

Your First Cup of JAVA

The following instructions will help you write your first program. These instructions are for users of Microsoft Windows platforms. We start with a checklist of what you need to write your first program. Next, we cover the steps to creating an application.

1 A Checklist

To write your first program, you need:

1. Java Standard Edition 8. You can download the SDK¹ and consult the installation instructions. (Make sure you download the SDK, not the JRE².)
2. A text editor. In this example, we'll use Sublime Text. You can easily adapt these instructions if you use a different text editor.

These two items are all you need to write your first Java program.

2 Creating Your First Application

Your first program, `HelloWorldApp`, will simply display the greeting “Hello world!”. To create this program, you will:

1. Create a source file. A source file contains text, written in the Java programming language, that you and other programmers can understand. You can use any text editor to create and edit source files.
2. Compile the source file into a **bytecode file**. The compiler, `javac`, takes your source file and translates its text into instructions that the **Java Virtual Machine** (Java VM) can understand. The compiler converts these instructions into a bytecode file.
3. Run the program contained in the bytecode file. The Java interpreter installed on your computer implements the Java VM. This interpreter takes your bytecode file and carries out the instructions by translating them into instructions that your computer can understand.

¹SDK = Software Development Kit

²JRE = Java Runtime Environment

2.1 Why Bytecodes are Cool

So, you’ve heard that with the Java programming language, you can “write once, run anywhere.” This means that when you compile your program, you don’t generate instructions for one specific platform. Instead, you generate Java bytecodes, which are instructions for the Java Virtual Machine (Java VM). If your platform — whether it’s Windows, UNIX, MacOS, or an Internet browser — has the Java VM, it can understand those bytecodes.

3 Create a Source File

1. Start Sublime Text. In a new document, type in the following code:

```
/**
 * The HelloWorldApp class implements an application
 * that displays "Hello World!" to the standard output.
 */
public class HelloWorldApp {
    public static void main(String[] args) {
        // Display "Hello World!"
        System.out.println("Hello World!");
    }
}
```

Be careful when you type! Type all code, commands, and file names exactly as shown. The Java compiler and interpreter are case-sensitive, so you must capitalize consistently.

`HelloWorldApp` \neq `helloworldapp`

2. Save this code to a file. From the menu bar, select File → Save As. In the Save As dialog box:
 - Using the “Save in” drop-down menu, specify the folder (directory) where you’ll save your file. In this example, the directory is `java` on the C drive.
 - In the “File name” text box, type “`HelloWorldApp.java`”, including the double quotation marks.
 - From the “Save as type” drop-down menu, choose “Text Document”.

Now click Save, and exit NotePad.

4 Compile the Source File

From the Start menu, select the command line application (type cmd in the search box). When the application launches, it should look like this:

```
C:\
```

The prompt shows your current directory. To compile your source code file, change your current directory to the directory where your file is located. For example, if your source directory is java on the C drive, you would type the following command at the prompt and press Enter:

```
cd c:\java
```

Now the prompt should change to `C:\java>`. If you enter `dir` at the prompt, you should see your file. Now you can compile. At the prompt, type the following command and press Enter:

```
javac HelloWorldApp.java
```

If your prompt reappears without error messages, congratulations. You have successfully compiled your program.

The compiler has generated a Java bytecode file, `HelloWorldApp.class`. At the prompt, type `dir` to see the new file that was generated. Now that you have a `.class` file, you can run your program.

5 Run the Program

In the same directory, enter at the prompt:

```
java HelloWorldApp
```

Now you should see:

```
C:\java>java HelloWorldApp
Hello World!
C:\java>
```

Congratulations! Your program works.