**前言**

　　废话不多说，以下针对JDK正确安装（及其环境变量配置完毕，即Java开发环境下）、Android Studio的安装，配置，以及创建工程、主题字体更换、窗口工具、布局、快捷方式等的基本使用逐一说明。

**安装java**

　　下载[Java](http://lib.csdn.net/base/javaee" \o "Java EE知识库" \t "_blank)安装包（jdk，网上有很多下载地址，最好去官网下：[https://www.java.com/zh\_CN/](https://www.java.com/zh_CN/" \t "_blank)），安装完后记得配置环境变量；

* + 在“系统变量”新建一个变量名为JAVA\_HOME的变量，变量值为你本地java的安装目录，我这里为：C:\Program Files\Java\jdk1.7.0\_80，设置这个的目的是作为下面两个环境变量的一个引用
  + 在“系统变量”选项区域中查看PATH变量，如果不存在，则新建变量PATH，否则选中该变量，单击“编辑”按钮，在“变量值”文本框的起始位置添加“%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;”
  + 在“系统变量”选项区域中查看CLASSPATH变量，如果不存在，则新建变量CLASSPATH，否则选中该变量，单击“编辑”按钮，在“变量值”文本框的起始位置添加“.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar;”。

　　成功安装判断方法：

键入命令“java -version”，“java”，“javac”几个命令，出现画面，说明环境变量配置成功；

　　说明：

* + JAVA\_HOME：该环境变量的值就是 Java 所在的目录，一些 Java 版的软件和一些 Java 的工具需要用到该变量，设置 PATH 和 CLASSPATH 的时候，也可以使用该变量以方便设置。
  + PATH：指定一个路径列表，用于搜索可执行文件的。执行一个可执行文件时，如果该文件不能在当前路径下找到，则依次寻找 PATH 中的每一个路径，直至找到。或者找完 PATH 中的路径也不能找到，则报错。Java 的编译命令 (javac)，执行命令 (java) 和一些工具命令 (javadoc, jdb 等) 都在其安装路径下的 bin 目录中。因此我们应该将该路径添加到 PATH 变量中。
  + CLASSPATH：也指定一个路径列表，是用于搜索 Java 编译或者运行时需要用到的类。在 CLASSPATH 列表中除了可以包含路径外，还可以包含 .jar 文件。Java 查找类时会把这个 .jar 文件当作一个目录来进行查找。通常，我们需要把 JDK 安装路径下的 jre/lib/rt.jar ([Linux](http://lib.csdn.net/base/linux): jre/lib/rt.jar) 包含在 CLASSPATH 中。
  + PATH 和 CLASSPATH 都指定路径列表，列表中的各项 (即各个路径) 之间使用分隔符分隔。在 Windows 下，分隔符是分号 (;)，而在 Linux 下，分隔符是冒号 (:)。
  + 注意，在 CLASSPATH 中包含了一个“当前目录 (.)”。包含了该目录后，就可以到任意目录下去执行需要用到该目录下某个类的 Java 程序，即使该路径并未包含在 CLASSPATH 中也可以。原因很简单：虽然没有明确的把该路径包含在 CLASSPATH 中，但 CLASSPATH 中的 “.” 在此时就代表了该路径。

