



Loops





Need Of Loop In Programming Language

Suppose there is a task where you want to print "Hello World" multiple times. Initially, you can write `System.out.println("Hello World")` to print it once. However, if you want to print it multiple times, instead of repeating the same code multiple times, you can utilize the concept of loops in the Java programming language. Loops allow you to write the code once and iterate over it based on the desired number of repetitions. This way, you can effectively reduce redundancy and avoid writing the same code repeatedly.



Syntax:

```
for (initialization expr; test expr; update exp)
{
    // body of the loop
    // statements we want to execute
}
```

For loop:

We can initialise the variable, check condition and increment/decrement value. It consists of four parts:

Initialization: It is the initial condition which is executed once when the loop starts. Here, we can initialise the variable, or we can use an already initialised variable. It is an optional condition.



Syntax:



```
for(initialization; condition; increment/decrement){  
  //statement or code to be executed  
}
```

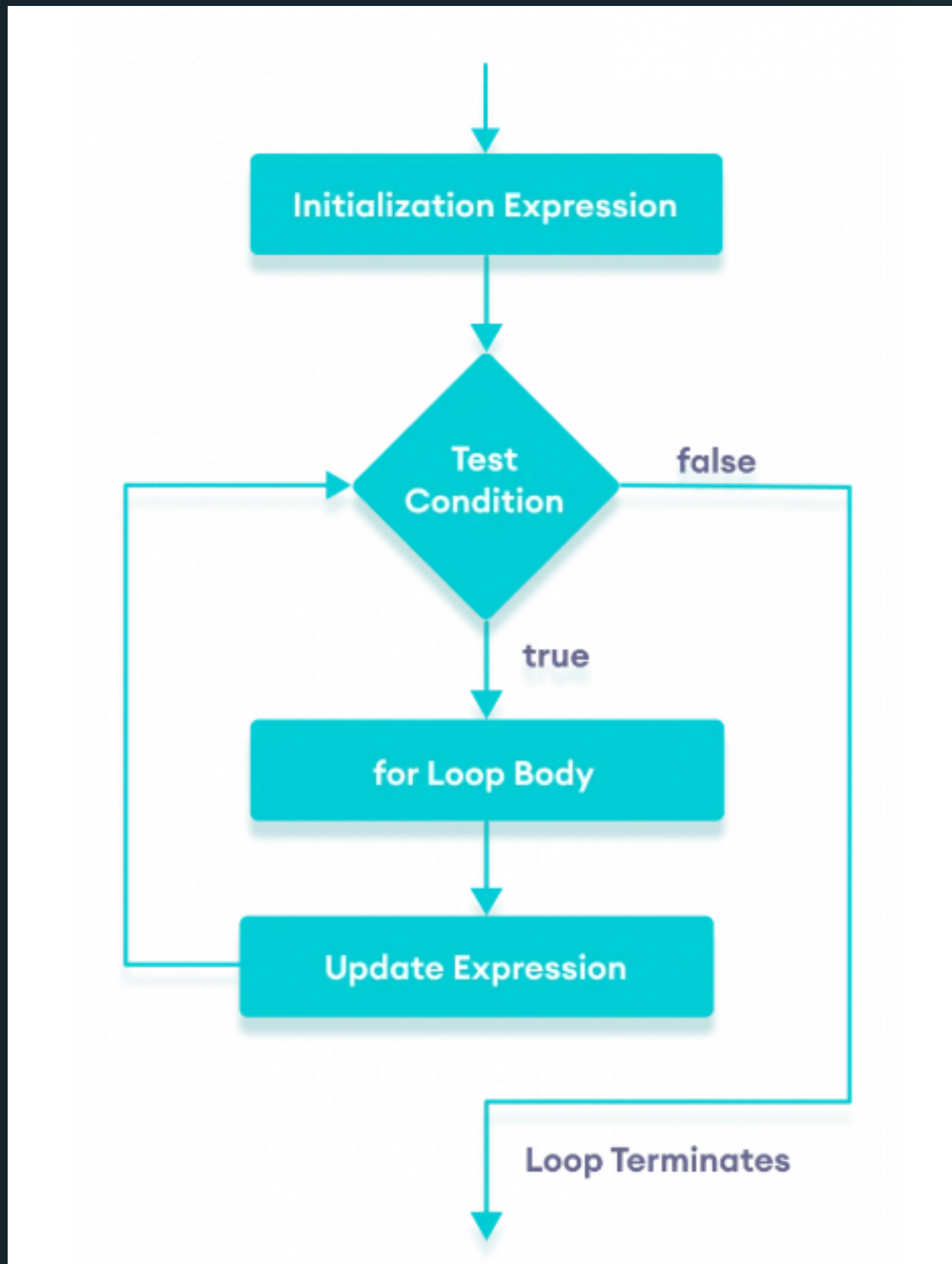
Condition: It is the second condition which is executed each time to test the condition of the loop. It continues execution until the condition is false. It must return a boolean value either true or false. It is an optional condition.

Increment/Decrement: It increments or decrements the variable value. It is an optional condition.

Statement: The statement of the loop is executed each time until the second condition is false.



Flowchart:





Example:



```
public class ForExample {  
    public static void main(String[] args) {  
        //Code of Java for loop  
        for(int i=1;i<=10;i++){  
            System.out.println("Hello World");  
        }  
    }  
}
```

Output:



```
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World
```



While Loop

So far, we have learned about the 'for' loop and its usage. Now, we are going to learn about another loop called the 'while' loop. The 'while' loop has the same purpose as the 'for' loop, but its syntax differs.

Here's the basic syntax of a while loop:



```
while (condition) {  
    // code to be executed repeatedly  
}
```



The condition is a boolean expression that is evaluated before each iteration of the loop. If the condition is true, the code inside the loop is executed. Then, the condition is evaluated again and if it's still true, the code is executed again, and so on, until the condition becomes false.

Syntax:

```
int i = 1;
while (i <= 5) {
    System.out.println(i);
    i++;
}
```




In this example, the loop starts with $i = 1$. The condition ($i \leq 5$) is true, so the code inside the loop (`System.out.println(i);` and `i++;`) is executed. The value of i is incremented by 1, so now $i = 2$. The condition is checked again and if it's true, the code inside the loop is executed again, and so on, until $i = 6$. At that point, the condition ($i \leq 5$) becomes false, and the loop terminates.

In general, you should use a for loop when you know how many times the loop should run. If you want the loop to break based on a condition other than the number of times it runs, you should use a while loop.