









Java JDK 配置教程

By: マンノカゲ

此教程将为大家图解在 windows 系统下，Java 环境 JDK 的简易配置方法，针对 win10 (win8) 和 win7 配置方法的解说 (~~注: 本教程注重简易和实用性, 并未描述所有的方法!!!~~): 首先你要先准备好 JDK, 目前推荐的版本号为 JDK11 ~~或其它~~ ~~长期支持~~ (后缀为 LTS) 版本, 官网提供 ~~安装版~~ 和 ~~压缩包~~ 两种方式, 压缩包可以免安装进行配置, 以下以安装版进行说明, 压缩版请跳过安装过程!

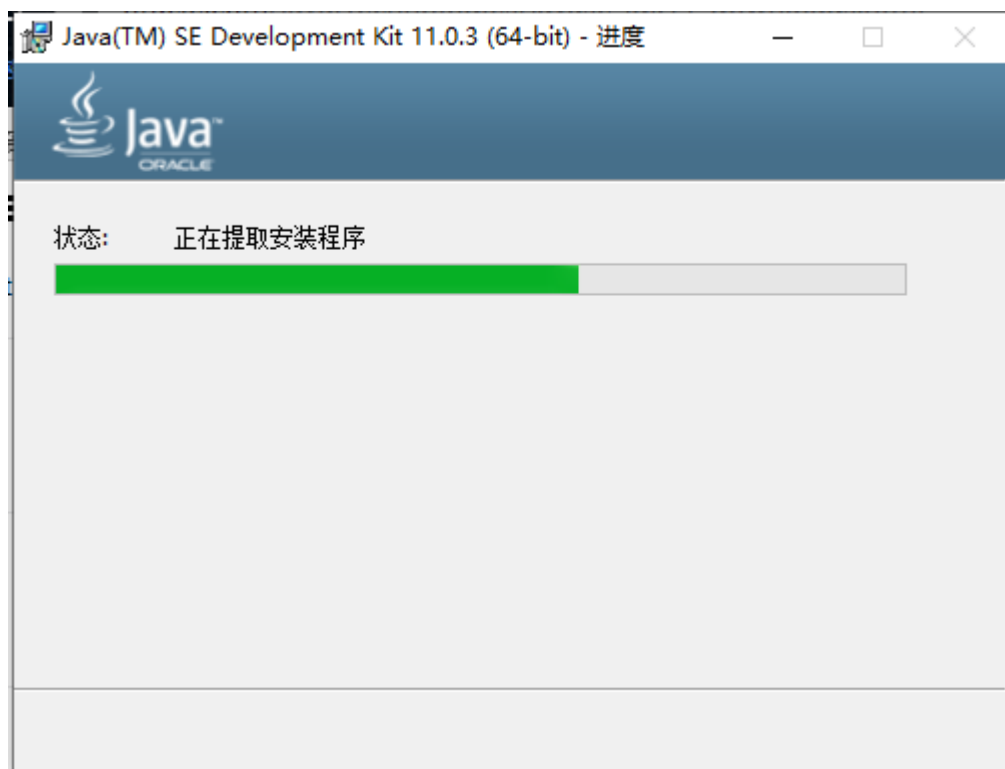
1: 首先访问 JDK 官网获取 JDK

<https://www.oracle.com/java/technologies/javase-jsp-downloads.html>, 推荐下载 JDK11;

Java SE Development Kit 11.0.4		
Product / File Description	File Size	Download
Linux	147.85 MB	 jdk-11.0.4_linux-x64_bin.deb
Linux	154.6 MB	 jdk-11.0.4_linux-x64_bin.rpm
Linux	172.01 MB	 jdk-11.0.4_linux-x64_bin.tar.gz
macOS	166.58 MB	 jdk-11.0.4_osx-x64_bin.dmg
macOS	166.95 MB	 jdk-11.0.4_osx-x64_bin.tar.gz
Solaris SPARC	188.21 MB	 jdk-11.0.4_solaris-sparcv9_bin.tar.gz
Windows	151.22 MB	 jdk-11.0.4_windows-x64_bin.exe ← 安装版
Windows	171.25 MB	 jdk-11.0.4_windows-x64_bin.zip ← 压缩版

