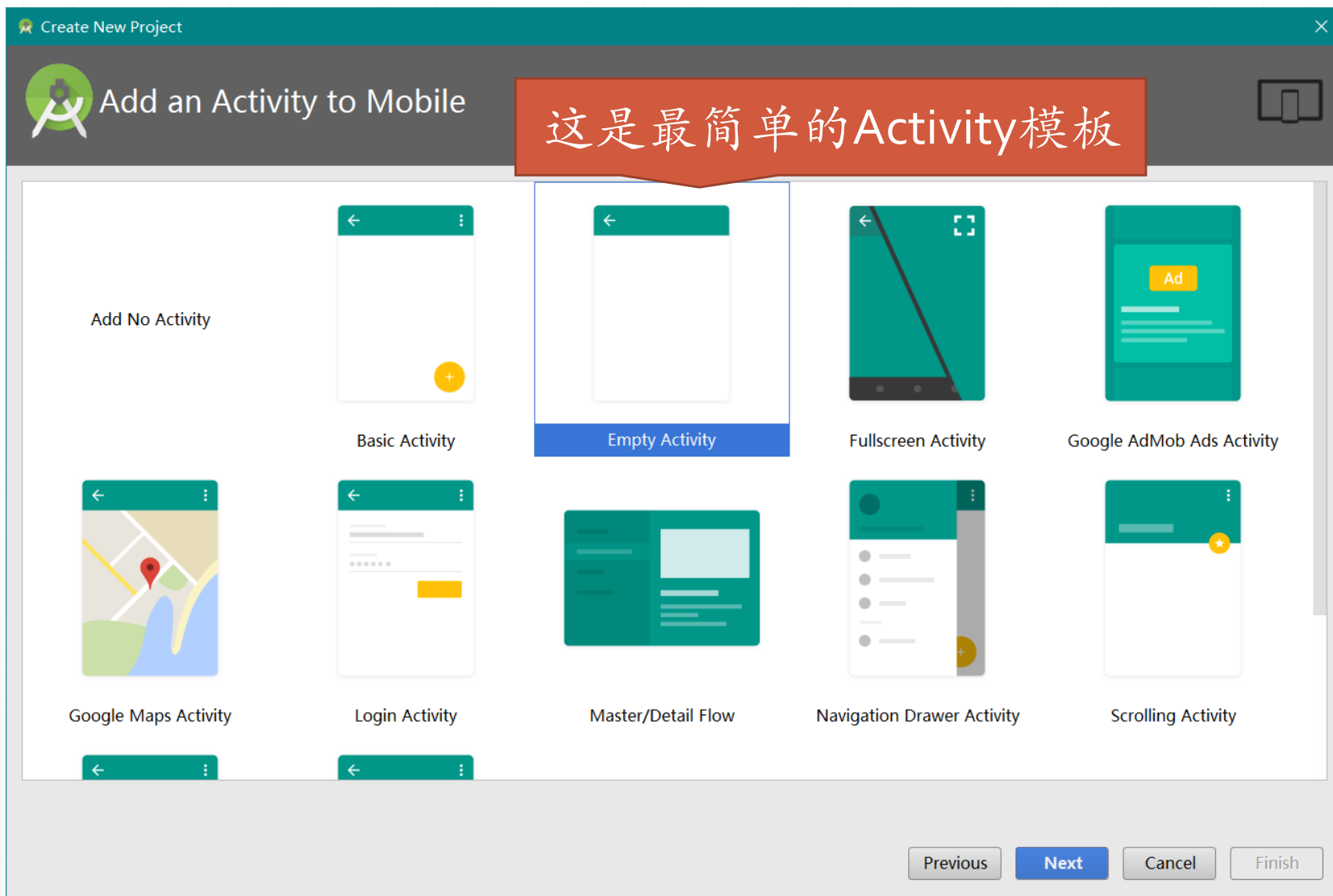
选择最低可用的Android版本

点击“Help me choose”，可以显示各版本的占用率.....



截止2016年10月Android
各版本的市场占有率

向项目中添加Activity



指定Activity和布局文件名

Create New Project

Customize the Activity

←

Empty Activity

Creates a new empty activity

Activity Name: MainActivity

☒ Generate Layout File

Layout Name: activity_main

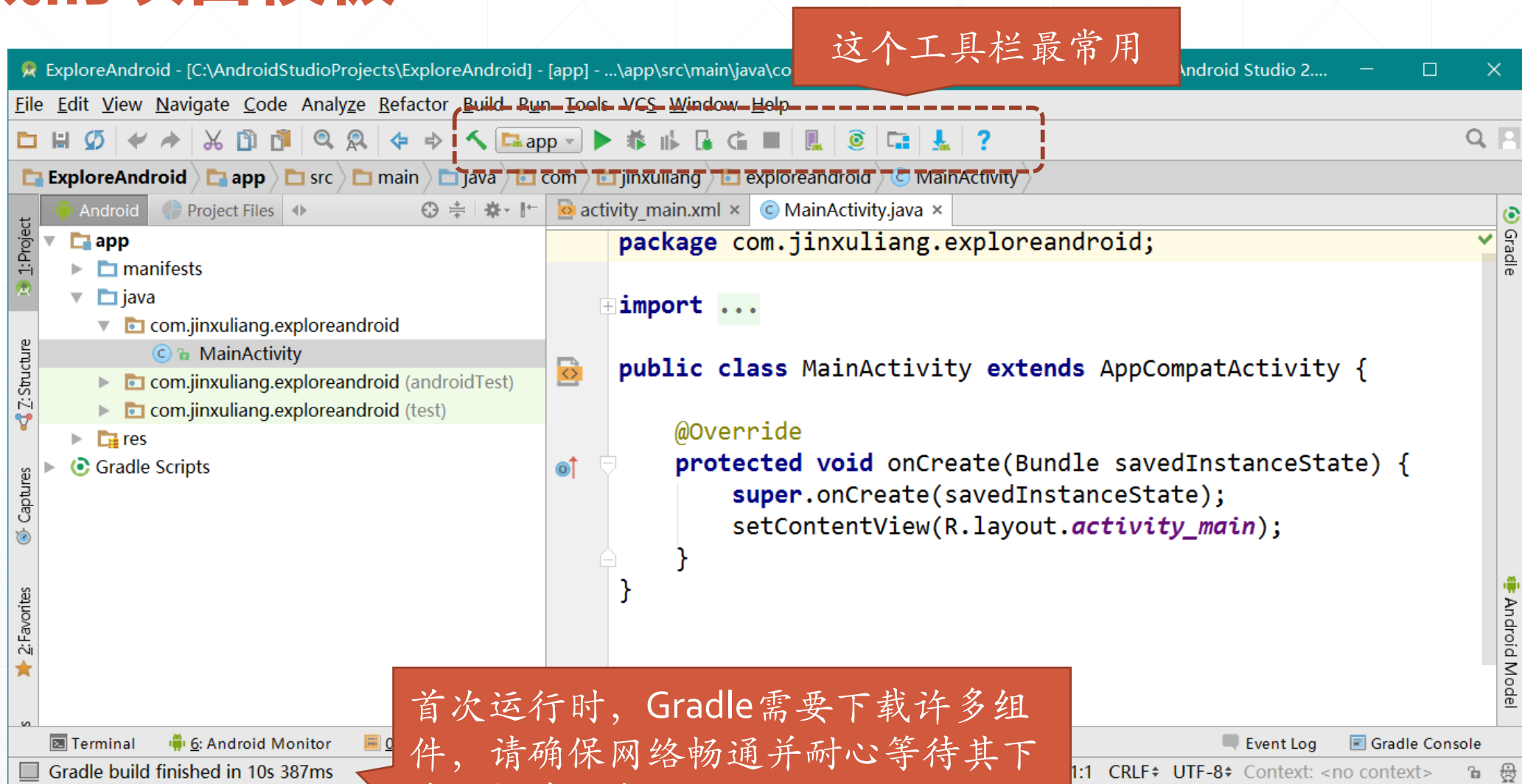
☒ Backwards Compatibility (AppCompat)

如果不需要在老版本Android手机上运行，可以不勾选此项

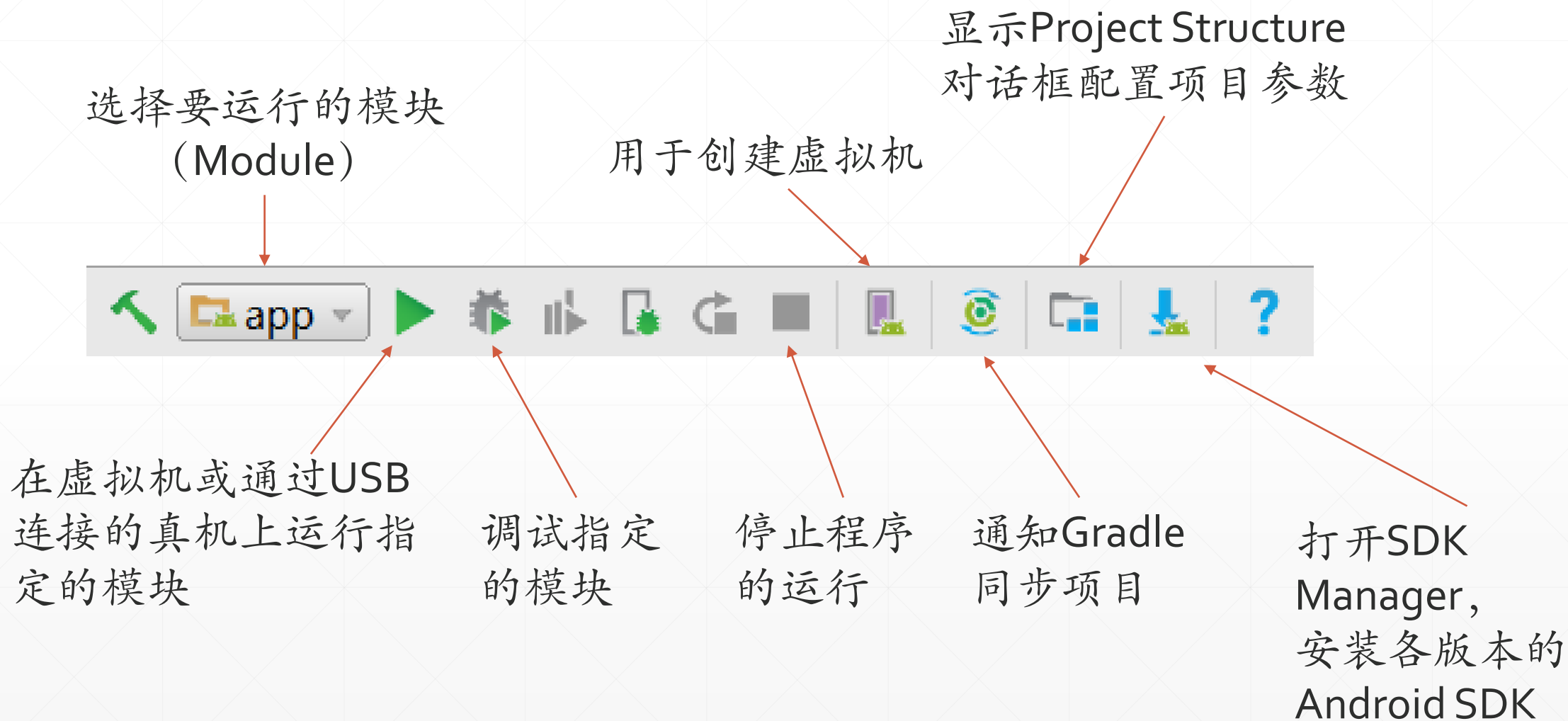
The name of the activity class to create

