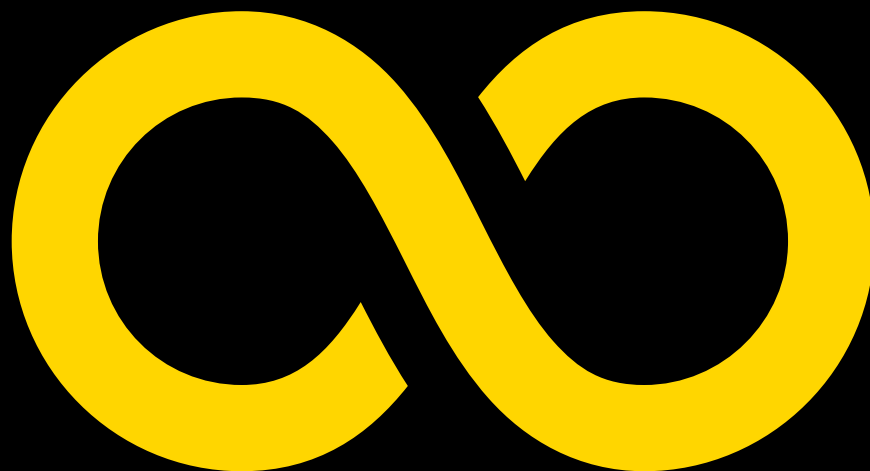


JavaScript

Loops



JavaScript Loops

01/08

- ▼ In JavaScript, **loops** are used to execute a block of code repeatedly until a specified condition is met.
- ▼ There are several types of loops available in JavaScript:
 - ▶ **for loops** through a block of code a number of times.
 - ▶ **while loops** through a block of code while a specified condition is true.
 - ▶ **do...while** loops also through a block of code while a specified condition is true.
 - ▶ **for...of** loops through the values of an iterable object.
 - ▶ **for...in** loops through the properties of an object.



JavaScript **for loop**

02/08

- ▼ The **for loop** is used when you know how many times you want to execute the block of code.
- ▼ It consists of three parts: **initialization**, **condition**, and **increment/decrement**.

```
for (let i = 1; i ≤ 5; i++) {  
    console.log(i);  
}
```

```
/*  
Output:  
1  
2  
3  
4  
5  
*/
```



JavaScript **while** loop

03/08

- ▼ The **while** loop is used when you want to execute a block of code as long as a condition is true.

```
let i = 1;

while (i ≤ 5) {
  console.log(i);
  i++;
}
```

```
/*
Output:
1
2
3
4
5
*/
```



JavaScript **do...while** loop

04/08

- ▼ The **do...while** loop is similar to the while loop, but the block of code is executed at least once before the condition is checked.

```
let i = 1;

do {
  console.log(i);
  i++;
} while (i ≤ 5);
```

```
/*
Output:
1
2
3
4
5
*/
```



JavaScript **for...of** loop

05/08

- ▼ The **for...of** loop is used to iterate over the elements of an iterable object (like arrays or strings).

```
const arr = [5, 10, 20];

for (const item of arr) {
  console.log(item);
}

/*
Output:
5
10
20
*/
```



JavaScript **for...in** loop

06/08

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

```
const person = {
  name: 'John Doe',
  age: 25,
  job: 'Software Engineer'
};

for (const key in person) {
  console.log(`${key}: ${person[key]}`);
}

/*
Output:
name: John Doe
age: 25
job: Software Engineer
*/
```



JavaScript loop control statements

07/08

▼ **break:** is used to **terminate the loop**, skipping the remaining iterations.

```
for (let i = 1; i ≤ 5; i++) {  
  if (i ≡ 3) {  
    break; // Terminate the loop if i is 3  
  }  
  console.log(i);  
}  
  
/*  
Output:  
1  
2  
*/
```

▼ **continue:** is used to **skip the current iteration** of the loop and continue with the next iteration.

```
for (let i = 1; i ≤ 5; i++) {  
  if (i % 2 ≡ 0) {  
    continue; // Skip the iteration if is odd  
  }  
  console.log(i);  
}  
  
/*  
Output:  
2  
4  
*/
```



JavaScript **nested loops**

08/08

- ▼ **Nested loops** in JavaScript refer to the situation where **one loop is present inside another loop**.
- ▼ This allows you to iterate over multiple dimensions of data, such as rows and columns of a matrix, or elements of a multidimensional array.
- ▼ Nested loops can be created using any type of loop construct in JavaScript, including for, while, and for...of loops.

```
for (let i = 1; i ≤ 2; i++) { // Outer loop for i
  for (let j = 1; j ≤ 3; j++) { // Inner loop for j
    console.log(`i: ${i}, j: ${j}`);
  }
}
```

/*

Output:

i: 1, j: 1

i: 1, j: 2

i: 1, j: 3

i: 2, j: 1

i: 2, j: 2

i: 2, j: 3

*/





Follow Us !



techglobal.school



techglobalschool



techglobalschool



techglobalschl



techglobalschool

www.techglobalschool.com