

`(x === 2) && (y === 3) // => true`: ambas as comparações são verdadeiro.&& é e  
`(x > 3) || (y < 3) // => false`: Nenhuma comparação é verdadeiro.|| é ou  
`! (x === y) // => true` :inverte um booleano  
valor

Se as expressões JavaScript são como frases, as declarações JavaScript são como frases completas.Declarações são o tópico do capítulo 5. Aproximadamente, Uma expressão é algo que calcula um valor, mas não faz

Qualquer coisa: não altera o estado do programa de forma alguma.Declarações, em Por outro lado, não tem um valor, mas eles alteram o estado.Você tem

Declarações variáveis ??e declarações de atribuição acima.O outro ampla categoria de declaração são estruturas de controle, como condicionais e loops.Você verá exemplos abaixo, depois de cobrirmos funções.

Uma função é um bloco nomeado e parametrizado de código JavaScript que Você define uma vez e pode invocar repetidamente.Funções não são cobertos formalmente até o capítulo 8, mas como objetos e matrizes, Você os verá muitas vezes antes de chegar a esse capítulo.Aqui estão

Alguns exemplos simples:

// As funções são blocos parametrizados de código JavaScript que Podemos invocar.

função plus1 (x) { // define uma função chamada "plus1"

com o parâmetro "X"

retornar x + 1; // retorna um valor maior que

O valor passou em

} // As funções são fechadas em Curly

aparelho ortodôntico

Plus1 (y) // => 4: y é 3, então este

Retornos de invocação 3+1

Deixe Square = function (x) { // As funções são valores e podem ser atribuído a vars