

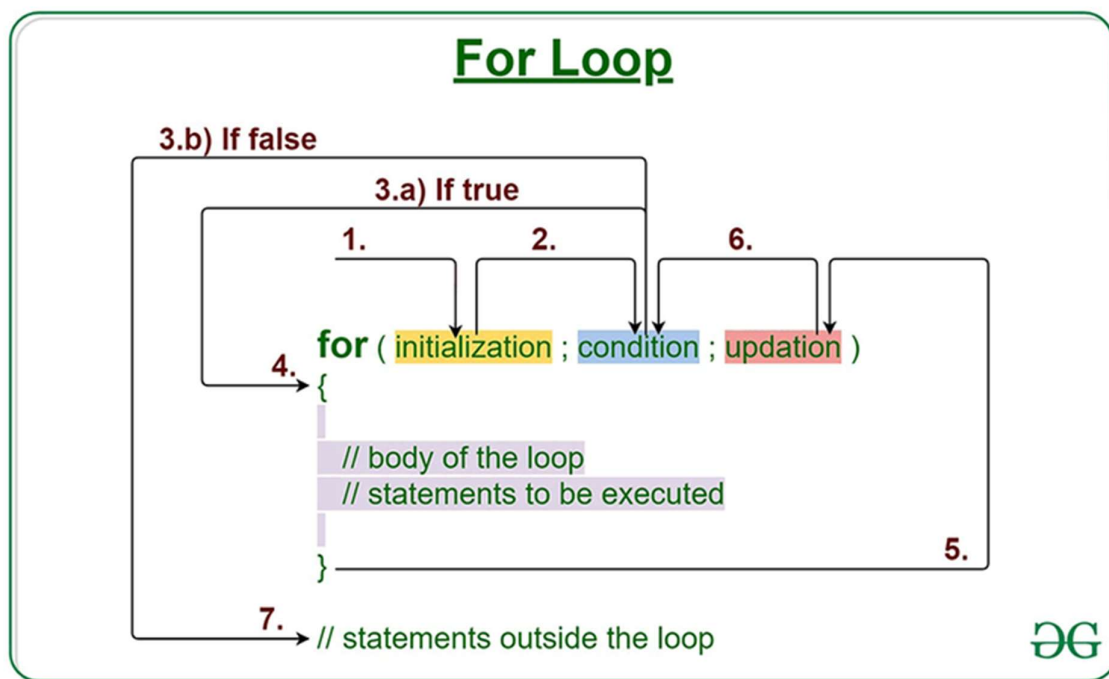
Looping statement

Looping in programming languages is a feature. Whenever we have to repeat certain statement, several times is called loop.

Types:

1. For loop
2. While loop
3. Do-while
4. For-each loop

For loop



Syntax:

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

Parts of Java For Loop

Java for loop is divided into various parts as mentioned below:

- Initialization Expression
- Test Expression
- Update Expression
-

1. Initialization Expression

In this expression, we have to initialize the loop counter to some value.

Example:

```
int i=1;
```

2. Test Expression

In this expression, we have to test the condition. If the condition evaluates to true then, we will execute the body of the loop and go to the update expression. Otherwise, we will exit from the for a loop.

Example:

```
i <= 10
```

3. Update Expression:

After executing the loop body, this expression increments/decrements the loop variable by some value.

Example:

```
i++;
```

How does a For loop work?

1. Control falls into the for loop. Initialization is done
2. The flow jumps to Condition
3. Condition is tested.
 - If the Condition yields true, the flow goes into the Body
 - If the Condition yields false, the flow goes outside the loop
4. The statements inside the body of the loop get executed.
5. The flow goes to the Updation
6. Updation takes place and the flow goes to Step 3 again
7. The for loop has ended and the flow has gone outside.

