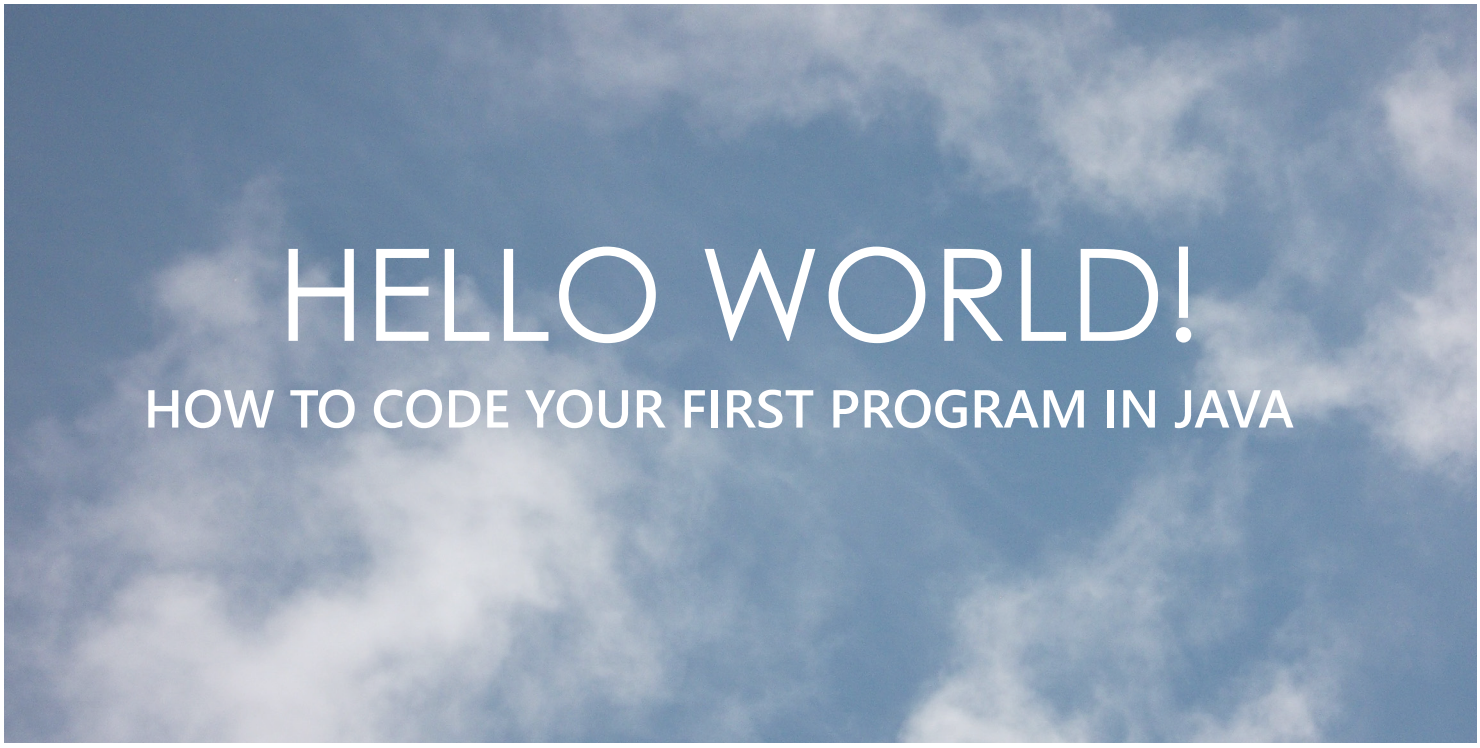


HELLO WORLD!

HOW TO CODE YOUR FIRST PROGRAM IN JAVA

WHY SHOULD YOU LEARN TO CODE?

WHAT YOU'LL NEED...

- **a Windows operating computer**
- **internet access**

Although we can't see it, code is all around us wherever we go. It's in the phone you use, the apps *on* it you use, your laptop, the websites you visit, even your car!

For something so widely used, only a small percentage of people in the world can actually understand and create code themselves. Many people are intimidated by code before they even know what it is!

Code is written in the form of programming languages, which are nothing more than text. If you can write a sentence, then you can code. It's not going anywhere anytime soon so you might as well learn it for yourself! This will show you everything you need to get started writing your very own programs.

GETTING STARTED

Before we can actually start writing, we need to download some tools first.

step 1: download the Java development kit

The Java development kit (JDK) is required to write and run Java programs on our computer. To install it search on google "download jdk 8" and click on the first result from oracle.com to reach the page with following section on it:

Java SE Development Kit 8u241

You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.