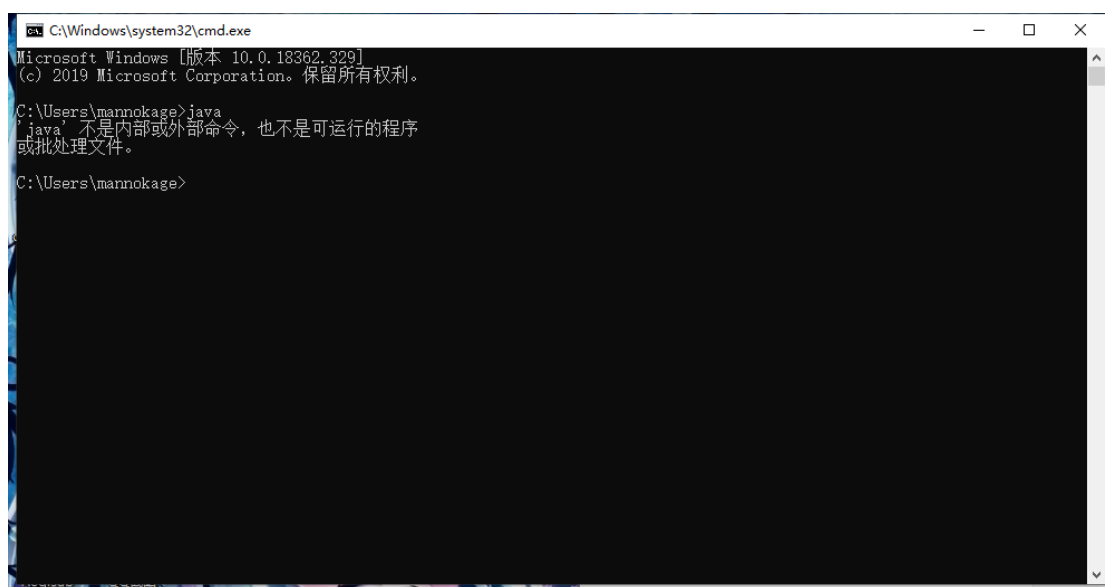
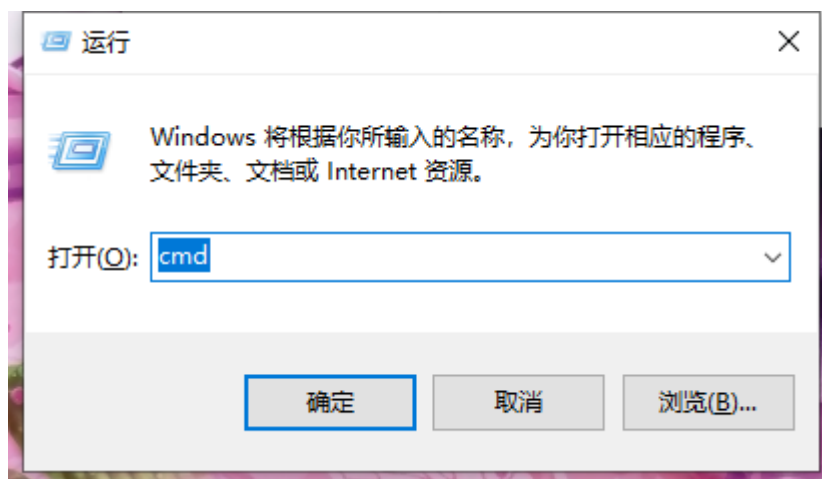
2: 打开已下载的安装包进行安装, 为了保证后续开发工程的安全性, 请将 JDK 安装到非系统盘 (一般为 C 盘) 的目录下; 下面采用默认 C 盘方式, 请视情况为 JDK 准备安装目录;





