

# Types of Loops in JavaScript

Save It !!



@js.guru\_



# Introduction

Mastering loops is key to efficient coding in JavaScript! Let's explore the various types of loops and how they can streamline your code. Here are the types of loops we'll cover:

## List of Loops:

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

Swipe left to dive into each loop type!

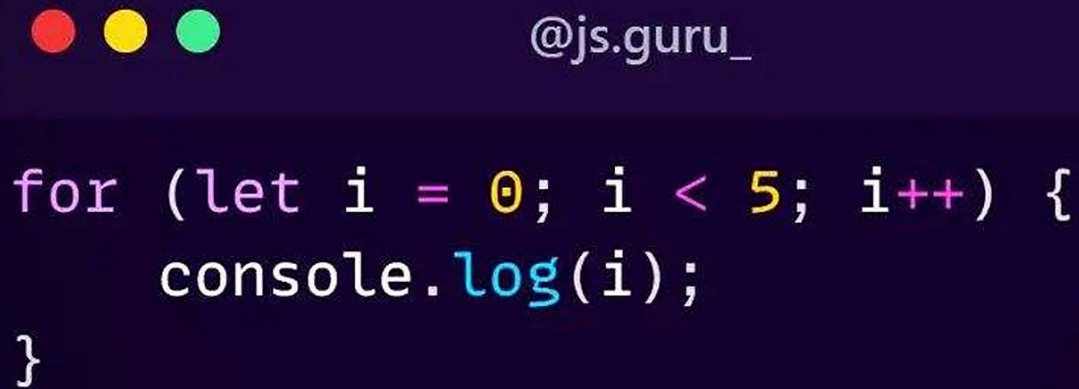


@js.guru\_



# 01. for Loop

The classic for loop is perfect for iterating over arrays or executing a block of code a specific number of times. With its flexibility and control, the for loop is a staple in JavaScript programming.



```
@js.guru_  
  
for (let i = 0; i < 5; i++) {  
    console.log(i);  
}
```



@js.guru\_





## 02. while Loop

Use the while loop when you need to repeat a block of code as long as a specified condition is true. Be cautious with your condition to avoid infinite loops!

```
@js.guru_  
  
let i = 0;  
while (i < 5) {  
    console.log(i);  
    i++;  
}
```




@js.guru\_



## 03. do...while Loop

Similar to the while loop, the do...while loop executes a block of code once, then repeats the loop as long as a specified condition is true. Unlike the while loop, the code block is executed at least once before the condition is checked.



```
let i = 0;
do {
  console.log(i);
  i++;
} while (i < 5);
```



@js.guru\_



## 04. for...in Loop

The for...in loop iterates over the enumerable properties of an object, allowing you to loop through the keys of an object or the indices of an array. Use caution when iterating over arrays to avoid unexpected behavior.

```
@js.guru_  
  
const person = {  
  name: 'JS Guru',  
  age: 30,  
  city: 'New York'  
};  
  
for (const key in person) {  
  console.log(key + ': ' + person[key]);  
}
```



@js.guru\_





## 05. for...of Loop

Introduced in ES6, the for...of loop provides a simple and concise way to iterate over iterable objects like arrays, strings, and collections. It's especially useful when you need to access the values directly without worrying about indices.

```
@js.guru_  
  
const fruits = ['apple', 'banana', 'orange'];  
  
for (const fruit of fruits) {  
    console.log(fruit);  
}
```

 fly.io



@js.guru\_

