

corda

1 + {} // => "1 [objeto objeto]": concatenação após

objeto para cordas

verdadeiro + verdadeiro // => 2: adição após boolean-to-number

2 + nulo // => 2: adição após o nulo se converte para 0

2 + indefinido // => nan: adição após convertidos indefinidos para

Nan

Finalmente, é importante observar que quando o operador + é usado com Strings e números, pode não ser associativa. Isto é, o resultado pode depender da ordem em que as operações são executadas.

Por exemplo:

1 + 2 + "camundongos cegos" // => "3 ratos cegos"

1 + (2 + "camundongos cegos") // => "12 camundongos cegos"

A primeira linha não tem parênteses e o operador + tem da esquerda para a direita associatividade, então os dois números são adicionados primeiro e sua soma é

concatenado com a string. Na segunda linha, parênteses alteram isso

Ordem de operações: o número 2 é concatenado com a string para

produzir uma nova corda. Então o número 1 é concatenado com o novo

string para produzir o resultado final.

#### 4.8.2 Operadores aritméticos unários

Operadores unários modificam o valor de um único operando para produzir um novo valor. Em JavaScript, todos os operadores unários têm alta precedência e

são todos bem associativos. Os operadores aritméticos unários descritos em

Esta seção (+, -, ++ e -) todos convertem seu único operando em um

número, se necessário. Observe que os caracteres de pontuação + e - são

usado como operadores unários e binários.