

# Lesson:

## Loop (While and Do While Loop)



# Topics Covered

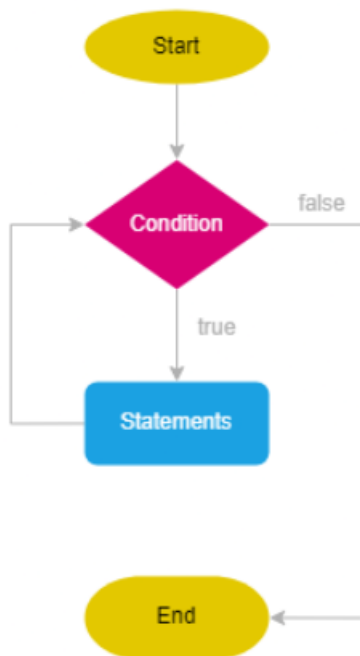
- While Loop
- DoWhile Loop
- For vs While vs Do While

## While loop

A while loop allows a block of code to be executed repeatedly as long as a specified condition is true.

```
JavaScript
while (condition) {
    // body of loop
}
```

### Understanding Flow



1. Loop starts
2. Condition is evaluated
  - a. If the condition is false, the while loop will terminate.
  - b. If the condition is true, the while loop's statements run, and continue until the condition evaluates to false.
3. Loop Ends

**Example 1:** To print numbers from 1 to 10 using while loop

JavaScript

```
let i = 1, n = 10;
while (i <= n)
{
    console.log(i);
    i=i+1;
}
```

/\*

Output

```
1
2
3
4
5
6
7
8
9
10
*/
```

**Example 2:** To print numbers from 10 to 1 in decreasing order.

JavaScript

```
let i = 1, n = 10;
while (n>=i)
{
    console.log(n);
    n=n-1;
}
```

/\* Output

```
10
9
8
7
6
5
4
3
2
1
*/
```

**Example 3:** Draw following pattern using while loop,

```
Unset
*
**
***
****
*****
```

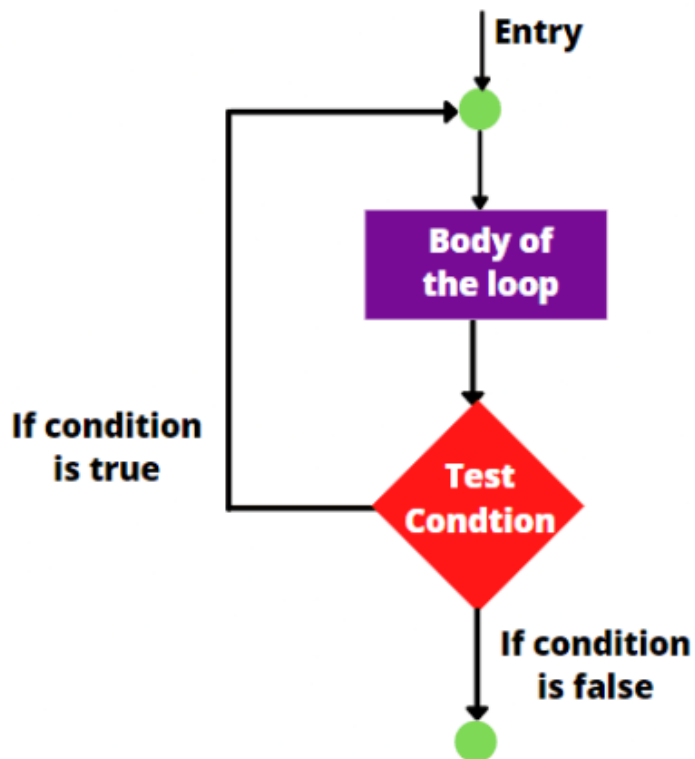
```
JavaScript
let row = 1;
while (row <= 5) {
  let pattern = '';
  let col = 1;
  while (col <= row) {
    pattern += '*';
    col++;
  }
  console.log(pattern);
  row++;
}
```

## Do while Loop

Do while is similar to while loop with one difference that the 1st iteration runs always, and later iterations run after the condition evaluates to true.

```
JavaScript
do {
  // body of loop
} while(condition);
```