Previous Next Cancel Finish

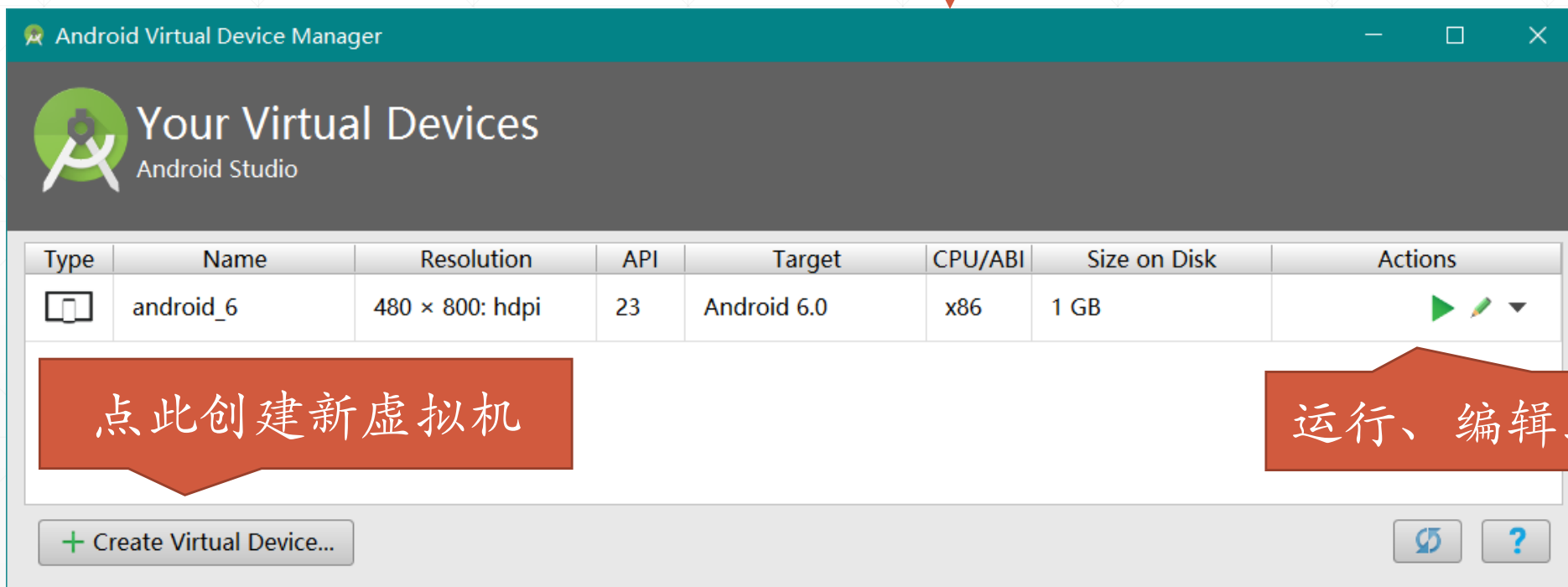
生成的项目模板



开发必用之工具栏命令



创建虚拟机

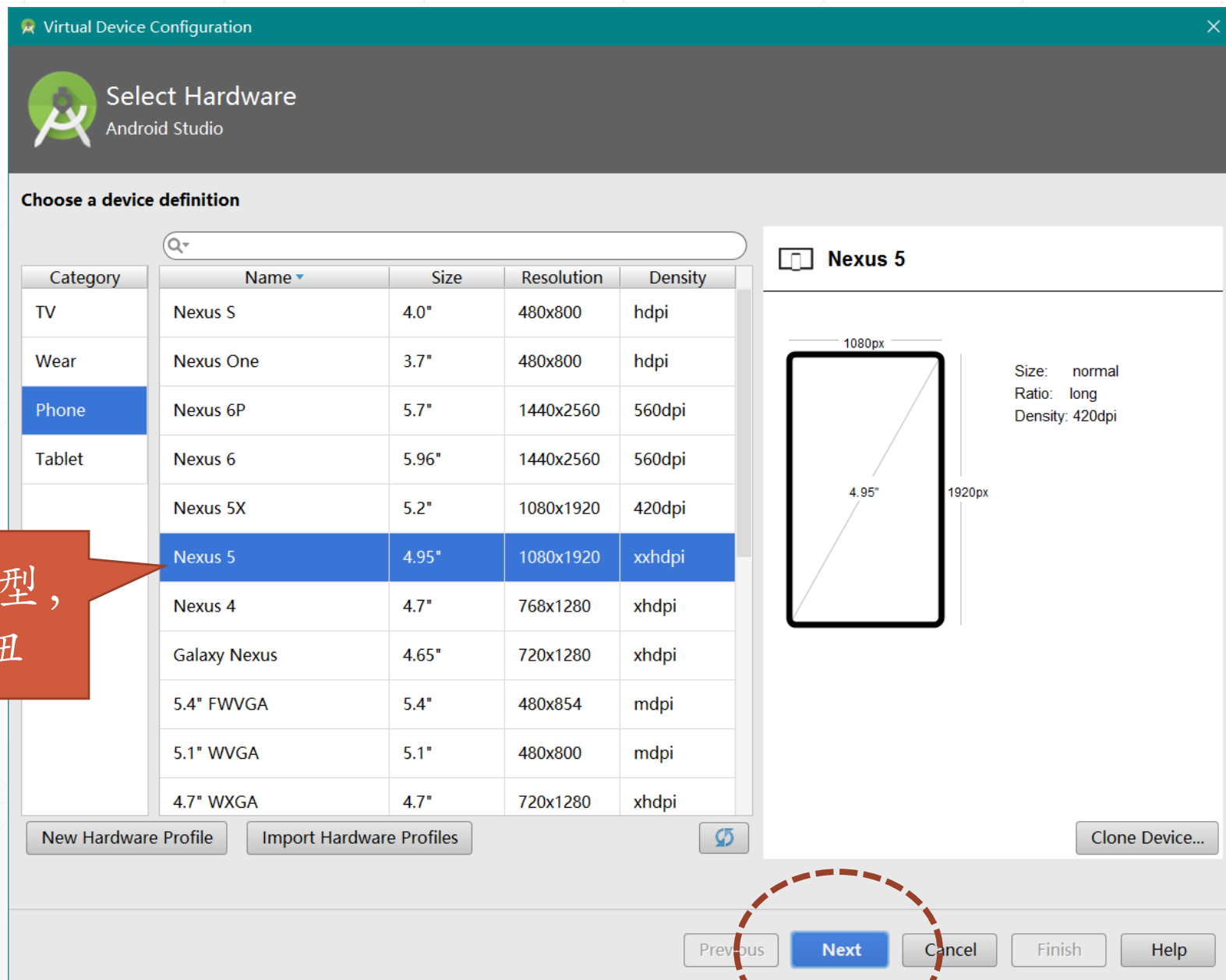


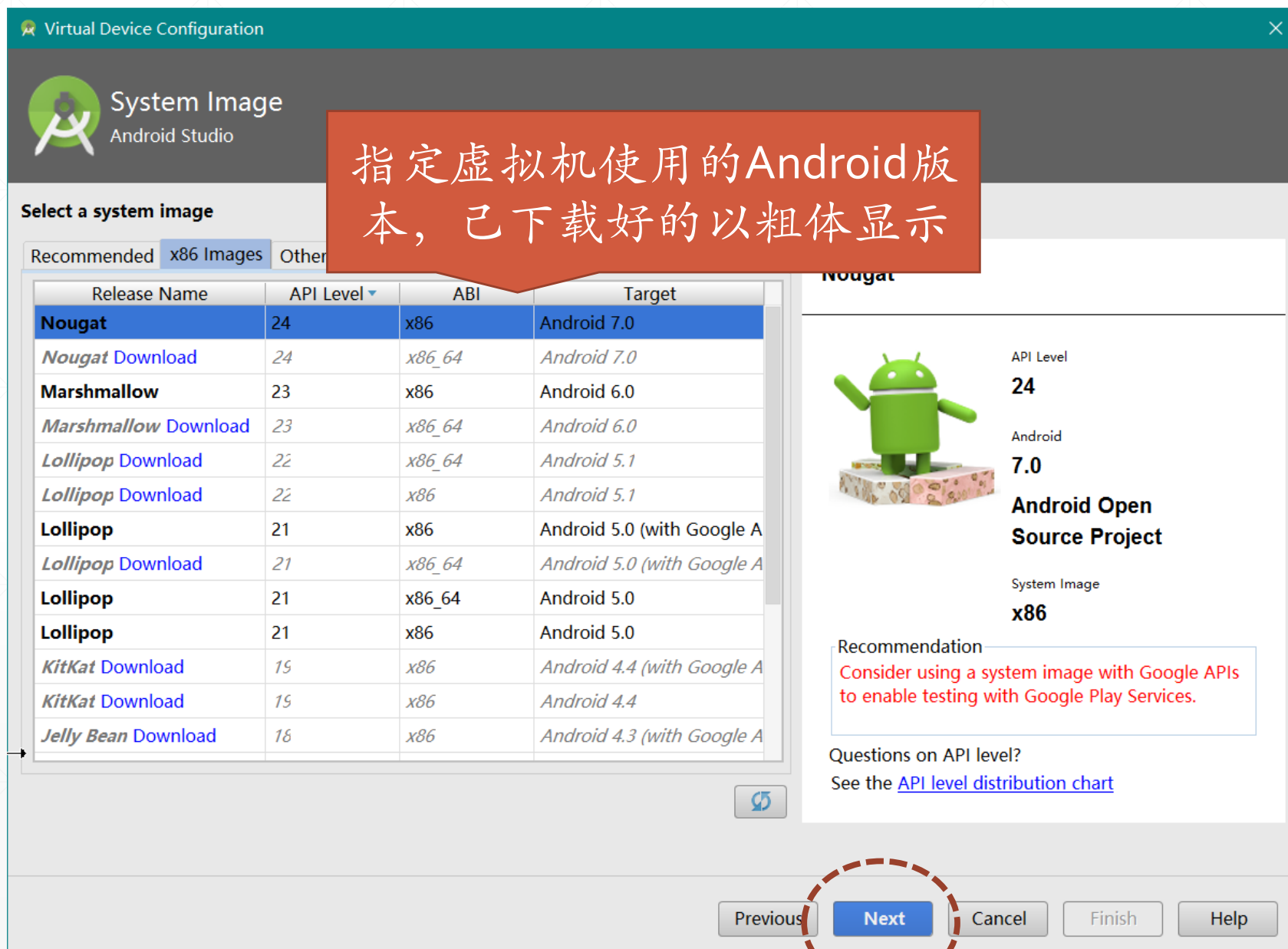
点此创建新虚拟机

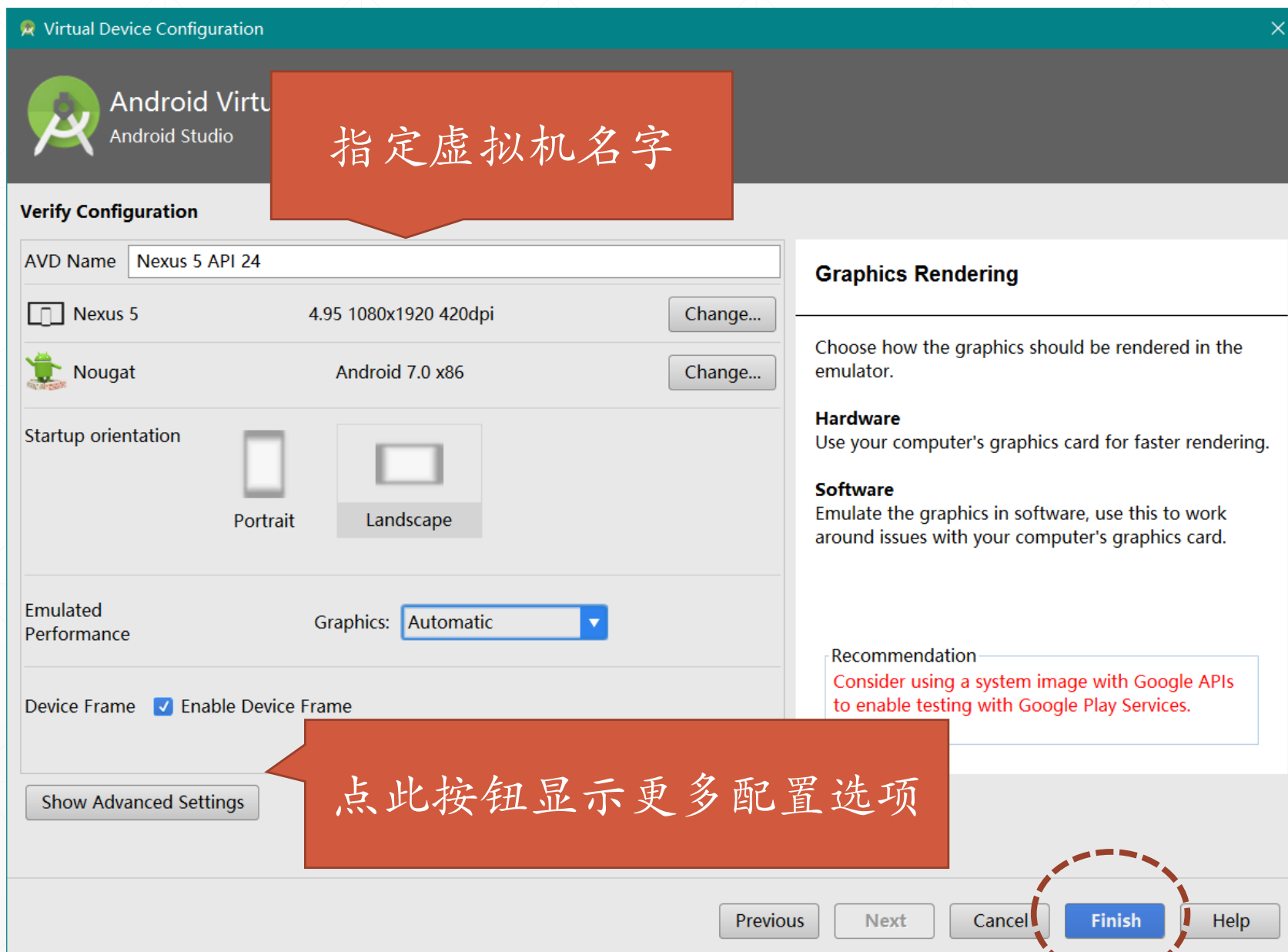
运行、编辑或删除虚拟机

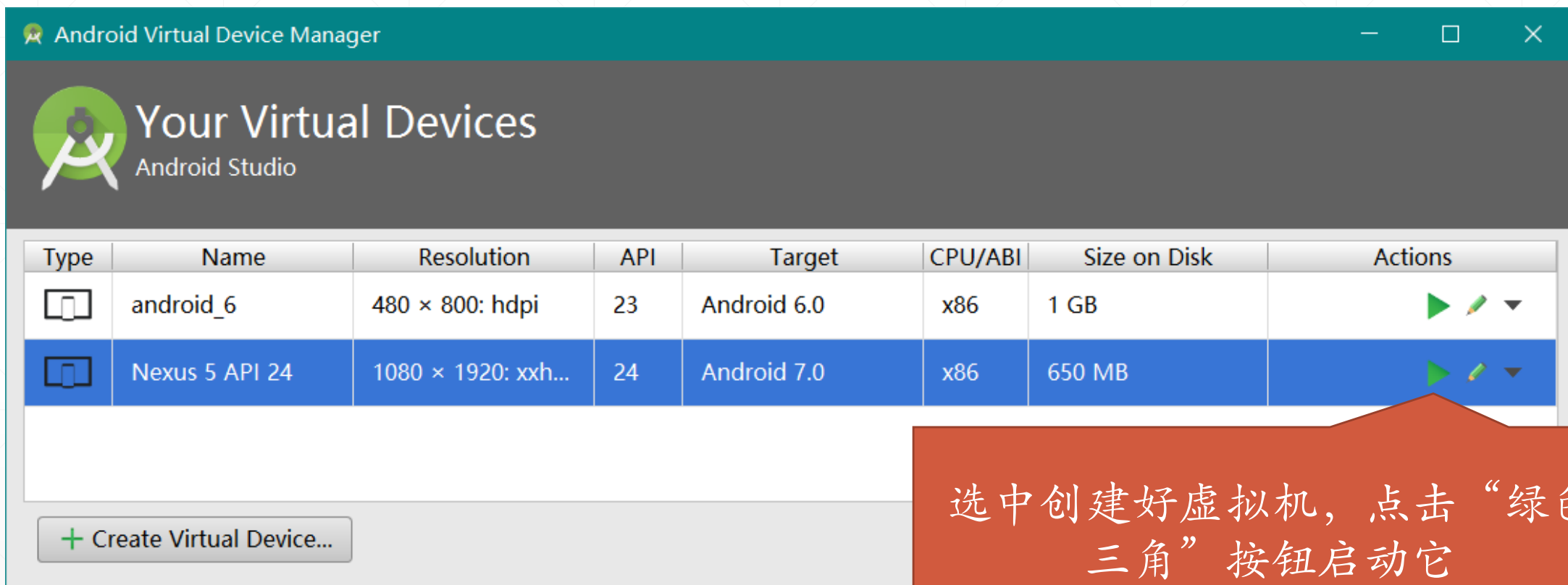
创建虚拟机

从列表中选一个机型，
点击“Next”按钮

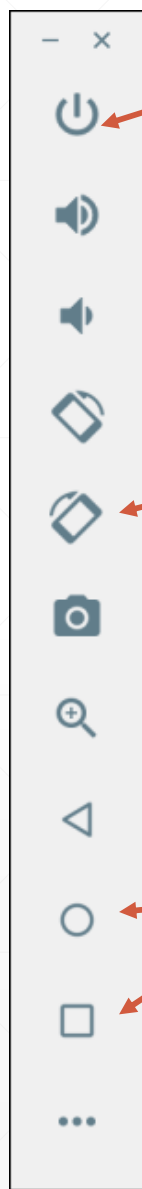
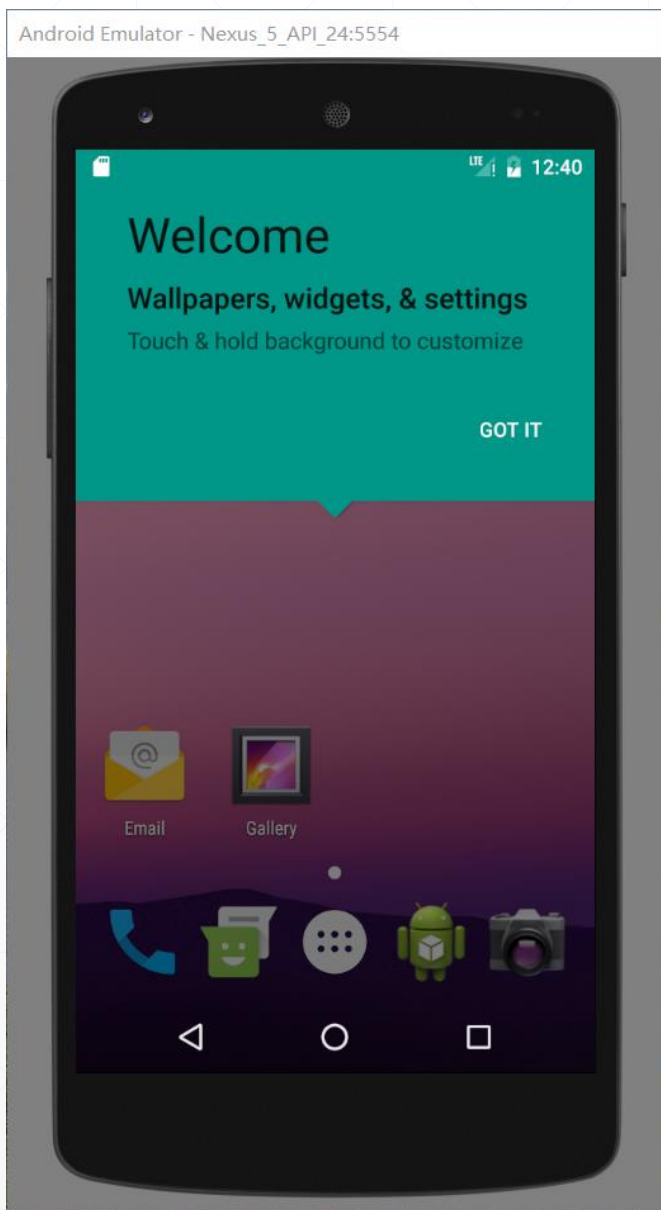








