**LOOPING STATEMENTS IN JAVA**

* Facilitates the execution of a set of instructions/functions repeatedly while some condition evaluates to true.
* Java provides three ways for executing the loops.
* **For**

A simple for loop is the same as [C](https://www.javatpoint.com/c-programming-language-tutorial)/[C++](https://www.javatpoint.com/cpp-tutorial). We can initialize the [variable](https://www.javatpoint.com/java-variables), check condition and increment/decrement value. It consists of four parts:

1. Initialization: It is the initial condition which is executed once when the loop starts. Here, we can initialize the variable, or we can use an already initialized variable. It is an optional condition.
2. Condition: It is the second condition which is executed each time to test the condition of the loop. It continues execution until the condition is false. It must return boolean value either true or false. It is an optional condition.
3. Increment/Decrement: It increments or decrements the variable value. It is an optional condition.
4. Statement: The statement of the loop is executed each time until the second condition is false.

**Syntax:**

1. for(initialization; condition; increment/decrement){
2. //statement or code to be executed
3. }

* **While**
* The [Java](https://www.javatpoint.com/java-tutorial) while loop is used to iterate a part of the [program](https://www.javatpoint.com/programs-list) repeatedly until the specified Boolean condition is true. As soon as the Boolean condition becomes false, the loop automatically stops.
* The while loop is considered as a repeating if statement. If the number of iteration is not fixed, it is recommended to use the while [loop](https://www.javatpoint.com/java-for-loop).

**Syntax:**

1. while (condition){
2. //code to be executed
3. I ncrement / decrement statement
4. }

* **Do…while**
* The Java do-whileloop is used to iterate a part of the program repeatedly, until the specified condition is true. If the number of iteration is not fixed and you must have to execute the loop at least once, it is recommended to use a do-while loop.
* Java do-while loop is called an exit control loop. Therefore, unlike while loop and for loop, the do-while check the condition at the end of loop body. The Java do-while loop is executed at least once because condition is checked after loop body

**Syntax:**

1. do{
2. //code to be executed / loop body
3. //update statement
4. }while (condition);

* **For-Each Loop**
* The for-each loop is used to traverse array or collection in Java. It is easier to use than simple for loop because we don't need to increment value and use subscript notation.
* It works on the basis of elements and not the index. It returns element one by one in the defined variable.

**Syntax:**

1. for(data\_type variable : array\_name){
2. //code to be executed
3. }

**BRANCHING STATEMENTS IN JAVA**

• **Break**

* When a break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.
* The Java break statement is used to break loop or [switch](https://www.javatpoint.com/java-switch) statement. It breaks the current flow of the program at specified condition. In case of inner loop, it breaks only inner loop.
* We can use Java break statement in all types of loops such as [for loop](https://www.javatpoint.com/java-for-loop), [while loop](https://www.javatpoint.com/java-while-loop) and [do-while loop](https://www.javatpoint.com/java-do-while-loop).

**Syntax:**

1. jump-statement;
2. break;

• **Continue**

* The continue statement is used in loop control structure when you need to jump to the next iteration of the loop immediately. It can be used with for loop or while loop.
* The Java continue statement is used to continue the loop. It continues the current flow of the program and skips the remaining code at the specified condition. In case of an inner loop, it continues the inner loop only.
* We can use Java continue statement in all types of loops such as for loop, while loop and do-while loop.

**Syntax:**

1. jump-statement;
2. continue;

**JAVA COMMENTS**

* Statements in a program that are not executed by the compiler and interpreter.
* Why do we use comments in a code?
* To make the program more readable by adding the details of the code.
* To maintain the code and to find the errors easily.
* To provide information or explanation about the variable, method, class, or any statement.
* To prevent the execution of program code while testing the alternative code.

**JAVA ARRAY**

* Java Array is an object which contains of a similar data type.
* Elements of an array are stored in a contiguous memory location.
* It is a data structure where we store similar elements.
* We can store only a fixed set of elements in a Java array.
* Array in Java is index-based, the first element of the array is stored at the 0th index.
* 2nd element is stored on 1st index and so on.



• **Single Dimensional**   
 **Declaration of Array**

Syntax:

**type var-name[]; OR type[] var-name;**

**Instantiating an Array in Java**

Syntax:

**var-name =new type [size];**

Example:

Int intArray[]; // declaring array

IntArray = new int[20]; // allocating memory to array

• **Multi-Dimensional**

Here data is stored in row and column based index (also known as matrix form).

**Declaration of Multidimentional Array**

Syntax:

**dataType[][] arrayRefVar; (or)**

**dataType [][]arrayRefVar; (or)**

**dataType arrayRefVar[][]; (or)**

**dataType []arrayRefVar[];**

**Example to Instantiate Multidimentional Array in Java**

int[][] arr=new int[3][3]; //3 row and 3 column

**Example to initialize Multidimensional Array in Java**

arr[0][0]=1;

arr[0][1]=2;

arr[0][2]=3;

arr[1][0]=4;

arr[1][1]=5;

arr[1][2]=6;

arr[2][0]=7;

arr[2][1]=8;

arr[2][2]=9;