**Android SDK下载和安装**

　　本地已有合适版本Android SDK，则无需再下载，或者可以使用SDK Manager更新SDK；

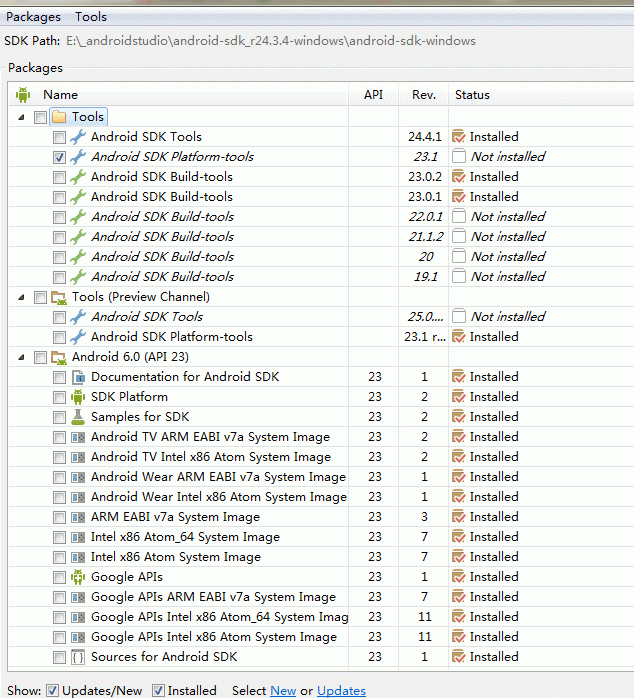
　　没有SDK，则需要下载。

　　这里说一下使用SDK Manager下载/更新SDK：

　　（1）如果没有安装SDK Manager，可以下载安装，地址 <https://github.com/inferjay/AndroidDevTools#sdk-tools>，下载完毕，直接运行安装就可以，安装过程中会自动匹配本机环境下的jdk；

进入到安装目录下：

　　（2）启动SDK Manager.exe，如下图：



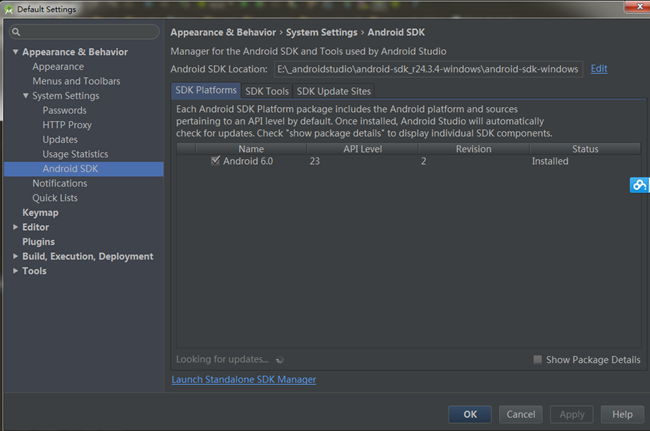
　　（3）更新时间比较长，也有可能更新不了，就得需要配个代理了。

**Android Studio的下载和安装（新建工程和调试运行）**

　　 提供一个国内镜像下载地址[https://github.com/inferjay/AndroidDevTools](https://github.com/inferjay/AndroidDevTools" \t "_blank)

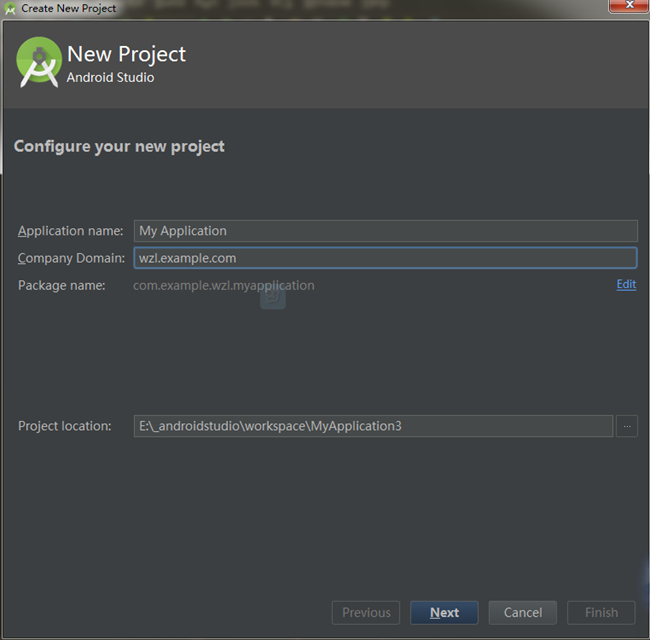
        下载完之后，如果是安装包，可以直接安装即可；如果是解压包，可以解压后，直接运行（安装目录下的studio.exe文件）。