选中创建好虚拟机，点击“绿色三角”按钮启动它



关闭虚拟机

旋转虚拟机

手机上的三个按键

显示更多选项

Extended controls

Location

Cellular

Battery

Phone

Directional pad

Fingerprint

Virtual sensors

Settings

Help

From
(650) 555-1212

HOLD CALL

CALL DEVICE

SMS message

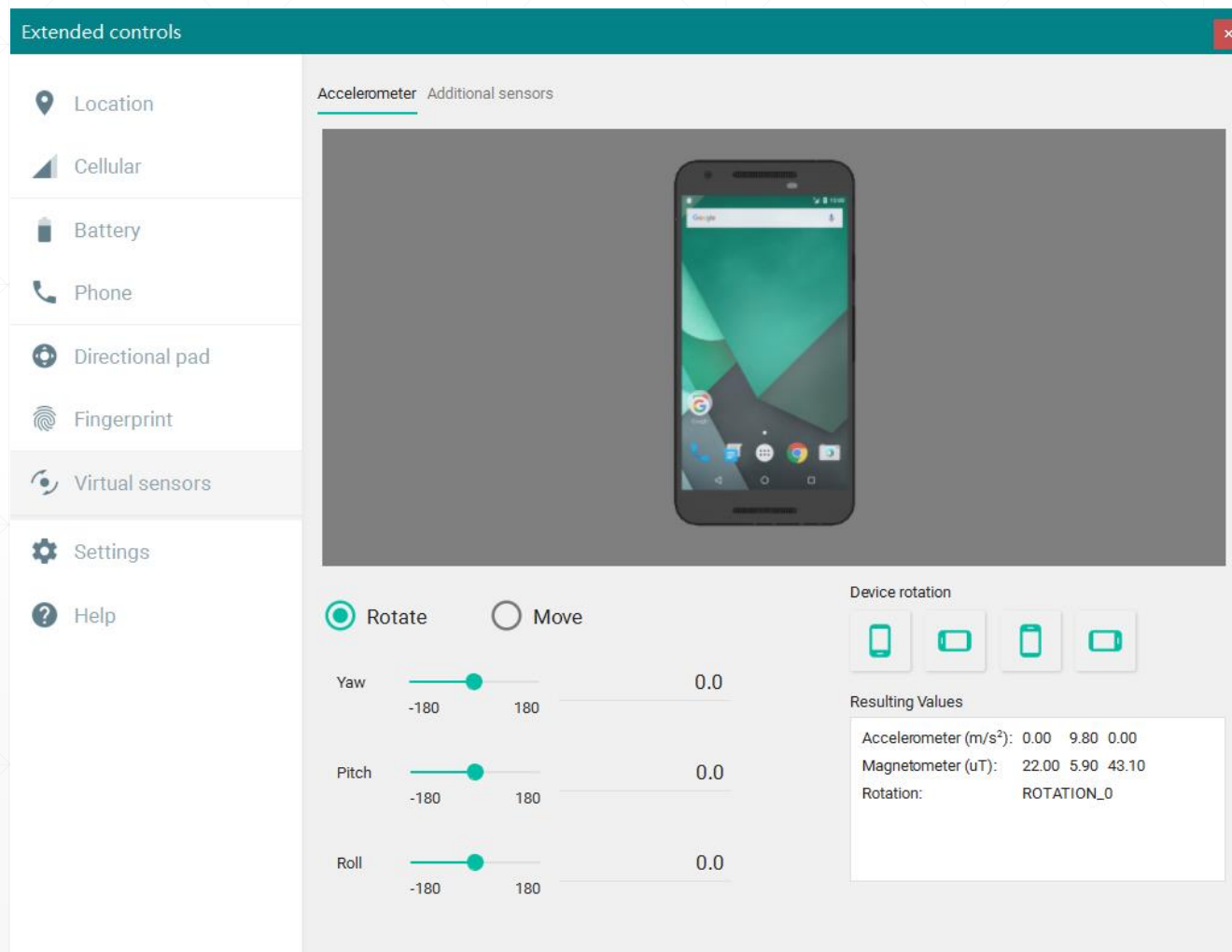
Don't forget the marshmallows!

SEND MESSAGE

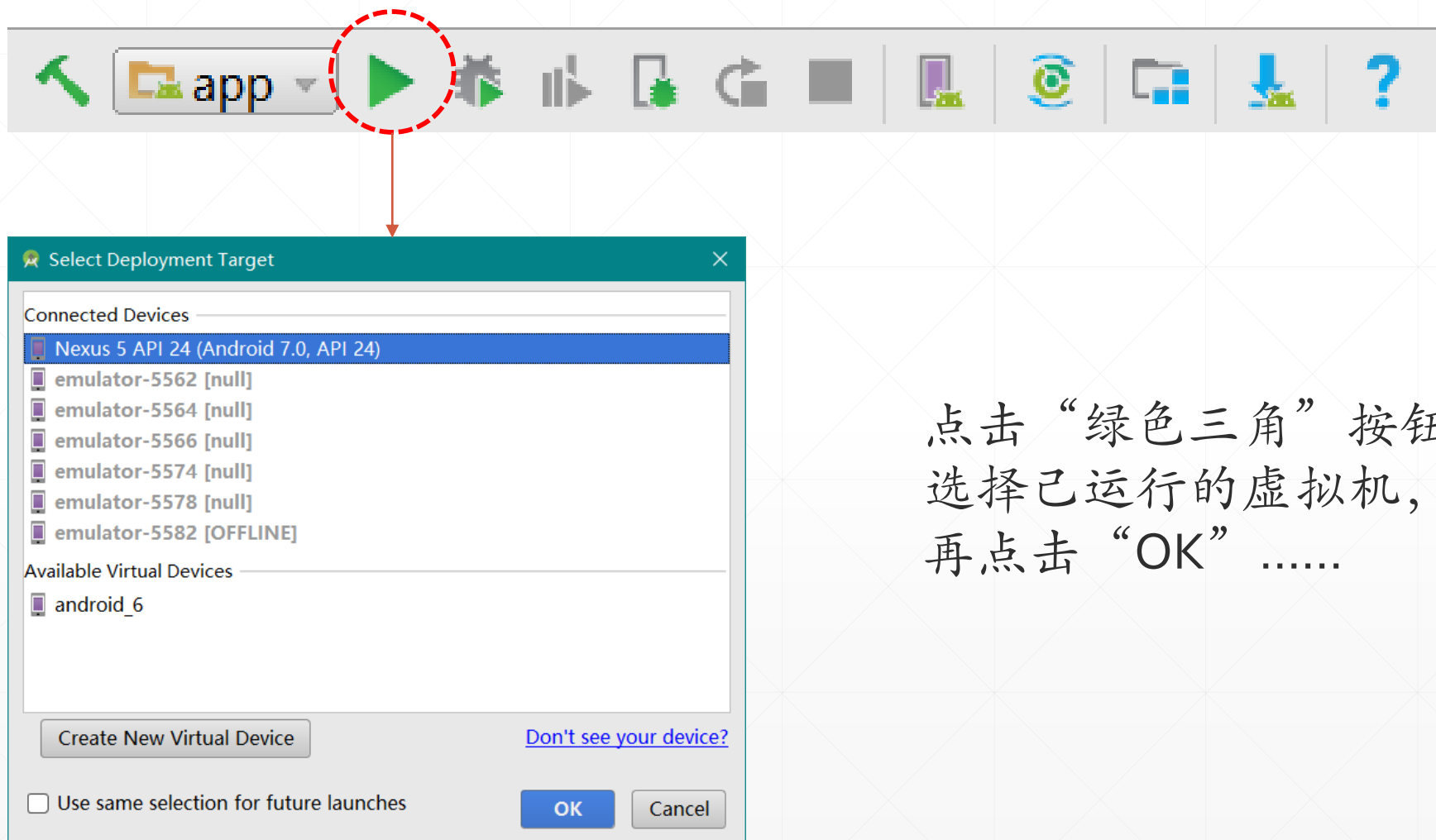
模拟电话拨入

模拟短信接收

模拟各种传感器



运行程序

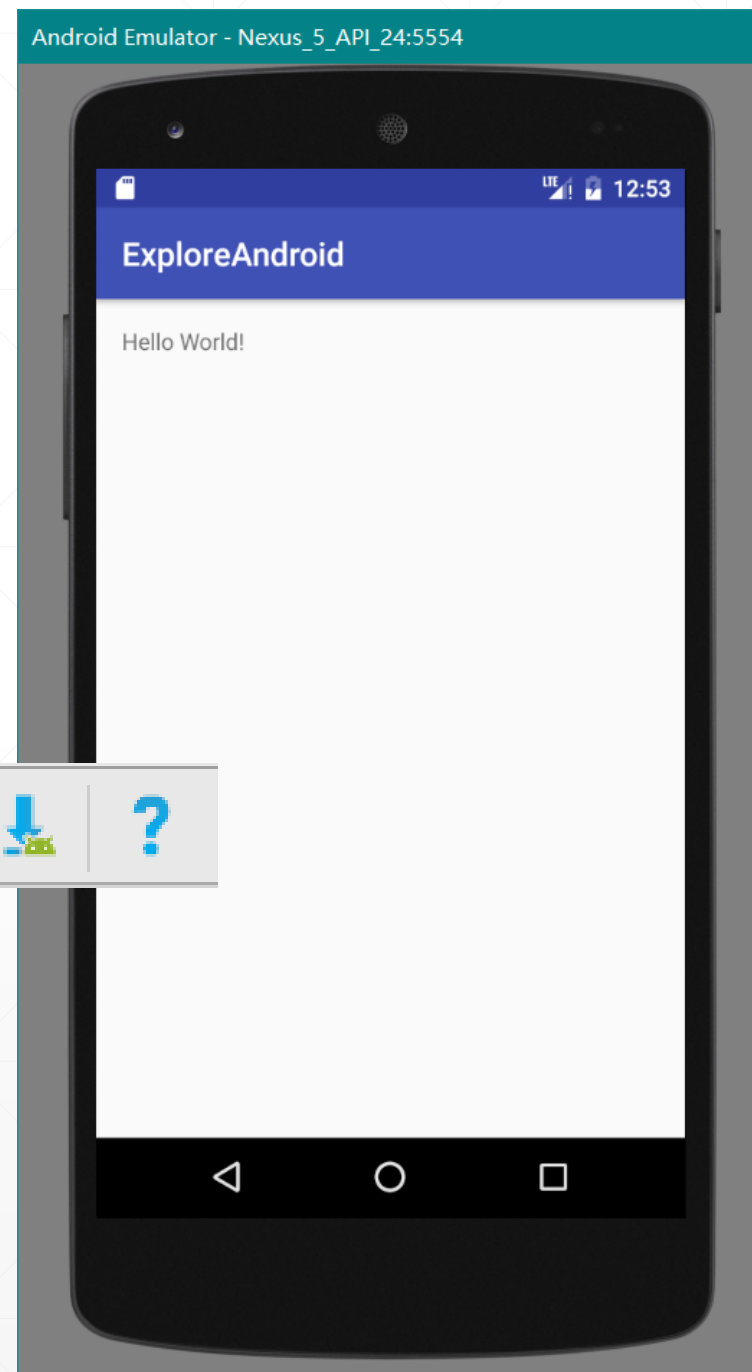


点击“绿色三角”按钮，
选择已运行的虚拟机，
再点击“OK”

等待Gradle构建项目，生成调试用的APK，再部署到虚拟机上，并自动运行之.....



