executado primeiro, escreva:
w = (x + y)*z;
Observe que as expressões de acesso e invocação de propriedades têm maior
Precedência do que qualquer um dos operadores listados na Tabela 4-1.Considere isso
expressão:
// meu é um objeto com uma propriedade chamada funções cujo
O valor é um
// Matriz de funções.Invocamos o número da função X, passando
argumento
// y, e então solicitamos o tipo de valor retornado.
tipo de my.fuções [x] (y)
Embora o tipo de seja um dos operadores de maior prioridade, o
Tipoof Operação é realizada no resultado do acesso à propriedade,
índice de matriz e invocação de funções, todos com maior prioridade
do que operadores.
Na prática, se você não tiver certeza sobre a precedência de seu
operadores, a coisa mais simples a fazer é usar parênteses para fazer o
Ordem de avaliação explícita.As regras que são importantes a saber são
estes: multiplicação e divisão são realizadas antes da adição e
subtração, e a atribuição tem muito baixa precedência e é quase
Sempre executado por último.
Quando novos operadores são adicionados ao JavaScript, eles nem sempre se encaixam
naturalmente nesse esquema de precedência.O ??Operador (§4.13.2) é
Mostrado na tabela como menor precedência que ||e &&, mas, de fato, seu
Precedência em relação a esses operadores não está definida e ES2020
Requer que você use explicitamente parênteses se você misturar ??com qualquer um deles ||