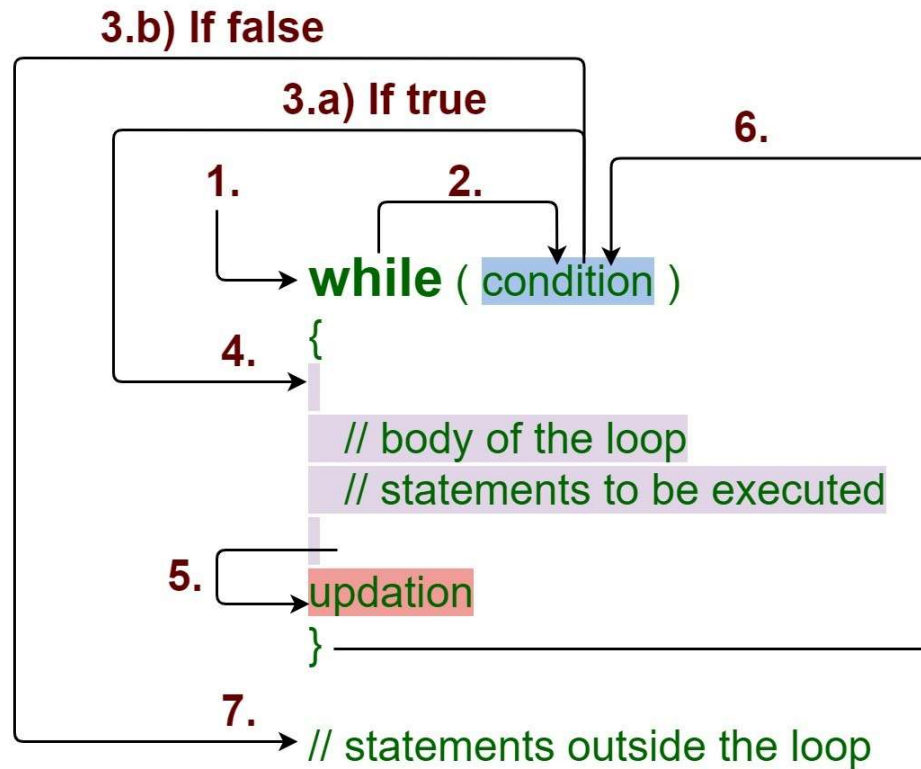


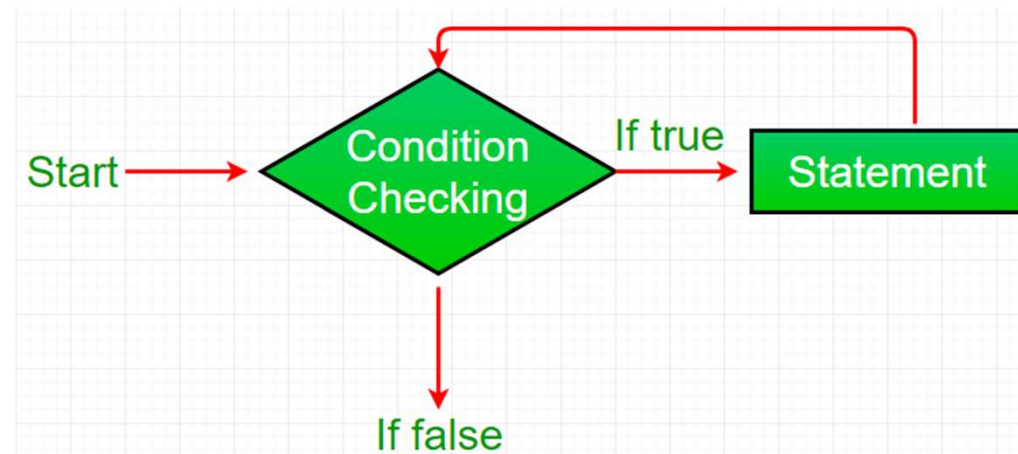
# Some Points in Java Loop

# While Loop



# How does a While loop executes?

1. Control falls into the while loop.
2. The execution flow jumps to Condition
3. Condition is tested.
  1. If Condition yields true, the flow goes into the Body.
  2. 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 do-while loop has ended and the flow has gone outside.