点此结束示例程序的运行，虚拟机重新显示桌面

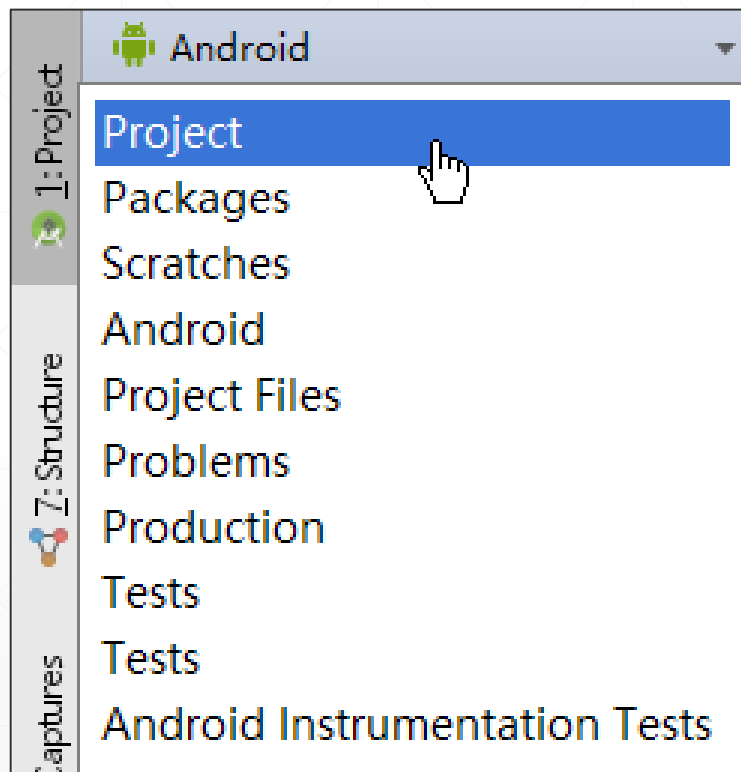


至此，使用Android Studio的一个完整的开发过程展示结束！

下面，我们展示一些Android Studio的使用技巧。

使用项目窗口（ Project Window ）

Project Window用于展示项目成员

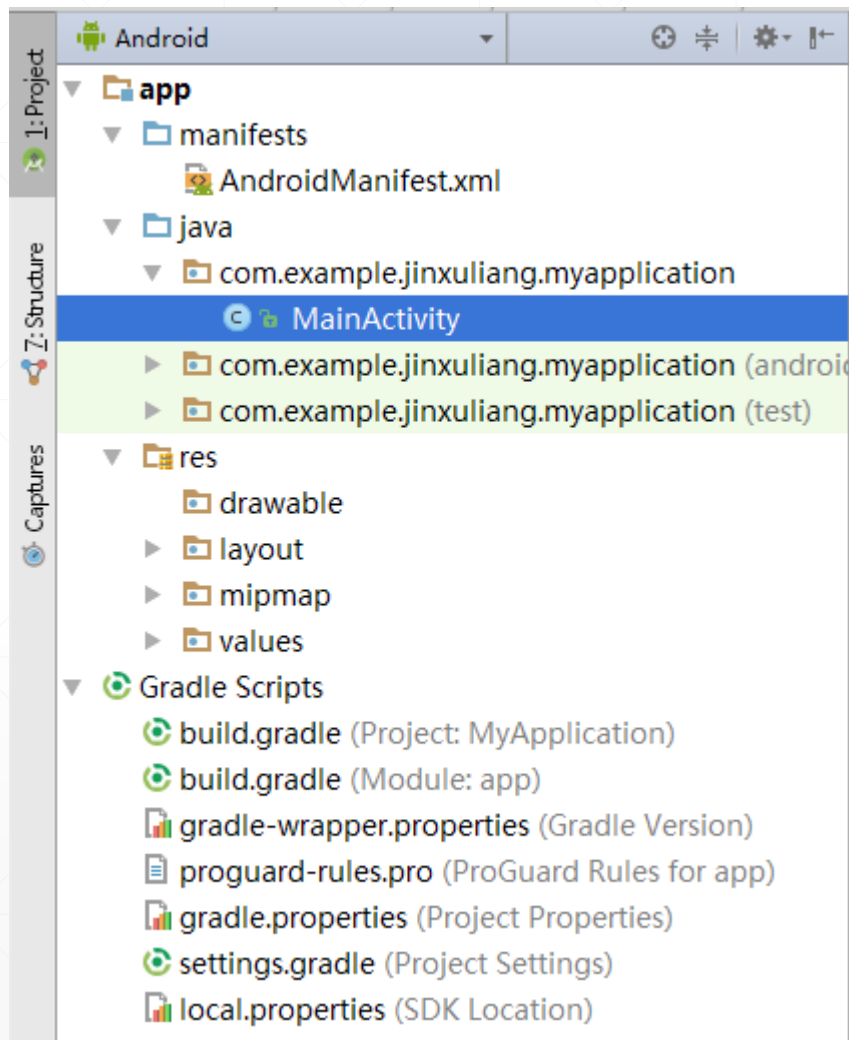


点击顶部下拉框，可以
切换视图

每一种视图，其实是以不同的
方式呈现项目中的各种成员

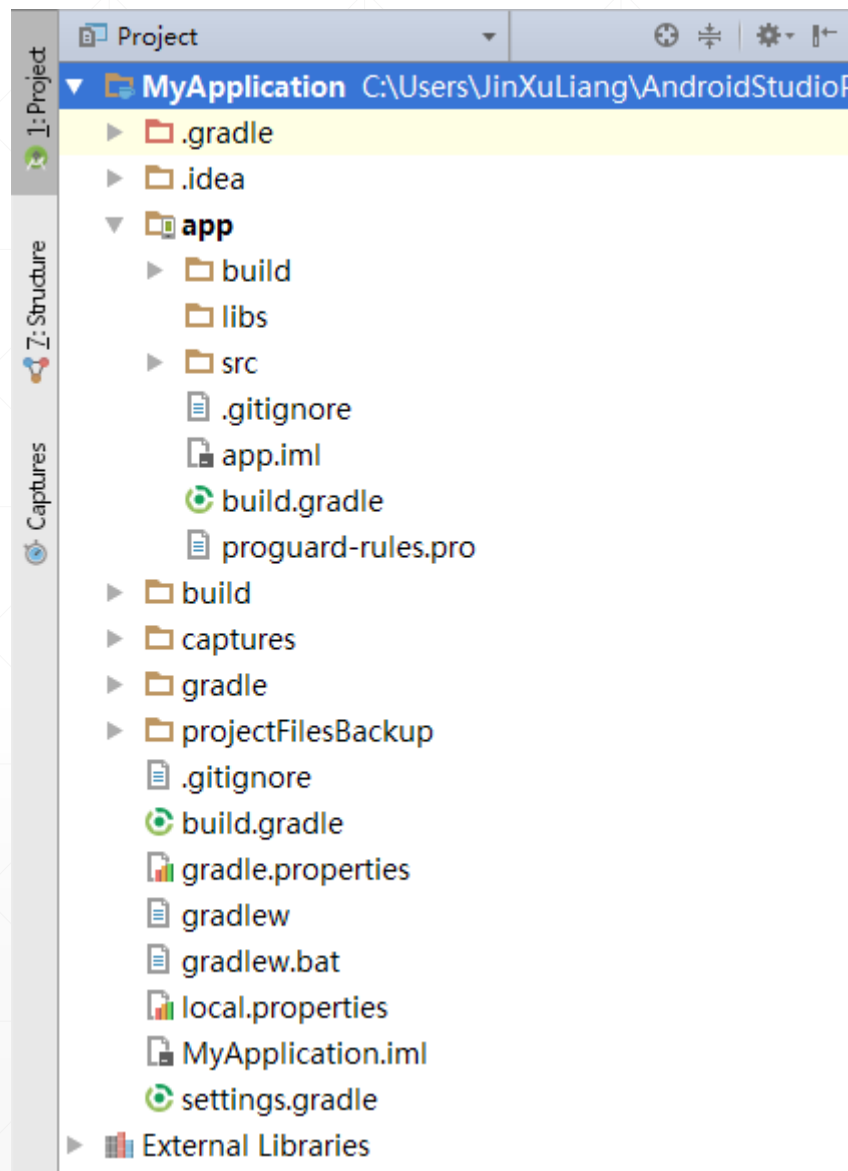
Android视图

这是最常用的视图，隐藏了日常开发很少用到的文件，将常用的文件分类以方便查找和访问。



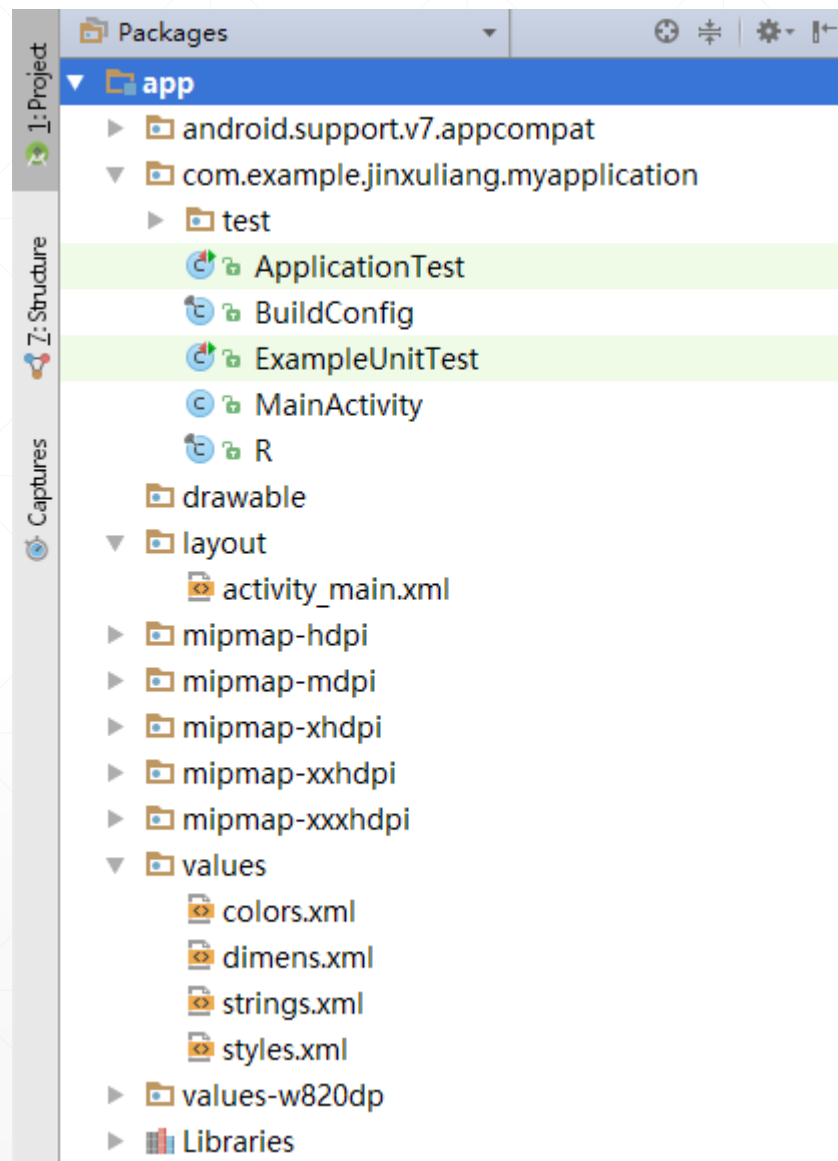
Project视图

展示真实的目录结构，与Windows资源管理器中看到的一样。

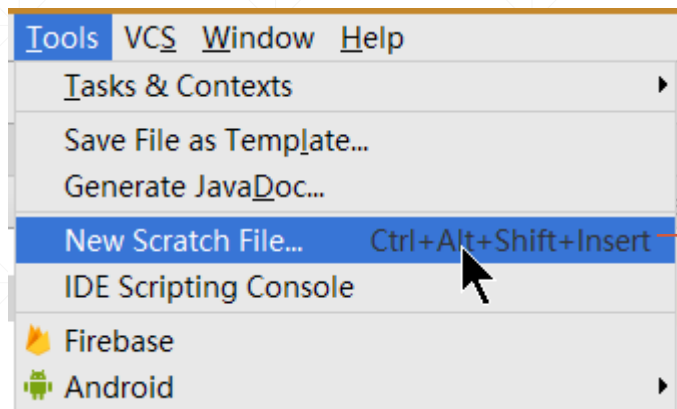


Packages视图

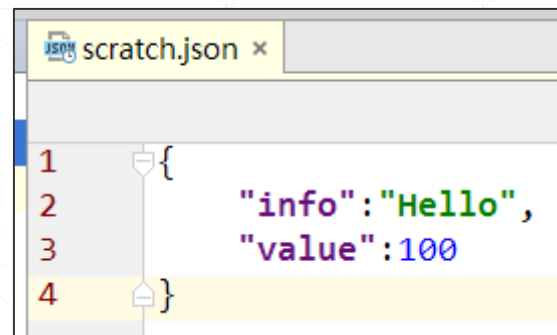
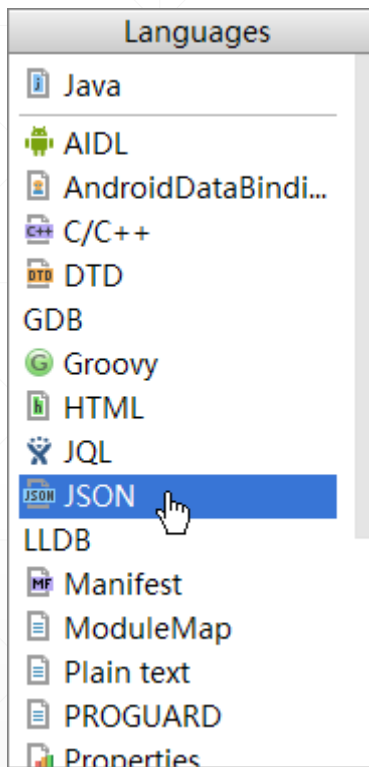
展示出项目中的所有包，第三方库和组件，以及相关的资源。



