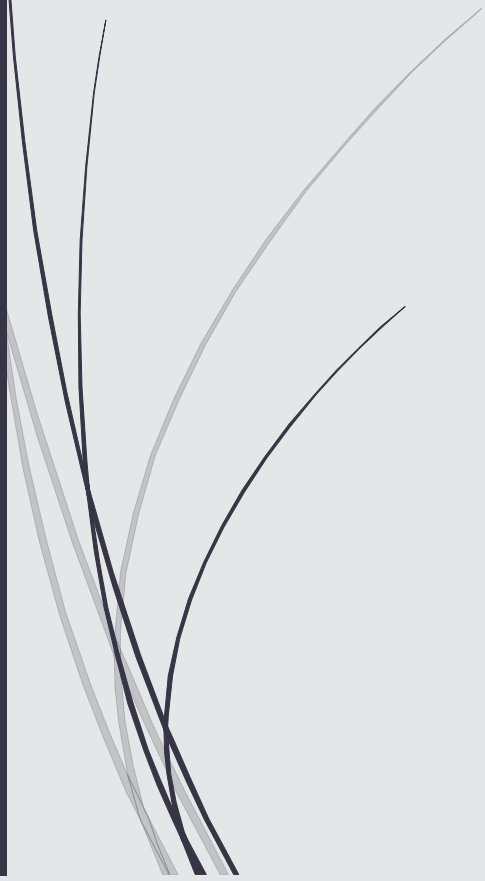
Java Lesson 1

Java From Scratch

Java Introduction
Java Install
Java Syntax



Content

What is Java?

Java is a popular programming language, created in 1995.

It is owned by Oracle, and more than **3 billion** devices run Java.

It is used for:

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

Why Use Java?

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming language in the world
- It has a large demand in the current job market
- It is easy to learn and simple to use
- It is open-source and free
- It is secure, fast and powerful
- It has a huge community support (tens of millions of developers)
- Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa

---

Java Getting Started

Java Install

Some PCs might have Java already installed.

To check if you have Java installed on a Windows PC, search in the start bar for Java or type the following in Command Prompt (cmd.exe):

```
C:\Users\Your Name>java -version
```

If Java is installed, you will see something like this (depending on version):

```
java version "11.0.1" 2018-10-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.1+13-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS, mixed mode)
```

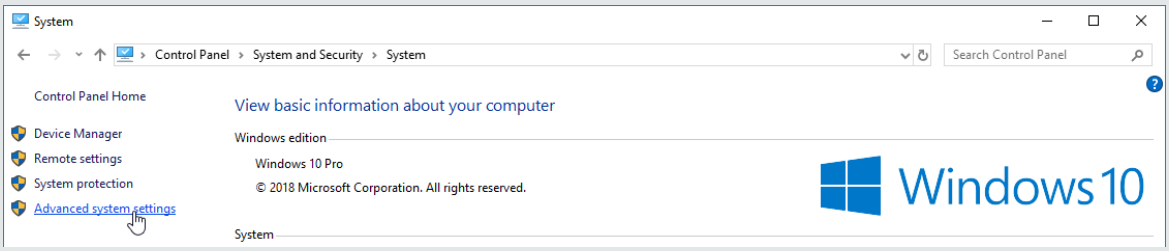
If you do not have Java installed on your computer, you can download it for free at oracle.com.

Note: In this tutorial, we will write Java code in a text editor. However, it is possible to write Java in an Integrated Development Environment, such as IntelliJ IDEA, Netbeans or Eclipse, which are particularly useful when managing larger collections of Java files.

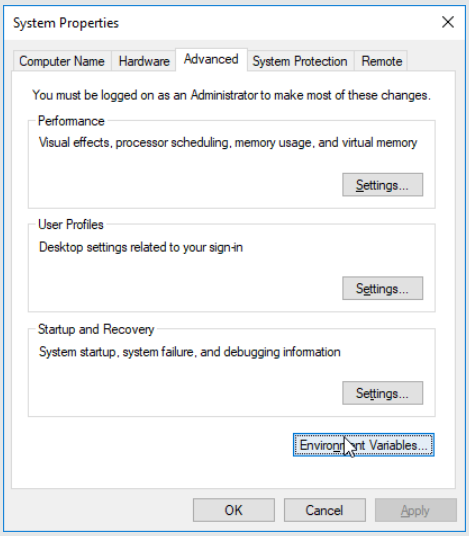
Setup for Windows

To install Java on Windows: -

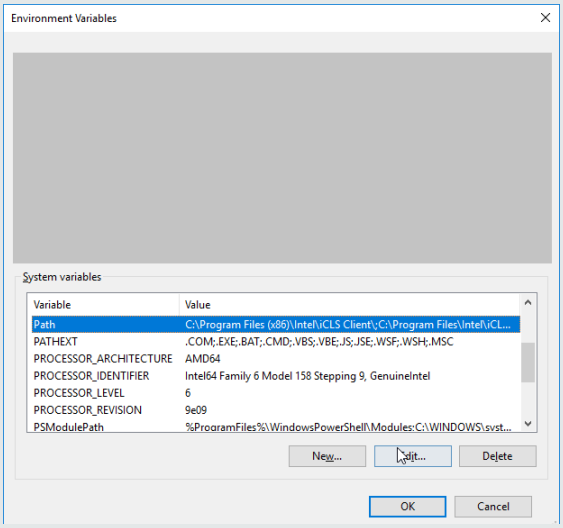
1. Go to "System Properties" (Can be found on Control Panel > System and Security > System > Advanced System Settings)



