

Loops (for, while, do-while)

Sure! Let's dive deep into **loops** in JavaScript — specifically the `for`, `while`, and `do...while` loops — and understand how they work with syntax, examples, and use cases.

What are Loops?

Loops allow you to execute a block of code **repeatedly** as long as a specified condition is true.

JavaScript has several types of loops, but here we'll focus on:

- `for` loop
- `while` loop
- `do...while` loop

1. `for` Loop

◆ Syntax:

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

- **Initialization:** runs once before the loop starts
- **Condition:** evaluated before each iteration
- **Increment/Decrement:** executed after each iteration

◆ Example:

```
for (let i = 1; i <= 5; i++) {  
    console.log("Count: " + i);
```

```
}
```

📌 Output:

```
Count: 1  
Count: 2  
Count: 3  
Count: 4  
Count: 5
```

2. while Loop

- Executes the block **while the condition is true.**
- The condition is evaluated **before** each iteration.

◆ Syntax:

```
while (condition) {  
    // code block  
}
```

◆ Example:

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

📌 Output:

```
Number is: 1  
Number is: 2
```

Number is: 3

3. do...while Loop

- Executes the code **at least once, then checks** the condition.
- Useful when the code must run **at least one time** no matter what.

◆ Syntax:

```
do {  
    // code block  
} while (condition);
```

◆ Example:

```
let i = 1;  
  
do {  
    console.log("Value: " + i);  
    i++;  
} while (i <= 2);
```

Output:

```
Value: 1  
Value: 2
```

Loop Comparison Table

Loop Type	Condition Checked	Minimum Runs	Best Use Case
	Before loop	0	Known number of iterations
	Before loop	0	Unknown number of iterations

do...while	After loop	1	Must run at least once
------------	------------	---	------------------------

⚠ Common Mistakes

- Forgetting to update loop variables → **infinite loop**
- Using `=` instead of `==` / `===` in condition
- Off-by-one errors in start/end values

🧪 Real-World Example: Sum of Numbers 1 to 5 using all 3 loops

Using `for` :

```
let sum = 0;
for (let i = 1; i <= 5; i++) {
  sum += i;
}
console.log("Sum:", sum); // 15
```

Using `while` :

```
let i = 1;
let sum = 0;
while (i <= 5) {
  sum += i;
  i++;
}
console.log("Sum:", sum);
```

Using `do...while` :

```
let i = 1;
let sum = 0;
```

```
do {  
    sum += i;  
    i++;  
} while (i <= 5);  
console.log("Sum:", sum);
```

Would you like to explore loop **nested loops**, **infinite loop examples**, or the `break` and `continue` statements next?