Scratches视图，可用于暂时保存一些代码

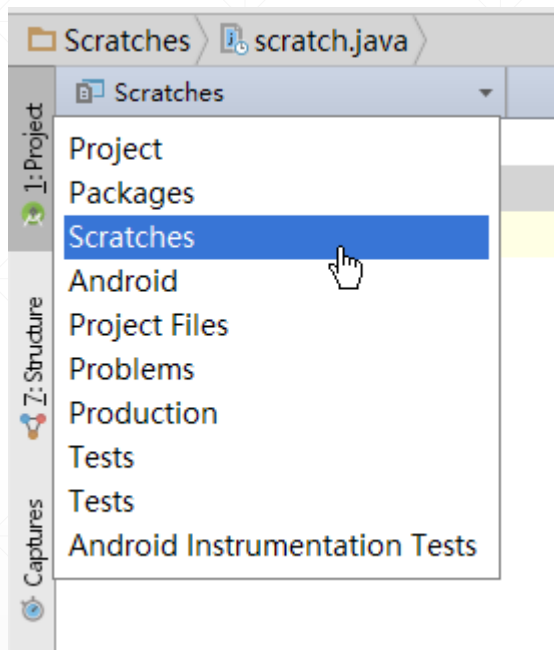


选择代码类型

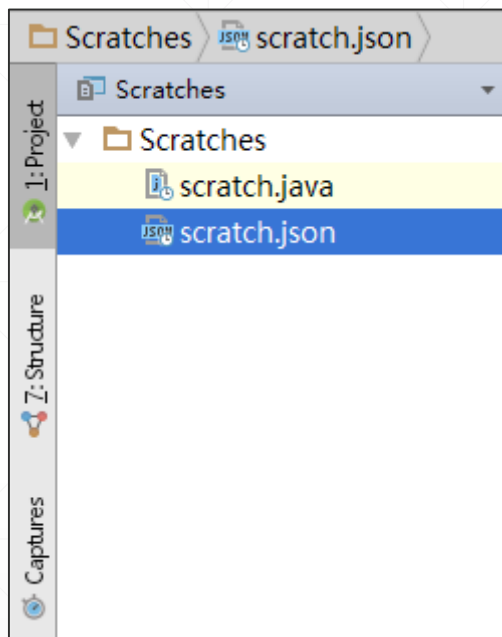


在打开的编辑窗口中输入或粘贴代码

使用Scratches视图



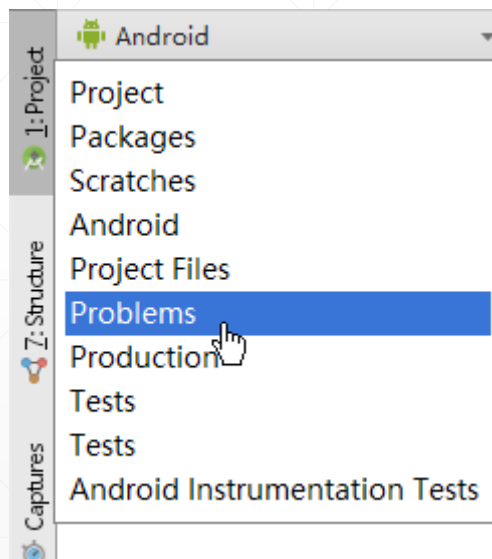
切换显示Scratches视图



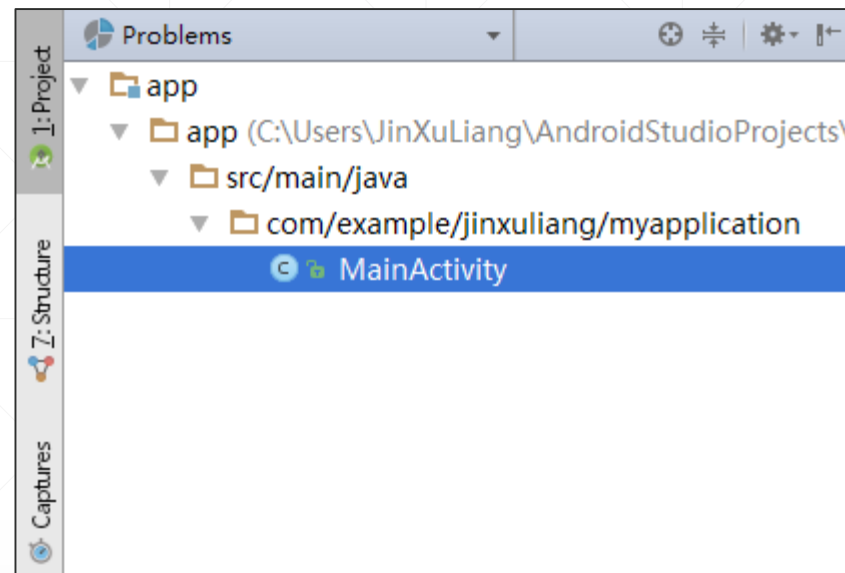
Scratches视图中展示了所有代码片断，可以随时新建和删除之

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        Toast.makeText(this, "中国人", Toast.LENGTH_LONG).show();  
        nouse();  
    }  
}
```

代码中有错



切换到Problems视图



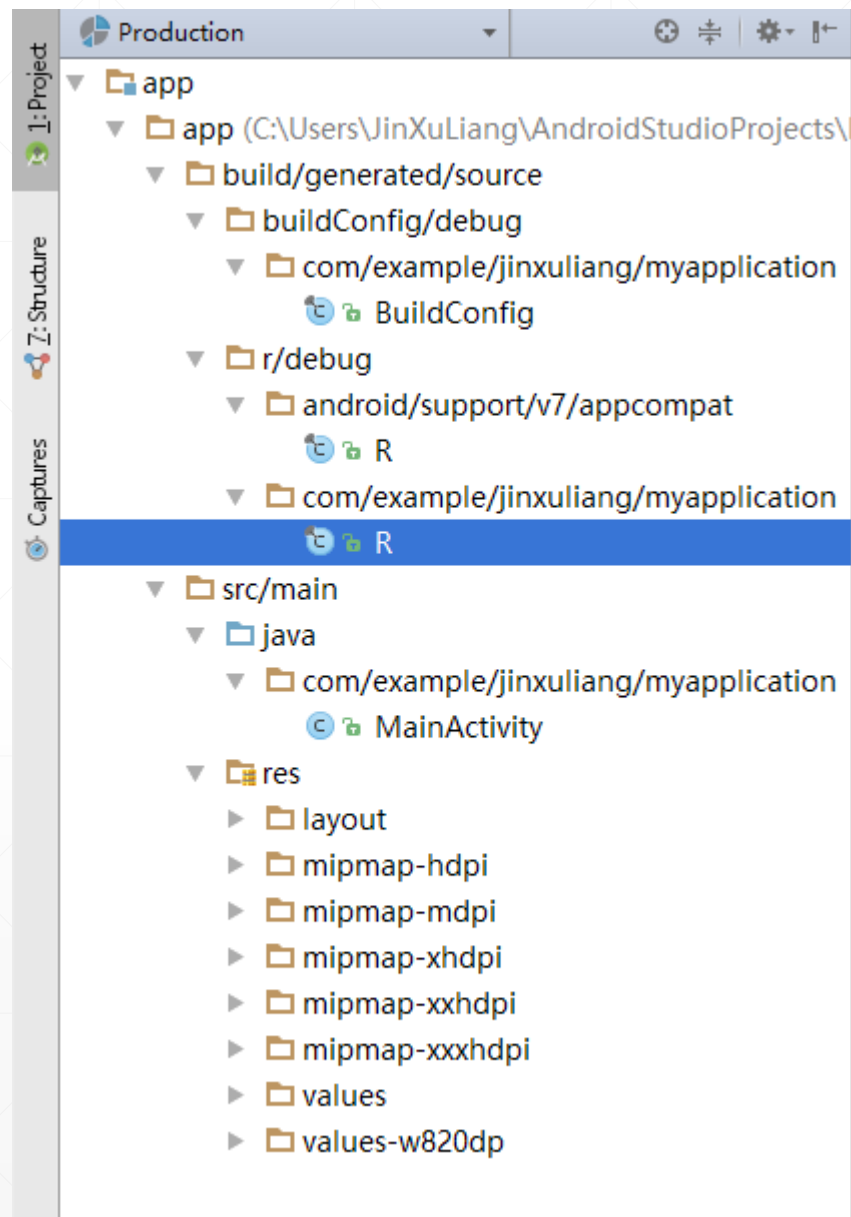
自动列出所有有错的文件，双击打开.....

使用Problems视图

Production视图

可以用于查看IDE自动生成和管理的一些文件（比如R.java）