○ Accept License Agreement ◉ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.94 MB	jdk-8u241-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.83 MB	jdk-8u241-linux-arm64-vfp-hflt.tar.gz
Linux x86	171.28 MB	jdk-8u241-linux-i586.rpm
Linux x86	186.1 MB	jdk-8u241-linux-i586.tar.gz
Linux x64	170.65 MB	jdk-8u241-linux-x64.rpm
Linux x64	185.53 MB	jdk-8u241-linux-x64.tar.gz
Mac OS X x64	254.06 MB	jdk-8u241-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	133.01 MB	jdk-8u241-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.24 MB	jdk-8u241-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.8 MB	jdk-8u241-solaris-x64.tar.Z
Solaris x64	92.01 MB	jdk-8u241-solaris-x64.tar.gz
Windows x86	200.86 MB	jdk-8u241-windows-i586.exe
Windows x64	210.92 MB	jdk-8u241-windows-x64.exe

Accept the license agreement, and then download the correct version for your computer (either Windows x86 or Windows x64).

Run the installer of whichever one you downloaded. Accept the default settings and follow the instructions to install the JDK.

What version of Windows do I have?

1. search "system properties" from the start menu and hit enter
2. scroll down to "System Type" - this is the version of Windows you have!

step 2: set up the JDK to be used

We will be running our programs using the Command Prompt on our computer (start menu --> "Command Prompt"). The Command Prompt searches through every location listed as an *Environmental Variable* in something called Path. We need to add the location of our JDK to this list in Path.

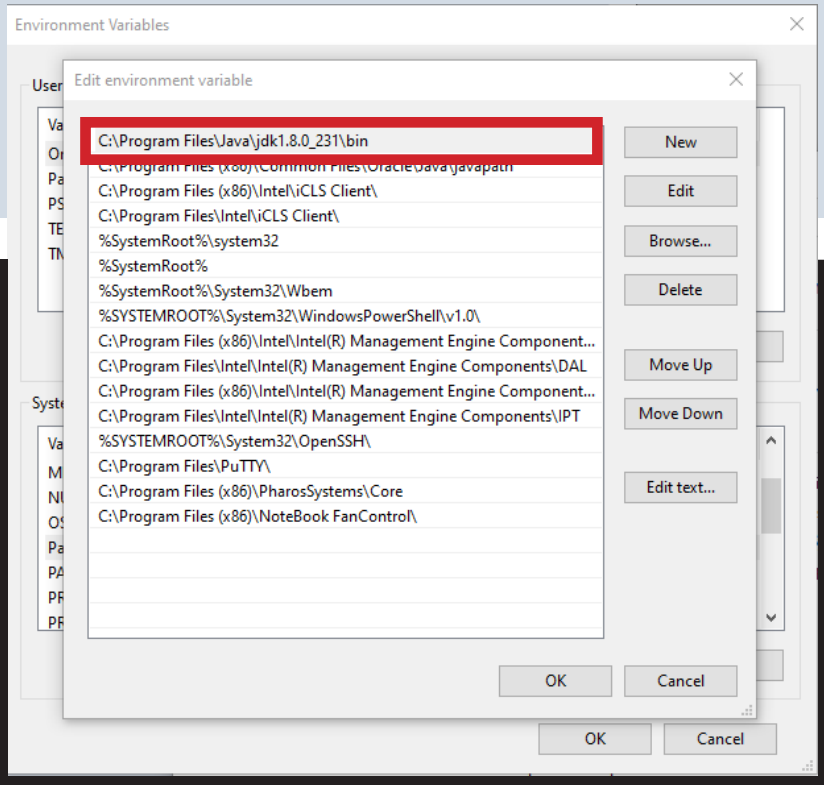
To find and copy the location of your JDK:

- Start menu --> "This PC" --> Windows (C:) --> Program Files OR Program Files (x86) --> Java --> jdk1.8.0_231 --> bin
- Click the Location at the top of your file explorer to highlight the text, and then type ctrl+c to copy the text to your clipboard.

Afterwords, add the location of JDK to the list of Environmental Variables:

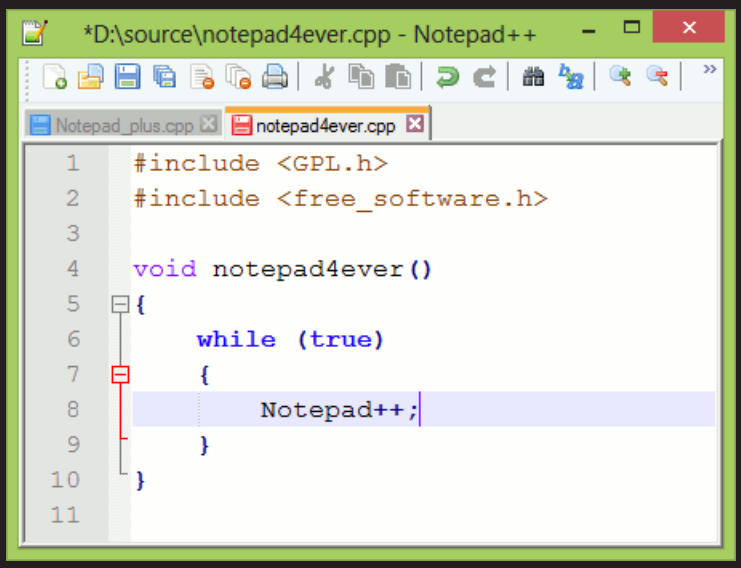
- Go to start menu --> "Control Panel" --> System --> Advanced System Settings
- Click the Advanced tab --> Environmental Variables
- In the bottom table labelled "System Variables," scroll down to the variable Path
- Click Path
- Click edit
- Click New
- Type ctrl+v to paste the location of the JDK we copied earlier.
- Click Move Up until the entry is all the way at the top of the list as shown below:
- Click Ok