        安装过程中，由于之前已经下载过SDK，所以跳过SDK的安装，一直next知道finish，运行打开Android studio，然后手动配置SDK，点击SDK Manager，配置已经下载好的SDK所在路径即可，如下图：

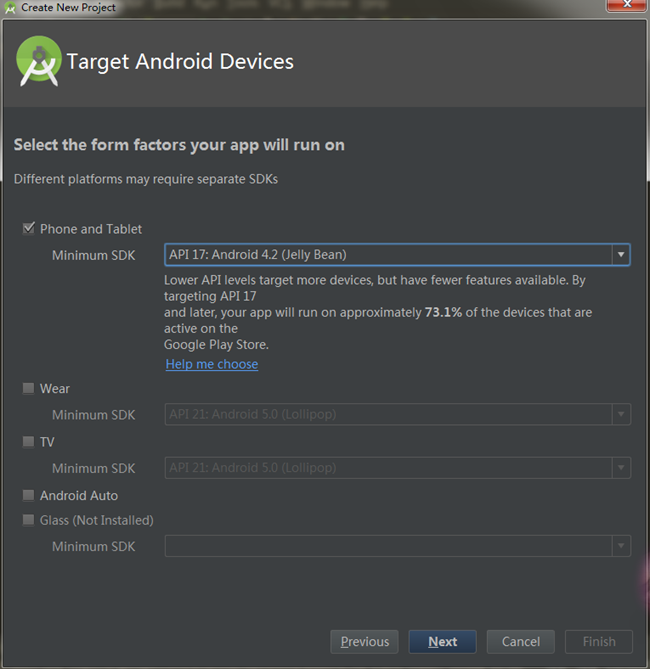


 　　配置好SDK之后就可以新建工程了，如以下系列图：

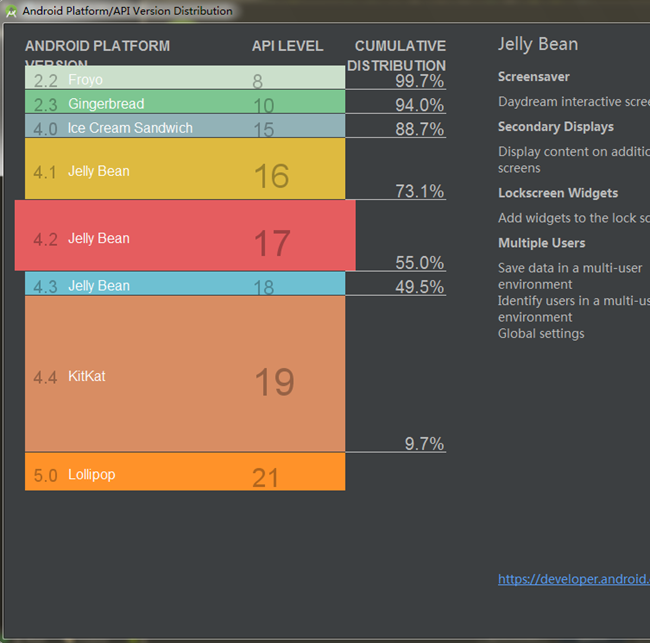
　　（1）file-->new-->new Project:



