Booleanos do 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_sort.asp&usg=ALkJrhhbxaRDCRHQ7PqivHuSttJXzXnu-A)[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_comparisons.asp&usg=ALkJrhj27Ds_37lhRLi_gJI0T73ZlSuNAg)

Um JavaScript Boolean representa um dos dois valores: **true** ou **false** .

Valores booleanos

Muitas vezes, na programação, você precisará de um tipo de dados que só pode ter um dos dois valores, como

* SIM NÃO
* LIGADO DESLIGADO
* VERDADEIRO FALSO

Para isso, o JavaScript possui um tipo de dados **booleano** . Isso só pode levar os valores **true** ou **false** .

A função booleana ()

Você pode usar a função booleana () para descobrir se uma expressão (ou uma variável) é verdadeira:

Exemplo

Boolean(10 > 9)        // 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_boolean_expression1&usg=ALkJrhhXmhJtQmJ6Dhd5uvbK_Wt0m0CIbQ)

Ou ainda mais fácil:

Exemplo

(10 > 9)              // also returns true  
10 > 9                // also 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_boolean_expression2&usg=ALkJrhgdVq36U8LdMN_SETrRyY-VJySsdA)

Comparações e condições

O capítulo JS Comparison fornece uma visão geral completa dos operadores de comparação.

O capítulo JS Condições fornece uma visão geral completa das declarações condicionais.

aqui estão alguns exemplos:

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| == | equal to | if (day == "Monday") |
| > | greater than | if (salary > 9000) |
| < | less than | if (age < 18) |

O valor booleano de uma expressão é a base para todas as comparações e condições de JavaScript.

Tudo com um "valor" é verdadeiro

Exemplos

100  
  
3.14  
  
-15  
  
"Hello"  
  
"false"  
  
7 + 1 + 3.14

[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_boolean&usg=ALkJrhi__PjtRDNOQyCTFuALuT2GyCJOxg)

Tudo sem um "valor" é falso

O valor booleano de **0** (zero) é **falso** :

var x = 0;  
Boolean(x);       // returns false

[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_boolean_zero&usg=ALkJrhiuq1UhyyziC_xBFaeT_6lVNSU0Cw)

O valor booleano de **-0** (menos zero) é **falso** :

var x = -0;  
Boolean(x);       // returns false

[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_boolean_minus&usg=ALkJrhgmEvS6Pc5GE9hhyFeuLoPlg0HhOA)

O valor booleano de **""** (string vazio) é **falso** :

var x = "";  
Boolean(x);       // returns false

[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_boolean_empty&usg=ALkJrhgZdOltn8ApetbwfePkBvSl56tGLA)

O valor booleano de **indefinido** é **falso** :

var x;  
Boolean(x);       // returns false

[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_boolean_undefined&usg=ALkJrhhcq2St5WD6uQeVYzI0hFlkPJWFFg)

O valor booleano de **nulo** é **falso** :

var x = null;  
Boolean(x);       // returns false

[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_boolean_null&usg=ALkJrhhR5q6MF23uiTObD4FZ8omBEKfP7A)

O valor booleano de **falso** é (você adivinhou) **falso** :

var x = false;  
Boolean(x);       // returns false

[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_boolean_false&usg=ALkJrhhBt_19rhXBKOUnHnyXy3b4dgqeFw)

O valor booleano de **NaN** é **falso** :

var x = 10 / "H";  
Boolean(x);       // returns false

[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_boolean_nan&usg=ALkJrhhaUiqZtP4cEzg8YjHcUGMLKfXOXw)

Booleanos podem ser objetos

Normalmente, os booleanos de JavaScript são valores primitivos criados a partir de literais:

**var x = false;**

Mas os booleanos também podem ser definidos como objetos com a palavra-chave nova:

**var y = novo booleano (falso);**

Exemplo

var x = false;  
var y = new Boolean(false);  
  
// typeof x returns boolean  
// typeof y 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_booleans_object&usg=ALkJrhjEPwf0fGGIGoqp6Tp2YQuUqPNftQ)

Não crie objetos booleanos. Isso retarda a velocidade de execução.   
A **nova** palavra-chave complica o código. Isso pode produzir alguns resultados inesperados:

Ao usar o operador ==, os booleanos iguais são iguais:

Exemplo

var x = false;               
var y = new Boolean(false);  
  
// (x == y) is true because x and y have equal values

[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_booleans_object1&usg=ALkJrhi3mM4ZN2kMTXgnpM0TZgp_ql-fHw)

Ao usar o operador ===, booleanos iguais não são iguais, porque o operador === espera igualdade tanto no tipo como no valor.

Exemplo

var x = false;               
var y = new Boolean(false);  
  
// (x === y) is false because x and y have different types

[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_booleans_object2&usg=ALkJrhgn7W62Dzu6y27Gie2NmAY_PZQgsg)

Ou pior ainda. Os objetos não podem ser comparados:

Exemplo

var x = new Boolean(false);               
var y = new Boolean(false);  
  
// (x == y) is false because objects cannot be compared

[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_booleans_object3&usg=ALkJrhiRCSqytDzqk9ssMblYsPHFgnvUIQ)

Observe a diferença entre (x == y) e (x === y).   
A comparação de dois objetos JavaScript sempre retornará falsa.

Referência Booleana Completa

Para uma referência completa, acesse nossa [Referência Booleana 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_boolean.asp&usg=ALkJrhhn6ft2OT5K5G8eQp4iXPmuaGWvpw) .

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