Métodos String String

[❮ 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_strings.asp&usg=ALkJrhjAzOUB93tq872mMxqSSXWIaDoYJQ)[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_numbers.asp&usg=ALkJrhj7CM2GlY7w9rR75lynj81dOtIJJQ)

Os métodos String ajudam você a trabalhar com strings.

String Methods and Properties

Valores primitivos, como "John Doe", não podem ter propriedades ou métodos (porque não são objetos).

Mas com JavaScript, métodos e propriedades também estão disponíveis para valores primitivos, porque o JavaScript trata valores primitivos como objetos ao executar métodos e propriedades.

String Length

A propriedade **length** retorna o comprimento de uma string:

Exemplo

var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
var sln = txt.length;

[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_string_length&usg=ALkJrhj_XZsL0OcudCx2cz9Pkx63uP3LPQ)

Encontrando uma String em uma String

O método **indexOf ()** retorna o índice de (a posição de) a **primeira** ocorrência de um texto especificado em uma string:

Exemplo

var str = "Please locate where 'locate' occurs!";  
var pos = str.indexOf("locate");

[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_string_indexof&usg=ALkJrhi5Sj-_ZJorxxKBR_31LQ2R_rJe4A)

O método **lastIndexOf ()** retorna o índice da **última** ocorrência de um texto especificado em uma seqüência de caracteres:

Exemplo

var str = "Please locate where 'locate' occurs!";  
var pos = str.lastIndexOf("locate");

[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_string_lastindexof&usg=ALkJrhj53bMkbGO8i7WvQR8UxCsXIiARYw)

Ambos os métodos indexOf () e lastIndexOf () retornam -1 se o texto não for encontrado.

O JavaScript conta as posições de zero.   
0 é a primeira posição em uma string, 1 é a segunda, 2 é a terceira ...

Ambos os métodos aceitam um segundo parâmetro como a posição inicial para a pesquisa:

Exemplo

var str = "Please locate where 'locate' occurs!";  
var pos = str.indexOf("locate",15);

[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_string_indexof_2&usg=ALkJrhhLaiv2Ou2bs7GnbxUcxtQFWvFKhw)

Procurando por uma String em uma String

O método **search ()** procura uma string para um valor especificado e retorna a posição da partida:

Exemplo

var str = "Please locate where 'locate' occurs!";  
var pos = str.search("locate");

[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_string_search_locate&usg=ALkJrhgpk7cjS5wmKvY76ct6dGVevV7rfg)

Você notou?

Os dois métodos, indexOf () e search (), são iguais?

Eles aceitam os mesmos argumentos (parâmetros) e retornam o mesmo valor?

Os dois métodos são bastante iguais. Estas são as diferenças:

* O método search () não pode ter um segundo argumento de posição inicial.
* O método search () pode ter valores de pesquisa muito mais poderosos (expressões regulares).

Você aprenderá mais sobre expressões regulares em um capítulo posterior.

Extraindo Peças de Cordas

Existem 3 métodos para extrair uma parte de uma string:

* fatia (começo, fim)
* substring (start, end)
* substr (início, comprimento)

O método slice ()

**slice ()** extrai uma parte de uma string e retorna a parte extraída em uma nova string.

O método leva 2 parâmetros: o índice inicial (posição) e o índice final (posição).

Este exemplo corta uma porção de uma string da posição 7 à posição 13:

Exemplo

var str = "Apple, Banana, Kiwi";  
var res = str.slice(7, 13);

O resultado de res será:

Banana

[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_string_slice&usg=ALkJrhjqS2NPaIj9rjxC1DZCh8zH4qkl2w)

Se um parâmetro for negativo, a posição será contada a partir do fim da string.

Este exemplo corta uma porção de uma string da posição -12 para a posição -6:

Exemplo

var str = "Apple, Banana, Kiwi";  
var res = str.slice(-12, -6);

O resultado de res será:

Banana

[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_string_slice_negative&usg=ALkJrhhFVBPjdnrp8NShs-lCnwj4t3wG4g)

Se você omitir o segundo parâmetro, o método cortará o restante da seqüência de caracteres:

Exemplo

var res = str.slice(7);

[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_string_slice_rest&usg=ALkJrhg3uvFPjHApTY_VGkm_Zc_XRDbD4g)

ou, contando desde o fim:

Exemplo

var res = str.slice(-12);

[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_string_slice_rest_negative&usg=ALkJrhjzApz8CBGYJQTLED7s5IQg6Oyb8Q)

Posições negativas não funcionam no Internet Explorer 8 e anteriores.

O método substring ()

**substring ()** é semelhante a slice ().

A diferença é que substring () não pode aceitar índices negativos.

Exemplo

var str = "Apple, Banana, Kiwi";  
var res = str.substring(7, 13);

O resultado de *res* será:

Banana

[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_string_substring&usg=ALkJrhiv255FqsupuRqrF5vmE6PrEJ2bsQ)

Se você omitir o segundo parâmetro, a substring () irá cortar o restante da string.

O método substr ()

**substr ()** é semelhante a slice ().

A diferença é que o segundo parâmetro especifica o **comprimento** da parte extraída.

Exemplo

var str = "Apple, Banana, Kiwi";  
var res = str.substr(7, 6);

O resultado de res será:

Banana

[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_string_substr&usg=ALkJrhjnxAtKTmfEENNqe4VLsU45Hw4pvw)

Se o primeiro parâmetro for negativo, a posição conta do final da string.

O segundo parâmetro não pode ser negativo, pois define o comprimento.

Se você omitir o segundo parâmetro, substr () irá cortar o restante da string.