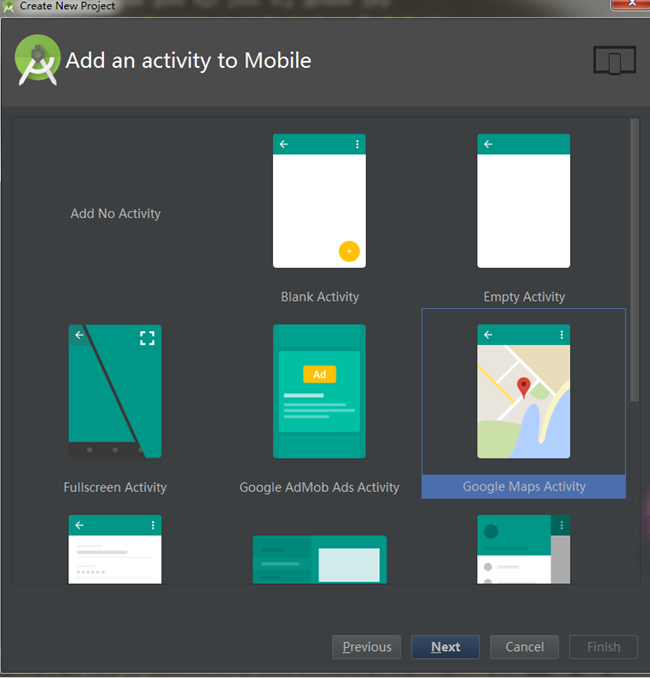
　　（2）next-->可以选择创建的工程的版本-->如果对版不了解，可以点击help me choose:



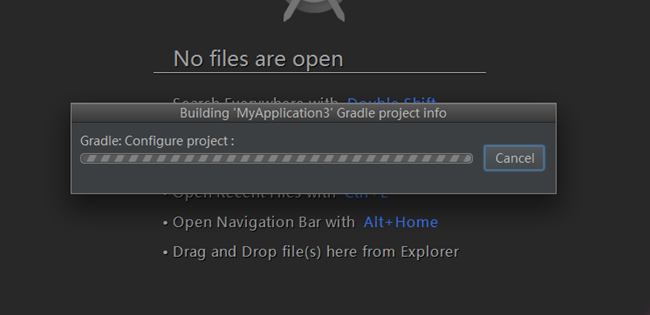
　　（3）点击help me choose 后即可显示当前的Android系统各版本信息，还有什么不明了的呢！（只看界面UI做的就很棒）



