Introduction to Java

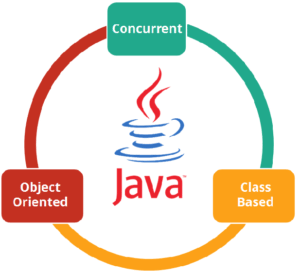
Java is a powerful general-purpose programming language. It is used to develop desktop and mobile applications, big data processing, embedded systems, and so on. According to Oracle, the company that owns Java, Java runs on 3 billion devices worldwide, which makes Java one of the most popular programming languages.

**About Java Programming**

* **Platform independent** - We can write Java code in one platform (operating system) and run on another platform without any modification.
* **Object-oriented** - Java is an object-oriented language. This helps to make our Java code more flexible and reusable.
* **Speed** - Well optimized Java code is nearly as fast as lower-level languages like C++ and much faster than Python, PHP, etc.

**Why Learn Java?**

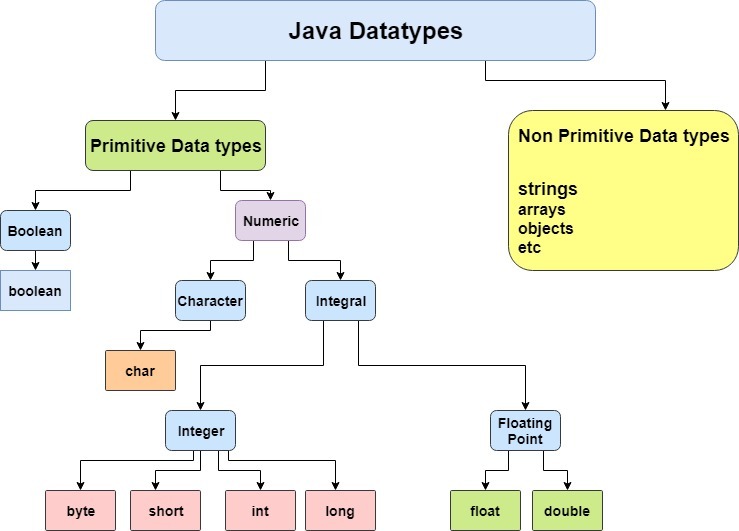
* Java is a platform-independent language. We can write Java code in one platform and run it in another platform
* Java is a general-purpose language with a wide range of applications. It's used for developing mobile and desktop applications, big data processing, embedded systems, and so on.
* Java is an object-oriented programming language. It helps in code reusability.



Java Features



Java Data Types



## ****Components in Java****

**JVM (Java Virtual Machine)**

It is an abstract machine. It is a specification that provides a run-time environment in which Java bytecode can be executed. It follows three notations:

* **Specification**: It is a document that describes the implementation of the Java virtual machine. It is provided by Sun and other companies.
* **Implementation**: It is a program that meets the requirements of JVM specification.
* **Runtime Instance**: An instance of JVM is created whenever you write a java command on the command prompt and run the class.

**JRE (Java Runtime Environment)**

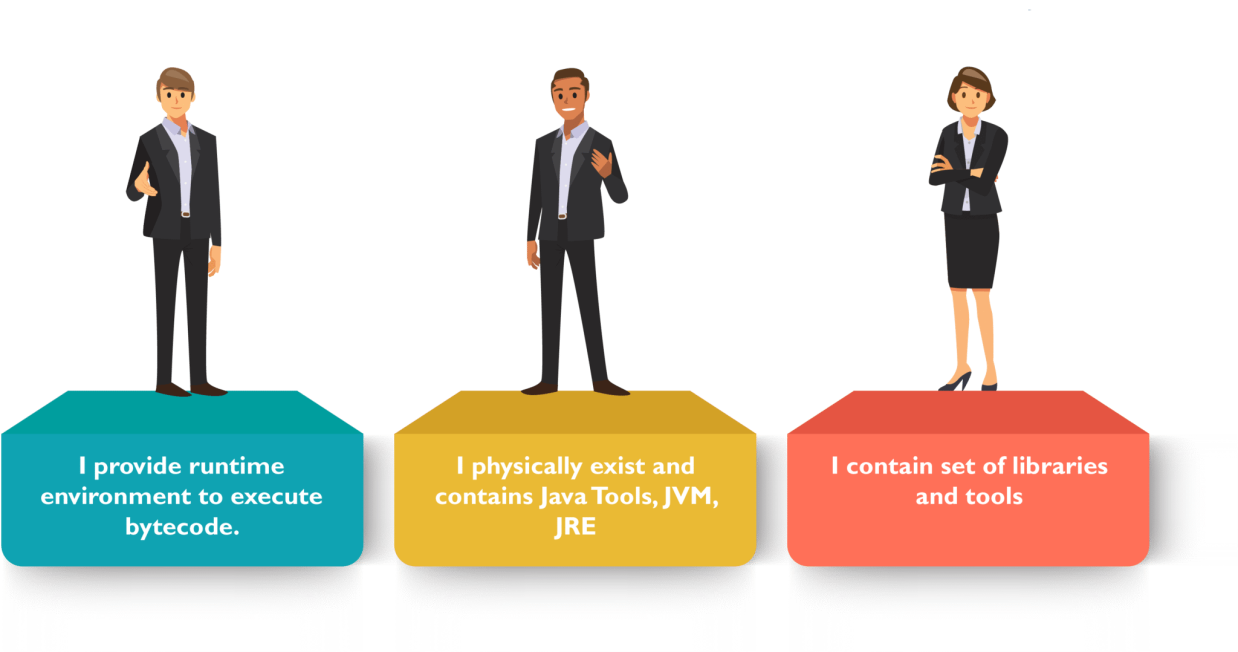
JRE refers to a runtime environment in which Java bytecode can be executed. It implements the JVM (Java Virtual Machine) and provides all the class libraries and other support files that JVM uses at runtime. So JRE is a software package that contains what is required to run a Java program. Basically, it’s an implementation of the JVM which physically exists.

