JavaScript para loop

[❮ Anterior](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_switch.asp&usg=ALkJrhhqmHTUIbkhtFsTrYCVUaID_rYRmA)[Próximo ❯](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_loop_while.asp&usg=ALkJrhhUNs9NMhGngJAl-SbUtQ-tF8mmeA)

Loops podem executar um bloco de código um número de vezes.

Loops de JavaScript

Loops são úteis, se você quiser executar o mesmo código uma e outra vez, cada vez com um valor diferente.

Muitas vezes, este é o caso quando se trabalha com arrays:

Em vez de escrever:

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

Você pode escrever:

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

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_loop_for&usg=ALkJrhir-10xSD2DJkoAnVQI535CkVvwcA)

Diferentes tipos de Loops

O JavaScript suporta diferentes tipos de loops:

* **para** - percorre um bloco de código várias vezes
* **para / in** -loops através das propriedades de um objeto
* **while** - percorre um bloco de código enquanto uma condição especificada é verdadeira
* **fazer / while** - também percorre um bloco de código enquanto uma condição especificada é verdadeira

The For Loop

O loop for geralmente é a ferramenta que você usará quando desejar criar um loop.

O loop for possui a seguinte sintaxe:

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

**A declaração 1** é executada antes do início do loop (o bloco de código).

**A instrução 2** define a condição para executar o loop (o bloco de código).

**A declaração 3** é executada cada vez que o loop (o bloco de código) foi executado.

Exemplo

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

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_loop_for_ex&usg=ALkJrhh_q7TbCUEz4RpVUXtDCJRAqHnlzg)

No exemplo acima, você pode ler:

A declaração 1 define uma variável antes do início do loop (var i = 0).

A instrução 2 define a condição para que o loop seja executado (eu devo ter menos de 5).

A Declaração 3 aumenta o valor (i ++) sempre que o bloco de código no loop foi executado.

Declaração 1

Normalmente você usará a indicação 1 para inicializar a variável usada no loop (i = 0).

Este não é sempre o caso, o JavaScript não se importa. A declaração 1 é opcional.

Você pode iniciar muitos valores na declaração 1 (separada por vírgula):

Exemplo

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

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_loop_for_om1&usg=ALkJrhhDaEnwQbGRdnuPqPrR4FL2bk5rcg)

E você pode omitir a instrução 1 (como quando seus valores são definidos antes do início do loop):

Exemplo

var i = 2;  
var len = cars.length;  
var text = "";  
for (; i < len; i++) {   
    text += cars[i] + "<br>";  
}

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_loop_for_om2&usg=ALkJrhgFpoC7LMwJcyzqUk7UA1wl_dfZUQ)

Declaração 2

Muitas vezes, a indicação 2 é usada para avaliar a condição da variável inicial.

Este não é sempre o caso, o JavaScript não se importa. A declaração 2 também é opcional.

Se a instrução 2 retornar verdadeiro, o loop começará novamente, se ele retornar falso, o loop terminará.

Se você omitir a indicação 2, você deve fornecer uma **pausa** no loop. Caso contrário, o loop nunca terminará.Isso irá travar seu navegador. Leia sobre quebras em um capítulo posterior deste tutorial.

Declaração 3

Muitas vezes, a instrução 3 incrementa o valor da variável inicial.

Este não é sempre o caso, o JavaScript não se importa, e a declaração 3 é opcional.

A Declaração 3 pode fazer qualquer coisa como incremento negativo (i--), incremento positivo (i = i + 15) ou qualquer outra coisa.

A Declaração 3 também pode ser omitida (como quando você incrementa seus valores dentro do loop):

Exemplo

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

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_loop_for_om3&usg=ALkJrhg6mNWPjauyszNktO35T-i4BoHFlw)

O For / In Loop

O JavaScript para / em guias de declaração através das propriedades de um objeto:

Exemplo

var person = {fname:"John", lname:"Doe", age:25};   
  
var text = "";  
var x;  
for (x in person) {  
    text += person[x];  
}

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_object_for_in&usg=ALkJrhiazLh4ztzFyUxv_u9Me4_Y8Frczw)

The While Loop

O loop while e o loop do / while serão explicados no próximo capítulo.