# Example

Write a Java program to find the summation of numbers from 1 to 10.

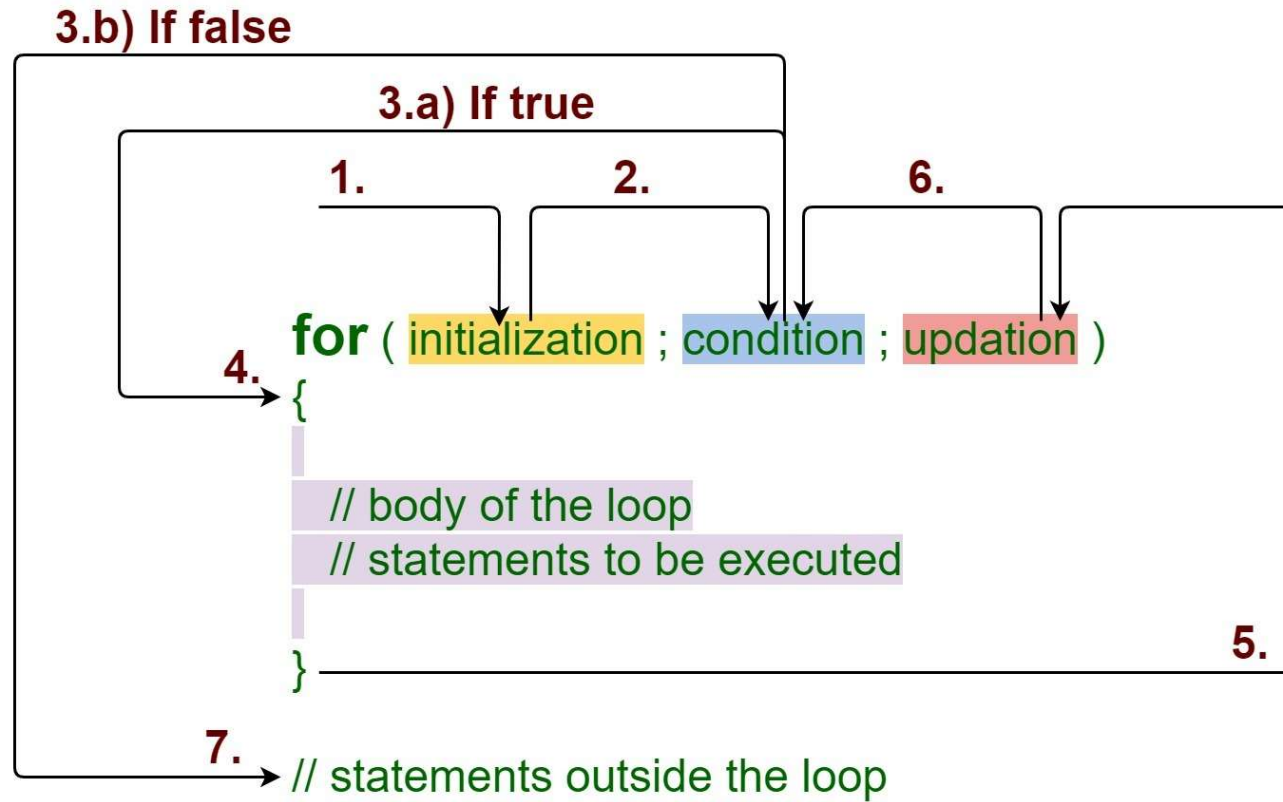
```
class whileLoopDemo {
    public static void main(String args[])
    {
        int x = 1, sum = 0;

        // Exit when x becomes greater than 10
        while (x <= 10) {
            // summing up x
            sum = sum + x;

            // Increment the value of x for
            // next iteration
            x++;
        }
        System.out.println("Sum: " + sum);
    }
}
```