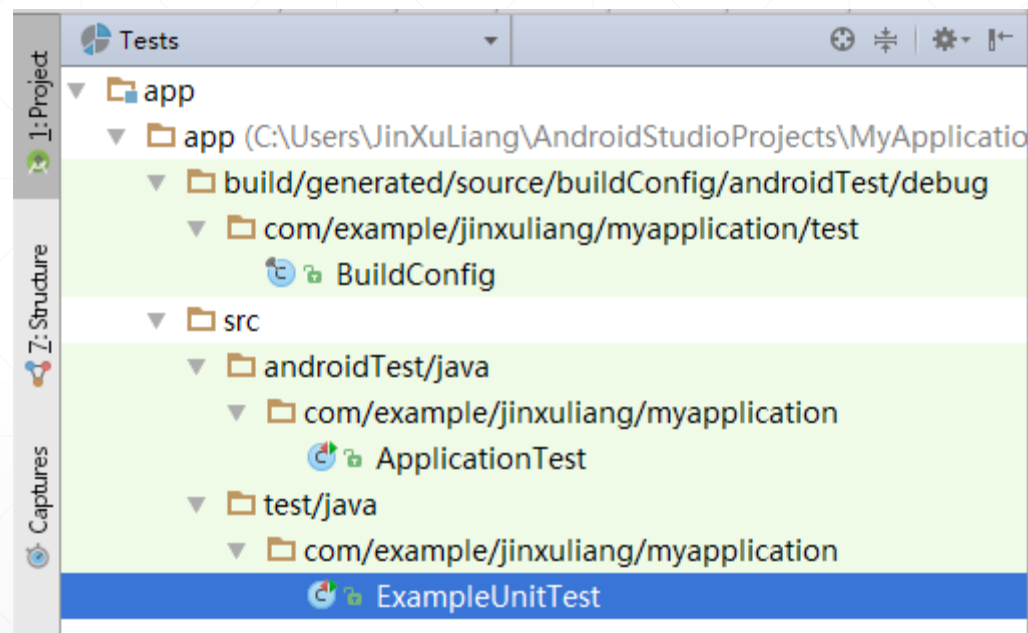
这些文件在发布时将会被打包到Apk中。



Tests视图

可以用于运行单元测试

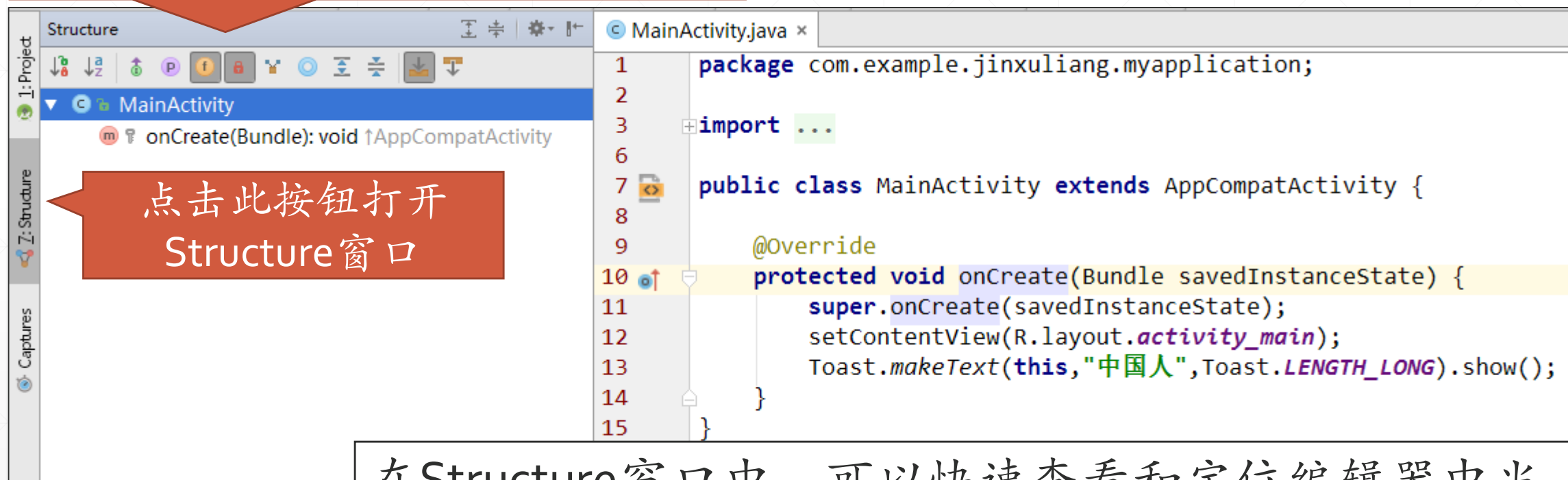
单元测试分为两类，一类是使用JUnit针对纯Java SE代码编写的测试，另一类是针对Android特定代码的测试，需要使用Android SDK所提供的特定测试组件。



常用操作与开发技巧

查看代码全貌

工具栏上的这些按钮，可以过滤要显示的成员，比如只显示public成员



在Structure窗口中，可以快速查看和定位编辑器中当前打开文件中的成员

快速查找特定的文件

Search Everywhere Double Shift

Project View Alt+1

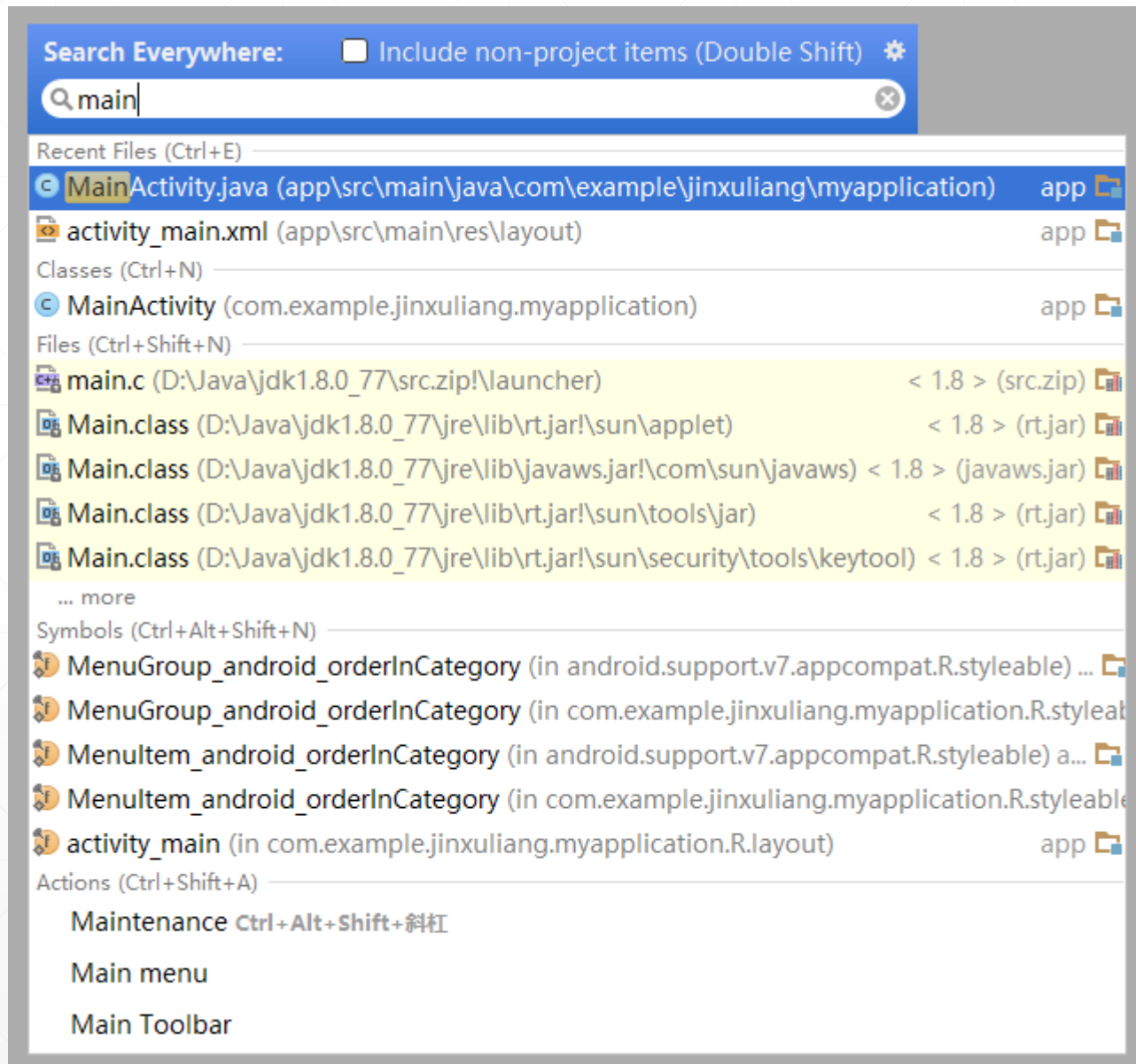
Go to File Ctrl+Shift+N

Recent Files Ctrl+E

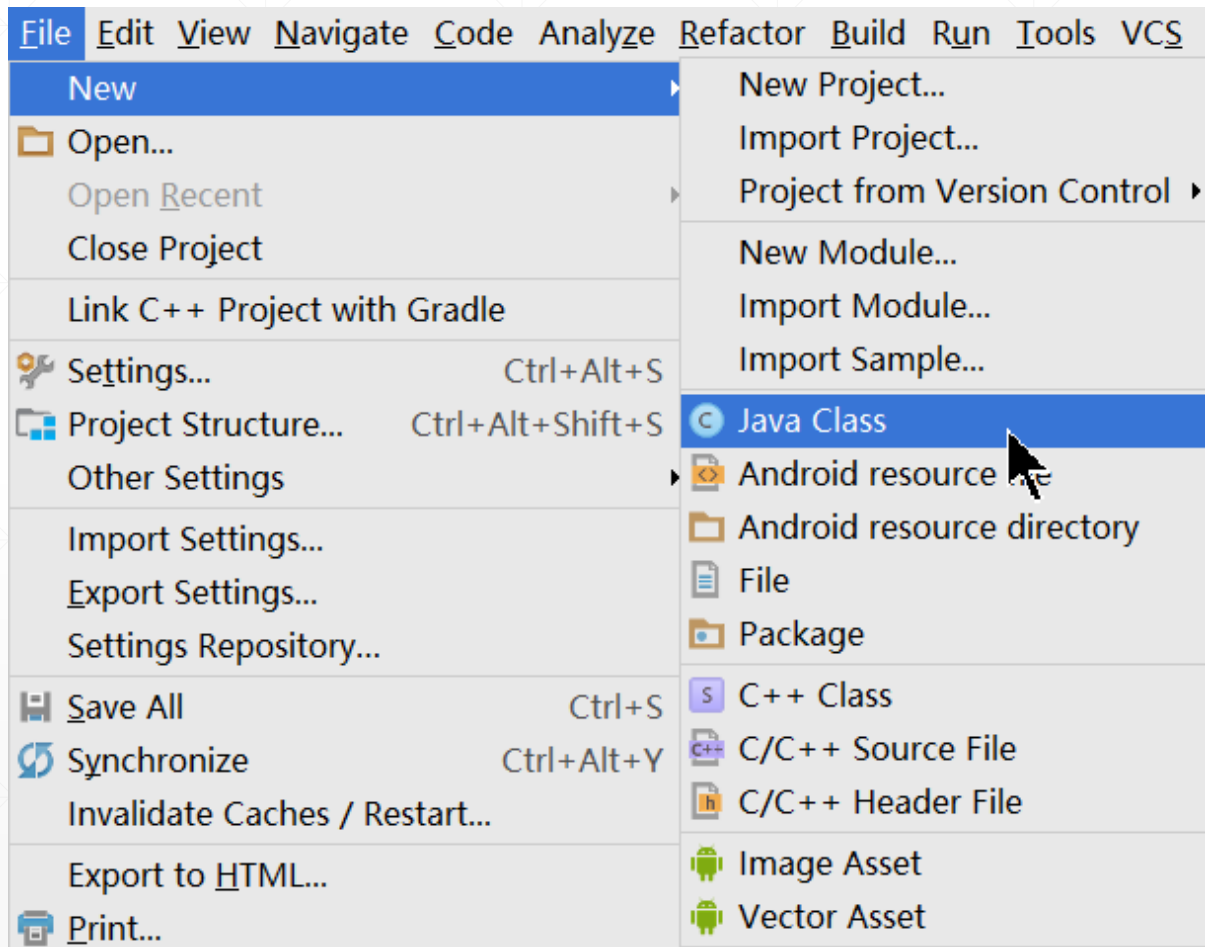
Navigation Bar Alt+Home

Drop files here to open

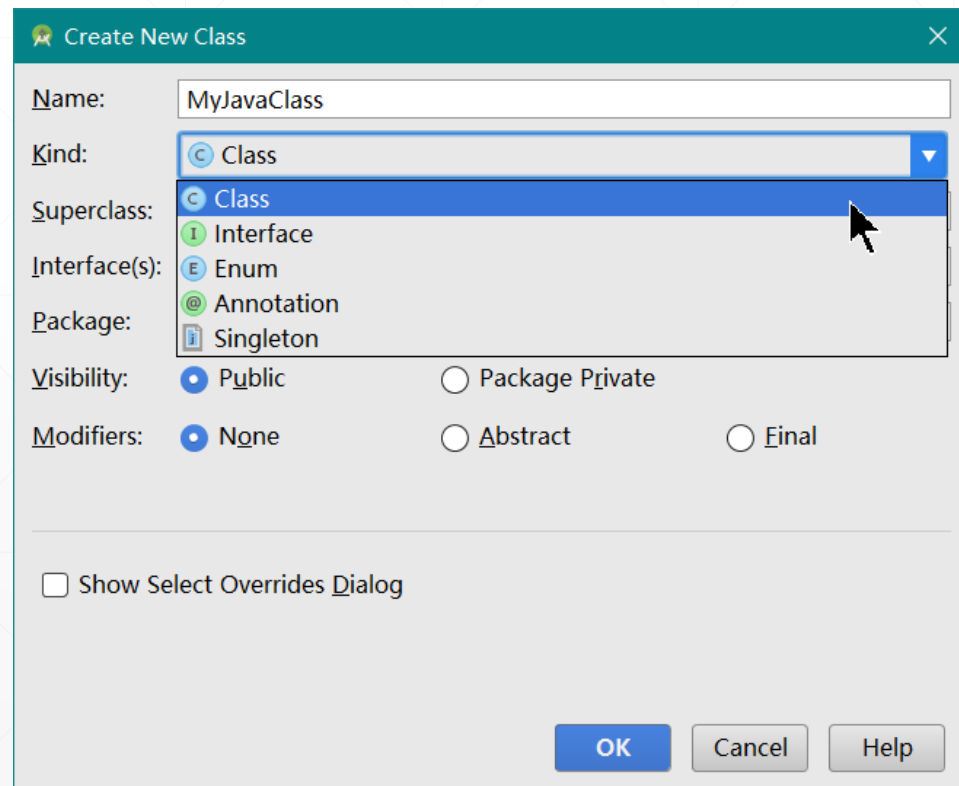
