

Segunda linha quebra como um semicolon porque pode continuar analisando o
Declaração mais longa `a = 3;`.

Essas regras de rescisão de declaração levam a alguns casos surpreendentes. Esse
O código parece duas declarações separadas separadas com uma nova linha:

Deixe `y = x + f`

`(a+b) .ToString ()`

Mas os parênteses na segunda linha de código podem ser interpretados como um
Invocação de função de `F` a partir da primeira linha, e JavaScript interpreta
O código como este:

Seja `y = x + f (a + b) .ToString ();`

É mais provável que não, essa não é a interpretação pretendida pelo
autor do código. Para trabalhar como duas declarações separadas, um
O semicolon explícito é necessário neste caso.

Em geral, se uma declaração começa com `(`, `[`, `/`, `+`, ou `-`, há uma chance
que isso poderia ser interpretado como uma continuação da declaração antes.

Declarações que começam com `/`, `+` e `-` são bastante raras na prática, mas
declarações começando com `(` e `[` não são incomuns, pelo menos em

Alguns estilos de programação JavaScript. Alguns programadores gostam de colocar
um semicolon defensivo no início de qualquer declaração para que
continuará a funcionar corretamente, mesmo que a declaração antes seja

Modificado e um semicolon de terminação anteriormente removido:

Seja `x = 0 // semicolon omitido aqui`

`; [x, x+1, x+2] .foreach (console.log) // defensivo; mantém isso`
declaração separada