

Please visit website: <http://cxyroad.com>

选择合适的JDK版本。例如，对于64位Windows系统，可以选择“jdk-10.0.1_windows-x64_bin.exe”等相应的安装文件。

!<https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/ee1d596db3f94cbabc814ba2b8fc2f74~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expire=1722004116&x-signature=XX95qc8vZ7N4hFA0vIFacGptz64%3D>

点击“Download”按钮开始下载JDK安装包。如果下载速度较慢，可以考虑使用国内镜像站点如www.injdk.cn/(<http://cxyroad.com/>“<https://www.injdk.cn/>”) 进行下载。

!<https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/07562e590863498a9e1215f376c77773~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expire=1722004116&x-signature=x3GFP45463maheTCYInhKI2Ptel%3D>

二、安装JDK

下载完成后，找到下载好的JDK安装包（例如：jdk-10.0.1_windows-x64_bin.exe），双击运行该安装程序。

1. **安装过程中的设置**：

* **默认安装路径**：通常情况下，可以选择默认的安装路径，但也可以根据个人习惯更改安装路径。建议将安装路径设置为用户文件夹中的某个专门的文件夹，如C:\Program Files\Java\jdk-10。

* **自定义安装选项**：在安装过程中，可以根据需要选择是否同时安装JRE（Java SE Runtime Environment）。如果需要，可以修改其安装目录。安装程序会自动进行安装，等待安装完成即可。

三、配置环境变量

右键点击“此电脑”或“我的电脑”，选择“属性”，然后点击“环境变量”。

* 新建一个名为“JAVA_HOME”的环境变量，其值为JDK的安装路径，例如：
： C:\Program Files\Java\jdk-10。

在系统变量中找到Path变量，右键点击并选择“编辑”。在Path变量的值中添加%JAVA_HOME%\bin；这样可以在任何地方直接调用Java命令。

打开命令提示符（cmd），输入`java -version`，如果显示了Java版本信息，则表示JDK已成功安装。

四、测试JDK是否安装成功

使用快捷键Win+R，输入`cmd`并回车。

在命令行中输入`java -version`，如果返回正确的Java版本信息，则说明JDK已经正确安装。

通过以上步骤，你就可以成功下载、安装并配置好JDK了。希望这些信息对你有所帮助！

如何解决JDK下载过程中遇到的常见问题？

在下载和安装JDK过程中，可能会遇到一些常见问题。以下是一些解决方案：

1. **检查网络连接**：

* 网络不稳定或连接超时是导致下载中断或失败的常见原因。请确保您的网络连接稳定，并在确保网络正常后尝试重新下载安装包。

2. **更换下 java -version

...

总结来说，在Linux系统上，用户需要手动下载、解压并配置环境变量；而在macOS系统上，则主要通过Homebrew进行自动化安装和配置。

JDK更新后的新特性有哪些，对现有项目的影晌如何？

JDK的更新通常会带来一系列新特性和改进，这些变化对现有项目的影响是多方面的。以下是一些常见的新特性和它们对现有项目可能产生的影响：

1. **性能提升**：

* 更好的垃圾回收机制和更快的类加载器可以显著提高应用程序的运行速度。例如，从Java 8升级到Java 11，G1GC的平均速度提升了16.1%，ParallelGC提升了4.5%。

* JDK 17引入了显著改进的垃圾回收器（GC），使得服务内存占用降低了一半左右，间接节约了服务器成本，并提高了代码运行效率。

2. **新特性和API改进**：

* JDK 8引入了Lambda表达式、Stream API、新的日期和时间API等，这些特性极大地提高了开发效率和代码可读性。

* JDK 11增加了局部变量类型推断、新的HTTP/2客户端支持以及Lambda表达式的新特性，进一步提升了开发效率。

* JDK 21引入了JEP 400模块化方式，使得开发者能够更轻松地组织和管理大型项目中的代码。

3. **支持最新技术和框架**：

* 许多新的技术和框架已经或即将开始依赖于JDK 11或以上版本，升级后可以保证应用程序能够充分利用这些新的技术和框架。

4. **长期支持版本（LTS）**：

* 比如JDK 11是一个长期支持版本，这意味着它将获得长期的更新和支持，有助于保持应用程序的稳定性和可靠性。

5. **行业趋势**：

* 随着技术的发展，许多新的技术和框架需要使用更高版本的JDK来实现其功能，因此及时升级JDK有助于保持应用的现代性和竞争力。

6. **非LTS版本的新特性**：

* 虽然非LTS版本如JDK 22可能不提供长期支持，但它们仍然包含一些引人注目的新特性，例如未命名变量和模式的最终迭代、字符串模板的第二预览版以及隐式声明的类等。

总结来说，JDK的更新不仅带来了性能上的提升和新特性，还为开发者提供了更多的工具和方法来提高开发效率和代码质量。对于现有项目，升级到新版本的JDK可以带来显著的性能提升和更好的兼容性，同时也能利用最新的技术和框架来优化和扩展应用。

如何使用命令行工具验证JDK是否成功安装并运行?

要验证JDK是否成功安装并运行，可以按照以下步骤进行：

1. **打开命令提示符**：

* 按下 `Win + R` 键，输入 `cmd` 并按回车键。

2. **输入验证命令**：

* 在命令行窗口中输入 `java -version` 和 `javac -version` 命令。这两个命令分别用于检查Java和Java编译器的版本信息。

3. **查看输出结果**：

* 如果正确配置了环境变量，这两个命令应分别显示JDK的版本信息。例如，如果看到类似“OpenJDK Runtime Environment (build 1.8.0_121) OpenJDK 64-bit”的输出，则表明Java已正确安装。

通过上述步骤，你可以确认JDK是否已经成功安装并配置好环境变量。

原文链接: <https://juejin.cn/post/7393304744695496713>