快速连按两下Shift键，
可以查找文件、类等



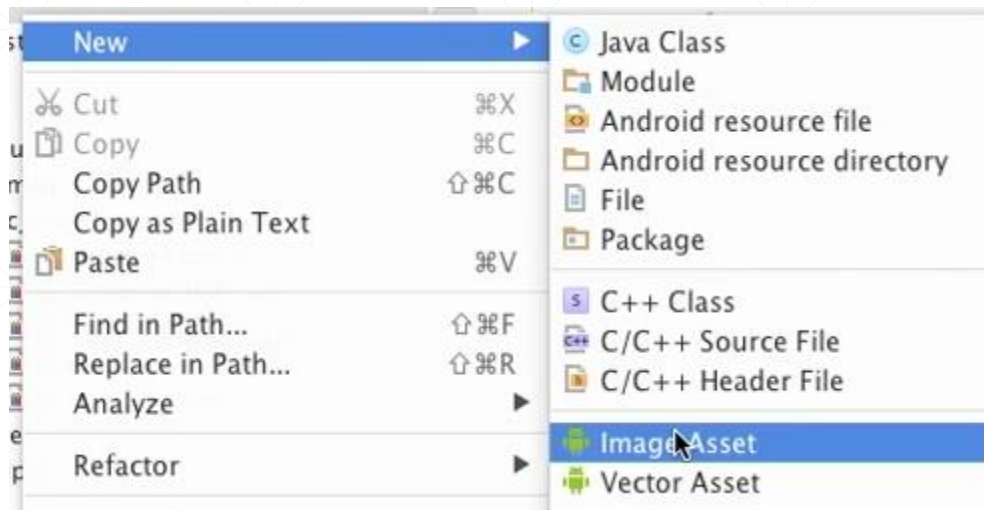
创建各种开发组件



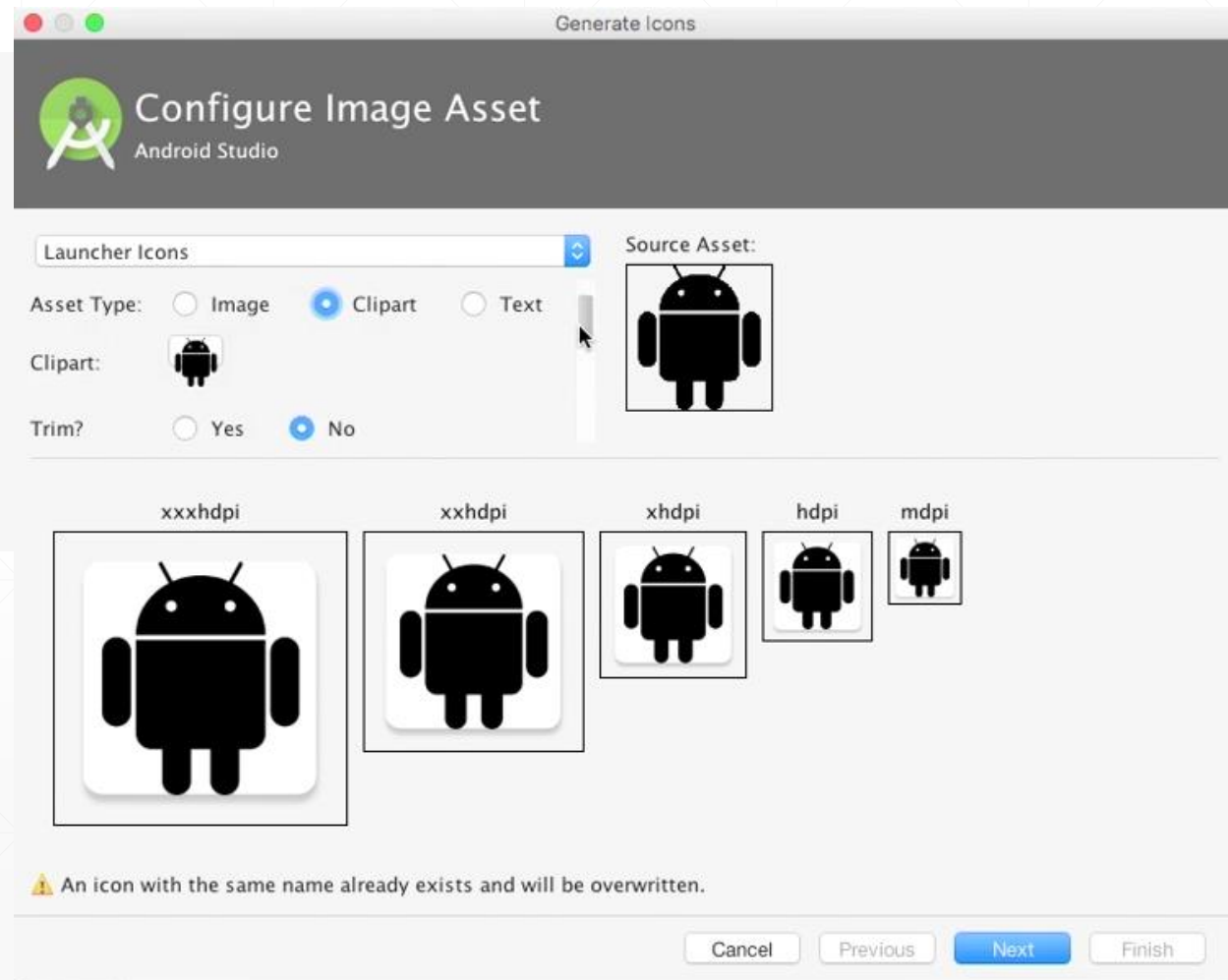
在File/New菜单中，包容
创建各种开发组件的命令

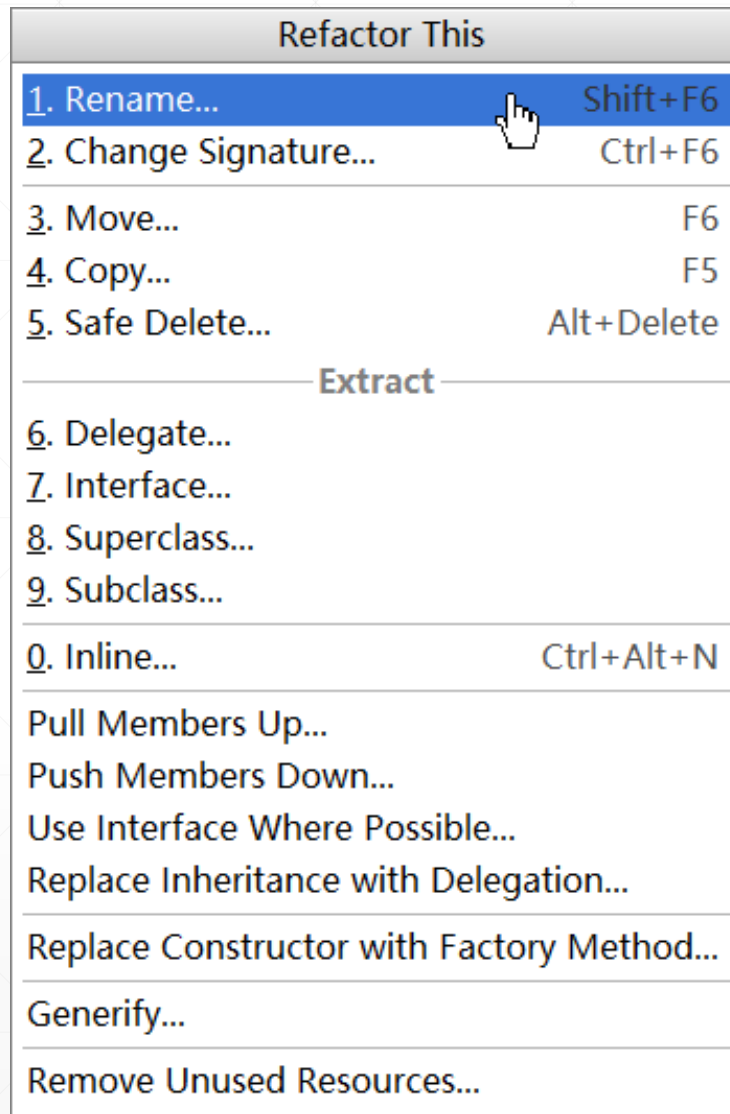
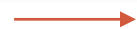
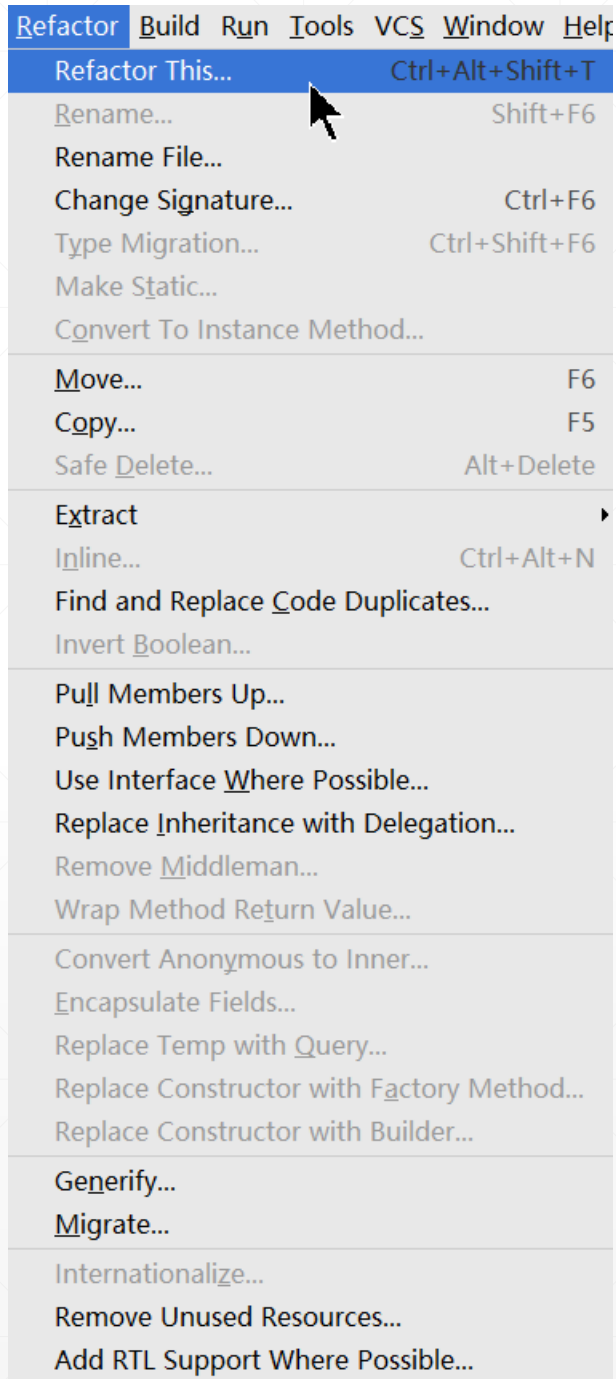


创建程序启动图标



使用Image Asset，可以创建显示在手机上的App图标





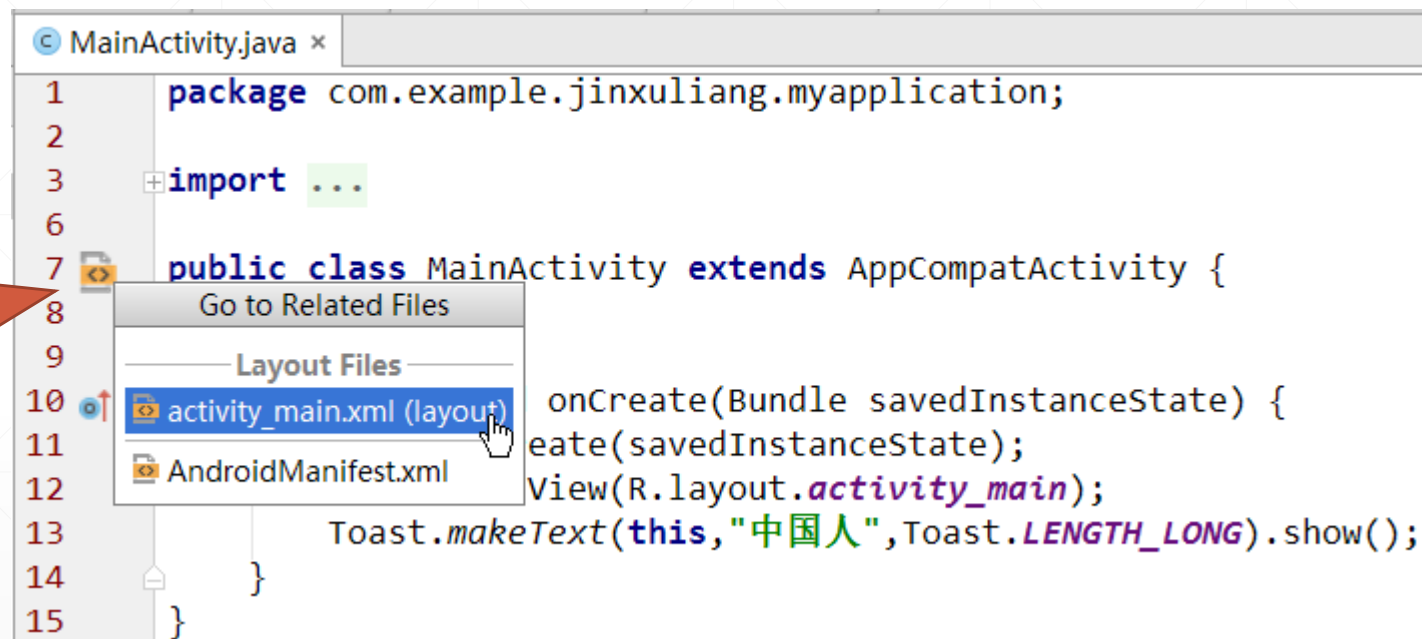
重构代码

Android Studio 集成了诸多重构功能，使用方便。

将光标放到代码中的特定位置，Ctrl+Alt+Shift+T，可以调出重构快捷菜单，集中了最常用的重构功能

切换到相关的文件

点击这个图标，可以快速地切换显示相关的文件



使用内置代码模板

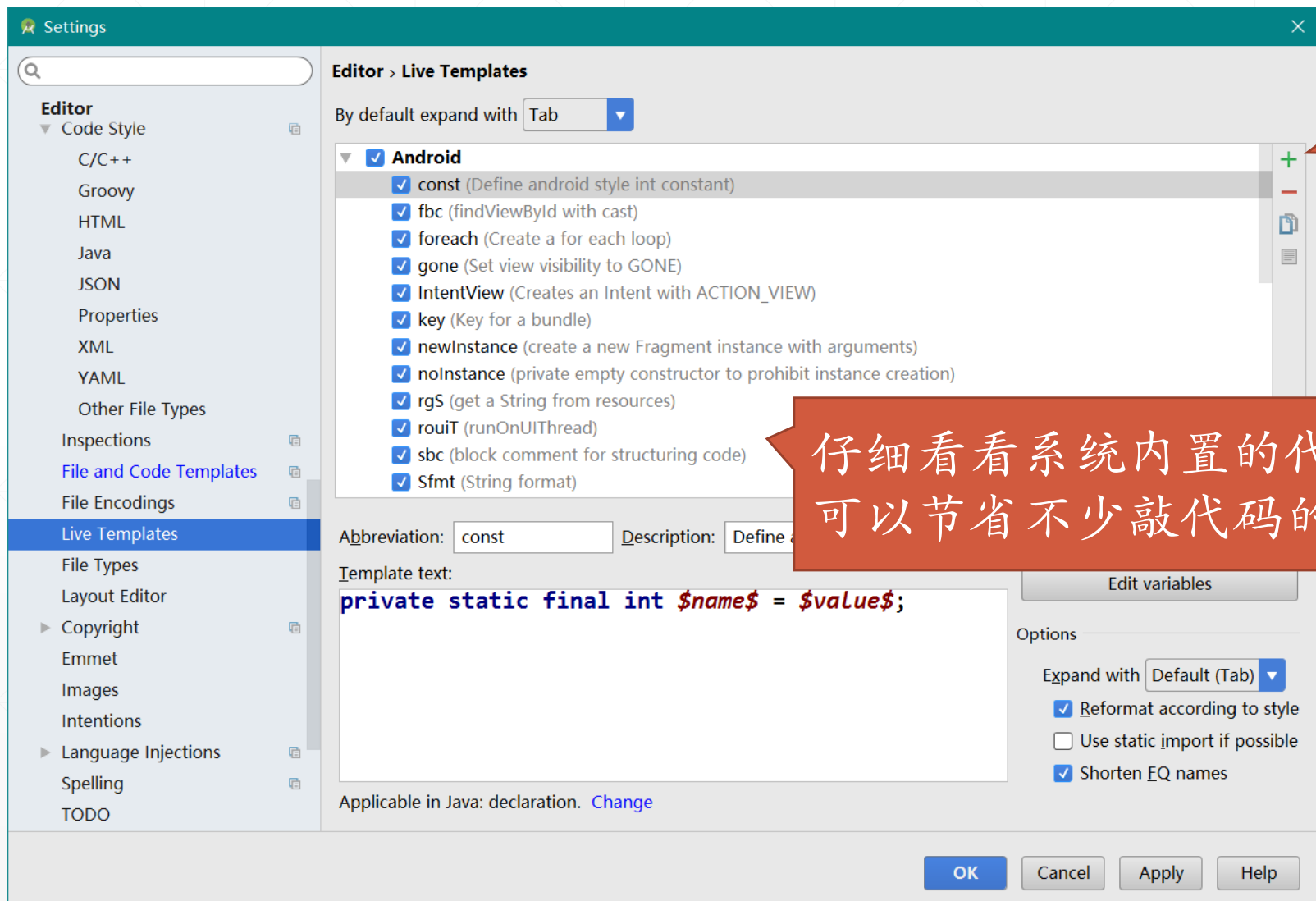
在代码中输入“const”再按Tab键，
会自动生成Java中的常量声明