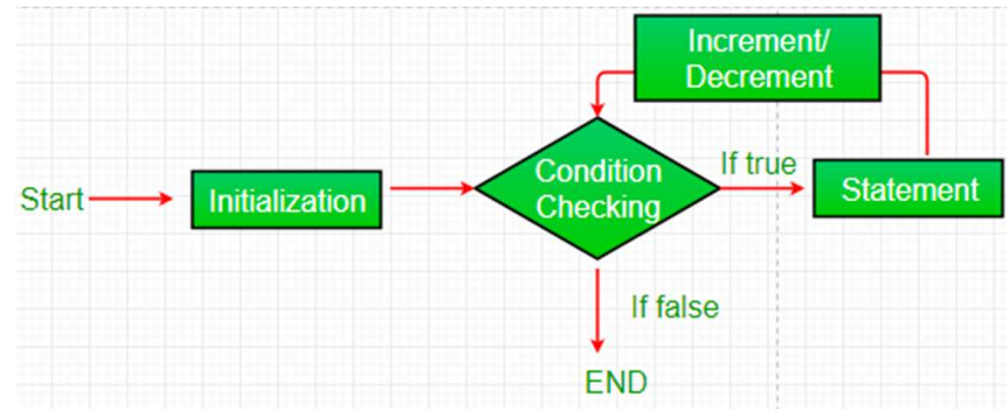
**Output:** Sum: 55

# For Loop



# How does a For loop executes?

1. Control falls into the for loop. Initialization is done
2. The execution flow jumps to Condition
3. Condition is tested.
  - a) If Condition yields true, the flow goes into the Body
  - b) If 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.



