

`1 + 2 // => 3`

`"Olá" + "" + "lá" // => "Olá"`

`"1" + "2" // => "12"`

Quando os valores de ambos os operandos são números ou são ambas as cordas,

Então é óbvio o que o operador + faz. Em qualquer outro caso, no entanto,

A conversão do tipo é necessária e a operação a ser realizada

Depende da conversão realizada. As regras de conversão para + têm

prioridade à concatenação da string: se qualquer um dos operandos for uma string ou

um objeto que se converte em uma string, o outro operando é convertido em um

String e concatenação são realizadas. A adição é realizada apenas se

Nenhuma operando é semelhante a uma corda.

Tecnicamente, o operador + se comporta assim:

Se um de seus valores de operando for um objeto, ele o converte em um

primitivo usando o algoritmo de objeto a princípio descrito em

§3.9.3. Os objetos de data são convertidos por seu `toSource()`

método e todos os outros objetos são convertidos via `valueOf()`,

Se esse método retornar um valor primitivo. No entanto, a maioria

Objetos não têm um método de valor útil `valueOf()`, então são

convertido via `toString()` também.

Após a conversão de objeto para primitivo, se qualquer um operando for um

string, a outra é convertida em uma corda e a concatenação é

realizado.

Caso contrário, ambos os operandos são convertidos em números (ou para

Nan) e adição é realizada.

Aqui estão alguns exemplos:

`1 + 2 // => 3`: adição

`"1" + "2" // => "12"`: concatenação

`"1" + 2 // => "12"`: concatenação após número a-