Métodos de número 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_numbers.asp&usg=ALkJrhj7CM2GlY7w9rR75lynj81dOtIJJQ)[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_math.asp&usg=ALkJrhjmx1sXoByo9wlm2AmCMtNyV3Rlaw)

Os métodos numéricos ajudam você a trabalhar com números.

Número Métodos e Propriedades

Valores primitivos (como 3.14 ou 2014), não podem ter propriedades e 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.

O método toString ()

**toString ()** retorna um número como uma string.

Todos os métodos de número podem ser usados ​​em qualquer tipo de números (literais, variáveis ​​ou expressões):

Exemplo

var x = 123;  
x.toString();            // returns 123 from variable x  
(123).toString();        // returns 123 from literal 123  
(100 + 23).toString();   // returns 123 from expression 100 + 23

[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_number_tostring&usg=ALkJrhh5QSIgzuLCSeyUW7TZLIwa3HM2Hg)

O método toExponential ()

**toExponential ()** retorna uma string, com um número arredondado e escrito usando a notação exponencial.

Um parâmetro define o número de caracteres atrás do ponto decimal:

Exemplo

var x = 9.656;  
x.toExponential(2);     // returns 9.66e+0  
x.toExponential(4);     // returns 9.6560e+0  
x.toExponential(6);     // returns 9.656000e+0

[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_number_toexponential&usg=ALkJrhj41QGy7g8KuNCl5WCylgBO5r9ujw)

O parâmetro é opcional. Se você não especificá-lo, o JavaScript não contornará o número.

O método toFixed ()

**toFixed ()** retorna uma string, com o número escrito com um número especificado de decimais:

Exemplo

var x = 9.656;  
x.toFixed(0);           // returns 10  
x.toFixed(2);           // returns 9.66  
x.toFixed(4);           // returns 9.6560  
x.toFixed(6);           // returns 9.656000

[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_number_tofixed&usg=ALkJrhiAS4CERijmB1nVv73uApc3WZqjHw)

toFixed (2) é perfeito para trabalhar com dinheiro.

O método toPrecision ()

**toPrecision ()** retorna uma string, com um número escrito com um comprimento especificado:

Exemplo

var x = 9.656;  
x.toPrecision();        // returns 9.656  
x.toPrecision(2);       // returns 9.7  
x.toPrecision(4);       // returns 9.656  
x.toPrecision(6);       // returns 9.65600

[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_number_toprecision&usg=ALkJrhhE8SwRKgl5qyOVfQHBu4lzTMyoag)

O método valueOf ()

**valueOf ()** retorna um número como um número.

Exemplo

var x = 123;  
x.valueOf();            // returns 123 from variable x  
(123).valueOf();        // returns 123 from literal 123  
(100 + 23).valueOf();   // returns 123 from expression 100 + 23

[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_valueof&usg=ALkJrhhGMvR2CXpMZfKZ0eswiyPYerE6tw)

Em JavaScript, um número pode ser um valor primitivo (typeof = número) ou um objeto (typeof = objeto).

O método valueOf () é usado internamente em JavaScript para converter objetos numéricos em valores primitivos.

Não há motivo para usá-lo em seu código.

Todos os tipos de dados JavaScript têm um método valueOf () e toString ().

Conversão de variáveis ​​em números

Existem 3 métodos de JavaScript que podem ser usados ​​para converter variáveis ​​em números:

* O método Number ()
* O método parseInt ()
* O método parseFloat ()

Esses métodos não são métodos de **número** , mas métodos **globais de** JavaScript.

Métodos globais

Os métodos globais de JavaScript podem ser usados ​​em todos os tipos de dados JavaScript.

Estes são os métodos mais relevantes, quando se trabalha com números:

|  |  |
| --- | --- |
| **Method** | **Description** |
| Number() | Returns a number, converted from its argument. |
| parseFloat() | Parses its argument and returns a floating point number |
| parseInt() | Parses its argument and returns an integer |

O Método Número ()

**Number ()** pode ser usado para converter variáveis ​​JavaScript em números:

Exemplo

x = true;  
Number(x);        // returns 1  
x = false;       
Number(x);        // returns 0  
x = new Date();  
Number(x);        // returns 1404568027739  
x = "10"  
Number(x);        // returns 10  
x = "10 20"  
Number(x);        // returns NaN

[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_global_number&usg=ALkJrhgUs6KNGo1A1y_KpWFS0x3_HtKKPw)

Usado em Date (), o método Number () retorna o número de milissegundos desde 1.1.1970.

O método parseInt ()

**parseInt ()** analisa uma string e retorna um número inteiro. Espaços são permitidos. Apenas o primeiro número é retornado:

Exemplo

parseInt("10");         // returns 10  
parseInt("10.33");      // returns 10  
parseInt("10 20 30");   // returns 10  
parseInt("10 years");   // returns 10  
parseInt("years 10");   // returns NaN

[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_global_parseint&usg=ALkJrhj-3w8Qv67fXgKPGKfWOmGcGNE-BA)

Se o número não puder ser convertido, NaN (Não é um Número) é retornado.

O método parseFloat ()

**ParseFloat ()** analisa uma string e retorna um número. Espaços são permitidos. Apenas o primeiro número é retornado:

Exemplo

parseFloat("10");        // returns 10  
parseFloat("10.33");     // returns 10.33  
parseFloat("10 20 30");  // returns 10  
parseFloat("10 years");  // returns 10  
parseFloat("years 10");  // returns NaN

[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_global_parsefloat&usg=ALkJrhgSlrcgInAVp6ZmYVzF9xi-QsXScQ)

Se o número não puder ser convertido, NaN (Não é um Número) é retornado.

Propriedades do número

|  |  |
| --- | --- |
| **Property** | **Description** |
| MAX\_VALUE | Returns the largest number possible in JavaScript |
| MIN\_VALUE | Returns the smallest number possible in JavaScript |
| NEGATIVE\_INFINITY | Represents negative infinity (returned on overflow) |
| NaN | Represents a "Not-a-Number" value |
| POSITIVE\_INFINITY | Represents infinity (returned on overflow) |

Exemplo

var x = Number.MAX\_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_number_max&usg=ALkJrhiWq3yz0mhS28fgka7Ri0x9wor3eA)

As propriedades do número pertencem ao wrapper de objeto do número do JavaScript chamado **Número** .

Essas propriedades só podem ser acessadas como **Number** .MAX\_VALUE.

Usando *myNumber* .MAX\_VALUE, onde *myNumber* é uma variável, expressão ou valor, retornará indefinido:

Exemplo

var x = 6;  
var y = x.MAX\_VALUE;    // y becomes undefined

[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_number_max_undefined&usg=ALkJrhgWQJQeKTgjt-1EKmGexUcVQftHkQ)

Referência de número de JavaScript completa

Para obter uma referência completa, acesse a [Referência do número de JavaScript completo](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_number.asp&usg=ALkJrhipIHZ5OXCKdR04wwrg9CWreFbrig) .

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