```
public class MyJavaClass {  
    con  
    const Define android style int co...  
    return x+y;  
}  
}
```

自动生成的代码框架

```
public class MyJavaClass {  
    private static final int MYCONST=481;  
    public int add(int x,int y){  
        return x+y;  
    }  
}
```

查看Android Studio的内置代码模板

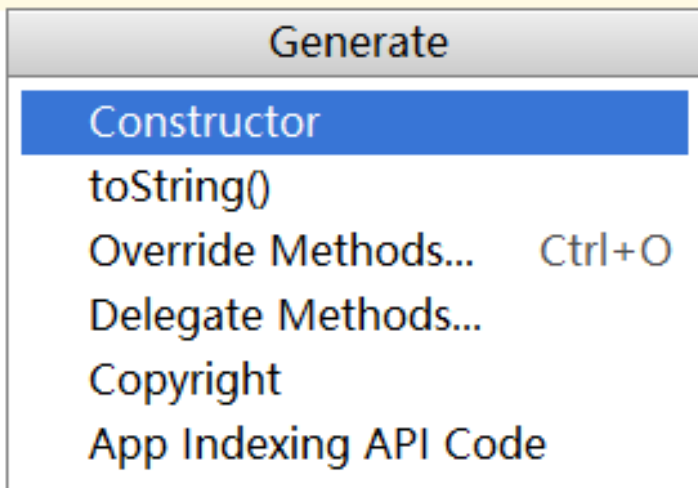


你也可以创建自己的代码模板

仔细看看系统内置的代码模板，
可以节省不少敲代码的时间

调出辅助代码编写快捷菜单

在代码空白处，按Alt + Insert键：

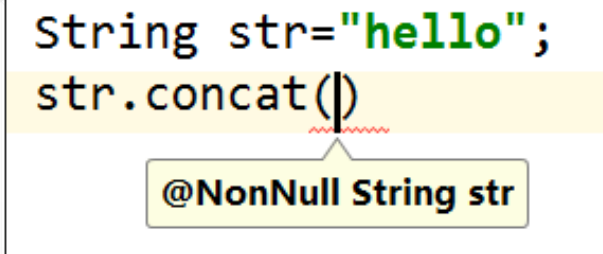


使用这个快捷菜单，可以快速
地创建类的构造方法、重写基
类的特定方法.....，很方便

一些有用的快捷键

✓ 选中多条语句，按 `ctrl + /`，可以批量地给它们注释或取消注释

✓ 立即显示智能提示：`ctrl + p`

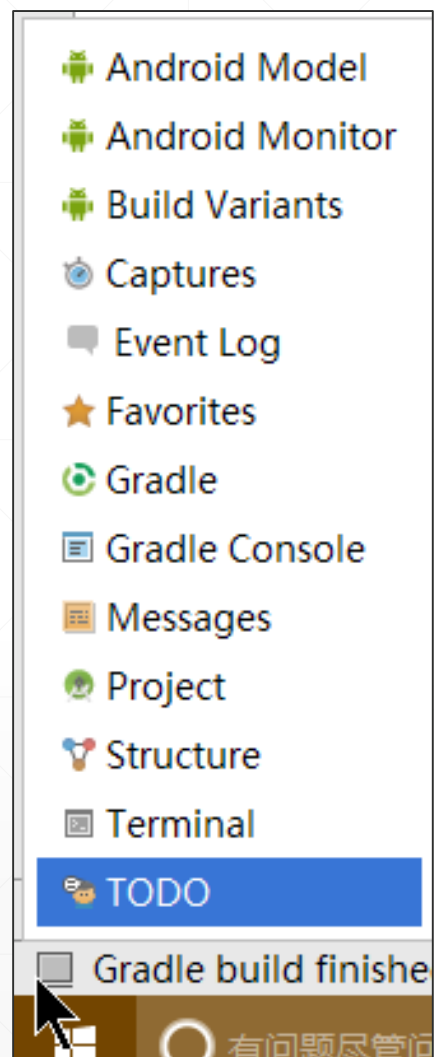


✓ 当代码格式混乱时，`Ctrl + Alt + L`，自动重新格式化代码。

✓ 上下移动整行或整个方法：`ctrl + shift + 上下光标键`

开发中常用窗口简介

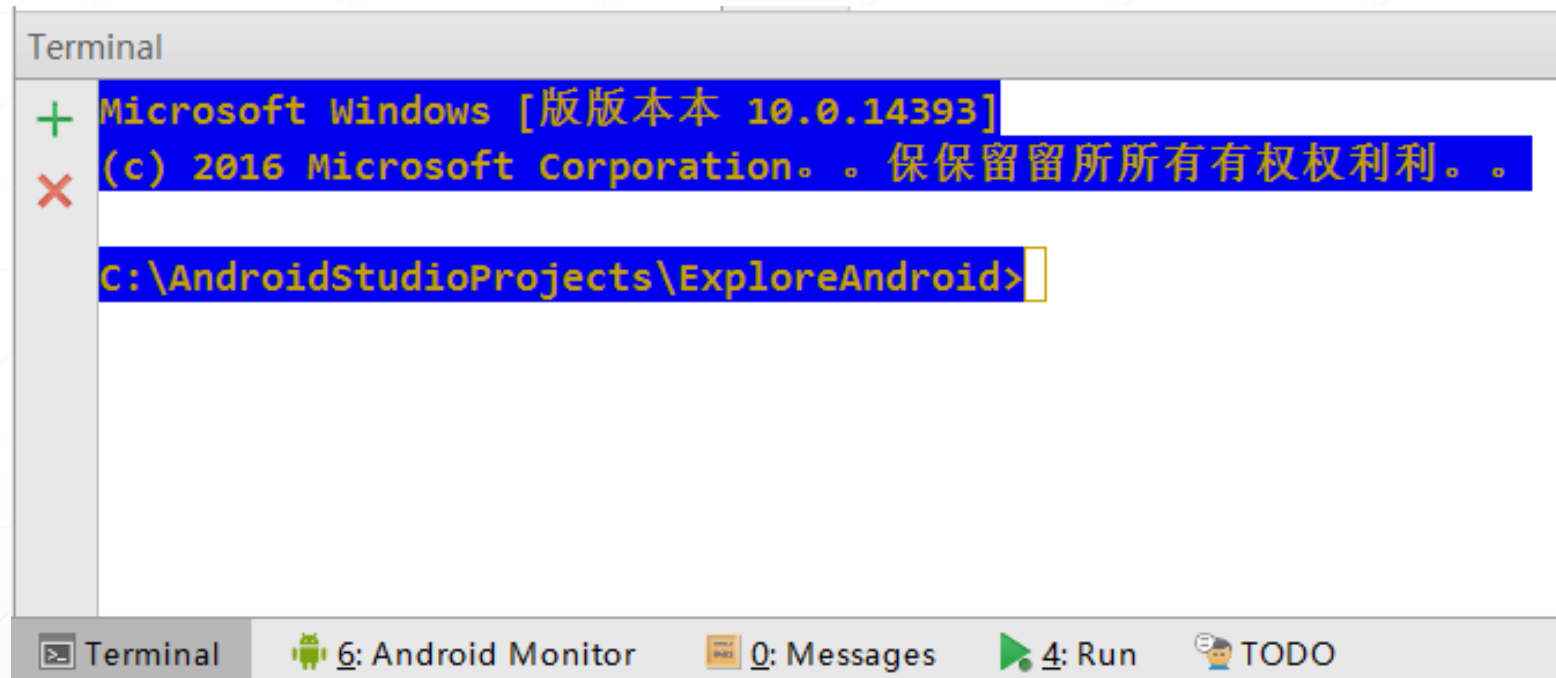
切换窗口



每一个窗口，都有特定的用途，下面我们介绍一些常用的窗口。

鼠标移到左下角的灰色方块，会弹出一个菜单，可以用于切换窗口

终端窗口



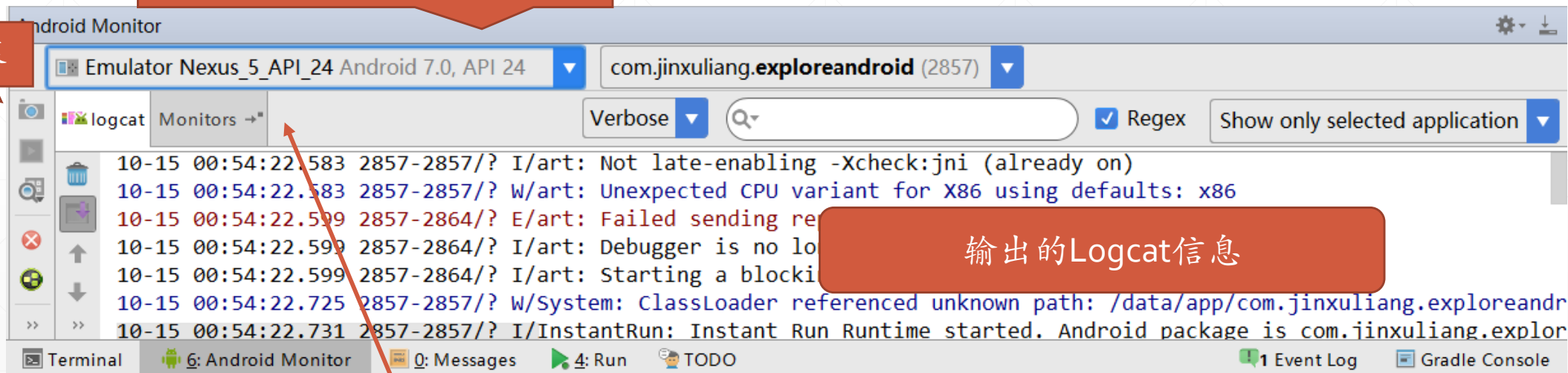
可以用于键入一些命令，与Window命令提示窗口用法一致，但当前版本的Android Studio在Windows 10下显示有BUG。

Monitor窗口

最重要的窗口，可以查看Log信息，占用内存，网速等，还可以截屏.....

列出当前所连接的手机，
从中选一个查看

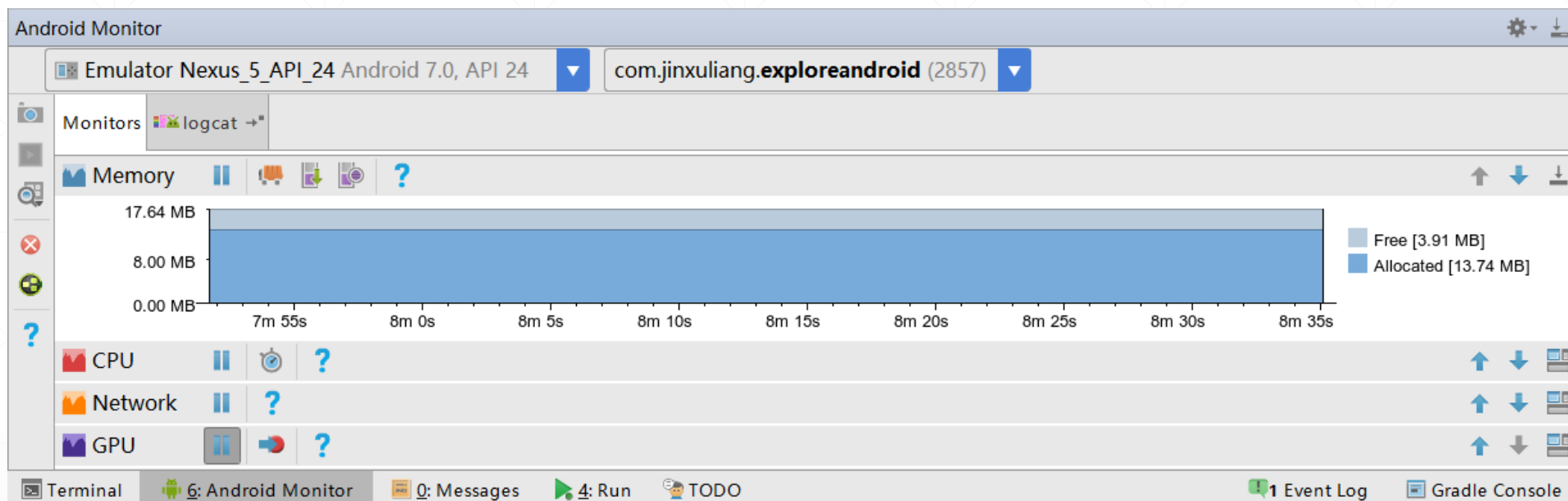
截手机屏幕



输出的Logcat信息

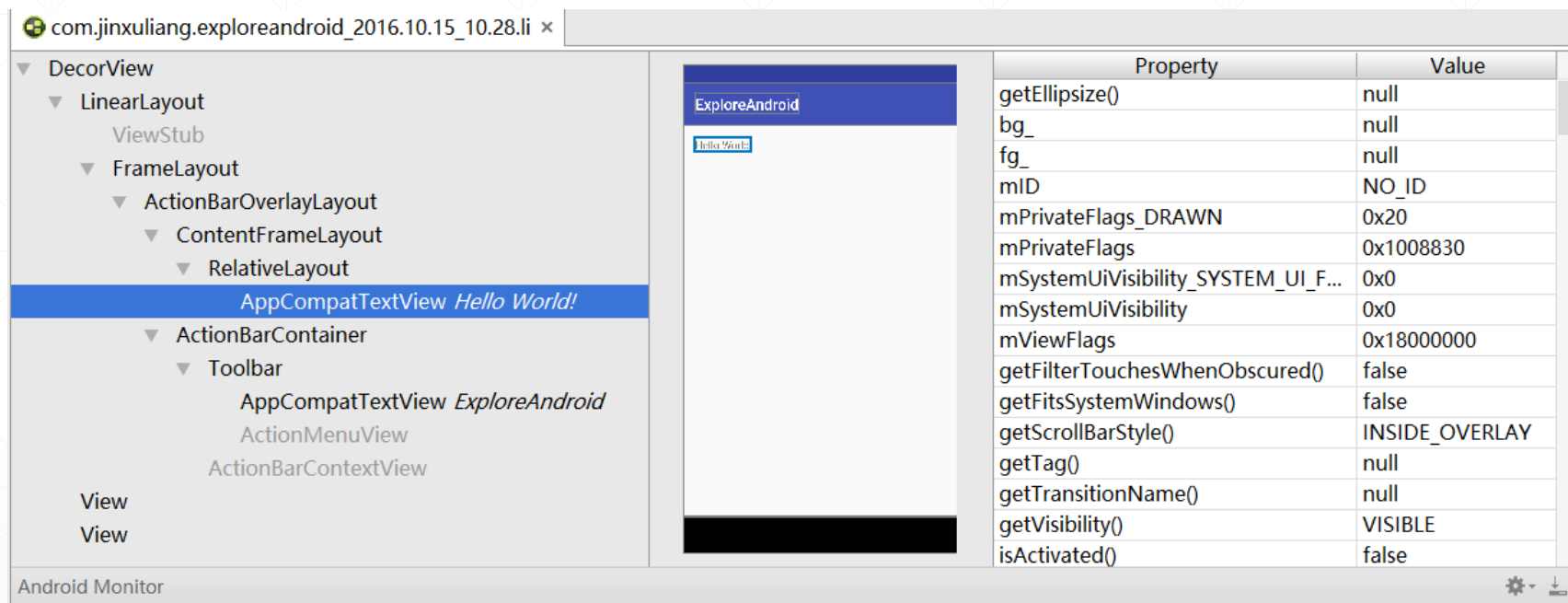
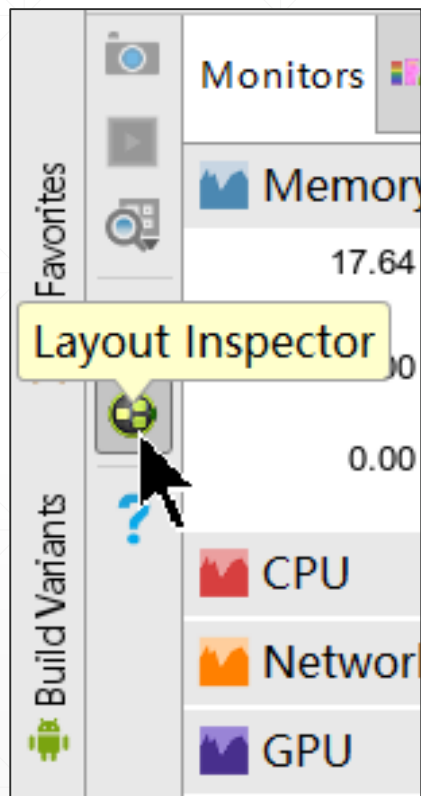
点击Monitor，查看内存占用情况等相关信息.....

Monitors选项卡



可以用于查看内存占用，CPU、网络 and GPU 的使用情况，
对于程序性能优化很有用。

Layout Inspector

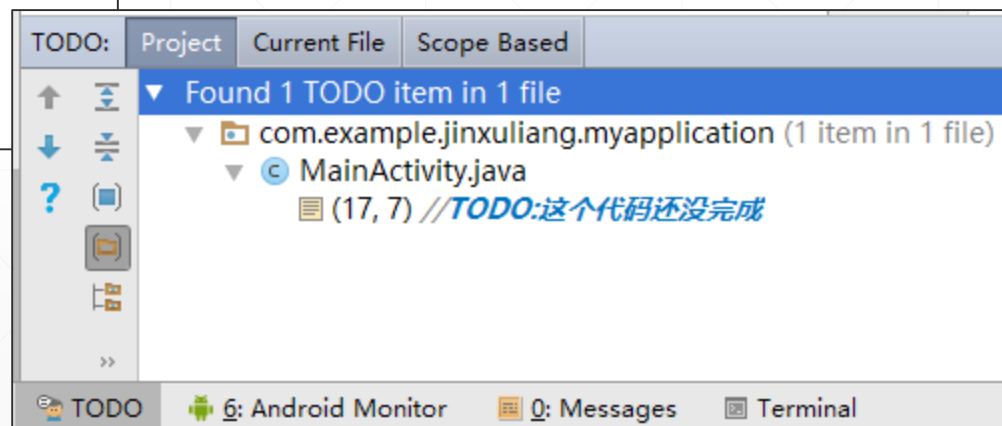


可以使用Layout Inspector查看真实的组件嵌套情况，从而优化界面装载时间。

TODO窗口

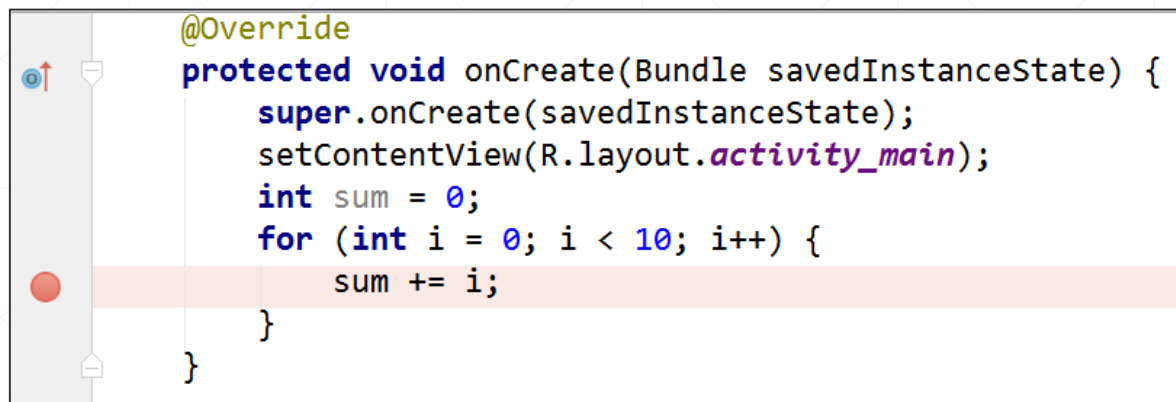
```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {...}  
  
    //TODO: 这个代码还没完成  
    @Override  
    protected void onStart() {  
        super.onStart();  
    }  
}
```

使用TODO窗口显示插入在代码中的“便签条”，以提醒自己完成特定的工作。



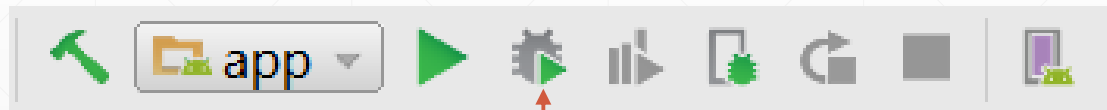
程序调试方法

在左部空白的地方点击，可以设置断点



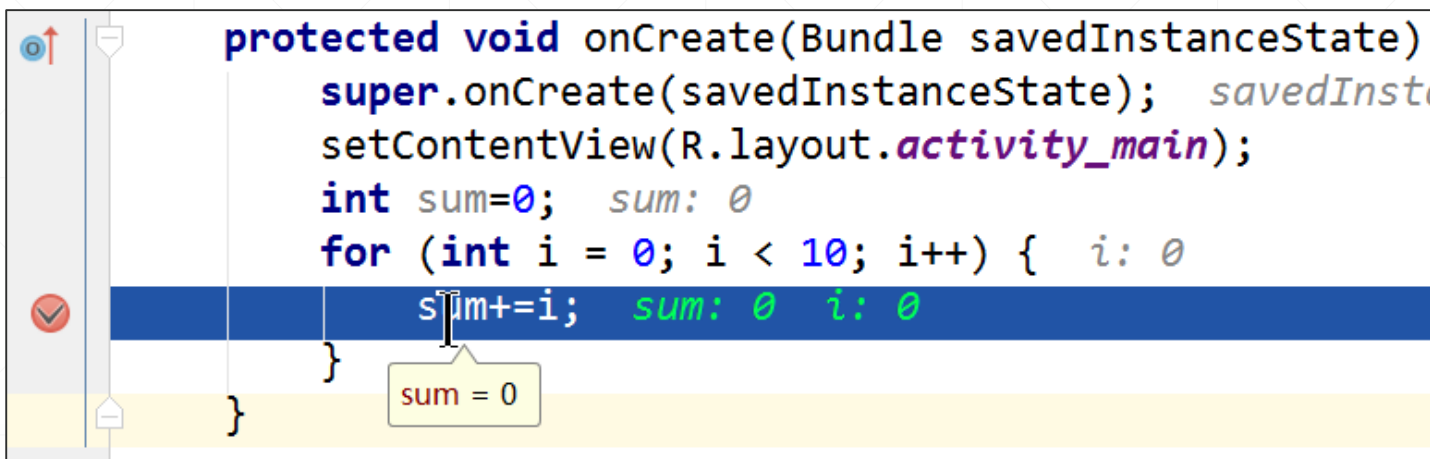
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    int sum = 0;
    for (int i = 0; i < 10; i++) {
        sum += i;
    }
}
```

The image shows a code editor window with a Java snippet. A red circle breakpoint is placed on the left margin next to the line `sum += i;`. The code is as follows:



点击此按钮启动
调试工作

暂停的程序

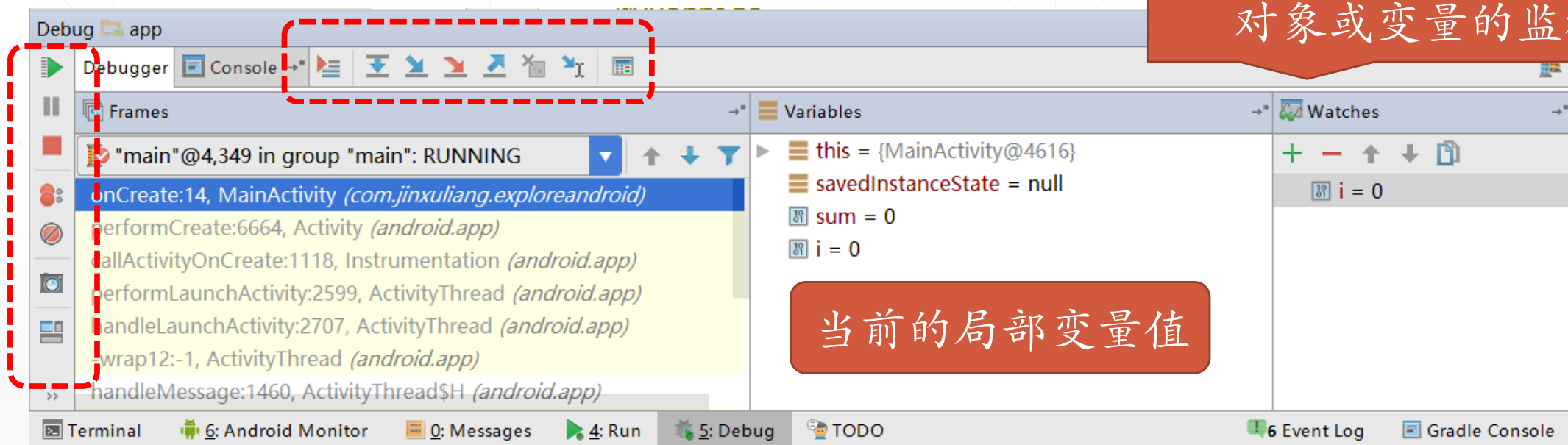


程序运行到断点处暂停，鼠标移到变量上，可以显示出变量的值

使用Debug窗口

完成单步执行等功能

点击“+”，添加对特定对象或变量的监视



继续与停止调试，查看断点等功能