

TUTORIAIS JAVASCRIPT

Como utilizar os operadores do JavaScript

Copyright 2013 – Todos os Direitos Reservados
Jorge Eider F. da Silva

Proibida a reprodução deste documento no todo ou em parte por quaisquer meios, seja digital, eletrônico ou impresso sem a expressa autorização do autor por escrito. Os infratores serão punidos de acordo com a Lei.

Como utilizar os operadores do JavaScript

Introdução

Nesse tutorial veremos quais são e como funcionam os vários operadores do JavaScript e como utilizá-los corretamente em nossas operações.

Operadores Aritméticos

Como o próprio nome indica, eles são utilizados para executar operações aritméticas. São os seguintes os operadores aritméticos:

- Adição (+)
- Subtração (-)
- Multiplicação (*)
- Divisão (/)
- Módulo (%)
- Incremento (++)
- Decremento (--)

Esses operadores podem ser combinados para formar expressões. Eles trabalham da mesma forma que outras linguagens de programação conhecidas. Veja a seguir como utilizar corretamente esses operadores e como eles funcionam em JavaScript.

Considerando que y assuma o valor 5, veremos como fica nossa tabela:

Operador	Descrição	Exemplo	Resultado de x	Resultado de y
+	Adição	$x=y+2$	7	5
-	Subtração	$x=y-2$	3	5
*	Multiplicação	$x=y*2$	10	5
/	Divisão	$x=y/2$	2.5	5
%	Módulo	$x=y\%2$	1	5
++	Incremento	$x=++y$	6	6
		$x=y++$	5	6
--	Decremento	$x=--y$	4	4
		$x=y--$	5	4

O Operador + (Adição)

Para utilizar a adição é usado o mesmo operador algébrico (+). Veja como utilizar esse operador.

Uso:

```
soma = var1 + var2
```

Onde:

var1 e **var2** são dois valores quaisquer, ou uma expressão que resulte em um valor válido.

Resultado:

Atribui o resultado da adição de **var1** e **var2** à variável **soma**.

Exemplo:

```
soma = 100 + 20  
soma = 120
```

O Operador - (Subtração)

Para utilizar a subtração é usado o mesmo operador algébrico (-). Veja como utilizar esse operador.

Uso:

```
subtrai = var1 - var2
```

Onde:

var1 e **var2** são dois valores quaisquer.

Resultado:

Atribui o resultado da subtração entre **var1** e **var2** à variável **subtrai**.

Exemplo:

```
subtrai = 100 - 20  
subtrai = 80
```

O Operador * (Multiplicação)

Em JavaScript, é utilizado o asterisco (*) como sinal de multiplicação entre os operandos, e não o **x**, como na álgebra.

Veja como utilizar esse operador.

Uso:

```
produto = var1 * var2
```

Onde:

var1 e **var2** são dois valores quaisquer.

Resultado:

Atribui o resultado da multiplicação de **var1** e **var2** à variável **produto**.

Exemplo:

```
produto = 10 * 20  
produto = 200
```

O Operador / (Divisão)

Em JavaScript, é utilizado a barra (/) como sinal de divisão entre os operandos, e não o sinal de dois pontos (:) como na álgebra. Uma divisão inteira produz um quociente inteiro.

Qualquer parte fracionária envolvida nessa divisão é simplesmente descartada. Não ocorrerá nenhum arredondamento.

Veja como utilizar esse operador.

Uso:

divisao = *var1* / *var2*

Onde:

var1 e **var2** são dois valores quaisquer.

Resultado:

Atribui o resultado da divisão de **var1** e **var2** à variável **divisao**.

Exemplo:

divisao = 100 / 20
divisao = 5

O Operador % (Módulo)

O operador módulo é utilizado para extrair o resto de uma divisão. Para isso é usado o sinal de porcentagem (%).

Veja como utilizar esse operador.

Uso:

resto = *var1* % *var2*

Onde:

var1 e **var2** são dois valores quaisquer.

Resultado:

Atribui o resto da divisão entre **var1** e **var2** à variável **resto**.

