

- » Downloading Java from the Oracle website
- » Installing Java
- » Identifying your version of Java

Chapter 1

Installing and Using Java Tools

Java development environments have two basic approaches. On the one hand, you can use a sophisticated integrated development environment (IDE) such as NetBeans or Eclipse. These tools combine a full-featured source editor that lets you edit your Java program files with integrated development tools, including visual development tools that let you create applications by dragging and dropping visual components onto a design surface.

At the other extreme, you can use just the basic command-line tools that are available free from Oracle's Java website (<https://java.oracle.com>). Then you can use any text editor you want to create the text files that contain your Java programs (called *source files*), and compile and run your programs by typing commands at a command prompt.



TIP

As a compromise, you may want to use a simple development environment, such as TextPad. TextPad is an inexpensive text editor that provides some nice features for editing Java programs (such as automatic indentation) and shortcuts for compiling and running programs. It doesn't generate any code for you or provide any type of visual design aids, however. TextPad is the tool we used to develop all the examples shown in this book. For information about downloading and using TextPad, see Chapter 2.



TIP

If you prefer a free alternative, you can also investigate Notepad++ at <https://notepad-plus-plus.org>.

You can also compile and run simple Java programs online at sites such as JDoodle (www.jdoodle.com/online-java-compiler) or Programiz (www.programiz.com/java-programming/online-compiler). At these sites, you can enter simple Java programs, compile them, and run them. They're a great way to dip your toes into the shallow end of the Java programming pool without having to install anything.

Downloading and Installing the Java Development Kit

Before you can start writing Java programs, you have to download and install the correct version of the Java Development Kit (JDK) for the computer system you're using. Oracle's Java website provides versions for Windows, macOS, and Unix. The following sections show you how to download and install the JDK.



TIP

If you prefer, you can download and install the open-source version of Java from <https://openjdk.org>.

Downloading the JDK

To get to the download page, point your browser to www.oracle.com/java/technologies. Then follow the appropriate links to download the latest version of Java SE for your operating system. (At the time we wrote this, the latest version was 18.0.1.1. However, Java 19 was available in early release form. By the time this book hits the shelves, Java 19 or later should be available from www.oracle.com/java/technologies.)

When you get to the Java download page, you'll need to select your operating system; Java is available for Linux, macOS, and Windows. In this chapter, we show you how to deploy Java to a Windows 10 or 11 computer. The procedures for deploying Java to Linux or macOS are similar.

The JDK download comes in three versions: a compressed.zip file, an executable installer (.exe), or a Windows installer package (.msi). All are about the same size (under 200MB). We find it easier to download and run the .exe installer.

Installing the JDK

After you download the JDK file, you can install it by running the executable file you downloaded. The procedure varies slightly depending on your operating system, but basically, you just run the JDK installation program file after you download it, as follows:

- » On a Windows system, open the folder in which you saved the installation program and double-click the installation program's icon.
- » On a Linux or macOS system, use console commands to change to the directory to which you downloaded the file and then run the program.

After you start the installation program, it prompts you for any information that it needs to install the JDK properly, such as which features you want to install and what folder you want to install the JDK in. You can safely choose the default answer for each option.

Perusing the JDK folders

When the JDK installs itself, it creates several folders on your hard drive. The locations of these folders vary depending on your system and how you installed Java. The three most likely places to find the Java home folder in Windows are the root of your C: drive, the folder C:\Program Files\Java, or the folder C:\Program Files (x86)\Java. Within one of these locations, you'll find the Java home folder, whose name starts with `jdk-` and ends with the version number. For version 19, the home folder is named `jdk-19`.

Table 1-1 lists the subfolders created in the JDK home folder. As you work with Java, you'll refer to these folders frequently.

TABLE 1-1 Subfolders of the JDK Home Folder

Folder	Description
<code>bin</code>	The compiler and other Java development tools
<code>conf</code>	Configuration file
<code>include</code>	This library contains files needed to integrate Java with programs written in other languages

TABLE 1-1 (continued)

Folder	Description
jmods	Modules for the Java Module System
legal	Copyright and license information for various Java components
lib	Library files, including the Java API class library

Setting the JAVA_HOME and path variables

After you install the JDK, you need to configure your operating system so that it can find the JDK command-line tools. To do that, you must set two environment variables: `JAVA_HOME`, which provides the location of the Java home folder, and `Path`, which lists the folders that the operating system uses to locate executable programs. Follow these steps:

- 1. Open File Explorer, right-click This PC, and choose Properties.**
This brings up the System Properties page.
- 2. Click the Advanced System Settings link.**
- 3. Click the Environment Variables button.**
The Environment Variables dialog box appears, as shown in Figure 1-1.
- 4. In the System Variables list, click the New button.**
The New System Variable dialog box, shown in Figure 1-2, appears.
- 5. Type `JAVA_HOME` in the Variable Name text box.**
- 6. Click the Browse Directory button, browse to the Java home folder, and then click OK.**
This action inserts the path to the home folder in the Variable Value text box.
- 7. Click OK.**
The `JAVA_HOME` variable is created.
- 8. Scroll to the Path variable in the System Variables list, select it, and then click the Edit button.**

