e são cobertos primeiro.O operador de adição recebe uma subseção própria
Porque também pode executar concatenação de string e tem alguns incomuns
Tipo Regras de conversão.Os operadores unários e os operadores bitwise
também são cobertos por subseções próprias.
A maioria desses operadores aritméticos (exceto como observado como segue) pode ser
usado com bigint (ver §3.2.5) operando ou com números regulares, como
desde que você não misture os dois tipos.
Os operadores aritméticos básicos são ** (exponenciação), *
(multiplicação), / (divisão), % (Modulo: restante após a divisão), +
(adição) e - (subtração).Como observado, discutiremos o operador +
em uma seção própria.Os outros cinco operadores básicos simplesmente avaliam
seus operando, converte os valores em números, se necessário, e depois
Calcule o poder, produto, quociente, restante ou diferença.Não-
operando numéricos que não podem se converter em números convertidos para a nan
valor.Se qualquer um opera é (ou converter para) nan, o resultado do
a operação é (quase sempre) nan.
O operador ** tem maior precedência que *, /e % (que por sua vez
tem maior precedência que + e -).Ao contrário dos outros operadores, **
Funciona a direita para a esquerda, então 2 ** 2 ** 3 é o mesmo que 2 ** 8, não 4 ** 3.
Há uma ambiguidade natural em expressões como -3 ** 2.Dependendo de
a relativa precedência de unário menos e exponenciação, que
A expressão pode significar (-3) ** 2 ou -(3 ** 2).Diferentes idiomas
lidar com isso de maneira diferente e, em vez de escolher lados, JavaScript simplesmente
torna um erro de sintaxe omitir parênteses neste caso, forçando você a
Escreva uma expressão inequívoca.** é a mais nova aritmética de JavaScript