## Understanding Flow



1. Loop Starts
2. Loop Body is executed first.
3. Condition is accessed
  - a. If Condition is false
  - b. If the condition is true, the while loop's statements run, and continue until the condition evaluates to false.
4. Loop Ends

**Example 1:** Write a program to print numbers from 1 to 7 in a line using do-while loop,

```

JavaScript
let result = '';

let i = 0;
do {
  i = i + 1;
  result = result + i;
} while (i < 7);
console.log(result);

/* Output
1234567
*/
  
```

For and while loop are used interchangeably, although they have different syntax they have equal capability, developer use different loops in different scenarios.

## For vs While vs Do While

Use a **for** loop when:

- Used when you know the number of iterations or the range of values you want to iterate over.
- The loop variable is automatically updated by the loop structure.

Use a **while** loop when:

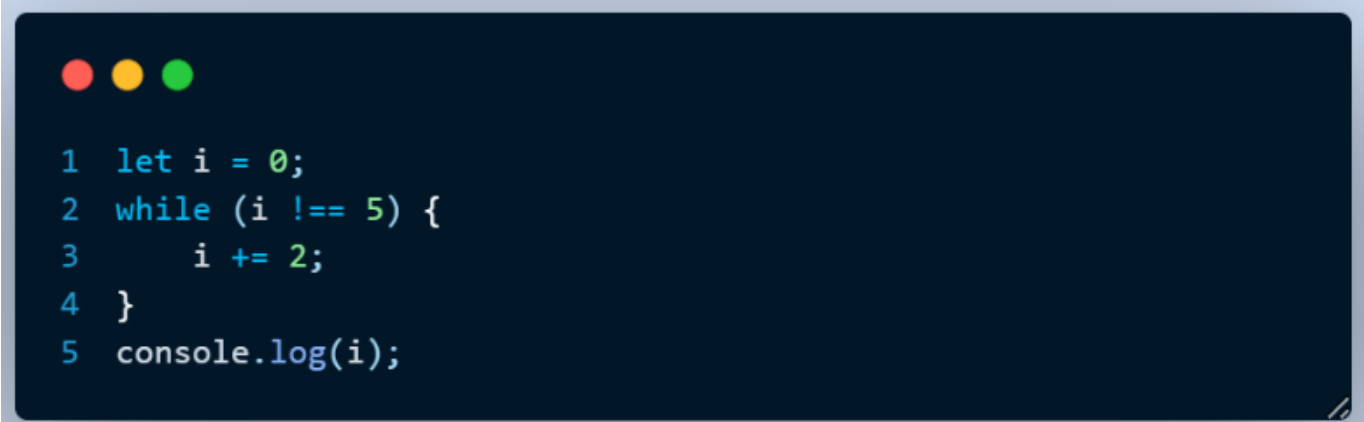
- Ideal for scenarios where the number of iterations is not known in advance or when the termination condition is based on a dynamic condition.
- You want to execute the code block only if the initial condition is true.
- The loop may not need to run at all if the condition is initially false.
- The loop termination depends on the condition being evaluated at the beginning of each iteration.

Use a **do-while** loop when:

- You want to execute the code block at least once, regardless of the initial condition.
- The loop termination depends on the condition being evaluated at the end of each iteration.

### Interview Point.

1. What is the output of the below code and comment on the output.



```
1 let i = 0;
2 while (i !== 5) {
3     i += 2;
4 }
5 console.log(i);
```

**Ans:** This code lands in an infinite loop and nothing gets printed.

In this code, the loop condition `i !== 5` will never become false because `i` is incremented by 2 in each iteration. As a result, `i` will never be equal to 5, and the loop will continue indefinitely.

2. What is the output of the below code and comment on the output.

```
1 let i = 0;
2 while (i < 3) {
3     i++;
4 }
5 console.log(i);
```

**Ans:** The output of this code is 5.

The loop increments *i* by 1 in each iteration until *i* becomes 3. The `console.log(i)` statement then prints the final value of *i*. The loop condition `i < 3` ensures that the loop continues until *i* is less than 3, and it terminates when *i* becomes 3.