到目前为止，JDK 已经成功安装到你的电脑了，你一定觉得已经全部 OK 了？其实，到上面为止只是一个基本的软件安装小 case 而已，一个隐性的问题会影响到你后期的使用，完

成之前的步骤后，你可以执行一些 *java 内部命令* 来检查你是否存在这个问题。(检查 JDK 环境配置方法 *win+r* 输入神奇代码 *cmd*，在刚刚打开的漆黑**命令提示符**界面输入 *java* 或 *java -version*，若出现 java 版本信息或 java 菜单导航信息即为验证成功)



未配置 java 前，输入 java 内部命令提示的错误信息界面

下面我们进入正题：JAVA 系统环境变量配置

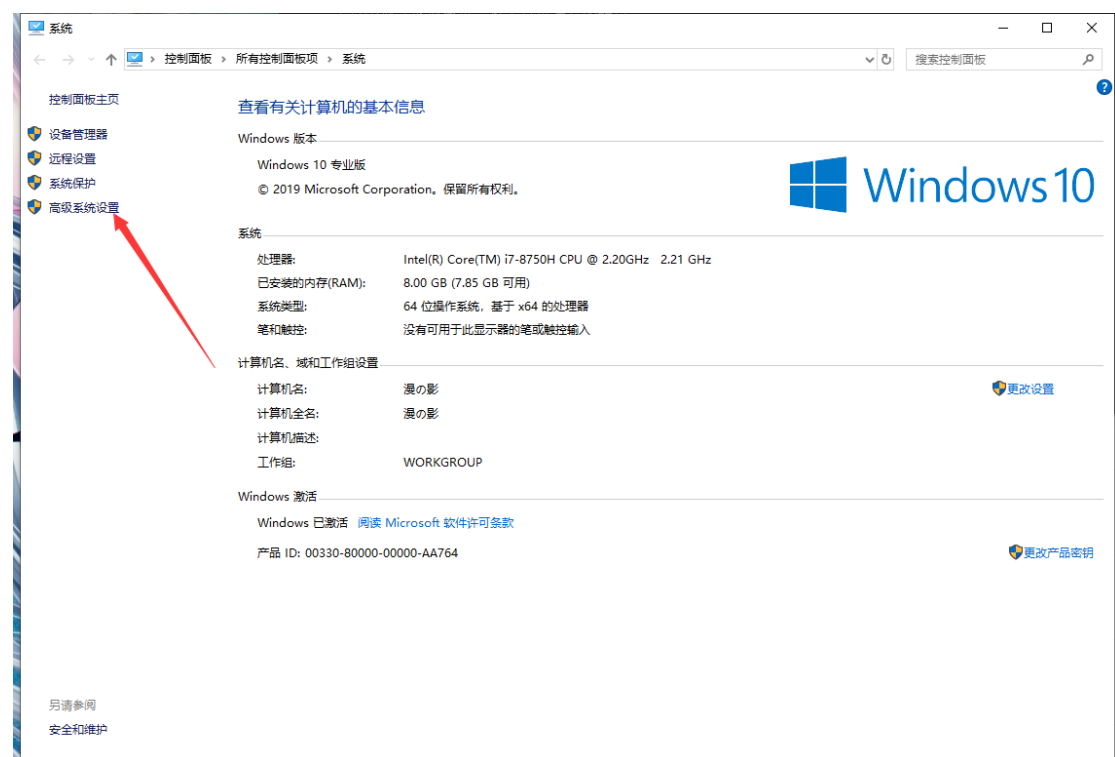
3-1: 系统环境的变量配置：下面我将分别就下面两种适用**配置方法**进行说明

方法①：在 win10 即将称霸年代，老旧的 win7 界面也被更加人性及醒目的可视化界面所替代（每错，就是 bug10 (ㄟ _ ㄟ)），下面为 **win10 专属**独特的可视化进行配置操作（win7，XP 请移步方法·②）：

