Sintaxe e Operadores JavaScript

Operadores

**Tipos de Operadores**

* Aritmética
* Atribuição
* Comparação
* Lógica
* Condicional

Atribuição

Um operador de atribuição atribui um valor ao operando à sua esquerda baseado no valor do operando à direita. O operador de atribuição básico é o igual (=), que atribui o valor do operando à direita ao operando à esquerda. Isto é, x = y atribui o valor de y a x.

|  |  |  |
| --- | --- | --- |
| Operador | Exemplo | Equivalente a |
| = | X = Y | X = Y |
| += | X += Y | X = X + Y |
| \*= | X \*= Y | X = X \* Y |
| /= | X /= Y | X = X / Y |
| %= | X %= Y | X = X % Y |

Aritmética

Operadores aritméticos tomam valores numéricos (sejam literais ou variáveis) como seus operandos e retornam um único valor númerico. Os operadores aritméticos padrão são os de soma (+), subtração (-), multiplicação (\*) e divisão (/). Estes operadores trabalham da mesma forma como na maioria das linguagens de programação quando utilizados com números de ponto flutuante (em particular, repare que divisão por zero produz um NaN (en-US)).

|  |  |
| --- | --- |
| Operador | Descrição |
| + | Adição |
| - | Subtração |
| \* | Multiplicação |
| \*\* | Exponencial |
| / | Divisão |
| % | Módulo |
| ++ | Incrementar |
| -- | Decrementar |

Comparação

Um operador de comparação compara seus operandos e retorna um valor lógico baseado em se a comparação é verdadeira. Os operandos podem ser numéricos, strings, lógicos ou objetos. Strings são comparadas com base em ordenação lexográfica utilizando valores Unicode. Na maioria dos casos, se dois operandos não são do mesmo tipo, o JavaScript tenta convertê-los para um tipo apropriado. Isto geralmente resulta na realização de uma comparação numérica. As únicas exceções a esta regra são os operadores === e o !==, que realizam comparações de igualdade e desigualdade "estritas". Estes operadores não tentam converter os operandos em tipos compatíveis antes de verificar a igualdade.

|  |  |
| --- | --- |
| Operador | Descrição |
| == | Igual a |
| === | Mesmo valor e tipo |
| != | Diferente |
| !== | Valor e tipo diferente |
| > | Maior que |
| < | Menor que |
| >= | Maior ou igual |
| <= | Menor ou igual |

Lógica

Operadores lógicos são utilizados tipicamente com valores booleanos (lógicos); neste caso, retornam um valor booleano. Entretanto, os operadores && e || na verdade retornam o valor de um dos operandos especificados, de forma que se esses operadores forem utilizados com valores não-booleanos, eles possam retornar um valor não-booleano.

|  |  |
| --- | --- |
| Operador | Descrição |
| && | “e” lógico |
| || | “ou” lógico |
| ! | “não” lógico |

Condicional

Uma forma mais simplificada do IF e Else

If(condicao) {

valor1;

Else

valor2;

}

O operador condicional é o único operador JavaScript que utiliza três operandos. O operador pode ter um de dois valores baseados em uma condição. A sintaxe é:

*condicao* ? *valor1* : *valor2*

Se condicao for verdadeira, o operador terá o valor de valor1. Caso contrário, terá o valor de valor2. Você pode utilizar o operador condicional em qualquer lugar onde utilizaria um operador padrão.

Por exemplo,

var status = (idade >= 18) ? "adulto" : "menor de idade";

Esta declaração atribui o valor "adulto" à variável status caso idade seja dezoito ou mais. Caso contrário, atribui o valor "menor de idade".

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Expressions\_and\_operators