　　（4）选择版本后next-->进入选择主Activity样式界面，下面只是列出几种，样式很多：

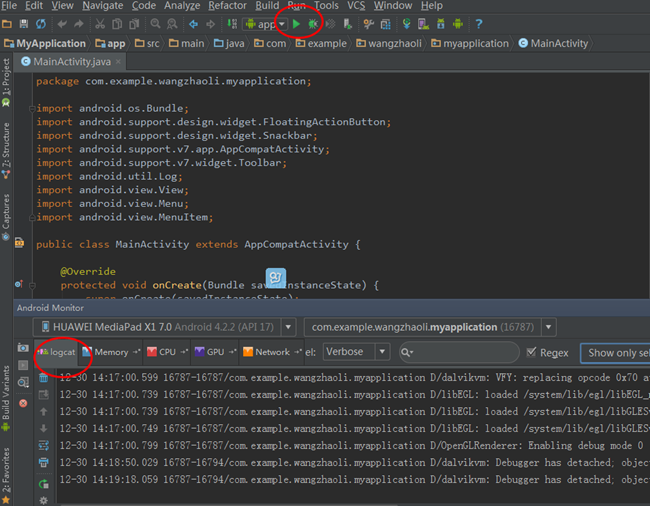


　　（5）继续next-->直到finish，则完成了该工程的创建，创建过程会稍长一些，需等待。



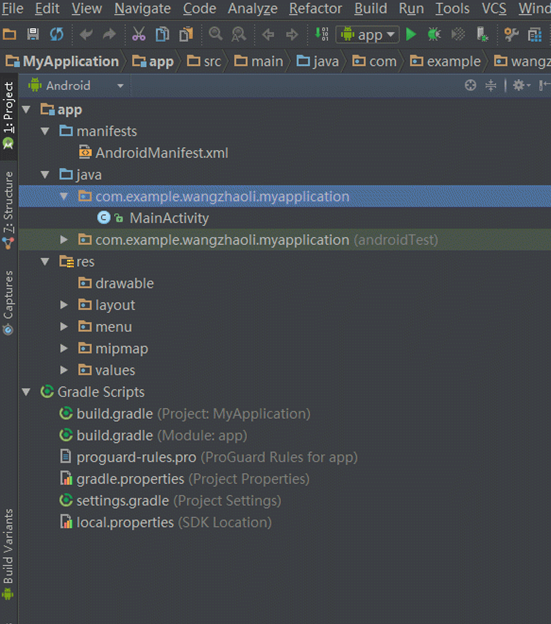
　　（6）以下就是创建完毕的工程界面，点击上面工具栏中的运行图标运行，下面的logcat中有日志信息，这些和Eclipse差不多，使用起来也会很顺手，就不再过多说明。

        后文会详细介绍一下Android Studio与Eclipse常用之处不同的地方和独有的优点。



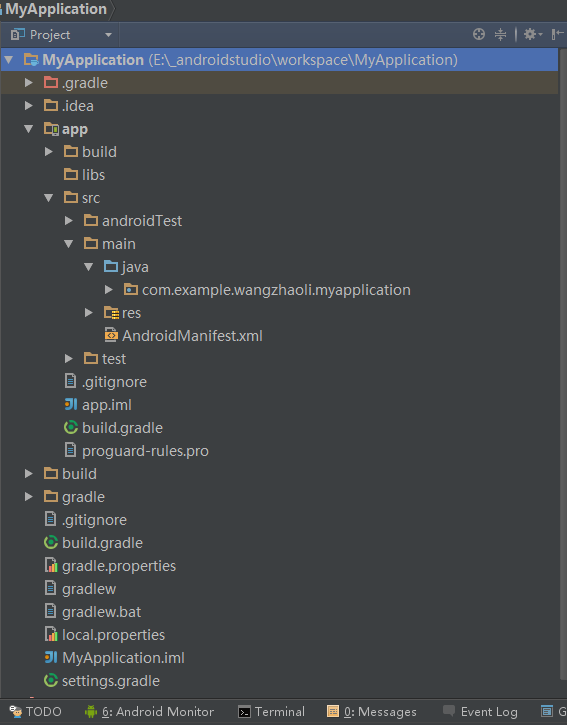
**Android Studio的工程目录结构**

    当我们新建一个项目的目录结构默认是这样的：



　　可以看到和Eclipse的目录结构有很大区别，Android Studio一个窗口只能有一个项目，而Eclipse则可以同时存在很多项目，如果你看着不习惯可以点击左上角的Android机器人进行切换：

       将"Android"切换为"Project"后如下图：



　　感觉这个和Eclipse中的结构很像。

      总的来说和Eclipse的区别有如下：

      1、Android Studio中有Project和Module的概念，前面说到Android  Studio中一个窗口只能有一个项目，即一个Project，代表一个 workspace，但是一个Project可以包含多个Module，比如你项目引用的Android Library, Java Library等，这些都可以看做是一个Module;

      2、上述目录中将java代码和资源文件（图片、布局文件等）全部归结为src，在src目录下有一个main的分组，同时划分出java和res两个文件夹，java文件夹则相当于Eclipse下的src文件夹，res目录结构则一样.

      关于工程目录结构只要使用习惯就可以了，不必过于多说啦，接下来说一下Android Studio的其他使用之处。

      3.主要文件和目录：

     （1）settings.gradle： 每个Moudle 要在这里注册，可以打开该文件看一下就会明白；

     （2）build.gradle：每个Project 的配置

     （3）app：每个Module

     （4）app/build.gradle：每个Module 配置文件，比如依赖的类库，SDK 版本

     （5）app/src/：源码和资源文件都在这里；我们写的文件都在这里面。

     （6）app/libs/：添加类库。