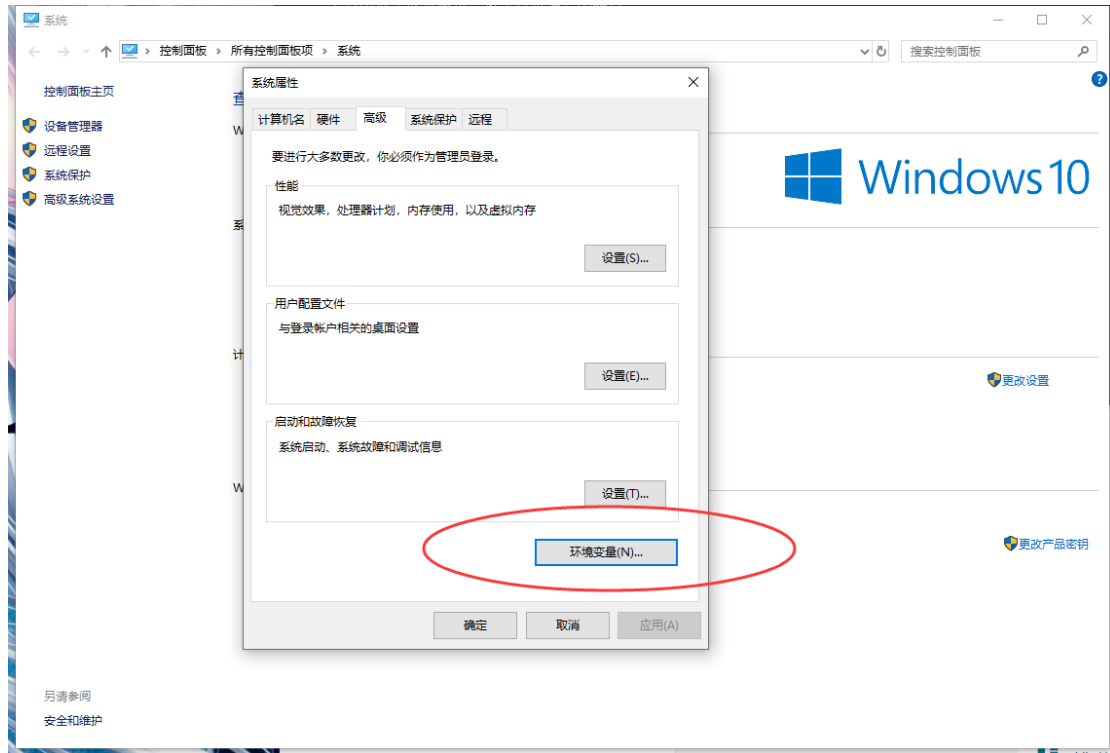


按住 Alt 键双击桌面的我的电脑图标（此处为简便的方法，也许你可以找到进入此操作页面的其他途径）进入一个包含你的计算机基本信息界面，依次点击**高级系统设**→**环境变量**→**Path 变量**

