LOOPING STATEMENTS IN JAVA

Velocity

Looping Statements in java

Loops are used to execute a set of instructions repeatedly when some conditions become true. There are three types of loops in Java.

1. For loop
2. While loop
3. Do while loop

1.For loop

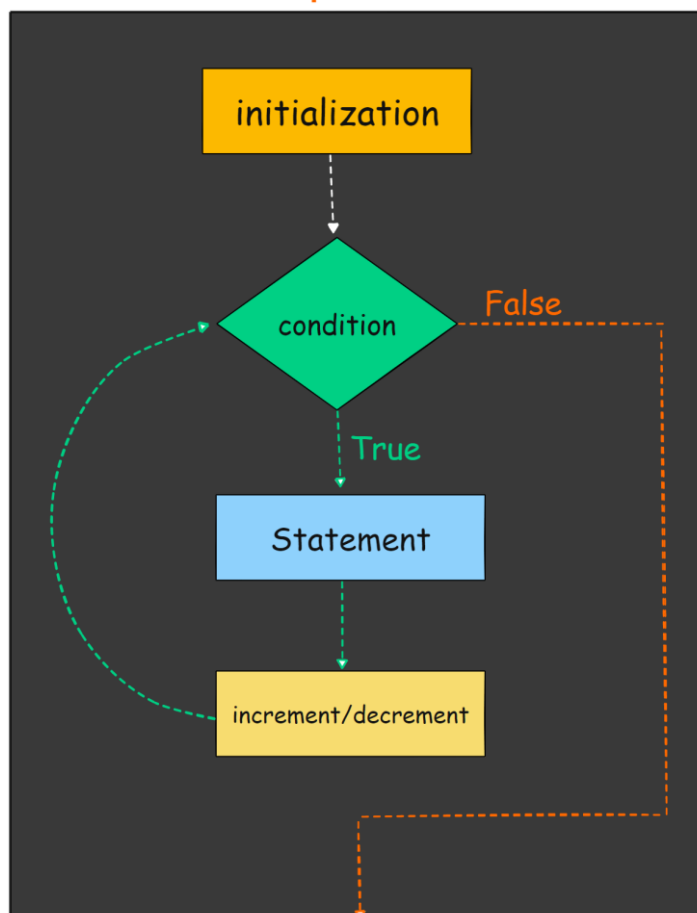For loop in java is used to iterate and evaluate a code multiple time.

When to use-

If the number of iterations is fixed, it is recommended to use for loop.

Syntax:

for (initialization; condition; increment/decrement) {

//statement or code to be executed

}

For Loop flow chart

Example- Different way to write the loops in java.

```java
public class Demo {

    public static void main(String[] args) {

        for (int i = 1; i <= 5; i++) {
            System.out.println("value of i>>" + i);
        }
    }
}
```

```java
public class Demo {

    public static void main(String[] args) {

        for (int i = 5; i>0; i--) {
            System.out.println("value of i>>" + i);
        }
    }
}
```

```java
public class Demo {

    public static void main(String[] args) {
        int i = 1;
        for (; i < 5; i++) {
            System.out.println("value of i>>" + i);
        }
    }
}
```

```java
1
2  public class Demo {
3
4      public static void main(String[] args) {
5
6          //Infinite loop
7          for (; ; ) {
8              System.out.println("infinite loop");
9          }
10         //Code after infinite loop will become unreachable
11 Unreachable code   System.out.println("Hello");
12     }
13 }
14
```

Note: Infinite Loop should not be designed in program

: The while loop evaluates a certain condition. If the condition is true, the code is executed. This process is continued until the specified condition turns out to be false.

When to use-

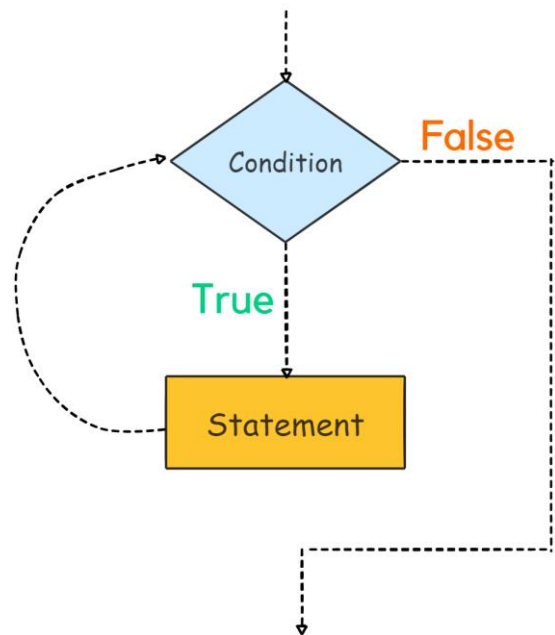If the number of iterations is not fixed, it is recommended to use while loop.

