# Expressões regulares 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_bitwise.asp&usg=ALkJrhgkHrAAFSEZyzYZRSlDaYO08cz0xA)[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_errors.asp&usg=ALkJrhhUtmkdaFtf3XbXmlUy0Ueqlc1-WA)

Uma expressão regular é uma seqüência de caracteres que forma um padrão de busca.

O padrão de pesquisa pode ser usado para operações de pesquisa de texto e substituição de texto.

## O que é uma expressão regular?

Uma expressão regular é uma seqüência de caracteres que forma um **padrão de busca** .

Quando você procura dados em um texto, você pode usar esse padrão de pesquisa para descrever o que você está procurando.

Uma expressão regular pode ser um único personagem, ou um padrão mais complicado.

Expressões regulares podem ser usadas para executar todos os tipos de operações de **pesquisa** de **texto** e **substituição de texto** .

## Sintaxe

/ pattern / modifiers ;

### Exemplo

var patt = /w3schools/i;

Exemplo explicado:

**/ w3schools / i** é uma expressão regular.

**O ensino médio** é um padrão (para ser usado em uma pesquisa).

**i** é um modificador (modifica a pesquisa para não ser sensível a maiúsculas e minúsculas).

## Usando Métodos de Cadeia

Em JavaScript, as expressões regulares são freqüentemente usadas com os dois **métodos de string** : search () e replace ().

**O método search ()** usa uma expressão para procurar uma correspondência e retorna a posição da partida.

**O método replace ()** retorna um string modificado onde o padrão é substituído.

## Usando a Pesquisa de Cadeia () Com uma Expressão Regular

### Exemplo

Use uma expressão regular para fazer uma pesquisa sem maiúsculas e minúsculas para "w3schools" em uma string:

var str = "Visit W3Schools";  
var n = str.search(/w3schools/i);

O resultado em n será:

6

[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_regexp&usg=ALkJrhjV1zu2SNfnLyMjpdzsfycoJ8rVZQ)

## Usando String search () com String

O método de busca também aceita uma string como argumento de pesquisa. O argumento de string será convertido em uma expressão regular:

### Exemplo

Use uma string para fazer uma pesquisa para "W3schools" em uma string:

var str = "Visit W3Schools!";  
var n = str.search("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_search&usg=ALkJrhi55b-qX3jLxveZR4CVSoGUmVbvwg)

## Use String replace () com uma Expressão Regular

### Exemplo

Use uma expressão regular insensível a maiúsculas e minúsculas para substituir a Microsoft por W3Schools em uma seqüência de caracteres:

var str = "Visit Microsoft!";  
var res = str.replace(/microsoft/i, "W3Schools");

O resultado em res será:

Visit 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_regexp&usg=ALkJrhgWKWRgGw3OA2f_ukd6Hyo_4KsW1A)

## Usando String replace () com uma String

O método replace () também aceita uma string como argumento de pesquisa:

var str = "Visit Microsoft!";  
var res = 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)

## Você notou?

Os argumentos de expressão regular (em vez de argumentos de string) podem ser usados ​​nos métodos acima.   
Expressões regulares podem tornar a sua pesquisa muito mais poderosa (caso insensível, por exemplo).

## Modificadores de expressões regulares

**Os modificadores** podem ser usados ​​para realizar buscas mais globais não sensíveis a maiúsculas e minúsculas:

|  |  |
| --- | --- |
| **Modifier** | **Description** |
| i | Perform case-insensitive matching |
| g | Perform a global match (find all matches rather than stopping after the first match) |
| m | Perform multiline matching |

## Padrões de Expressão Regular

**Brackets** são usados ​​para encontrar uma variedade de caracteres:

|  |  |
| --- | --- |
| **Expression** | **Description** |
| [abc] | Find any of the characters between the brackets |
| [0-9] | Find any of the digits between the brackets |
| (x|y) | Find any of the alternatives separated with | |

**Metacaracteres** são caracteres com um significado especial:

|  |  |
| --- | --- |
| **Metacharacter** | **Description** |
| \d | Find a digit |
| \s | Find a whitespace character |
| \b | Find a match at the beginning or at the end of a word |
| \uxxxx | Find the Unicode character specified by the hexadecimal number xxxx |

**Quantificadores** definem quantidades:

|  |  |
| --- | --- |
| **Quantifier** | **Description** |
| n+ | Matches any string that contains at least one n |
| n\* | Matches any string that contains zero or more occurrences of n |
| n? | Matches any string that contains zero or one occurrences of n |

## Usando o objeto RegExp

Em JavaScript, o objeto RegExp é um objeto de expressão regular com propriedades e métodos predefinidos.

## Usando teste ()

O método test () é um método de expressão RegExp.

Ele procura uma string para um padrão e retorna verdadeiro ou falso, dependendo do resultado.

O exemplo a seguir procura uma string para o caractere "e":

### Exemplo

var patt = /e/;  
patt.test("The best things in life are free!");

Uma vez que existe um "e" na string, a saída do código acima será:

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_regexp_test&usg=ALkJrhjE4l7B0vTU3oTUvPpZn3eMkzh37g)

Você não precisa colocar a expressão regular em uma variável primeiro. As duas linhas acima podem ser encurtadas para uma:

/e/.test("The best things in life are free!");

## Usando o exec ()

O método exec () é um método de expressão RegExp.

Ele procura uma string para um padrão especificado e retorna o texto encontrado.

Se nenhuma correspondência for encontrada, ela retornará *nula.*

O exemplo a seguir procura uma string para o caractere "e":

### Exemplo 1

/e/.exec("The best things in life are free!");

Uma vez que existe um "e" na string, a saída do código acima será:

e

[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_regexp_exec&usg=ALkJrhiigQxw9dcTxIImbQcfLLFTA00ANQ)

## Complete RegExp Reference

Para obter uma referência completa, acesse nossa [Referência completa do JavaScript RegExp](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_regexp.asp&usg=ALkJrhgtj3DEJGN5yNxOZZmmlEccHCF1Tg) .

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