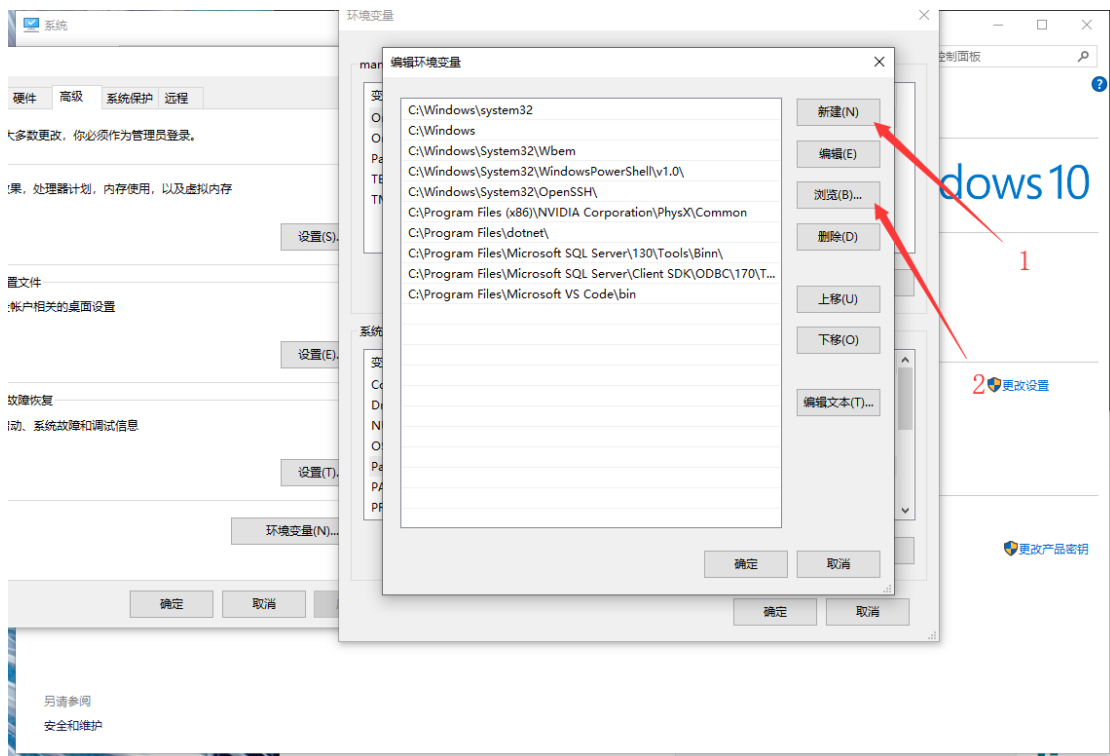
高级系统设置：

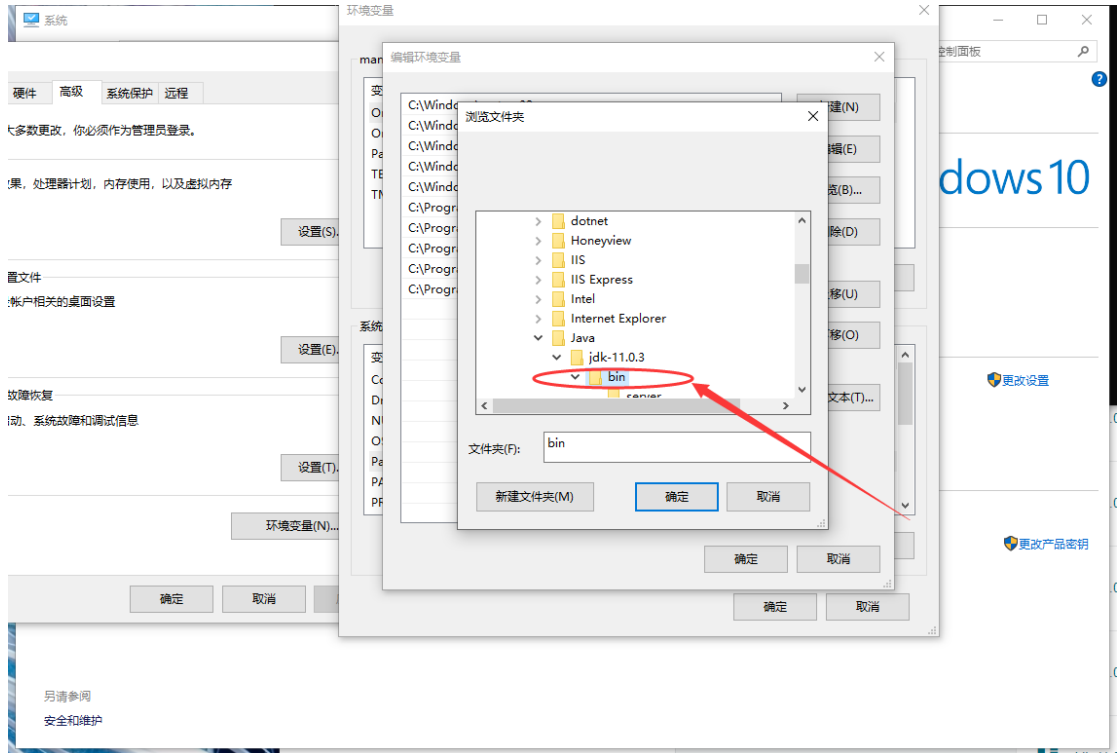


环境变量



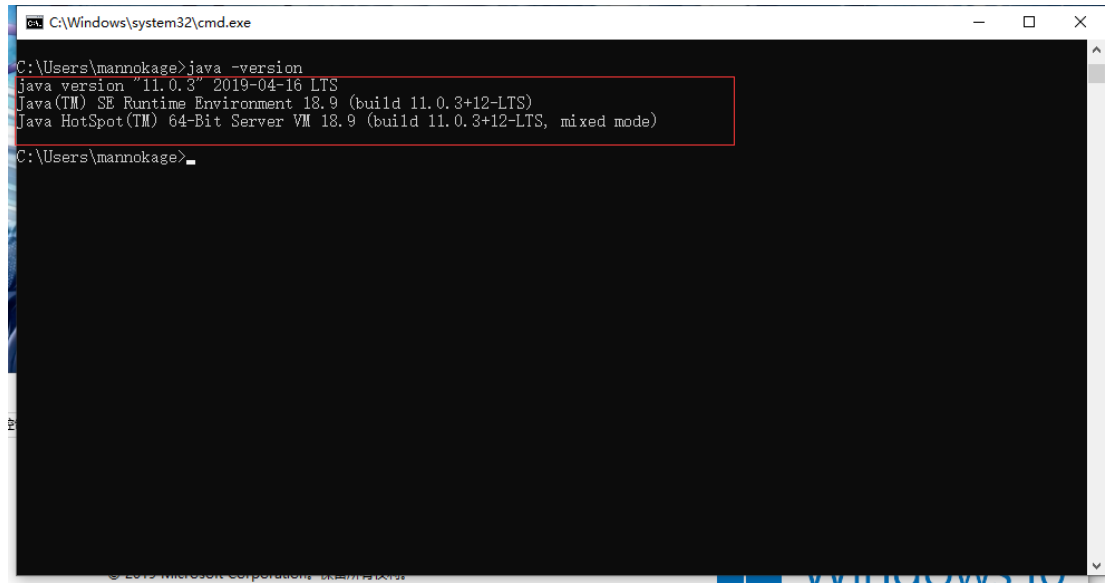
选择**新建**→**浏览**，找到自己的 JDK 安装目录下 **bin** 目录，单击确定即可。





(注: JDK5 以上无需另行配置 CLASS PATH 和 JAVA_HOME 目录, JDK 会自行分配在 JDK 安装目录中!!!!)

在 **命令提示符** 下输入验证 JAVA 版本的指令 **java -version** (java 和 - 间有一个空格, - 和 version 则没有) 会出现 java 的版本信息, 或者输入 **java** 出现 java 的操作导航菜单。



```
C:\Windows\system32\cmd.exe

C:\Users\mannokage>java -version
java version "11.0.3" 2019-04-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.3+12-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.3+12-LTS, mixed mode)

C:\Users\mannokage>_
```

