

Delete é um operador unário que tenta excluir a propriedade do objeto ou elemento da matriz especificado como seu operando. Como a tarefa, Incremento e operadores de decréscimo, o exclusão é normalmente usado para o seu Efeito colateral da exclusão da propriedade e não para o valor que ele retorna. Alguns

Exemplos:

Seja o = {x: 1, y: 2}; // Comece com um objeto
excluir o.x; // Exclua uma de suas propriedades
"x" em o // => false: a propriedade não
existe mais

Seja a = [1,2,3]; // Comece com uma matriz
exclua um [2]; // exclua o último elemento do
variedade

2 em a // => false: o elemento da matriz 2 não
existe mais

A.Length // => 3: Observe que o comprimento da matriz

Não muda, no entanto

Observe que uma propriedade excluída ou elemento da matriz não é apenas definido como o valor indefinido. Quando uma propriedade é excluída, a propriedade deixa de existir. Tentando ler uma propriedade inexistente retorna indefinida, Mas você pode testar a existência real de uma propriedade com o in operador (§4.9.3). A exclusão de um elemento de matriz deixa um "buraco" na matriz e não muda o comprimento da matriz. A matriz resultante é escassa (§7.3).

Delete espera que seu operando seja um LValue. Se não é um lvalue, o O operador não toma ação e retorna verdadeiro. Caso contrário, exclua Tentativas de excluir o LValue especificado. excluir retorna verdadeiro se for Exclui com sucesso o LValue especificado. Nem todas as propriedades podem ser Excluído, no entanto: propriedades não confundíveis (§14.1) são imunes