

Às vezes, você pode precisar realizar uma conversão explícita, ou você pode preferir tornar as conversões explícitas para manter seu código mais claro. A maneira mais simples de realizar uma conversão de tipo explícita é usar as funções booleanas (), number () e string ():

Número ("3") // => 3

String (false) // => "false": ou use false.toString ()

Booleano ([]) // => true

Qualquer valor que não seja nulo ou indefinido tem um método toString (), e o resultado desse método geralmente é o mesmo retornado pela função String ().

Como um aparte, observe que os booleanos (), number () e string ()

As funções também podem ser invocadas - com novo - como construtor. Se você usa elas dessa maneira, você terá um objeto "wrapper" que se comporta exatamente como um valor primitivo de booleano, número ou string. Esses objetos de envoltório são um legado histórico desde os primeiros dias de JavaScript, e há

Nunca realmente qualquer um bom motivo para usá-los.

Certos operadores de JavaScript realizam conversões de tipo implícito e são

Às vezes usado explicitamente para fins de conversão de tipo. Se um

operando do operador + é uma string, ele converte o outro em um

string. O operador UNARY + converte seu operando em um número. E o

UNARY!O operador converte seu operando em um booleano e nega.

Esses fatos levam aos seguintes idiomas de conversão de tipo que você pode

Veja em algum código:

x + "" // => string (x)