**JDK(Java Development Kit)**

It is the tool necessary to:-

* Compile
* Document
* Package Java programs.

The JDK completely includes JRE which contains tools for Java programmers. The Java Development Kit is provided free of charge. Along with JRE, it includes an interpreter/loader, a compiler (javac), an archiver (jar), a documentation generator (Javadoc) and other tools needed in Java development. In short, it contains JRE + development tools.



### Java Flow Control

In computer programming, it's often desirable to execute a certain section of code based upon whether the specified condition is true or false (which is known only during the run time). For such cases, control flow statements are used.

**Java if (if-then) Statement**

In Java, the syntax of the **if-then** statement is:

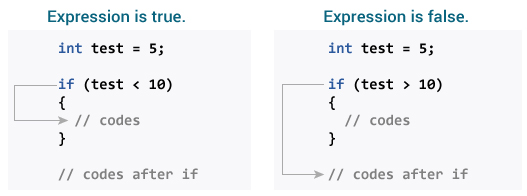
if (expression) {

// statements

}

Here expression is a boolean expression. A boolean expression returns either true or false.

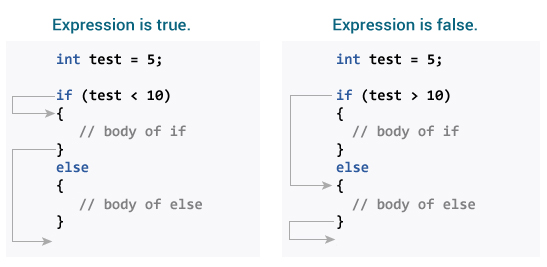
* if the expression is evaluated to true, statement(s) inside the body of if (statements inside parenthesis) are executed
* if the expression is evaluated to false, statement(s) inside the body of if are skipped from execution



**Java if...else (if-then-else) Statement**

The if statement executes a certain section of code if the test expression is evaluated to true. However, if the test expression is evaluated to false, it does nothing.

In this case, we can use an optional else block. Statements inside the body of else block are executed if the test expression is evaluated to false. This is known as the **if-then-else** statement in Java.



**Java if..else..if Statement**

In Java, we have an **if...else...if** ladder, that can be used to execute one block of code among multiple other blocks.

if (expression1) {

// codes

}

else if(expression2) {

// codes

}

else if (expression3) {

// codes

}

.

.

else {

// codes

}

Here, if statements are executed from the top towards the bottom. As soon as the test expression is true, codes inside the body of that the if statement is executed. Then, the control of the program jumps outside the if-else-if ladder.

If all test expressions are false, codes inside the body of else is executed.

**Java Nested if..else Statement**

In Java, it is also possible to if..else statements inside a if..else statement. It's called nested if...else statement.

# Java switch Statement

In Java, we have used the [if..else..if ladder](https://www.programiz.com/java-programming/if-else-statement" \l "if-else-ladder" \o "Java if..else..if ladder) to execute a block of code among multiple blocks. However, the syntax of if...else...if ladders are too long.

Hence, we can use the switch statement as a substitute for long **if...else...if** ladders. The use of switch statements makes our code more readable.

The syntax of the switch statement is:

switch (variable/expression) {

case value1:

// statements of case1

break;

case value2:

// statements of case2

break;

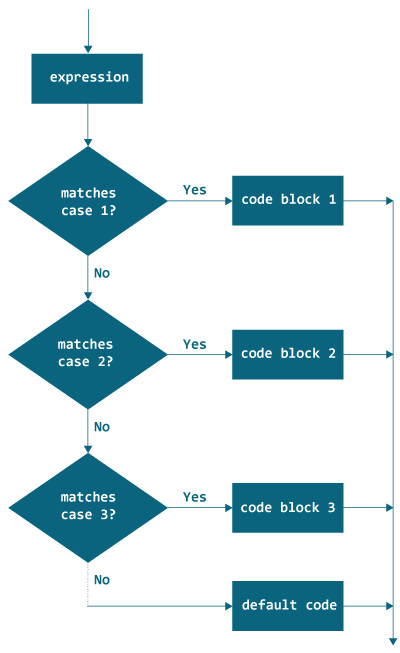
.. .. ...

