

Training Day 22 Report

22 July 2025

Loops and Conditions in JavaScript

1. Introduction

In JavaScript, **loops** and **conditions** are fundamental concepts that help control the flow of a program.

- **Conditional statements** are used to **make decisions** based on certain conditions.
- **Loops** are used to **execute a block of code repeatedly** as long as a condition remains true.

Together, they make programs **dynamic, logical, and efficient**.

Conditional Statements in JavaScript

2. What are Conditional Statements?

Conditional statements allow your program to make decisions and execute different code blocks depending on whether a condition is true or false.

The main conditional statements in JavaScript are:

1. if statement
2. if...else statement
3. if...else if...else statement
4. switch statement
5. ternary operator (?:)

3. The if Statement

The if statement executes a block of code **only if** a given condition is true.

Syntax:

```
if (condition) {  
    // code to execute if condition is true  
}
```

Example:

```
let age = 18;

if (age >= 18) {
    console.log("You are eligible to vote.");
}
```

Output:

You are eligible to vote.

4. The if...else Statement

This statement executes one block of code if the condition is true, and another block if it is false.

Syntax:

```
if (condition) {
    // executes if true
} else {
    // executes if false
}
```

Example:

```
let marks = 40;

if (marks >= 50) {
    console.log("You passed!");
} else {
    console.log("You failed!");
}
```

Output:

You failed!

5. The if...else if...else Statement

Used when multiple conditions need to be checked in sequence.

Syntax:

```
if (condition1) {
```

```
    // code if condition1 is true
} else if (condition2) {
    // code if condition2 is true
} else {
    // code if none are true
}
```

Example:

```
let score = 85;
```

```
if (score >= 90) {
    console.log("Grade A");
} else if (score >= 75) {
    console.log("Grade B");
} else if (score >= 50) {
    console.log("Grade C");
} else {
    console.log("Fail");
}
```

Output:

Grade B

6. The switch Statement

The switch statement is used when you want to compare a variable with multiple values.

Syntax:

```
switch(expression) {
    case value1:
        // code
        break;
    case value2:
```

```
    // code

    break;

default:

    // code
}
```

Example:

```
let day = 3;
```

```
switch (day) {

  case 1:

    console.log("Monday");

    break;

  case 2:

    console.log("Tuesday");

    break;

  case 3:

    console.log("Wednesday");

    break;

  default:

    console.log("Other day");

}
```

Output:

Wednesday

7. Ternary Operator

The **ternary operator** provides a shorthand way of writing an if...else statement.

Syntax:

```
condition ? expressionIfTrue : expressionIfFalse;
```

Example:

```
let age = 20;

let result = (age >= 18) ? "Adult" : "Minor";

console.log(result);
```

Output:

Adult

Loops in JavaScript

8. What are Loops?

A **loop** allows you to run the same block of code multiple times.

This is useful when you need to **repeat tasks**, like printing numbers, processing arrays, or iterating over objects.

Main loop types in JavaScript:

1. for loop
2. while loop
3. do...while loop
4. for...in loop
5. for...of loop

9. The for Loop

The for loop is used when the number of iterations is **known**.

Syntax:

```
for (initialization; condition; increment/decrement) {

    // code to execute

}
```

Example:

```
for (let i = 1; i <= 5; i++) {

    console.log("Number: " + i);

}
```

Output:

Number: 1

Number: 2

Number: 3

Number: 4

Number: 5

10. The while Loop

Used when the number of iterations is **unknown** — it runs as long as the condition is true.

Syntax:

```
while (condition) {  
    // code to execute  
}
```

Example:

```
let i = 1;  
while (i <= 3) {  
    console.log("Hello " + i);  
    i++;  
}
```

Output:

Hello 1

Hello 2

Hello 3

11. The do...while Loop

This loop executes the code **at least once**, even if the condition is false.

Syntax:

```
do {  
    // code to execute  
} while (condition);
```

Example:

```
let i = 5;
```

```
do {
```

```
    console.log("Number is " + i);  
  
    i++;  
} while (i < 5);
```

Output:

Number is 5

(The loop runs once before checking the condition.)

12. The for...in Loop

Used to iterate over **object properties**.

Example:

```
let person = { name: "Shubhdeep", age: 20, city: "Ludhiana" };
```

```
for (let key in person) {  
    console.log(key + ": " + person[key]);  
}
```

Output:

name: Shubhdeep

age: 20

city: Ludhiana

13. The for...of Loop

Used to iterate over **arrays or strings**.

Example:

```
let fruits = ["Apple", "Banana", "Mango"];
```

```
for (let fruit of fruits) {  
    console.log(fruit);  
}
```

Output:

Apple

Banana

Mango

14. Break and Continue Statements

Break Statement

Used to **exit** a loop before it finishes all iterations.

Example:

```
for (let i = 1; i <= 10; i++) {  
    if (i === 5) break;  
    console.log(i);  
}
```

Output:

1

2

3

4

Continue Statement

Used to **skip** the current iteration and continue with the next one.