

1 + 2 // => 3: adição.

"1" + "2" // => "12": concatenação.

"1" + 2 // => "12": 2 é convertido em "2".

11 < 3 // => Falso: Comparação numérica.

"11" < "3" // => true: comparação de string.

"11" < 3 // => False: Comparação numérica, "11" convertida a 11.

"um" < 3 // => false: comparação numérica, "um" convertido para nan.

Finalmente, observe que o <= (menor ou igual) e >= (maior que ou igual) operadores não confiam na igualdade ou em operadores estritos de igualdade. Para determinar se dois valores são "iguais". Em vez disso, o menos do que operador ou igual é simplesmente definido como "não maior que" e o O operador maior do que ou igual é definido como "não menor que". O que a exceção ocorre quando um operando é (ou converte para) nan, em que Caso, todos os quatro operadores de comparação retornam falsos.

4.9.3 O operador no

O operador in espera um operando do lado esquerdo que é uma corda, símbolo ou valor que pode ser convertido em uma string. Espera um operando do lado direito isso é um objeto. Ele avalia para verdadeiro se o valor do lado esquerdo for o nome de uma propriedade do objeto do lado direito. Por exemplo:

deixe Point = {x: 1, y: 1}; // Defina um objeto

"X" no ponto // => true: objeto tem propriedade chamado "X"

"Z" no ponto // => false: objeto não tem "z" propriedade.

"ToString" no ponto // => true: objeto herda

Método da ToString

deixe dados = [7,8,9]; // Uma matriz com elementos (índices) 0, 1 e 2

"0" em dados // => true: a matriz tem um elemento

"0"