Conversão de tipo 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_break.asp&usg=ALkJrhjlKbv5pnC5V2O8yGDZLo6MdoegQA)[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_bitwise.asp&usg=ALkJrhgkHrAAFSEZyzYZRSlDaYO08cz0xA)

Number () converte para um número, String () converte para um String, Boolean () converte para um booleano.

Tipos de dados JavaScript

Em JavaScript, existem 5 tipos de dados diferentes que podem conter valores:

* corda
* número
* boleano
* objeto
* função

Existem 3 tipos de objetos:

* Objeto
* Encontro
* Array

E 2 tipos de dados que não podem conter valores:

* nulo
* Indefinido

O tipo de Operador

Você pode usar o **tipoof** operador para encontrar o tipo de dados de uma variável JavaScript.

Exemplo

typeof "John"                 // Returns "string"   
typeof 3.14                   // Returns "number"  
typeof NaN                    // Returns "number"  
typeof false                  // Returns "boolean"  
typeof [1,2,3,4]              // Returns "object"  
typeof {name:'John', age:34}  // Returns "object"  
typeof new Date()             // Returns "object"  
typeof function () {}         // Returns "function"  
typeof myCar                  // Returns "undefined" \*  
typeof null                   // 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_typeof_all&usg=ALkJrhj-SvBsn91beXzc9cvy1STdv6pG3A)

Observe:

* O tipo de dados de NaN é o número
* O tipo de dados de uma matriz é objeto
* O tipo de dados de uma data é objeto
* O tipo de dados de nulo é objeto
* O tipo de dados de uma variável **indefinida** é **indefinido** \*
* O tipo de dados de uma variável que não recebeu um valor também é **indefinido** \*

Você não pode usar o **tipoof** para determinar se um objeto JavaScript é uma matriz (ou uma data).

O tipo de dados de tipo

O tipo de operador não é uma variável. É um operador. Operadores (+ - \* /) não têm nenhum tipo de dados.

Mas, o tipo de operador sempre **retorna uma string** (contendo o tipo de operando).

A propriedade do construtor

A propriedade do construtor retorna a função do construtor para todas as variáveis ​​de JavaScript.

Exemplo

"John".constructor                // Returns function String()  {[native code]}  
(3.14).constructor                // Returns function Number()  {[native code]}  
false.constructor                 // Returns function Boolean() {[native code]}  
[1,2,3,4].constructor             // Returns function Array()   {[native code]}  
{name:'John',age:34}.constructor  // Returns function Object()  {[native code]}  
new Date().constructor            // Returns function Date()    {[native code]}  
function () {}.constructor        // Returns function Function(){[native code]}

[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_constructor_all&usg=ALkJrhiFsBKSt5Lwxm6ZJkFkFIpNRpsMMg)

Você pode verificar a propriedade do construtor para descobrir se um objeto é uma matriz (contém a palavra "Array"):

Exemplo

function isArray(myArray) {  
    return myArray.constructor.toString().indexOf("Array") > -1;  
}

[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_array_isarray&usg=ALkJrhhFCcXvDY_SpFOUPM5pSDIXYJ45nQ)

Ou ainda mais simples, você pode verificar se o objeto é uma função de matriz:

Exemplo

function isArray(myArray) {  
    return myArray.constructor === Array;  
}

[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_type_isarray&usg=ALkJrhiiLVCZ-yDIzsjBAOY-aGSPQwqBXw)

Você pode verificar a propriedade do construtor para descobrir se um objeto é uma Data (contém a palavra "Data"):

Exemplo

function isDate(myDate) {  
    return myDate.constructor.toString().indexOf("Date") > -1;  
}

[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_date_isdate&usg=ALkJrhgVaU5USdh-V2tsAsYFO6bV8Fp7Mw)

Ou ainda mais simples, você pode verificar se o objeto é uma função Data:

Exemplo

function isDate(myDate) {  
    return myDate.constructor === Date;  
}

[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_type_isdate&usg=ALkJrhhJsteYvlQahtziUub8CcrVE-jI7A)

Conversão de tipo JavaScript

As variáveis ​​de JavaScript podem ser convertidas para uma nova variável e outro tipo de dados:

* Com o uso de uma função JavaScript
* **Automaticamente** pelo próprio JavaScript

Convertendo Números para Cordas

O método global **String ()** pode converter números em strings.

Ele pode ser usado em qualquer tipo de números, literais, variáveis ​​ou expressões:

Exemplo

String(x)         // returns a string from a number variable x  
String(123)       // returns a string from a number literal 123  
String(100 + 23)  // returns a string from a number from an expression

[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_string&usg=ALkJrhhAbMxIMgWTwU0rH4Wex_rCo-iopA)

O método Number **toString ()** faz o mesmo.

