# Matrizes de classificação 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_array_methods.asp&usg=ALkJrhjb0cSvDudUryrRbIUXnar_DmKx5g)[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_booleans.asp&usg=ALkJrhgJ9yVhiJDIueZKx5YuQRHnZgBcMQ)

## Classificando uma matriz

O método **sort ()** classifica uma matriz em ordem alfabética:

### Exemplo

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.sort();            // Sorts the elements of 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_sort&usg=ALkJrhjd-SW9ooILoVpGJXonbITroCNBZA)

## Inverter uma matriz

O método **reverso ()** invoca os elementos em uma matriz.

Você pode usá-lo para classificar uma matriz em ordem decrescente:

### Exemplo

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.sort();            // Sorts the elements of fruits   
fruits.reverse();         // Reverses the order of the 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_sort_reverse&usg=ALkJrhjJ1ShU-4j8U8TC09wo6dsS9AKf-g)

## Numeric Sort

Por padrão, a função sort () classifica valores como **strings** .

Isso funciona bem para cordas ("Apple" vem antes de "Banana").

No entanto, se os números são classificados como strings, "25" é maior do que "100", porque "2" é maior do que "1".

Por causa disso, o método sort () produzirá resultados incorretos ao classificar números.

Você pode corrigir isso fornecendo uma **função de comparação** :

### Exemplo

var points = [40, 100, 1, 5, 25, 10];  
points.sort(function(a, b){return a - b});

[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_sort2&usg=ALkJrhg6qB5uunCZZAoRuOTpPqISWiXn_g)

Use o mesmo truque para ordenar uma matriz descendente:

### Exemplo

var points = [40, 100, 1, 5, 25, 10];  
points.sort(function(a, b){return b - a});

[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_sort3&usg=ALkJrhjkVPJEJ9OcdqD_jAdIg4hlXsCtsQ)

## A função de comparação

O objetivo da função de comparação é definir uma ordem de classificação alternativa.

A função de comparação deve retornar um valor negativo, zero ou positivo, dependendo dos argumentos:

function(a, b){return a-b}

Quando a função sort () compara dois valores, ele envia os valores para a função de comparação e classifica os valores de acordo com o valor retornado (negativo, zero, positivo).

**Exemplo:**

Ao comparar 40 e 100, o método sort () chama a função de comparação (40,100).

A função calcula 40-100 e retorna -60 (um valor negativo).

A função de ordenação classificará 40 como um valor inferior a 100.

Você pode usar esse fragmento de código para experimentar com seleção numérica e alfabeticamente:

<button onclick="myFunction1()">Sort Alphabetically</button>  
<button onclick="myFunction2()">Sort Numerically</button>  
  
<p id="demo"></p>  
  
<script>  
var points = [40, 100, 1, 5, 25, 10];  
document.getElementById("demo").innerHTML = points;  
  
function myFunction1() {  
    points.sort();  
    document.getElementById("demo").innerHTML = points;  
}  
function myFunction2() {  
    points.sort(function(a, b){return a - b});  
    document.getElementById("demo").innerHTML = points;  
}  
</script>

[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_sort_alpha&usg=ALkJrhh9Yl4vf0y-3ANYrg8wIGVp8tHSrw)

## Classificando uma matriz em ordem aleatória

### Exemplo

var points = [40, 100, 1, 5, 25, 10];  
points.sort(function(a, b){return 0.5 - Math.random()});

[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_sort_random&usg=ALkJrhjPZBAHvf7y6wYNcrF49yrS20Lqdw)

## Encontre o valor de matriz mais alto (ou menor)

Não há funções internas para encontrar o valor máximo ou mínimo em uma matriz.

No entanto, depois de ter ordenado uma matriz, você pode usar o índice para obter os valores mais altos e mais baixos.

Classificar ascendente:

### Exemplo

var points = [40, 100, 1, 5, 25, 10];  
points.sort(function(a, b){return a - b});  
// now points[0] contains the lowest value  
// and points[points.length-1] contains the highest value

[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_sort_low&usg=ALkJrhj6ldSOLS-bAMgGNAblXX3z9-qKBw)

Ordenando descendente:

### Exemplo

var points = [40, 100, 1, 5, 25, 10];  
points.sort(function(a, b){return b - a});  
// now points[0] contains the highest value  
// and points[points.length-1] contains the lowest value

[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_sort_high&usg=ALkJrhjVuHhjdKTDVC4ficCj0i3ff9JfMg)

Classificar uma matriz inteira é um método muito ineficiente se você quiser apenas encontrar o valor mais alto (ou menor).

## Usando Math.max () em uma matriz

Você pode usar Math.max.apply para encontrar o número mais alto em uma matriz:

### Exemplo

function myArrayMax(arr) {  
    return Math.max.apply(null, arr);  
}

[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_sort_math_max&usg=ALkJrhi8ngQ2T-BQGBemcJs1tZ61DxKnlw)

Math.max.apply ([1, 2, 3]) é equivalente a Math.max (1, 2, 3).

## Usando Math.min () em uma matriz

Você pode usar Math.min.apply para encontrar o número mais baixo em uma matriz:

### Exemplo

function myArrayMin(arr) {  
    return Math.min.apply(null, arr);  
}

[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_sort_math_min&usg=ALkJrhg2YPi5VJ6-t4dP6pyPiYceC0qkqg)

Math.min.apply ([1, 2, 3]) é equivalente a Math.min (1, 2, 3).

## Meus métodos Min / Max JavaScript

A solução mais rápida é usar um método "caseiro".

Esta função percorre uma matriz que compara cada valor com o valor mais alto encontrado:

### Exemplo (Encontrar Máx)

function myArrayMax(arr) {  
    var len = arr.length  
    var max = -Infinity;  
    while (len--) {  
        if (arr[len] > max) {  
            max = arr[len];  
        }  
    }  
    return max;  
}

[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_sort_max&usg=ALkJrhjTZ84i_HR19x-BWNyM08A3YFIO-g)

Esta função percorre uma matriz que compara cada valor com o menor valor encontrado:

### Exemplo (Localizar Min)

function myArrayMin(arr) {  
    var len = arr.length  
    var min = Infinity;  
    while (len--) {  
        if (arr[len] < min) {  
            min = arr[len];  
        }  
    }  
    return min;  
}

[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_sort_min&usg=ALkJrhhmDPVvZ6_D1uk-bmlReLH194n_NQ)

## Classificando Arrays de Objetos

As matrizes de JavaScript geralmente contêm objetos:

### Exemplo

var cars = [  
{type:"Volvo", year:2016},  
{type:"Saab", year:2001},  
{type:"BMW", year:2010}];

Mesmo que os objetos tenham propriedades de diferentes tipos de dados, o método sort () pode ser usado para ordenar a matriz.

A solução é escrever uma função de comparação para comparar os valores da propriedade:

### Exemplo

cars.sort(function(a, b){return a.year - b.year});

[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_sort_object1&usg=ALkJrhjDRZFfnsj4a5rhvw9L_QxQeHixXg)

Comparar propriedades de string é um pouco mais complexo:

### Exemplo

 cars.sort(function(a, b){  
    var x = a.type.toLowerCase();  
    var y = b.type.toLowerCase();  
    if (x < y) {return -1;}  
    if (x > y) {return 1;}  
    return 0;  
});