

## What is a Loop in JavaScript?

A **loop** in JavaScript is a programming structure that **repeats a block of code** as long as a certain condition is true.

□ It's used when you want to **do the same task multiple times** — like printing numbers, going through arrays, or processing data.

**JavaScript loop example so you understand exactly how they work, step by step.**

---

### □ 1. for Loop

#### □ Code:

```
for (let i = 1; i <= 3; i++) {  
  console.log("for loop:", i);  
}
```

#### □ How it works:

Step	Description
let i = 1	Initialization: Start the loop with variable <i>i</i> = 1.
i <= 3	Condition: As long as <i>i</i> is <b>less than or equal to 3</b> , run the loop.
console.log(...)	Loop body: This runs <b>each time</b> the condition is true.
i++	Increment: After each run, increase <i>i</i> by 1.

#### □ Loop runs like this:

- *i* = 1: condition true → print for loop: 1
  - *i* = 2: condition true → print for loop: 2
  - *i* = 3: condition true → print for loop: 3
  - *i* = 4: condition false → loop ends.
-

## □ 2. while Loop

### □ Code:

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

### □ How it works:

Step	Description
let i = 1	Set up a counter variable outside the loop.
while (i <= 3)	This checks the condition <b>before</b> each loop.
console.log(...)	If true, the code inside the loop runs.
i++	Don't forget to increment i or the loop will run forever!

### □ Loop runs like this:

- i = 1: condition true → print while loop: 1, then i++ → 2
  - i = 2: condition true → print while loop: 2, then i++ → 3
  - i = 3: condition true → print while loop: 3, then i++ → 4
  - i = 4: condition false → loop stops.
-

### □ 3. do...while Loop

#### □ Code:

```
let i = 1;
do {
  console.log("do...while loop:", i);
  i++;
} while (i <= 3);
```

#### □ How it works:

Step	Description
let i = 1	Initialize a variable.
do { ... }	The loop body runs <b>first</b> (at least once).
while (i <= 3)	Then check if the condition is true to run again.

#### □ Loop runs like this:

- First run:  $i = 1 \rightarrow$  print do...while loop: 1,  $i++ \rightarrow 2$
- Check condition:  $i = 2 \leq 3 \rightarrow$  true  $\rightarrow$  repeat
- Second run: print do...while loop: 2,  $i++ \rightarrow 3$
- Third run: print do...while loop: 3,  $i++ \rightarrow 4$
- Now  $i = 4 \rightarrow$  condition false  $\rightarrow$  loop ends

□ Even if the condition was false to begin with, this loop **runs at least once**.

---

## □ 4. `for...of` Loop

### □ Code:

```
let fruits = ["apple", "banana", "cherry"];

for (let fruit of fruits) {
  console.log("for...of loop:", fruit);
}
```

### □ How it works:

Part	Description
<code>fruits</code>	An array of strings.
<code>for (let fruit of fruits)</code>	Loop through each <b>item</b> in the array.
<code>fruit</code>	On each loop, this variable holds the current item.

### □ Loop runs like this:

- First iteration: `fruit = "apple"` → print
- Second: `fruit = "banana"` → print
- Third: `fruit = "cherry"` → print

□ This loop is **perfect for arrays or strings**.

---

## □ 5. for...in Loop

### □ Code:

```
let person = { name: "Alice", age: 30, city: "Paris" };

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

### □ How it works:

Part	Description
person	A JavaScript object with 3 properties.
for (let key in person)	Loops through the <b>property names</b> (keys) of the object.
person[key]	Accesses the value of each property.

### □ Loop runs like this:

- First: key = "name" → value: "Alice"
- Second: key = "age" → value: 30
- Third: key = "city" → value: "Paris"

### □ Output:

```
name: Alice
age: 30
city: Paris
```

- Use for...in to loop over **object keys**, not arrays.
- 

## □ Summary of Differences:

<b>Loop Type</b>	<b>Best For</b>	<b>Runs At Least Once?</b>
<code>for</code>	Fixed number of times	<input type="checkbox"/>
<code>while</code>	Condition-based, unknown count	<input type="checkbox"/>
<code>do...while</code>	Same as <code>while</code> , but runs at least once	<input type="checkbox"/>
<code>for...of</code>	Arrays, strings, iterable data	<input type="checkbox"/>
<code>for...in</code>	Objects — keys and values	<input type="checkbox"/>