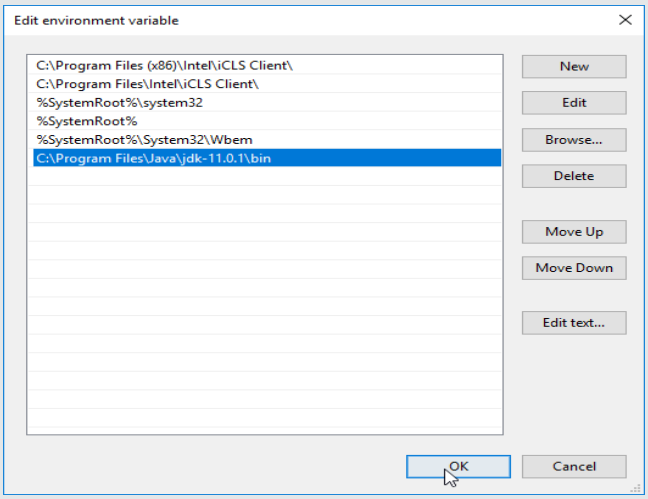
2. Click on the "Environment variables" button under the "Advanced" tab



3. Then, select the "Path" variable in System variables and click on the "Edit" button



4. Click on the "New" button and add the path where Java is installed, followed by **\bin**. By default, Java is installed in C:\Program Files\Java\jdk-11.0.1 (If nothing else was specified when you installed it). In that case, You will have to add a new path with: **C:\Program Files\Java\jdk-11.0.1\bin**
Then, click "OK", and save the settings



5. At last, open Command Prompt (cmd.exe) and type **java -version** to see if Java is running on your machine

Write the following in the command line (cmd.exe):

```
C:\Users\Your Name>java -version
```

If Java was successfully installed, you will see something like this (depending on version):

```
java version "11.0.1" 2018-10-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.1+13-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS, mixed mode)
```

Java Quickstart

In Java, every application begins with a class name, and that class must match the filename.
Let's create our first Java file, called Main.java, which can be done in any text editor (like Notepad).
The file should contain a "Hello World" message, which is written with the following code:

Main.java

```
public class Main {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
```

Don't worry if you don't understand the code above - we will discuss it in detail in later chapters. For now, focus on **how** to run the code above.

Save the code in Notepad as "Main.java". Open Command Prompt (cmd.exe), navigate to the directory where you saved your file, and type "javac Main.java":

```
C:\Users\Your Name>javac Main.java
```

This will compile your code. If there are no errors in the code, the command prompt will take you to the next line. Now, type "java Main" to run the file:

```
C:\Users\Your Name>java Main
```

The output should read:

```
Hello World
```

Congratulations! You have written and executed your first Java program.

Java Syntax

In the previous chapter, we created a Java file called `Main.java`, and we used the following code to print "Hello World" to the screen:

Main.java Example explained

Every line of code that runs in Java must be inside a class. In our example, we named the class **Main**. A class should always start with an uppercase first letter.

Note: Java is case-sensitive: "MyClass" and "myclass" has different meaning.

The name of the java file **must match** the class name. When saving the file, save it using the class name and add ".java" to the end of the filename. To run the example above on your computer, make sure that Java is properly installed: Go to the Get Started Chapter for how to install Java. The output should be:

```
Hello World
```

The main Method

The `main()` method is required and you will see it in every Java program:

```
public static void main(String[] args)
```

Any code inside the `main()` method will be executed. Don't worry about the keywords before and after main. You will get to know them bit by bit while reading this tutorial.

For now, just remember that every Java program has a `class` name which must match the filename, and that every program must contain the `main()` method.

System.out.println()

Inside the `main()` method, we can use the `println()` method to print a line of text to the screen:

```
public static void main(String[] args) {
  System.out.println("Hello World");
}
```

Note: The curly braces `{}` marks the beginning and the end of a block of code.

`System` is a built-in Java class that contains useful members, such as `out`, which is short for "output". The `println()` method, short for "print line", is used to print a value to the screen (or a file).

Don't worry too much about `System`, `out` and `println()`. Just know that you need them together to print stuff to the screen.

You should also note that each code statement must end with a semicolon `(;)`.

Exercise:

Insert the missing part of the code below to output "Hello World".

```
public class MyClass {
  public static void main(String[] args) {
    ____.____.____("Hello World");
  }
}
```

Our Java Lessons

Java From Scratch

- Lesson 1 PDF (Java Getting Started)
- Lesson 2 PDF (Java Output, Comments, and Variables)
- Lesson 3 PDF (Java Data Types and Casting)
- Lesson 4 PDF (Java Operators and Strings)
- Lesson 5 PDF (Java Math and Booleans)
- Lesson 6 PDF (JAVA IF ELSE AND SWITCH)
- Lesson 7 PDF (Java While Loop and For Loop)
- Lesson 8 PDF (Java Arrays)