# Operadores 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_variables.asp&usg=ALkJrhhDzgZLlB-xJBU42_lagfdUkOwXnw)[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_arithmetic.asp&usg=ALkJrhjOpPURz1rMExwQmRTYZ732DHCzOA)

### Exemplo

Atribua valores a variáveis ​​e adicione-os juntos:

var x = 5;         // assign the value 5 to x  
var y = 2;         // assign the value 2 to y  
var z = x + y;     // assign the value 7 to z (x + y)

[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_oper&usg=ALkJrhibTbcJ6XMtaspOvEPhkg4P5WNhrQ)

O operador de **atribuição** (=) atribui um valor a uma variável.

### Tarefa

var x = 10;

[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_oper_equal&usg=ALkJrhipGGEJm3wmDjpKYqoOQPMOCQ04eA)

O operador de **adição** (+) adiciona números:

### Adicionando

var x = 5;  
var y = 2;  
var z = x + y;

[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_oper_add&usg=ALkJrhjWlj5Aw8jDLa2UxJaBPKgn6tiJ-A)

O operador de **multiplicação** (\*) multiplica números.

### Multiplicando

var x = 5;  
var y = 2;  
var z = x \* y;

[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_oper_mult&usg=ALkJrhgRXqG0OT3gbHbC4V-TPUSrYfRMng)

## Operadores Aritméticos de JavaScript

Os operadores aritméticos são usados ​​para executar aritmética em números:

|  |  |
| --- | --- |
| **Operator** | **Description** |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| / | Division |
| % | Modulus |
| ++ | Increment |
| -- | Decrement |

Operadores aritméticos são totalmente descritos no capítulo [**JS Arithmetic**](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_arithmetic.asp&usg=ALkJrhjOpPURz1rMExwQmRTYZ732DHCzOA) .

## Operadores de atribuição de JavaScript

Os operadores de atribuição atribuem valores às variáveis ​​JavaScript.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Same As** |
| = | x = y | x = y |
| += | x += y | x = x + y |
| -= | x -= y | x = x - y |
| \*= | x \*= y | x = x \* y |
| /= | x /= y | x = x / y |
| %= | x %= y | x = x % y |

O operador de **atribuição de adição** (+ =) adiciona um valor a uma variável.

### Tarefa

var x = 10;  
x += 5;

[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_oper_plusequal&usg=ALkJrhhb1IuM6frd8gNNKw0TEv7nG5VdHw)

Os operadores de atribuição são totalmente descritos no capítulo de [**Atribuição JS**](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_assignment.asp&usg=ALkJrhgDWJbLDQUlWzrFSJF9on8JN8ZhBg) .

## Operadores de Cadeia de JavaScript

O operador + também pode ser usado para adicionar (concatenar) strings.

### Exemplo

txt1 = "John";  
txt2 = "Doe";  
txt3 = txt1 + " " + txt2;

O resultado de txt3 será:

John Doe

[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_oper_concatenate&usg=ALkJrhhP_9TjBkjPvWgB7W0m-B4y4uYL9A)

O operador + = atribuição também pode ser usado para adicionar (concatenar) strings:

### Exemplo

txt1 = "What a very ";  
txt1 += "nice day";

O resultado de txt1 será:

What a very nice day

[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_oper_concat4&usg=ALkJrhhnX7OTcMk4nUXfWA5pcF1r9_H5Bg)

Quando usado em strings, o operador + é chamado de operador de concatenação.

## Adicionando cordas e números

Adicionando dois números, retornará a soma, mas adicionando um número e uma seqüência de caracteres retornará uma string:

### Exemplo

x = 5 + 5;  
y = "5" + 5;  
z = "Hello" + 5;

O resultado de x , y e z será:

10  
55  
Hello5

[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_oper_concat5&usg=ALkJrhi6c43ahXrDHglkwrobFXU-S-StTQ)

Se você adicionar um número e uma string, o resultado será uma string!

## Operadores de comparação de JavaScript

|  |  |
| --- | --- |
| **Operator** | **Description** |
| == | equal to |
| === | equal value and equal type |
| != | not equal |
| !== | not equal value or not equal type |
| > | greater than |
| < | less than |
| >= | greater than or equal to |
| <= | less than or equal to |
| ? | ternary operator |

Os operadores de comparação estão completamente descritos no capítulo [**Comparações JS**](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_comparisons.asp&usg=ALkJrhj27Ds_37lhRLi_gJI0T73ZlSuNAg) .

## Operadores lógicos de JavaScript

|  |  |
| --- | --- |
| **Operator** | **Description** |
| && | logical and |
| || | logical or |
| ! | logical not |

Os operadores lógicos são totalmente descritos no capítulo [**Comparações JS**](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_comparisons.asp&usg=ALkJrhj27Ds_37lhRLi_gJI0T73ZlSuNAg) .

## Operadores de tipo JavaScript

|  |  |
| --- | --- |
| **Operator** | **Description** |
| typeof | Returns the type of a variable |
| instanceof | Returns true if an object is an instance of an object type |

Os operadores de tipo são totalmente descritos no capítulo [**Conversão de tipo JS**](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_type_conversion.asp&usg=ALkJrhjCCrCXWL-i-KsoENwDbjPajQ-Ing) .

## Operadores Bitwise do Bit

Os operadores de bit funcionam em números de 32 bits.

Qualquer operando numérico na operação é convertido em um número de 32 bits. O resultado é convertido de volta para um número de JavaScript.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operator** | **Description** | **Example** | **Same as** | **Result** | **Decimal** |
| & | AND | 5 & 1 | 0101 & 0001 | 0001 | 1 |
| | | OR | 5 | 1 | 0101 | 0001 | 0101 | 5 |
| ~ | NOT | ~ 5 | ~0101 | 1010 | 10 |
| ^ | XOR | 5 ^ 1 | 0101 ^ 0001 | 0100 | 4 |
| << | Zero fill left shift | 5 << 1 | 0101 << 1 | 1010 | 10 |
| >> | Signed right shift | 5 >> 1 | 0101 >> 1 | 0010 | 2 |
| >>> | Zero fill right shift | 5 >>> 1 | 0101 >>> 1 | 0010 | 2 |

Os exemplos acima usam exemplos não assinados de 4 bits. Mas o JavaScript usa números assinados de 32 bits.   
Por isso, em JavaScript, ~ 5 não retornará 10. Ele retornará -6.   
~ 00000000000000000000000000000101 retornará 11111111111111111111111111111010

Os operadores Bitwise são descritos completamente no capítulo [**JS Bitwise**](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) .