

e se o valor à direita for falsamente, o valor geral deve ser falsidade. Então, quando o valor à esquerda é verdadeiro, o operador de && avalia e retorna o valor à direita:

Seja o = {x: 1};

Seja p = nulo;

O && o.x // => 1: o é verdade, então retornar o valor de O.x

p && p.x // => null: p é falsidade, então devolva e não

Avalie P.X

É importante entender que && pode ou não avaliar seu direito

operando lateral. Neste exemplo de código, a variável p é definida como nula e

A expressão P.X, se avaliada, causaria um TypeError. Mas o

Usos de código && de maneira idiomática para que o P.X seja avaliado apenas se P for

Verdade - não nula ou indefinida.

O comportamento de && às vezes é chamado de curto -circuito, e você pode

às vezes veja o código que explora propositalmente esse comportamento para

Execute o código condicionalmente. Por exemplo, as duas linhas a seguir de

O código JavaScript tem efeitos equivalentes:

if (a === b) stop (); // Invocar Stop () Somente se A === B

(a === b) && stop (); // Isso faz a mesma coisa

Em geral, você deve ter cuidado sempre que escrever uma expressão com

efeitos colaterais (atribuições, incrementos, decréscimos ou função

invocações) no lado direito de &&. Se esses efeitos colaterais

Ocorre depende do valor do lado esquerdo.

Apesar da maneira um tanto complexa que esse operador realmente funciona, é

é mais comumente usado como um simples operador de álgebra booleano que