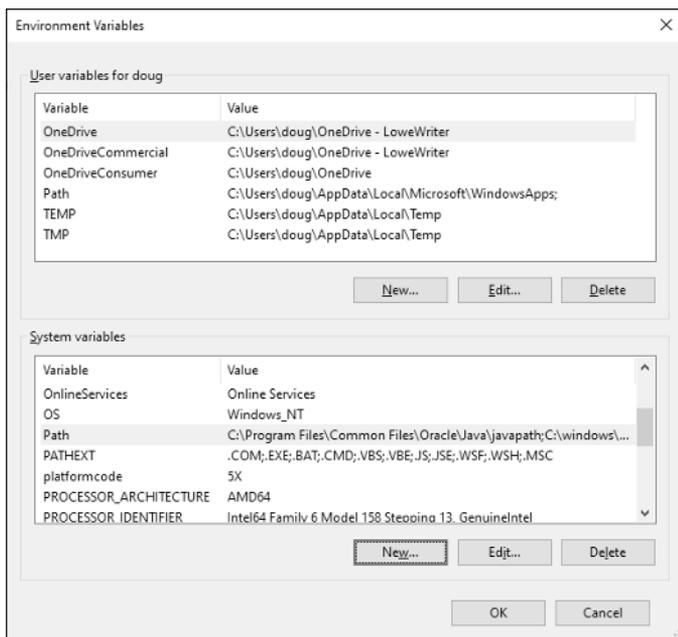


FIGURE 1-1: The Environment Variables dialog box.



FIGURE 1-2: Creating the JAVA_HOME variable.

This brings up a handy dialog box that lets you add or remove paths to the Path variable or change the order of the paths, shown in Figure 1-3.

9. Peruse the list of entries in the Path variable. If you find one that references a previous version of Java, delete it.

Specifically, look for an entry that begins with `C:\Program Files\Java`. If you find such an entry, select it and then click Delete.

10. Click the New button.

This opens a line for you to create a new path entry.

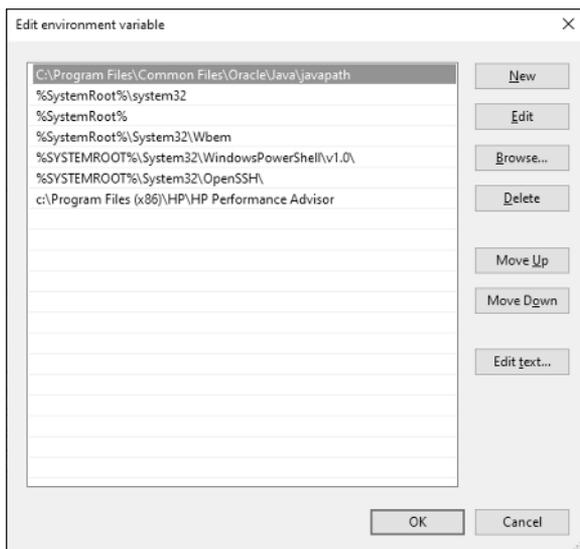


FIGURE 1-3: Editing the Path variable.

11. Type %JAVA_HOME%\bin as the new path entry.

The percent signs insert the value of the JAVA_HOME variable in your new path entry. For example, if JAVA_HOME is set to C:\Program Files\Java\jdk19, the new path entry will be C:\Program Files\Java\jdk19\bin.

12. Click OK three times to exit.

The first OK gets you back to the Environment Variables dialog box; the second OK gets you back to the System Properties dialog box; and the third OK closes the System Properties dialog box.

For Linux, the procedure depends on which shell you're using. For more information, consult the documentation for the shell you're using.

Confirming Your Java Version

After you've installed Java, it's a good idea to confirm that you've installed the correct version. To do that, follow these steps:

- 1. Press the Windows key, type `cmd`, and press Enter to open a command prompt.**
- 2. Type `java -version` and press Enter.**

This command instructs Java to display its version number. You'll see output similar to this:

```
openjdk version "19-ea" 222-09-20
OpenJDK Runtime Environment (build 19-ea+27-2074)
OpenJDK 64-Bit Server VM (build 19-ea+27-2074, mixed
mode, sharing)
```

- 3. Confirm that the first line of the output reflects the version you installed.**

In this example, Java version 19-ea is installed. The `ea` indicates that we're using the early-access version of Java 19, which is the version we used as we wrote this book. By the time you read this, you'll see a slightly different version of Java 19 or later. Or, you may see a Java 18 or even a Java 17 version number. Any of these versions will work for the coding examples in this book, unless we specifically mention that a specific Java version is required.

- 4. Close the command window.**