.. .. ...

default:

// default statements

}



# Java while and do...while Loop

In computer programming, loops are used to repeat a specific block of code until a certain condition is met (test expression is false). For example,

Imagine we need to print a sentence 50 times on your screen. Well, we can do it by using the print statement 50 times (without using loops). How about you need to print a sentence one million times? You need to use loops. With loops, we can simply write the print statement one time and run it for any number of times.

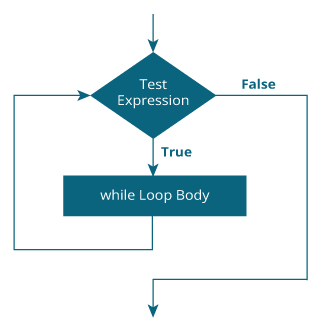
It's just a simple example showing the importance of loop in computer programming. There are 3 types of loops in Java: [for loop](https://www.programiz.com/java-programming/for-loop), while loop, and do-while loop.

### How while loop works?

* statements inside the while loop are executed.
* then, the test expression is evaluated again.

This process goes on until the test expression is evaluated to false. If the test expression is evaluated to false,

* the while loop is terminated.



## Java do...while Loop

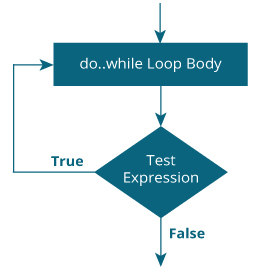
The do...while loop is similar to while loop with one key difference. The body of do...while loop is executed for once before the test expression is checked.

### How do...while loop works?

The body of do...while loop is executed once (before checking the test expression). Only then, the test expression is checked.

If the test expression is evaluated to true, codes inside the body of the loop are executed, and the test expression is evaluated again. This process goes on until the test expression is evaluated to false.

When the test expression is false, the do..while loop terminates.



# Java for Loop

In computer programming, loops are used to repeat a specific block of code until a certain condition is met (test expression is false). For example,

## Java for Loop

The syntax of for Loop in Java is:

for (initialization; testExpression; update)

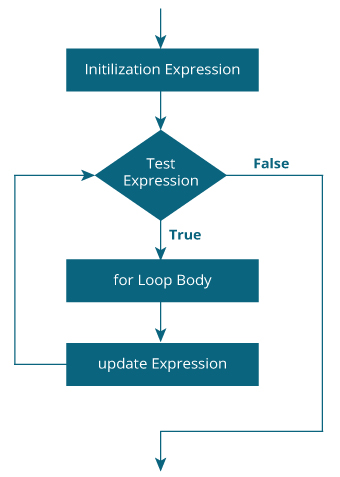
{

// codes inside for loop's body

}

### Working of for loop

1. The **initialization** expression is executed only once.
2. Then, the **test** expression is evaluated. Here, test expression is a boolean expression.
3. If the test expression is evaluated to true,  
   Codes inside the body of for loop is executed.  
   Then the **update** expression is executed.  
   Again, the test expression is evaluated.  
   If the test expression is true, codes inside the body of for loop is executed and update expression is executed.  
   This process goes on until the test expression is evaluated to false.
4. If the test expression is evaluated to false, for loop terminates.



# Java Arrays

An array is a collection of similar types of data. It is a container that holds data (values) of one single type. For example, you can create an array that can hold 100 values of int type.

In Java, arrays are a fundamental construct that allows you to store and access a large number of values conveniently.

## How to declare an array?

In Java, here is how we can declare an array.

dataType[] arrayName;

* dataType - it can be [primitive data types](https://www.programiz.com/java-programming/variables-primitive-data-types#data-types) like int, char, double, byte, etc. or [Java objects](https://www.programiz.com/java-programming/class-objects)
* arrayName - it is an [identifier](https://www.programiz.com/java-programming/keywords-identifiers#identifiers)

Let's take an example,

double[] data;

Here, data is an array that can hold values of type double.

**But, how many elements can array this hold?**

Good question! We have to allocate memory for the array. The memory will define the number of elements that the array can hold.

data = new Double[10];

Here, the size of the array is 10. This means it can hold 10 elements (10 double types values). The size of an array is also known as the length of an array.

**Note**: Once the length of the array is defined, it cannot be changed in the program.

Let's take another example:

int[] age;

age = new int[5];

Here, age is an array. It can hold 5 values of int type.

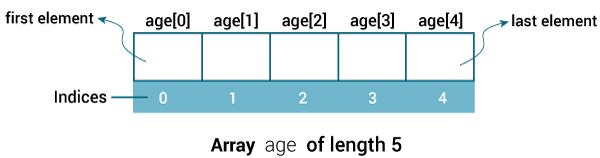
In Java, we can declare and allocate memory of an array in one single statement. For example,

int[] age = new int[5];

**Java Array Index**

In Java, each element in an array is associated with a number. The number is known as an array index. We can access elements of an array by using those indices. For example,

int[] age = new int[5];



Java Array Index