

Drawing Mode



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[0] + "<br>";  
text += cars[1] + "<br>";  
text += cars[2] + "<br>";  
text += cars[3] + "<br>";  
text += cars[4] + "<br>";  
text += cars[5] + "<br>";
```

You can write:

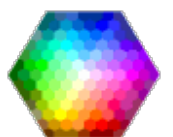
```
var i;  
for (i = 0; i < cars.length; i++) {
```

Need
coupo

Get It H

hona

COLOR
PICKER



```
text += cars[i] + "<br>";  
}
```

[Try it Yourself »](#)

HOW TO

Tabs
Dropdowns
Accordions
Side Navigation
Top Navigation
Modal
Boxes
Progress Bars
Parallax
Login Form
HTML Includes
Google Maps
Range Sliders
Tooltips
Slideshow
Filter List
Sort List

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
- **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

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.

SHARE



Example

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

[Try it Yourself »](#)

CERTIFIC

HTML
CSS
JavaScript
PHP
jQuery
Bootstrap
XML

[Read More](#)

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.

Need a coupon?

[HTML](#)[CSS](#)[JAVASCRIPT](#)[SQL](#)[PHP](#)[BOOTSTRAP](#)[HOW TO](#)

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>";  
}
```

JS Date Get Methods

JS Date Set Methods

JS Math

JS Random

JS Booleans

JS Comparisons

JS Conditions

JS Switch

JS Loop For

JS Loop While

JS Break

JS Type Conversion

JS Bitwise

JS RegExp

JS Errors

JS Scope

JS Hoisting

JS Strict Mode

JS this Keyword

JS Let

JS Const

JS Debugging

JS Style Guide

JS Best Practices

JS Mistakes

JS Performance

JS Reserved Words

JS Versions

JS Version ES5

JS Version ES6

JS JSON

JS Forms

JS Forms

Forms API

JS Objects

}

Try it Yourself »

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

Example

```
var i = 2;
var len = cars.length;
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



Free W
Toc



Improve gr
word choi
sentence s
in your wri
free



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 do anything like negative increment (`i--`), positive increment (`i = i + 15`), or anything else.

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) {
    text += person[x];
}
```

```
}
```

[Try it Yourself »](#)

The While Loop

The `while` loop and the `do/while` loop will be explained in the next chapter.


Test Yourself With Exercises

Exercise:

Create a loop that runs from 0 to 9.

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

[Submit Answer »](#)[Start the Exercise](#)[◀ Previous](#)[Next ▶](#)



AccuWeather APIs

Try It Out - Get Free Access to a Sampling of Our Weather API Endpoints

OPEN

- REPORT ERROR
- PRINT PAGE
- FORUM
- ABOUT

Top 10 Tutorials

- HTML Tutorial
- CSS Tutorial
- JavaScript Tutorial
- How To Tutorial
- W3.CSS Tutorial
- Bootstrap Tutorial
- SQL Tutorial
- PHP Tutorial
- jQuery Tutorial

Top 10 References

- HTML Reference
- CSS Reference
- JavaScript Reference
- W3.CSS Reference
- Bootstrap Reference
- SQL Reference
- PHP Reference
- HTML Colors
- jQuery Reference

Top 10 Examples

- HTML Examples
- CSS Examples
- JavaScript Examples
- How To Examples
- W3.CSS Examples
- Bootstrap Examples
- PHP Examples
- jQuery Examples
- Angular Examples

Web Certificates

- HTML Certificate
- CSS Certificate
- JavaScript Certificate
- jQuery Certificate
- PHP Certificate
- Bootstrap Certificate
- XML Certificate

[Python Tutorial](#)[Python Reference](#)[XML Examples](#)

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, cookie and privacy policy. Copyright 1999-2019 by Refsnes Data. All Rights Reserved.

Powered by W3.CSS.

