JavaScript loops are fundamental control flow structures that allow you to repeatedly execute a block of code. They are essential for automating repetitive tasks and processing collections of data. Here's a breakdown of the different types of loops in JavaScript:

1. for Loop:

The for loop is commonly used when you know in advance how many times you want to iterate.

```
for (initialization; condition; increment/decrement) {
 // Code to be executed repeatedly
}
```

- initialization: Executed once before the loop starts. Typically used to declare and initialize a loop counter variable.
- condition: Evaluated before each iteration. If the condition is true, the loop body is executed. If false, the loop terminates.
- increment/decrement: Executed after each iteration. Typically used to update the loop counter.

```
for (let i = 0; i < 10; i++) {
  console.log(i); // Output: 0, 1, 2, ..., 9
  }

  // Example with an array:
  const colors = ["red", "green", "blue"];
  for (let i = 0; i < colors.length; i++) {
  console.log(colors[i]); // Output: "red", "green", "blue"
  }
```

2. while Loop:

The while loop is used when the number of iterations is not known in advance and depends on a condition.

```
  while (condition) {
  // Code to be executed repeatedly as long as the condition is true
   }
```

- The condition is evaluated before each iteration. If true, the loop body is executed. If false, the loop terminates. It's very important to make sure the condition eventually becomes false to avoid an infinite loop!

```
let count = 0;
while (count < 5) {
console.log(count); // Output: 0, 1, 2, 3, 4
count++; // Increment the counter (important to avoid an infinite loop)
}
```

3. do...while Loop:

The do...while loop is similar to the while loop, but the loop body is executed *at least once* before the condition is checked.

```
do {
    // Code to be executed repeatedly
    } while (condition);
    JavaScript
    let i = 10;
    do {
    console.log(i); // Output: 10 (even though the condition is initially false)
    i++;
    } while (i < 5); // Condition is checked after the first iteration
```

4. for...in Loop (for Objects):

The for...in loop is used to iterate over the *enumerable properties* of an object.

```
    const person = {
 name: "Alice",
 age: 30,
 city: "New York"
 };

 for (let key in person) {
 console.log(key + ": " + person[key]); // Output: "name: Alice", "age: 30", "city: New York"
 }
```

5. for...of Loop (for Iterables):

The for...of loop is used to iterate over *iterable objects*, such as arrays, strings, maps, sets, etc.

JavaScript

```
const colors = ["red", "green", "blue"];

for (let color of colors) {
console.log(color); // Output: "red", "green", "blue"
}

const message = "Hello";
for (let char of message) {
console.log(char); // Output: "H", "e", "l", "l", "o"
}
```

Loop Control Statements:

- **break:** Terminates the loop completely.

```
for (let i = 0; i < 10; i++) {
 if (i === 5) {
    break; // Exit the loop when i is 5
 }
 console.log(i); // Output: 0, 1, 2, 3, 4
 }
```

- **continue:** Skips the rest of the current iteration and proceeds to the next iteration.

```
for (let i = 0; i < 10; i++) {
if (i % 2 === 0) { // Skip even numbers
   continue;
}
console.log(i); // Output: 1, 3, 5, 7, 9
}
```

Choosing the Right Loop:

- for: When you know the number of iterations in advance.
- while: When the number of iterations depends on a condition.
- do...while: When you need to execute the loop body at least once.
- for...in: When you need to iterate over object properties.
- for...of: When you need to iterate over iterable objects (arrays, strings, etc.).

Understanding and using loops effectively is essential for writing efficient and powerful JavaScript code. Practice using the different types of loops and loop control statements to become comfortable with these important concepts.

1. Write a Javascript Program to print all the factors of a given in the descending order.
2. Write a Javascript Program to print first ten multiples of a given number in the descending order.
3. Write a Javascript Program to print total number of factors (count) of a given number.
4. Write a Javascript Program to test a given number is a Fibonacci or not?
5. Write a Javascript Program to print all the even numbers up to a given number in the descending order.
6. Write a Javascript Program to print all the odd numbers in the descending order up to a given number.
7. Write a Javascript program to print all the even numbers which are divisible by 3 up to a given number.
8. Write a Javascript Program to print all the even numbers which are divisible by 9 up to a given number.
9. Write a Javascript Program to print all the odd numbers which are divisible by 3.