Arrays de JavaScript

[❮ 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_date_methods.asp&usg=ALkJrhgr4ErxfO5Ic5hr-czK9wV0hQWXrQ)[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_array_methods.asp&usg=ALkJrhjb0cSvDudUryrRbIUXnar_DmKx5g)

Arrays de JavaScript são usados ​​para armazenar vários valores em uma única variável.

Exemplo

var cars = ["Saab", "Volvo", "BMW"];

[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_array&usg=ALkJrhjH-eldMaTd1pvL2YXfbEyTP3FRZg)

O que é uma matriz?

Uma matriz é uma variável especial, que pode conter mais de um valor por vez.

Se você tem uma lista de itens (uma lista de nomes de carros, por exemplo), armazenar os carros em variáveis ​​individuais pode ser assim:

var car1 = "Saab";  
var car2 = "Volvo";  
var car3 = "BMW";

No entanto, e se você quiser percorrer os carros e encontrar um específico? E se você não tivesse 3 carros, mas 300?

A solução é uma matriz!

Uma matriz pode armazenar muitos valores com um único nome, e você pode acessar os valores consultando um número de índice.

Criando uma matriz

Usar uma matriz literal é a maneira mais fácil de criar uma matriz de JavaScript.

Sintaxe:

var *array\_name* = [ *item1* , *item2* , ...];

Exemplo

var cars = ["Saab", "Volvo", "BMW"];

[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_array&usg=ALkJrhjH-eldMaTd1pvL2YXfbEyTP3FRZg)

Espaços e quebras de linha não são importantes. Uma declaração pode abranger várias linhas:

Exemplo

var cars = [  
    "Saab",  
    "Volvo",  
    "BMW"  
];

[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_array_newlines&usg=ALkJrhi5qPa3m8Dw6NH1WaxjASwKj0q6OQ)

Colocar uma vírgula após o último elemento (como "BMW",) é inconsistente em todos os navegadores.

O IE 8 e anteriores falharão.

Usando a palavra-chave JavaScript nova

O exemplo a seguir também cria uma matriz e atribui valores a ela:

Exemplo

var cars = new Array("Saab", "Volvo", "BMW");

[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_array_new&usg=ALkJrhgi0ktbpsJGVxIczzZhK1_ULtqfzQ)

Os dois exemplos acima fazem exatamente o mesmo. Não é necessário usar o novo Array ().   
Por simplicidade, legibilidade e velocidade de execução, use o primeiro (método literal da matriz).

Acesse os elementos de uma matriz

Você se refere a um elemento de matriz referente ao **número de índice** .

Esta declaração acessa o valor do primeiro elemento em carros:

var name = cars[0];

Esta declaração modifica o primeiro elemento em carros:

cars[0] = "Opel";

Exemplo

var cars = ["Saab", "Volvo", "BMW"];  
document.getElementById("demo").innerHTML = cars[0];

[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_array_element&usg=ALkJrhiITSEDiNydSgdXbo9ZH9aED25Udw)

[0] é o primeiro elemento em uma matriz. [1] é o segundo. Os índices de matriz começam com 0.

Acesse a matriz completa

Com o JavaScript, pode acessar a matriz completa consultando o nome da matriz:

Exemplo

var cars = ["Saab", "Volvo", "BMW"];  
document.getElementById("demo").innerHTML = cars;

[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_array_full&usg=ALkJrhiVoUq9j39sdRM7PhIn2yA-q9Nx5w)

Arrays são objetos

Arrays são um tipo especial de objetos. O **tipo de** operador em JavaScript retorna "objeto" para arrays.

Mas, matrizes de JavaScript são melhor descritas como arrays.

Arrays usam **números** para acessar seus "elementos". Neste exemplo, a **pessoa [0]** retorna John:

Array:

var person = ["John", "Doe", 46];

[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_array_array&usg=ALkJrhhBfKSXC2bsdXQZ62ISQJluTOdftA)

Os objetos usam **nomes** para acessar seus "membros". Neste exemplo, **person.firstName** retorna John:

Objeto:

var person = {firstName:"John", lastName:"Doe", age:46};

[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_array_object&usg=ALkJrhh2ZWN0Gz_OMDiA3501byPebW0tnA)

Elementos de matriz podem ser objetos

As variáveis ​​de JavaScript podem ser objetos. Arrays são tipos especiais de objetos.

Por isso, você pode ter variáveis ​​de diferentes tipos na mesma matriz.

Você pode ter objetos em uma matriz. Você pode ter funções em uma matriz. Você pode ter arrays em uma matriz:

myArray[0] = Date.now;  
myArray[1] = myFunction;  
myArray[2] = myCars;

Array Properties and Methods

A força real das matrizes de JavaScript são as propriedades e métodos da matriz incorporada:

Exemplos

var x = cars.length;   // The length property returns the number of elements  
var y = cars.sort();   // The sort() method sorts arrays

Os métodos de matrizes são abordados nos próximos capítulos.

O comprimento Propriedade

A propriedade **length** de uma matriz retorna o comprimento de uma matriz (o número de elementos da matriz).

Exemplo

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.length;                       // the length of fruits is 4

[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_array_length&usg=ALkJrhhodzDmMVtKE_VWxUGHmIOo6P-IYA)

A propriedade length é sempre mais do que o índice de matriz mais alto.