Exemplo

x.toString()  
(123).toString()  
(100 + 23).toString()

[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)

No capítulo [Número Métodos](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_number_methods.asp&usg=ALkJrhhLW5r1rthV-07AzrRkSCufdl_KTQ) , você encontrará mais métodos que podem ser usados ​​para converter números em strings:

|  |  |
| --- | --- |
| **Method** | **Description** |
| toExponential() | Returns a string, with a number rounded and written using exponential notation. |
| toFixed() | Returns a string, with a number rounded and written with a specified number of decimals. |
| toPrecision() | Returns a string, with a number written with a specified length |

Convertendo Booleanos para Strings

O método global **String ()** pode converter booleans em strings.

String(false)        // returns "false"  
String(true)         // returns "true"

O método booleano **toString ()** faz o mesmo.

false.toString()     // returns "false"  
true.toString()      // returns "true"

Converting Dates to Strings

O método global **String ()** pode converter datas para strings.

String(Date())      // returns "Thu Jul 17 2014 15:38:19 GMT+0200 (W. Europe Daylight Time)"

O método Data **toString ()** faz o mesmo.

Exemplo

Date().toString()   // returns "Thu Jul 17 2014 15:38:19 GMT+0200 (W. Europe Daylight Time)"

No capítulo [Métodos de data](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_date_methods.asp&usg=ALkJrhgr4ErxfO5Ic5hr-czK9wV0hQWXrQ) , você encontrará mais métodos que podem ser usados ​​para converter datas em strings:

|  |  |
| --- | --- |
| **Method** | **Description** |
| getDate() | Get the day as a number (1-31) |
| getDay() | Get the weekday a number (0-6) |
| getFullYear() | Get the four digit year (yyyy) |
| getHours() | Get the hour (0-23) |
| getMilliseconds() | Get the milliseconds (0-999) |
| getMinutes() | Get the minutes (0-59) |
| getMonth() | Get the month (0-11) |
| getSeconds() | Get the seconds (0-59) |
| getTime() | Get the time (milliseconds since January 1, 1970) |

Convertendo Cordas para Números

O método global **Number ()** pode converter strings em números.

Cordas contendo números (como "3.14") convertidos em números (como 3.14).

Cordas vazias convertidas para 0.

Qualquer outra coisa converte para NaN (Não é um número).

Number("3.14")    // returns 3.14  
Number(" ")       // returns 0   
Number("")        // returns 0  
Number("99 88")   // returns NaN

No capítulo [Número Métodos](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_number_methods.asp&usg=ALkJrhhLW5r1rthV-07AzrRkSCufdl_KTQ) , você encontrará mais métodos que podem ser usados ​​para converter seqüências de caracteres em números:

|  |  |
| --- | --- |
| **Method** | **Description** |
| parseFloat() | Parses a string and returns a floating point number |
| parseInt() | Parses a string and returns an integer |

O Operário Unary +

O **operador unary +** pode ser usado para converter uma variável em um número:

Exemplo

var y = "5";      // y is a string  
var x = + y;      // x is a number

[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_typeof3&usg=ALkJrhhmdMMyQ8PY3aN1-goT8NL2I5S7DQ)

Se a variável não puder ser convertida, ela ainda se tornará um número, mas com o valor NaN (Não é um número):

Exemplo

var y = "John";   // y is a string  
var x = + y;      // x is a number (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_typeof4&usg=ALkJrhjUdHR_-MXsKMzciUaJZ42KiLJycw)

Conversão de booleanos em números

O método global **Number ()** também pode converter booleans em números.

Number(false)     // returns 0  
Number(true)      // returns 1

Conversão de datas em números

O método global **Number ()** pode ser usado para converter datas em números.

d = new Date();  
Number(d)          // returns 1404568027739

O método de data **getTime ()** faz o mesmo.

d = new Date();  
d.getTime()        // returns 1404568027739

Conversão Automática de Tipo

Quando o JavaScript tenta operar em um tipo de dados "errado", ele tentará converter o valor para um tipo "correto".

O resultado nem sempre é o que você espera:

5 + null    // returns 5         because null is converted to 0  
"5" + null  // returns "5null"   because null is converted to "null"  
"5" + 2     // returns 52        because 2 is converted to "2"  
"5" - 2     // returns 3         because "5" is converted to 5  
"5" \* "2"   // returns 10        because "5" and "2" are converted to 5 and 2

[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_type_convert_auto&usg=ALkJrhgTchRh2IFmhYuCd0Ck1rRsTSfWMA)

Conversão Automática de Cadeia

O JavaScript chama automaticamente a função toString () da variável quando você tenta "exibir" um objeto ou uma variável:

document.getElementById("demo").innerHTML = myVar;  
  
// if myVar = {name:"Fjohn"}  // toString converts to "[object Object]"  
// if myVar = [1,2,3,4]       // toString converts to "1,2,3,4"  
// if myVar = new Date()      // toString converts to "Fri Jul 18 2014 09:08:55 GMT+0200"

