JavaScript For Loop

‹ Previous Next ›

Loops can execute a block of code a number of times.

JavaScript Loops

Loops are handy, if you want to run the same code over and over again, each time with a different value.

Often this is the case when working with arrays:

Instead of writing:

```
text += cars[2] + "<br>";
text += cars[3] + "<br>";
text += cars[4] + "<br>";
text += cars[5] + "<br>";
```

You can write:

```
var i;
for (i = 0; i < cars.length; i++) {
  text += cars[i] + "<br>";
}
```

Try it Yourself »

Different Kinds of Loops

JavaScript supports different kinds of loops:

- `for` - loops through a block of code a number of times
- `for/in` - loops through the properties of an object
- `for/of` - loops through the values of an iterable object
- `while` - loops through a block of code while a specified condition is true
- `do/while` - also loops through a block of code while a specified condition is true

The `for` loop has the following syntax:

```
for (statement 1; statement 2; statement 3) {
  // code block to be executed
}
```

Statement 1 is executed (one time) before the execution of the code block.

Statement 2 defines the condition for executing the code block.

Statement 3 is executed (every time) after the code block has been executed.

Example

```
for (i = 0; i < 5; i++) {
  text += "The number is " + i + "<br>";
}
```

Try it Yourself »

From the example above, you can read:

Statement 1 sets a variable before the loop starts (var i = 0).

Statement 2 defines the condition for the loop to run (i must be less than 5).

Statement 3 increases a value (i++) each time the code block in the loop has been executed.

Statement 1

Normally you will use statement 1 to initialize the variable used in the loop (i = 0).

This is not always the case, JavaScript doesn't care. Statement 1 is optional.

You can initiate many values in statement 1 (separated by comma):

Example

```
for (i = 0, len = cars.length, text = ""; i < len; i++) {
  text += cars[i] + "<br>";
}
```

Try it Yourself »

And you can omit statement 1 (like when your values are set before the loop starts):

Example

```
var text = "";
for (; i < len; i++) {
  text += cars[i] + "<br>";
}
```

Try it Yourself »

Statement 2

Often statement 2 is used to evaluate the condition of the initial variable.

This is not always the case, JavaScript doesn't care. Statement 2 is also optional.

If statement 2 returns true, the loop will start over again, if it returns false, the loop will end.

If you omit statement 2, you must provide a **break** inside the loop. Otherwise the loop will never end. This will crash your browser. Read about breaks in a later chapter of this tutorial.

Statement 3

Often statement 3 increments the value of the initial variable.

This is not always the case, JavaScript doesn't care, and statement 3 is optional.

Statement 3 can also be omitted (like when you increment your values inside the loop):

Example

```
var i = 0;
var len = cars.length;
for (; i < len; ) {
  text += cars[i] + "<br>";
  i++;
}
```

Try it Yourself »

The For/In Loop

The JavaScript `for/in` statement loops through the properties of an object:

Example

```
var person = {fname:"John", lname:"Doe", age:25};

var text = "";
var x;
for (x in person) {
```

Try it Yourself »

The For/Of Loop

The JavaScript `for/of` statement loops through the values of an iterable objects

`for/of` lets you loop over data structures that are iterable such as Arrays, Strings, Maps, NodeLists, and more.

The `for/of` loop has the following syntax:

```
for (variable of iterable) {
  // code block to be executed
}
```

variable - For every iteration the value of the next property is assigned to the variable. *Variable* can be declared with `const`, `let`, or `var`.

iterable - An object that has iterable properties.

Looping over an Array

Example

```
var x;

for (x of cars) {
  document.write(x + "<br >");
}
```

Try it Yourself »

Looping over a String

Example

```
var txt = 'JavaScript';
var x;

for (x of txt) {
  document.write(x + "<br >");
}
```

Try it Yourself »

The While Loop

Test Yourself With Exercises

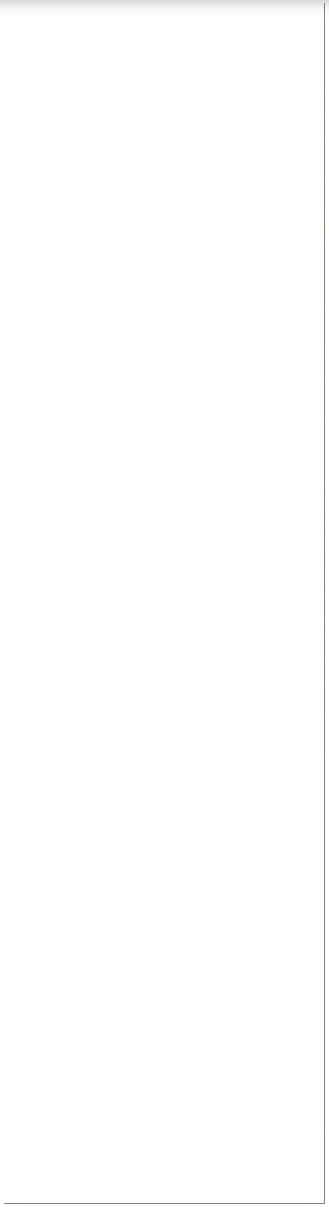
Exercise:

Create a loop that runs from 0 to 9.

```
var i;
    (  =  ;  <  ;  ) {
  console.log(i);
}
```

Submit Answer »

Start the Exercise



COLOR PICKER

HTML
CSS
JavaScript
SQL
Python
PHP
jQuery
Bootstrap
XML

Read More »