While loop flow diagram

Syntax-

Initialization;

While (condition) {

Increment/decrement operator;

}

Example-1

```java
public class Demo {

    public static void main(String[] args) {

        int i = 1; // Initialization
        while (i <= 10) {
            System.out.println("value of i>>" + i); //Statement
            i++; //Increment
        }

    }
}
```

```java
public class Demo {

    public static void main(String[] args) {

        int i = 10; // Initialization
        while (i>0) {
            System.out.println("value of i>>" + i); //Statement
            i--; //Decrement
        }

    }
}
```

```java
1
2  public class Demo {
3
4⊖     public static void main(String[] args) {
5
6          int i = 10;
7          // Infinite loop
8          while (true) {
9              System.out.println("value of i>>" + i); // Statement
10
11          }
12  Unreachable code  System.out.println(" Hello");
13      }
14 }
```

Note: Infinite Loop should not be designed in program

3.Do while loop: The do-while loop is similar to the while loop, the only difference being that the condition in the do-while loop is evaluated after the execution of the loop body. This guarantees that the loop is executed at least once.

When to use-

If the number of iterations is not fixed and you must have to execute the loop at least once, it is recommended to use the do-while loop

Syntax-

Initialization;

do {

Statement;

increment or decrement;

} while (condition);

do-while flow chart

```java
public class Demo {

    public static void main(String[] args) {

        int i = 1; // Initialization
        do {
            System.out.println(i); // Statement
            i++; // Increment
        } while (i <= 10);// Condition

    }
}
```
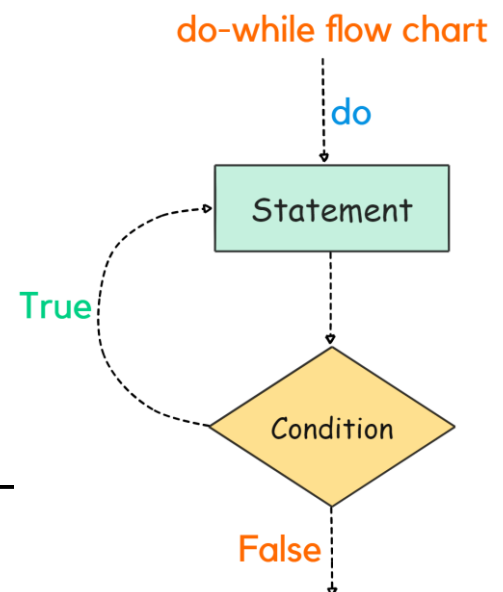
```java
public class Demo {

    public static void main(String[] args) {

        char ch = 'A';
        do {
            System.out.println(ch + " ");
            ch++;
        } while (ch <= 'Z');

    }
}
```

Comparison	for loop	while loop	do-while loop
Intro	The Java for loop is a control flow statement that iterates a part of the program's multiple times.	The Java while loop is a control flow statement that executes a part of the programs repeatedly on the basis of given Boolean condition.	The Java do while loop is a control flow statement that executes a part of the programs at least once and the further execution depends upon the given Boolean condition.
When to use	If the number of iterations is fixed.	If the number of iterations is not fixed.	If the number of iterations is not fixed and you must have to execute the loop at least once.

Syntax	**for**(init;condition;incr/<u>decr</u>){
	// code to be executed
	    }
		while(condition) {
		//code to be executed
		    }
			do {
			//code to be executed
			    }
			while(condition);

Example	//for loop
	for(**int**=1;i<=10;i++){
	System.***out***.println(i);
	}
		//while loop
		int i=1;
		while(i<=10){
		System.***out***.println(i);
		i++;
		}
			//do-while loop
			int i=1;
			do{
			System.***out***.println(i);
			i++;
			}**while**(i<=10);

Syntax for infinitive loop	for(;;){
	//code to be executed
	}
		while(true){
		//code to be executed
		}
			do{
			//code to be executed
			}while(true);