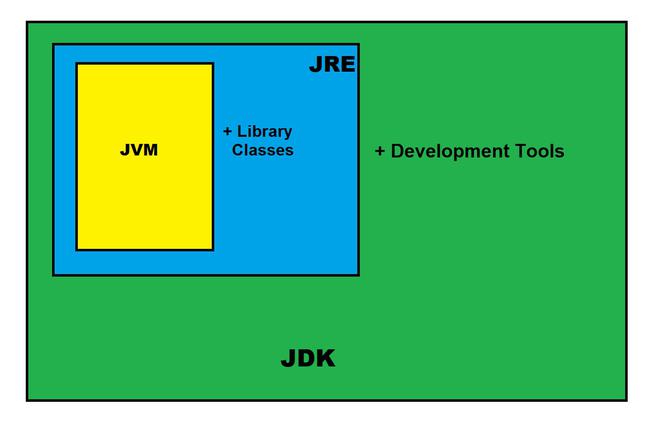
JAVA

Developed by Sun Microsystems in 1995, Java is a highly popular, object-oriented programming language. This platform independent programming language is utilized for Android development, web development, artificial intelligence, cloud applications.



**Most Popular JDKs:**

* **Oracle JDK:** the most popular JDK and the main distributor of Java11,
* **OpenJDK:** Ready for use: JDK 15, JDK 14, and JMC,
* **Azul Systems Zing:** efficient and low latency JDK for Linux os,
* **Azul Systems:** based Zulu brand for Linux, Windows, Mac OS X,
* **IBM J9 JDK:** for AIX, Linux, Windows, and many other OS,
* **Amazon Corretto:** the newest option with the no-cost build of OpenJDK and long-term support.

### Compile and Run Java Code using JDK:

Your Java text segment is converted into **bytecode** after compilation which carries the **.class** extension.

**The Jar component:**

JDK contains many useful tools and among them, the most popular after javac is the jar tool. The jar file is nothing but a full pack of Java classes. After creating the .class files, you can put them together in a .jar, which compresses and structures them in a predictable fashion. Now, let’s convert our Hello.class to a jar file.

Before proceeding, please note that you should be in the same directory where the Hello.java file was saved. Now type the command given below in the command line.

**Creating a .jar file**

*C:\Users\Pinaki\Documents>”c:\Program Files\Java\jdk-11.0.9\bin\jar.exe” –create –file Hello.jar Hello.class*

Now you can notice that Hello.jar file had been created in the same directory using Hello.class file and jar.exe. You can use the jar file by adding it to your classpath and executing the program inside it. Here the -cp stands for classpath which helps to add the jar to the same classpath.

**Executing the .jar file**

*java -cp hello\_world.jar hello\_world*

**Important Components of JDK**

Below there is a comprehensive list of mostly used components of Jdk which are very useful during the development of a java application.

| **Component** | **Use** |
| --- | --- |
| javac | Java compiler converts source code into Java bytecode |
| java | The loader of the java apps. |
| javap | Class file disassembler, |
| javadoc | Documentation generator, |
| jar | Java Archiver helps manage JAR files. |
| appletviewer | Debugging of Java applets without a web browser, |
| xjc | Accepts an XML schema and generates Java classes, |
| apt | Annotation-processing tool, |
| jdb | Debugger, |
| jmc | Java Mission Control, |
| JConsole | Monitoring and Management Console, |
| pack200 | JAR compression tool, |
| extcheck | Utility tool to detects JAR file conflicts, |
| idlj | IDL-to-Java compiler, |
| keytool | The keystore manipulating tool, |
| jstatd | jstat daemon (experimental) |
| jstat | JVM statistics monitoring tool |
| jshell | jshell introduced in java 9. |
| jstack | Prints Java stack traces(experimental) |
| jrunscript | Java command-line script shell. |
| jhat | Java Heap Analysis Tool (experimental) |
| jpackage | Generate self-contained application bundles. |
| javaws | Web Start launcher for JNLP applications, |
| javah | C header and stub generator, |
| jarsigner | jar signing and verification tool |
| jinfo | configuration information(experimental) |
| javafxpackager | Package and sign JavaFX applications |

**Data Types in Java**

**Data types in Java** are of different sizes and values that can be stored in the variable that is made as per convenience and circumstances to cover up all test cases. Java has two categories in which data types are segregated

1. **Primitive Data Type:** such as boolean, char, int, short, byte, long, float, and double
2. **Non-Primitive Data Type or Object Data type:** such as String, Array, etc.

**Java Programming Tutorial: FAQs**

**Why use Java ?**

Java is simple to learn programming language because doesn’t contain concepts like : Pointers and operator overloading and it is secure and portable.

**What are the major concepts in Java?**

There are 4 major concept in Java that is abstraction, encapsulation, polymorphism, and inheritance. Along with this Java also works with three OOPs concept

**How Java different to C++ ?**

| **C++** | **JAVA** |
| --- | --- |
| C++ is platform dependent. | Java is platform independent. |
| C++ uses compiler only. | Java uses compiler and interpreter both. |
| C++ support pointers and operator overloading. | Java doesn’t support pointers and operator overloading concept. |
| C++ not support multithreading concept. | Java supports multithreading concept. |

**Why Java is so popular programming language?**

Java is based on object model hence it is one the popular programming language.

**What are the scope of Java Technologies for Web Applications?**

Java Technologies for Web Applications are a set of Java-based technologies that are used to develop web applications. These technologies include:

* Java Servlet API
* JavaServer Pages
* JavaServer Faces
* Enterprise JavaBeans
* JDBC (Java Database Connectivity)
* Java Messaging Service (JMS):
* JavaMail API:
* JAX-WS