Expressões relacionais sempre avaliam como verdadeiro ou falso, então quando
Usado assim, o próprio operador && retorna verdadeiro ou falso.
Os operadores relacionais têm maior precedência que && (e ||), então
Expressões como essas podem ser escritas com segurança sem parênteses.
Mas && não exige que seus operando sejam valores booleanos.Lembre -se disso
Todos os valores de JavaScript são "verdadeiros" ou "falsamente".(Veja §3.4 para obter detalhes.
Os valores falsamente são falsos, nulos, indefinidos, 0, -0, nan e
"".Todos os outros valores, incluindo todos os objetos, são verdadeiros.) O segundo nível
em que && pode ser entendido é como um booleano e operador para
valores verdadeiros e falsamente.Se ambos os operandos são verdadeiros, o operador retorna
um valor verdadeiro.Caso contrário, um ou ambos os operando devem ser falsamente, e o
O operador retorna um valor falsamente.Em JavaScript, qualquer expressão ou
declaração que espera que um valor booleano funcione com um verdadeiro ou falsamente
valor, então o fato de que && nem sempre retorna verdadeiro ou falso
não causar problemas práticos.
Observe que esta descrição diz que o operador retorna ?um verdadeiro
valor ?ou? um valor falsamente ?, mas não especifica o que é esse valor.Para
Isso, precisamos descrever && no terceiro e último nível.Este operador
Começa avaliando seu primeiro operando, a expressão à sua esquerda.Se o
O valor à esquerda é falsamente, o valor de toda a expressão também deve ser
falsy, então && simplesmente retorna o valor à esquerda e nem mesmo
Avalie a expressão à direita.
Por outro lado, se o valor à esquerda for verdadeiro, então o geral
O valor da expressão depende do valor do lado direito.Se
O valor à direita é verdadeiro, então o valor geral deve ser verdade,