Elementos da matriz de looping

A melhor maneira de percorrer uma matriz, está usando um loop "for":

Exemplo

var fruits, text, fLen, i;  
  
fruits = ["Banana", "Orange", "Apple", "Mango"];  
fLen = fruits.length;  
text = "<ul>";  
for (i = 0; i < fLen; i++) {  
    text += "<li>" + fruits[i] + "</li>";  
}

[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_array_loop&usg=ALkJrhg_0j4p8fAVkWJr6oBcPbLzqs97Uw)

Adicionando elementos de matriz

A maneira mais fácil de adicionar um novo elemento a uma matriz é usar o método push:

Exemplo

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.push("Lemon");                // adds a new element (Lemon) to fruits

[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_array_add_push&usg=ALkJrhj-epEEIQ3qtDm-mOrmAowA2kVwYw)

Novo elemento também pode ser adicionado a uma matriz usando a propriedade length:

Exemplo

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits[fruits.length] = "Lemon";     // adds a new element (Lemon) to fruits

[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_array_add&usg=ALkJrhgXFsR_BRf6eBm2yLg-VySbYq1CJA)

**ATENÇÃO !**

A adição de elementos com índices altos pode criar "buracos" indefinidos em uma matriz:

Exemplo

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits[6] = "Lemon";                 // adds a new element (Lemon) to fruits

[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_array_holes&usg=ALkJrhh69rcoAhFQi66PjmB73xmQqCvDSg)

Arrays associativos

Muitas linguagens de programação suportam arrays com índices nomeados.

Arrays com índices nomeados são chamados matrizes associativas (ou hashes).

O JavaScript **não** suporta arrays com índices nomeados.

Em JavaScript, os **arrays** sempre usam **índices numerados** .

Exemplo

var person = [];  
person[0] = "John";  
person[1] = "Doe";  
person[2] = 46;  
var x = person.length;         // person.length will return 3  
var y = person[0];             // person[0] will return "John"

[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_array_associative_1&usg=ALkJrhjugd7ZST_SNKSOVWn1Iq3GuHqViA)

**ATENÇÃO !!**   
Se você usa índices nomeados, o JavaScript redefinirá a matriz para um objeto padrão.   
Depois disso, alguns métodos e propriedades de matriz produzirão **resultados incorretos** .

Exemplo:

var person = [];  
person["firstName"] = "John";  
person["lastName"] = "Doe";  
person["age"] = 46;  
var x = person.length;         // person.length will return 0  
var y = person[0];             // person[0] will return undefined

[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_array_associative_2&usg=ALkJrhgOVPed5H5PCkMluTt7end65odVvA)

A Diferença entre Arrays e Objetos

Em JavaScript, os **arrays** usam **índices numerados** .

Em JavaScript, os **objetos** usam **índices nomeados** .

Arrays são um tipo especial de objetos, com índices numerados.

Quando usar Arrays. Quando usar objetos.

* O JavaScript não suporta arrays associativos.
* Você deve usar **objetos** quando quiser que os nomes dos elementos sejam **strings (texto)** .
* Você deve usar **arrays** quando quiser que os nomes dos elementos sejam **números** .

Evite nova matriz ()

Não há necessidade de usar o construtor de matriz incorporado do JavaScript **novo** Array ().

**Use [] em vez disso.**

Essas duas declarações diferentes criam uma nova matriz vazia chamada pontos:

var points = new Array();         // Bad  
var points = [];                  // Good

Essas duas declarações diferentes criam uma nova matriz contendo 6 números:

var points = new Array(40, 100, 1, 5, 25, 10); // Bad  
var points = [40, 100, 1, 5, 25, 10];          // Good

[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_array_literal&usg=ALkJrhhH_uMiH1jpjd02SlgosYfd0MQynQ)

A **nova** palavra-chave só complica o código. Também pode produzir alguns resultados inesperados:

var points = new Array(40, 100);  // Creates an array with two elements (40 and 100)

E se eu remover um dos elementos?

var points = new Array(40);       // Creates an array with 40 undefined elements !!!!!

[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_array_new_error&usg=ALkJrhiMMTTmmtzwpEvQjGjnzpBsRYskPg)

Como reconhecer uma matriz

Uma questão comum é: como eu sei se uma variável é uma matriz?

O problema é que o operador de JavaScript **typeof** retorna "objeto":

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
  
typeof fruits;             // returns object

[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_array_typeof&usg=ALkJrhi8RwmxkEP8h8Q0HiT07wm6GyNziw)

O operador tipoof retorna o objeto porque uma matriz de JavaScript é um objeto.

Solução 1:

Para resolver este problema, o ECMAScript 5 define um novo método **Array.isArray ()** :

Array.isArray(fruits);     // returns true

[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_array_isarray_method&usg=ALkJrhiGhMUy0gxnZd6CLY0d-Rp6mhfebA)

O problema com esta solução é que o ECMAScript 5 **não** é **suportado em navegadores mais antigos** .

Solução 2:

Para resolver este problema, você pode criar sua própria função isArray ():

function isArray(x) {  
    return x.constructor.toString().indexOf("Array") > -1;  
}

[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_array_isarray&usg=ALkJrhhFCcXvDY_SpFOUPM5pSDIXYJ45nQ)

A função acima sempre retorna true se o argumento for uma matriz.

Ou mais precisamente: ele retorna verdadeiro se o protótipo do objeto contiver a palavra "Array".

Solução 3:

O **instante do** operador retorna true se um objeto for criado por um determinado construtor:

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
  
fruits instanceof Array     // returns true

[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_array_instanceof&usg=ALkJrhgvB9nxnSi7tyj7S4YbMqYcqayezg)