Substituindo Conteúdo de Cadeia

O método **replace ()** substitui um valor especificado por outro valor em uma string:

Exemplo

str = "Please visit Microsoft!";  
var n = str.replace("Microsoft", "W3Schools");

[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_string_replace&usg=ALkJrhiiY0Jcn34_bNDQU9L0YXKfmfvhTg)

O método replace () não altera a string em que é chamado. Ele retorna uma nova string.

Por padrão, a função replace () substitui **apenas a primeira** partida:

Exemplo

str = "Please visit Microsoft and Microsoft!";  
var n = str.replace("Microsoft", "W3Schools");

[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_string_replace_first&usg=ALkJrhgo7lMKa2Q--jVJgSI-shKk86Jn_A)

Para substituir todas as correspondências, use uma **expressão regular** com um sinalizador **/ g** (correspondência global):

Exemplo

str = "Please visit Microsoft and Microsoft!";  
var n = str.replace(/Microsoft/g, "W3Schools");

[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_string_replace_global&usg=ALkJrhjyazIWGImNLx0-APLdEgL0IBLTDA)

Por padrão, a função replace () é sensível a maiúsculas e minúsculas. Escrever MICROSOFT (com maiúsculas) não funcionará:

Exemplo

str = "Please visit Microsoft!";  
var n = str.replace("MICROSOFT", "W3Schools");

[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_string_replace_case&usg=ALkJrhgA4OAyYIt3HVgr9oZNM9up-ou78w)

Para substituir a insensibilidade a maiúsculas e minúsculas, use uma **expressão regular** com uma bandeira **/ i**(insensível):

Exemplo

str = "Please visit Microsoft!";  
var n = str.replace(/MICROSOFT/i, "W3Schools");

[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_string_replace_insensitive&usg=ALkJrhh9iZqeoiXTVottYi9Z8yia3-CSyw)

Você aprenderá muito mais sobre expressões regulares no capítulo [JavaScript Expressões regulares](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_regexp.asp&usg=ALkJrhhUmvXKu92_bnuhRFq3hMcpiZuSew) .

Convertendo para maiúsculas e minúsculas

Uma string é convertida em maiúsculas com **toUpperCase ()** :

Exemplo

var text1 = "Hello World!";       // String  
var text2 = text1.toUpperCase();  // text2 is text1 converted to upper

[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_string_toupper&usg=ALkJrhiGXG52VTHupALSYHTfHePpUS8afA)

Uma string é convertida em minúsculas com **toLowerCase ()** :

Exemplo

var text1 = "Hello World!";       // String  
var text2 = text1.toLowerCase();  // text2 is text1 converted to lower

[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_string_tolower&usg=ALkJrhiUANofU8pYiuIWC2nGXKVx4FSgDw)

O método concat ()

**concat ()** junta duas ou mais strings:

Exemplo

var text1 = "Hello";  
var text2 = "World";  
var text3 = text1.concat(" ", text2);

[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_string_concat&usg=ALkJrhheUfSz9ZjXKS3_GhuMqWN4AIK4Jw)

O método **concat ()** pode ser usado em vez do operador mais. Essas duas linhas fazem o mesmo:

Exemplo

var text = "Hello" + " " + "World!";  
var text = "Hello".concat(" ", "World!");

Todos os métodos de string retornam uma nova string. Eles não modificam a string original.   
Dito formalmente: as strings são imutáveis: as strings não podem ser alteradas, apenas substituídas.

Extraindo caracteres de cadeia

Existem 2 métodos **seguros** para extrair caracteres de seqüência de caracteres:

* CARTA (posição)
* charCodeAt (posição)

O método charAt ()

O método **charAt ()** retorna o caractere em um índice especificado (posição) em uma string:

Exemplo

var str = "HELLO WORLD";  
str.charAt(0);            // returns H

[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_string_charat&usg=ALkJrhiga3WPCml03VPu7nWF-2ymDxln4g)

O método charCodeAt ()

O método **charCodeAt ()** retorna o unicode do caractere em um índice especificado em uma string:

Exemplo

var str = "HELLO WORLD";  
  
str.charCodeAt(0);         // returns 72

[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_string_charcodeat&usg=ALkJrhi53oUUWkH7fcB1Jny9JzmQ-i7Dzg)

Acessar um String como uma matriz é inseguro

Você pode ter visto um código como esse, acessando uma string como uma matriz:

var str = "HELLO WORLD";  
  
str[0];                   // returns H

Isso é **inseguro** e **imprevisível:**

* Não funciona em todos os navegadores (não no IE5, IE6, IE7)
* Isso faz com que as cordas parecem arrays (mas não são)
* str [0] = "H" não dá um erro (mas não funciona)

Se você quiser ler uma string como uma matriz, converta-a em uma matriz primeiro.

Convertendo uma Cadeia para uma matriz

Uma string pode ser convertida em uma matriz com o método **split ()** :

Exemplo

var txt = "a,b,c,d,e";   // String  
txt.split(",");          // Split on commas  
txt.split(" ");          // Split on spaces  
txt.split("|");          // Split on pipe

[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_string_split&usg=ALkJrhg3-VCj_VQtXpMLxXpkcX33BP_b7w)

Se o separador for omitido, a matriz retornada conterá toda a cadeia no índice [0].

Se o separador for "", a matriz retornada será uma matriz de caracteres únicos:

Exemplo

var txt = "Hello";       // String  
txt.split("");           // Split in characters

[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_string_split_char&usg=ALkJrhhAvkS7lCqxax0on-27eWWidRj-0A)

Referência de Cadeia Completa

Para uma referência completa, acesse a [Referência de Cadeia de JavaScript Completa](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/jsref/jsref_obj_string.asp&usg=ALkJrhihJFxlMtBOplWZzpTlN2_wFy2ofw) .

A referência contém descrições e exemplos de todas as propriedades e métodos de cadeia.