We're done with the hardest part. *Phew!*

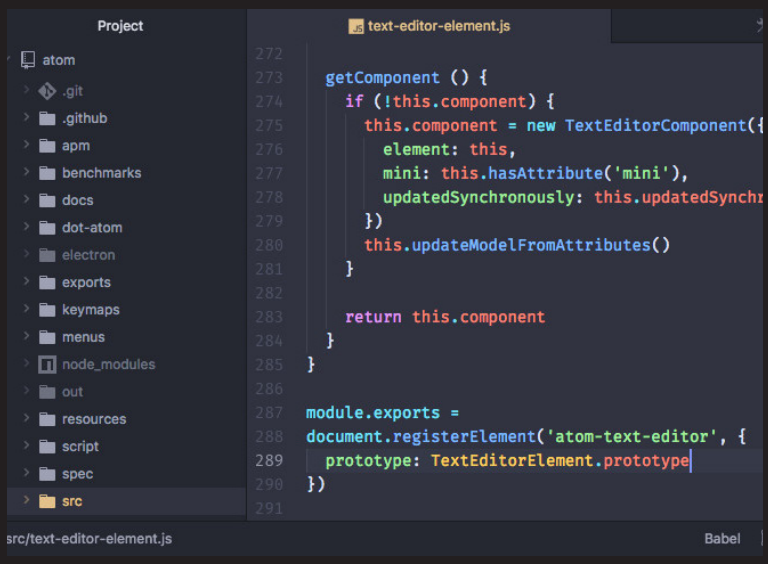


step 3: download a text editor

Next, let's find a text-editor to write your program with. These are just like Notepad on your computer except with more features. Popular free ones include Notepad++ and Atom.



notepad++



atom

You can search either of these on google to find their website and follow the instructions there to download them.

Once you have one downloaded, open it up to start coding!

Now that we have everything set up, we can finally begin writing our **Hello World** program.

hello world!

This refers to a program that simply prints the text “Hello World!” onto the screen. This exercise is commonly used as an introduction to a new programming language.

WRITING YOUR PROGRAM

step 1: save your file

Before we write any code, save the file to a convenient location. Let's name the file **HelloWorld.java** and save it right to our Desktop.

step 2: write your program

To write your program, type the following text exactly as written:

```
public class HelloWorld {

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

System.out.println

This is what we call a statement. Instead of a period (".") you end all statements with a semi-colon (";"). This one tells the system to print out a line of text.

System.out.println("Hello World");

This says that we want to print out a line of text, and we want that text to say **Hello World!**

What about all the other stuff?

public class HelloWorld { ... }

In Java, every line of code that can actually be run needs to be added inside of a class. The brackets are used to define the scope of your class, or where it begins and ends. Notice how when we declare our class, we must declare it inside a file with the same name (in this case, HelloWorld.java).

public static void main(String[] args) { ... }

This is a *method* within our *class* called main. This is the entry point of our program where our statements go (within it's brackets). You don't need to worry about what it all means yet, just that it must have this exact wording in order to work.

A method is just a list of statements, such as our statement that prints "Hello World" to the screen. **Every Java program must have a main method.** However, you can have many methods within your class alongside the main method.

A **public** method means it is accessible by anyone and by other methods within your class. If you were writing a method with sensitive data, you would delclare it as **private** instead. **void** simply means the method does not return any type of value at the end. If say you had a method that added two numbers together and returned the answer, for instance, the return type of that method would be **int**, which stands for Integer, instead of **void**.

step 3: save your file

Save your file once again to update the changes we made.

step 4: run your program

You've finally made it! Open the Command Prompt and type "cd desktop" and hit enter. Next type "javac HelloWorld.java" and finally "java HelloWorld".

The phrase **Hello World!** should print out to the screen. You've coded your first program! It may not do much, but you now have the tools at your disposal to code anything you set your mind to.

So what now?

With programming, the sky is the limit. You could try and make your own calculator, you could make your own text-based game, you could make a random number or name generator, basically anything you can imagine you can turn into code!

There are tons of resources online that are great introductions to getting you started with making more complex programs, or learning other languages besides Java. A great example is **www.codeacademy.com**. Code Academy is completely free and an amazing tool for novice coders like yourself.

Now get out there and start coding!