以上为键入 `java -version` 后显示的 JDK 版本信息


```

C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.18362.329]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\Users\mannokage>java
用法: java [options] <主类> [args...]
      (执行类)
或 java [options] -jar <jar 文件> [args...]
      (执行 jar 文件)
或 java [options] -m <模块> [/<主类>] [args...]
      (执行模块中的主类)
或 java [options] --module <模块> [/<主类>] [args...]
      (执行模块中的主类)
或 java [options] <源文件> [args]
      (执行单个源文件程序)

将主类、源文件、-jar <jar 文件>、-m 或
--module <模块> /<主类> 后的参数作为参数
传递到主类。

其中，选项包括：

-cp <目录和 zip/jar 文件的类搜索路径>
-classpath <目录和 zip/jar 文件的类搜索路径>
--classpath <目录和 zip/jar 文件的类搜索路径>
      使用：分隔的，用于搜索类文件的目录，JAR 档案
      和 ZIP 档案列表。
-p <模块路径>
--module-path <模块路径>...
      用：分隔的目录列表，每个目录
      都是一个包含模块的目录。
--upgrade-module-path <模块路径>...
      用：分隔的目录列表，每个目录
      都是一个包含模块的目录，这些模块
      用于替换运行时映像中的可升级模块
--add-modules <模块名称> [, <模块名称> ...]
      除了初始模块之外要解析的根模块。
      <模块名称> 还可以为 ALL-DEFAULT, ALL-SYSTEM,
      ALL-MODULE-PATH。
--list-modules
      列出可观察模块并退出
-d <module name>
--describe-module <模块名称>
      描述模块并退出
--dry-run
      创建 VM 并加载主类，但不执行 main 方法。
      此 --dry-run 选项对于验证诸如
      模块系统配置这样的命令行选项可能非常有用。
--validate-modules
      验证所有模块并退出
--validate-modules
      --validate-modules 选项对于查找
      模块路径中模块的冲突及其他错误可能非常有用。
-D<名称>=<值>
      设置系统属性
-verbose:[class|module|gc|jni]
      启用详细输出
-version
      将产品版本输出到错误流并退出
--version
      将产品版本输出到输出流并退出
-showversion
      将产品版本输出到错误流并继续
--show-version
      将产品版本输出到输出流并继续
--show-module-resolution
      在启动过程中显示模块解析输出
-? -h -help
      将此帮助消息输出到错误流
--help
      将此帮助消息输出到输出流
-X
      将额外选项的帮助输出到错误流
--help-extra
      将额外选项的帮助输出到输出流
-ea[:<程序包名称>...][:<类名>]
-enableassertions[:<程序包名称>...][:<类名>]
      按指定的粒度启用断言
-da[:<程序包名称>...][:<类名>]
-disableassertions[:<程序包名称>...][:<类名>]
      按指定的粒度禁用断言
-esa | -enablesystemassertions
      启用系统断言
-dsa | -disablesystemassertions
      禁用系统断言
-agentlib:<库名>[=<选项>]
      加载本机代理库 <库名>，例如 -agentlib:jdwp
      另请参阅 -agentlib:jdwp=help
-agentpath:<路径名>[=<选项>]
      按完整路径名加载本机代理库
-javaagent:<jar 路径>[=<选项>]
      加载 Java 编程语言代理，请参阅 java.lang.instrument
-splash:<图像路径>
      使用指定的图像显示启动屏幕
      自动支持和使用 HiDPI 缩放图像
      (如果可用)。应始终将未缩放的图像文件名 (例如，image.ext)
      作为参数传递给 -splash 选项。
      将自动选取提供的最合适的缩放
      图像。
      有关详细信息，请参阅 SplashScreen API 文档
@argument 文件
      一个或多个包含选项的参数文件
-disable-@files
      阻止进一步扩展参数文件
--enable-preview
      允许类依赖于此发行版的预览功能
要为长选项指定参数，可以使用 --<名称>=<值> 或
--<名称> <值>。

```

```

C:\Users\mannokage>java -version
java version "11.0.3" 2019-04-16 LTS

```

以上为键入 *java* 后 *java* 操作指令向导

方法②：这是一个更加方便并且 **win10** 和 **win7** 及落伍的 **XP** 均通用的办法, 在方法中①涉及的命令提示符下键入：

set path=%path%;C:\Program Files\Java\jdk-11.0.3\bin 后回车，此处验证方式雷同方法①，此处便不再累赘！

（注：黄色字体显示的路径为 java 默认的安装目录，请选择自己安装或解压时的目录）

至此，你已经完成基本的 JDK 环境配置，选择适合自己的 IDE 进行 JAVA 学习吧！

終わり