Números e booleanos também são convertidos, mas isso não é muito visível:

// if myVar = 123             // toString converts to "123"  
// if myVar = true            // toString converts to "true"  
// if myVar = false           // toString converts to "false"

Tabela de Conversão de Tipo de JavaScript

Esta tabela mostra o resultado da conversão de valores de JavaScript diferentes para Number, String e Boolean:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Original Value** | **Converted to Number** | **Converted to String** | **Converted to Boolean** | **Try it** |
| false | 0 | "false" | false | [Try it »](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_type_convert_false&usg=ALkJrhiRjf9UGaL6vuh1-_TnGx-bVvrFjg) |
| true | 1 | "true" | true | [Try it »](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_type_convert_true&usg=ALkJrhhD7w8hEC9icSiKl4ab1JgTCaIrGA) |
| 0 | 0 | "0" | false | [Try it »](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_type_convert_number_0&usg=ALkJrhgCtWjCuQMDgA7Q6Lulgv_VyCgnow) |
| 1 | 1 | "1" | true | [Try it »](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_type_convert_number_1&usg=ALkJrhjq-lWEovu2c5g3o43zxb48idzE2g) |
| "0" | 0 | "0" | **true** | [Try it »](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_type_convert_string_0&usg=ALkJrhh-bkrog33bo_EQepSwzG7oiHjduw) |
| "000" | 0 | "000" | **true** | [Try it »](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_type_convert_string_000&usg=ALkJrhjbeXIWudeaH61YjVR9nBMKlUk8Wg) |
| "1" | 1 | "1" | true | [Try it »](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_type_convert_string_1&usg=ALkJrhhO5X0p96ACvUcuLD6kuo5g4JMm9Q) |
| NaN | NaN | "NaN" | false | [Try it »](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_type_convert_nan&usg=ALkJrhhMIjbF_w6kXQ7Ez3s_vnoU6hWm1w) |
| Infinity | Infinity | "Infinity" | true | [Try it »](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_type_convert_infinity&usg=ALkJrhhXBTYU0mAs6N_QU1yZ34FOgmZwXw) |
| -Infinity | -Infinity | "-Infinity" | true | [Try it »](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_type_convert_infinity_minus&usg=ALkJrhgTn82zc6gZoPxAsxHeFKRgqggv2g) |
| "" | **0** | "" | **false** | [Try it »](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_type_convert_string_empty&usg=ALkJrhjZwzIzzZyIiSZamx6aR-GTGKEOKQ) |
| "20" | 20 | "20" | true | [Try it »](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_type_convert_string_number&usg=ALkJrhiC17CDI8EapRaJeUnOI5p0MLoOgA) |
| "twenty" | NaN | "twenty" | true | [Try it »](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_type_convert_string_text&usg=ALkJrhgLDdn6QPXvV8YhhQaVh_V5KNBm9A) |
| [ ] | **0** | "" | true | [Try it »](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_type_convert_array_empty&usg=ALkJrhihHZvxHff3hWB7yO3LCgh2Jg_kHQ) |
| [20] | **20** | "20" | true | [Try it »](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_type_convert_array_one_number&usg=ALkJrhiJDgqkv3eQnedMCUgArNR6gyiKww) |
| [10,20] | NaN | "10,20" | true | [Try it »](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_type_convert_array_two_numbers&usg=ALkJrhiVnGIPk1EtYuaGl1IR9xQdzJDsiw) |
| ["twenty"] | NaN | "twenty" | true | [Try it »](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_type_convert_array_one_string&usg=ALkJrhi5eoU8m2X2zbEiSoTgNznemkakXA) |
| ["ten","twenty"] | NaN | "ten,twenty" | true | [Try it »](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_type_convert_array_two_strings&usg=ALkJrhj30UqXSejogamJuvGm44zErB-SOA) |
| function(){} | NaN | "function(){}" | true | [Try it »](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_type_convert_function&usg=ALkJrhhjps2Z2eudbPFgGuRRFDHpJHVjoQ) |
| { } | NaN | "[object Object]" | true | [Try it »](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_type_convert_object&usg=ALkJrhi5_7AtCmyBN4SGiKYB43rByjfMxA) |
| null | **0** | "null" | false | [Try it »](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_type_convert_null&usg=ALkJrhiOkt6T7oz368mNIlHQkHkQwHOVBw) |
| undefined | NaN | "undefined" | false | [Try it »](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_type_convert_undefined&usg=ALkJrhhj4ZCo4RbXU6oCbF7fl9DsDfzXXw) |

Os valores entre aspas indicam valores de seqüência de caracteres.

**Os valores vermelhos** indicam valores (alguns) programadores podem não esperar.