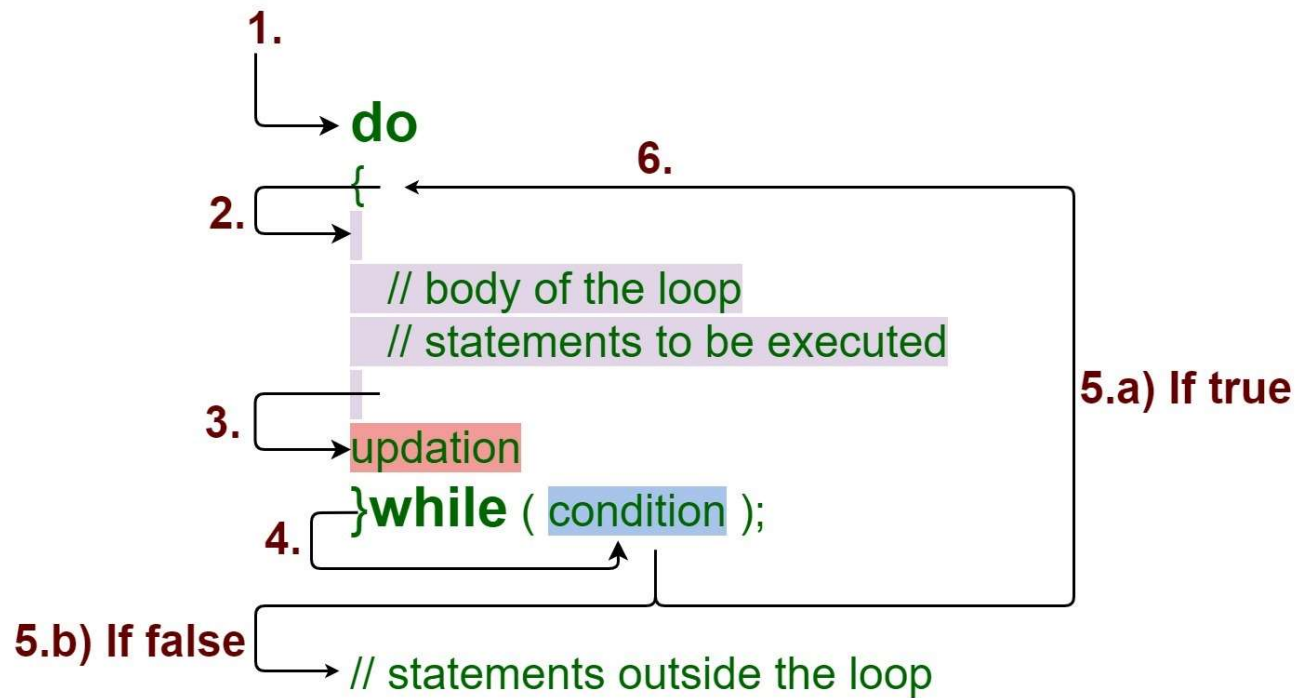
# Example

Write a Java program to find the summation of numbers from 1 to 20.

```
class forLoopDemo {  
    public static void main(String args[])  
    {  
        int sum = 0;  
  
        // for loop begins  
        // and runs till x <= 20  
        for (int x = 1; x <= 20; x++) {  
            sum = sum + x;  
        }  
        System.out.println("Sum: " + sum);  
    }  
}
```

**Output:** Sum: 210

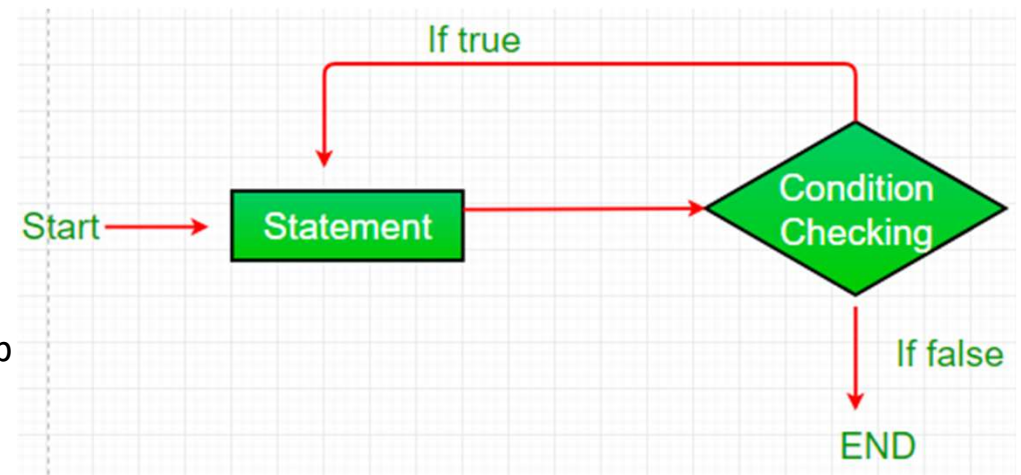
# Do - While Loop





# How does a do-While loop executes?

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 execution flow jumps to Condition
5. Condition is tested.
  - a) If Condition yields true, goto Step 6.
  - b) If Condition yields false, the flow goes outside the loop
6. Flow goes back to Step 2.



# Example

Write a Java program to find the summation of numbers from 11 to 21.

```
class dowhileloopDemo {
    public static void main(String args[])
    {
        int x = 21, sum = 0;
        do {

            // The line will be printed even
            // if the condition is false
            sum += x;
            x--;

        } while (x > 10);

        System.out.println("Sum: " + sum);
    }
}
```

**Output:** Sum: 176

```
int i = 0;
while (i < 3) {
    System.out.println(i);
    i = i + 1;
}
for (int i=0; i<3; i = i+1)
    System.out.println(i);
}
```

- Initialization
- Loop predicate
- Loop body
  - Some statement that affects the loop's predicate

# break vs continue

- You have already seen the ***break*** statement used to "jump out" of a switch statement.
- The ***break*** statement can also be used to jump out of a loop.
- The ***continue*** statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.
- **Example**
  - In class

- For loop forward and backward with more than one step.
- Example
  - In class

# Practice 1

Write a program to input 5 numbers from keyboard and find their sum and average

Input the 5 numbers : 1 2 3 4 5

Expected Output :

Input the 5 numbers :

1

2

3

4

5

The sum of 5 no is : 15

The Average is : 3.0

## Practice 2

- Get a number from user and check if it is a prime number or not

# Practice 3

- Guess a random number between 1-100



# Practice 4

- Print out

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5