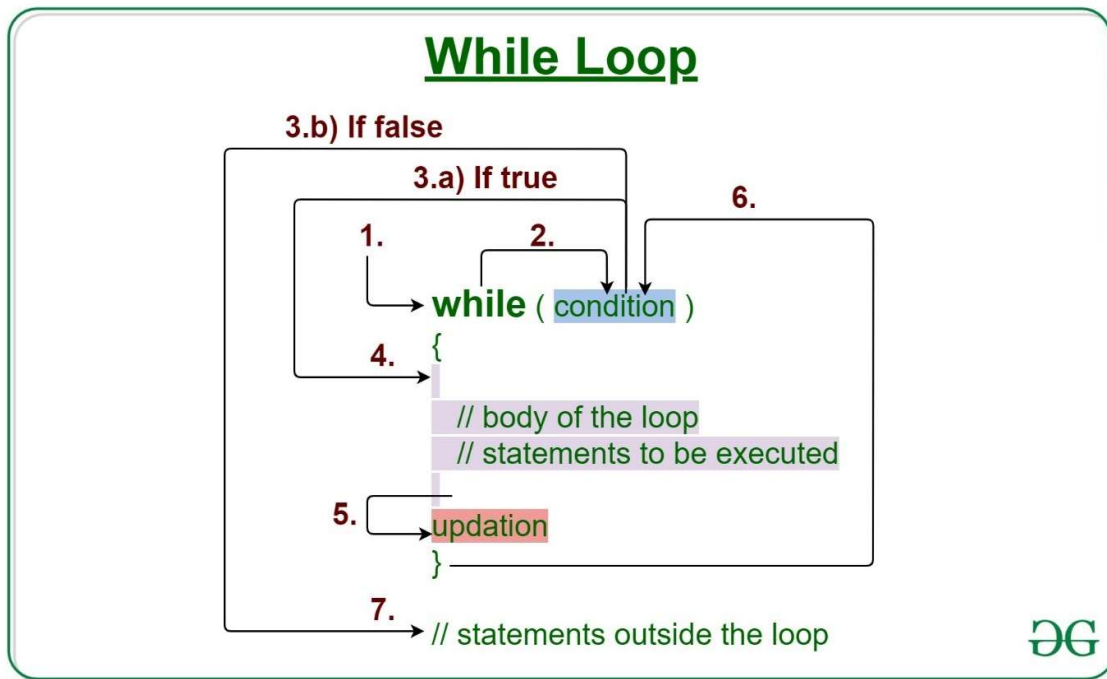
```

class Test {
    public static void main(String[] args)
    {
        for (int i = 1; i <= 10; i++) {
            System.out.println(i);
        }
    }
}

```

while loop

Java while loop is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition. The while loop can be thought of as a repeating if statement. While loop in Java comes into use when we need to repeatedly execute a block of statements. The while loop is considered as a repeating if statement. If the number of iterations is not fixed, it is recommended to use the while loop.



Syntax:

```

while (test_expression)
{
    // statements

    update_expression;
}

```

Parts of Java While Loop

The various parts of the While loop are:

1. Test Expression: In this expression, we have to test the condition. If the condition evaluates to true then we will execute the body of the loop and go to update expression. Otherwise, we will exit from the while loop.

Example:

```
i <= 10
```

2. Update Expression: After executing the loop body, this expression increments/decrements the loop variable by some value.

Example:

```
i++;
```

How Does a While loop execute?

1. Control falls into the while loop.
2. The flow jumps to Condition
3. Condition is tested.
 - If Condition yields true, the flow goes into the Body.
 - If Condition yields false, the flow goes outside the loop
4. The statements inside the body of the loop get executed.
5. Updation takes place.
6. Control flows back to Step 2.
7. The while loop has ended and the flow has gone outside.

