



What is JDK





Introduction :

- It is a bundle of software development tools and supporting libraries combined with the Java Runtime Environment and Java Virtual Machine.
- The JDK provides the tools necessary to write Java programs that can be executed and run by the JVM and JRE.
- **JDK contains:**
- Java Runtime Environment (JRE),
- An interpreter/loader (Java),
- A compiler (javac),
- An archiver (jar) and many more.





- **Compile with the JDK :**
- We'll start by compiling and running a Java program the old-fashioned way. with text files and console commands.

```
class Demo {  
  
    public static void main(String[] args){  
  
        System.out.println("JDK!");  
    }  
}
```





- use the JDK compiler to turn this text file into an executable program. Compiled code in Java is known as bytecode, and carries the .class extension.
- we'll use the javac command, passing the Demo.java file as the argument to the command.

```
javac Demo.java
```

- That should result in a successful compile. It will just output the new file. Any errors will result in console output.
- now we have the Demo.class file in the same directory as Demo.java. run it by typing: **java Demo**, which will result in the output.





- **Create a JAR file :**
- The javac is the star of the JDK, but the /bin directory contains other tools you will need. Probably the most prominent after javac is the jar tool.
- A JAR (.jar) file is a packaged set of Java classes. Once the compiler has created the .class files, we can put them together in a .jar, which compresses and structures them in a predictable fashion.
- command to convert Demo.class to a Demo.jar file.
jar cf Demo.jar Demo.class
- Now we'll have an Demo.jar file in the directory. We can make use of the .jar by adding it to our classpath and executing the program inside.

