JS JavaScript
TUTORIAL

JavaScript for Loop

Summary: in this tutorial, you will learn how to use the JavaScript `for` loop statement to create a loop with various options.

Introduction to the JavaScript for loop statement

The `for` loop statement creates a loop with three optional expressions. The following illustrates the syntax of the `for` loop statement:

```
for (initializer; condition; iterator) {
    // statements
}
```

1) initializer

The `for` statement executes the `initializer` only once the loop starts. Typically, you declare and initialize a local loop variable in the initializer.

2) condition

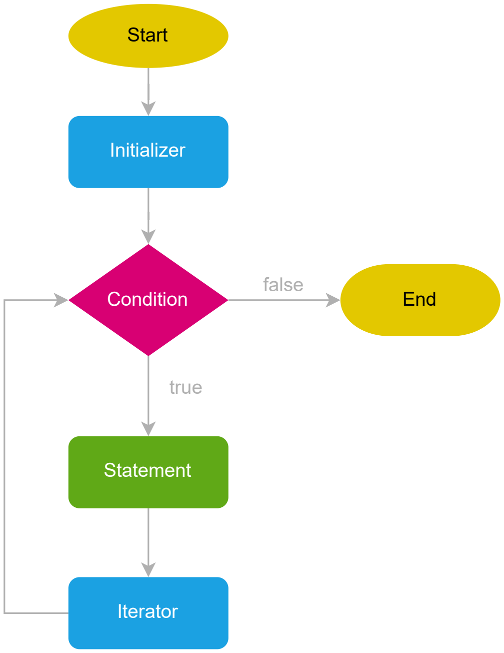
The `condition` is a boolean expression that determines whether the `for` should execute the next iteration.

The `for` statement evaluates the `condition` before each iteration. If the condition is `true` (or is not present), it executes the next iteration. Otherwise, it'll end the loop.

3) iterator

The `for` statement executes the `iterator` after each iteration.

The following flowchart illustrates the `for` loop:



In the `for` loop, the three expressions are optional. The following shows the `for` loop without any expressions:

```
for ( ; ; ) {
   // statements
}
```

JavaScript for loop examples

Let's take some examples of using the `for` loop statement.

1) A simple JavaScript for loop example

The following example uses the `for` loop statement to show numbers from 1 to 4 to the console:

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

Output:

```
1
2
3
4
```

How it works.

- First, declare a variable `counter` and initialize it to 1.

- Second, display the value of `counter` in the console if `counter` is less than 5.
- Third, increase the value of `counter` by one in each iteration of the loop.

2) Using the JavaScript for loop without the initializer example

The following example uses a `for` loop that has no initializer expression:

```
let j = 1;
for (; j < 10; j += 2) {
  console.log(j);
}
```

Output:

```
1
3
5
7
9
```

3) Using the JavaScript for loop without the condition example

Similar to the `initializer` expression, the `condition` expression is optional. If you omit the `condition` expression, you need to use a `break` statement to terminate the loop.

```
for (let j = 1; ; j += 2) {
  console.log(j);
  if (j > 10) {
    break;
  }
}
```

Output:

```
1
3
5
7
9
11
```

3) Using the JavaScript for loop statement without any expression example

All three expressions of the `for` loop statements are optional. Therefore, you can omit all of them. For example:

```
let j = 1;
for (;;) {
  if (j > 10) {
    break;
  }
  console.log(j);
  j += 2;
}
```

Output:

```
1
3
5
7
9
```

4) Using the JavaScript for loop without the loop body example

JavaScript allows the `for` statement to have an empty statement. In this case, you place a semicolon (`;`) immediately after the `for` statement.

For example, the following uses a for loop to calculate the sum of 10 numbers from 1 to 10:

```
let sum = 0;
for (let i = 0; i <= 9; i++, sum += i);
console.log(sum);
```

Output:

```
55
```

Summary

- Use the JavaScript `for` statement to create a loop that executes a block of code using various options.

Quiz