// Java program to illustrate while loop.

```
class whileLoopDemo {  
    public static void main(String args[])  
    {  
        // initialization expression  
        int i = 1;  
        // test expression  
        while (i < 6) {  
            System.out.println("Hello World");  
            // update expression  
            i++;  
        }  
    }  
}
```

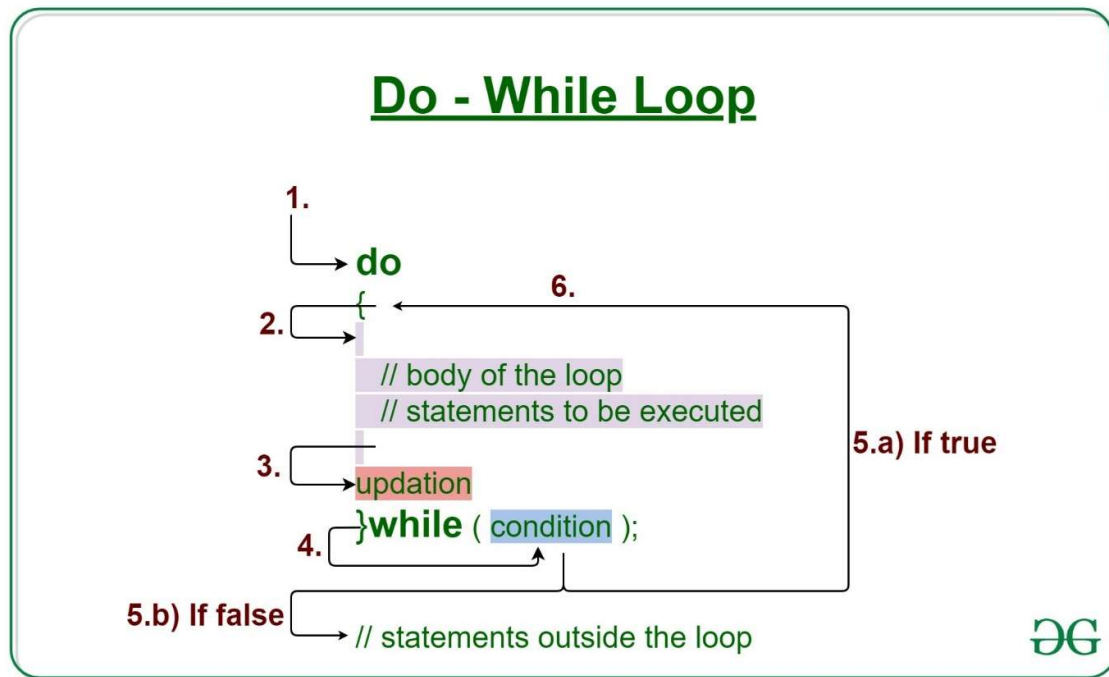
do-while loop with Examples

Loops in Java come into use when we need to repeatedly execute a block of statements. Java **do-while loop** is an **Exit control loop**. Therefore, unlike **for** or **while** loop, a do-while check for the condition after executing the statements of the loop body.

Syntax:

```
do  
{  
    // Loop Body  
    Update_expression  
}
```

```
// Condition check
while (test_expression);
```



Components of do-while Loop

A. Test Expression: In this expression, we have to test the condition. If the condition evaluates to true then we will execute the body of the loop and go to update expression. Otherwise, we will exit from the while loop. For example:

```
i <= 10
```

B. Update Expression: After executing the loop body, this expression increments/decrements the loop variable by some value. For example:

```
i++;
```

Execution of do-While loop

1. Control falls into the do-while loop.
2. The statements inside the body of the loop get executed.
3. Updation takes place.
4. The flow jumps to Condition
5. Condition is tested.
 1. If Condition yields true, go to Step 6.
 2. If Condition yields false, the flow goes outside the loop

6. The flow goes back to Step 2.

For-each loop in Java

For-each is another array traversing technique like for loop, while loop, do-while loop introduced in Java5.

- It starts with the keyword **for** like a normal for-loop.
- Instead of declaring and initializing a loop counter variable, you declare a variable that is the same type as the base type of the array, followed by a colon, which is then followed by the array name.
- In the loop body, you can use the loop variable you created rather than using an indexed array element.

Hope you all understood about control flow