Exemplo:

resto = 14 % 4 (14 dividido por 4 tem como quociente 3 e resto 2)
resto = 2

Vejamos um exemplo prático:

Exemplo 1

Nesse exemplo vamos criar duas variáveis e atribuir valores para elas, em seguida utilizaremos algumas operações com elas para você saber como funcionam esses operadores. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js7-exemplo1.html**.

```
<!DOCTYPE html>
<html>
  Tutorial JavaScript7 - Exemplo 1 - Como utilizar os operadores do JavaScript
```

```

<head>
  <title>Tutorial JavaScript7 - Exemplo 1 - Como utilizar os operadores do JavaScript</title><p>
</head>
<body>

<script>
  var valor1, valor2; // São declaradas as variáveis valor1 e valor2.
  valor1 = 15;        // valor1 assume o valor de 15.
  valor2 = 7;         // valor2 assume o valor de 7.
  document.write("valor1 é igual a: " + valor1 + "<br>");
  document.write("valor2 é igual a: " + valor2 + "<br>");
  document.write("valor1 + valor2 é igual a: " + (valor1 + valor2) + "<br>");
  document.write("valor1 - valor2 é igual a: " + (valor1 - valor2) + "<br>");
  document.write("valor1 multiplicado por valor2 é igual a: " + (valor1 * valor2) + "<br>");
  document.write("valor1 dividido por valor2 é igual a: " + (valor1 / valor2) + "<br>");
  document.write("O resto de valor1 com o valor2 é igual a: " + (valor1 % valor2));
</script>

</body>
</html>

```

2. Esse código após executado no browser exibirá o seguinte:

```

Tutorial JavaScript7 - Exemplo 1 - Como utilizar os operadores do JavaScript

valor1 é igual a: 15
valor2 é igual a: 7
valor1 + valor2 é igual a: 22
valor1 - valor2 é igual a: 8
valor1 multiplicado por valor2 é igual a: 105
valor1 dividido por valor2 é igual a: 2.142857142857143
O resto de valor1 com o valor2 é igual a: 1

```

O Operador ++ (Incremento)

No JavaScript esse operador é muito similar ao estilo e funcionalidade de outras linguagens como, por exemplo, o C, o C++ e o Java. Veja como utilizar esse operador.

Uso:

incrementa = *var1*++ (pós-incremento)
incrementa = ++*var1* (pré-incremento)

Onde:

var1 é um valor numérico qualquer.

Resultado:

Incrementa ou adiciona **var1** em uma unidade. Se esse operador for usado após o operando (por exemplo: **var1++**), será retornado o valor antes de incrementar. Se usado antes (por exemplo: **++var1**), então retornará o valor após incrementar.

Exemplo:

```
var1 = 10
var1++      - var1 é igual a 10
++var1     - var1 é igual a 11
```

Esse operador é utilizado em substituição à expressão: `var1 = var1 + 1`.

O Operador -- (Decremento)

Esse operador funciona da mesma forma que o operador anterior. Veja como utilizar esse operador.

Uso:

```
decrementa = var1-- (pós-decremento)
decrementa = --var1 (pré-decremento)
```

Onde:

var1 é um valor numérico qualquer.

Resultado:

Decrementa ou subtrai **var1** em uma unidade. Se esse operador for usado após o operando (por exemplo: **var1--**), será retornado o valor antes de decrementar. Se usado antes (por exemplo: **--var1**), então retornará o valor após decrementar.

Exemplo:

```
var1 = 10
var1--      - var1 é igual a 10
--var1     - var1 é igual a 9
```

Esse operador é utilizado em substituição à expressão: `var1 = var1 - 1`.

Vejamos um exemplo prático:

Exemplo 2

Nesse exemplo vamos criar duas variáveis e atribuir valores para elas, em seguida utilizaremos algumas operações com elas para você entender como funcionam esses operadores. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js7-exemplo2.html**.

```
<!DOCTYPE html>
<html>
  Tutorial JavaScript7 - Exemplo 2 - Como utilizar os operadores do JavaScript
<head>
  <title>Tutorial JavaScript7 - Exemplo 2 - Como utilizar os operadores do JavaS-
  cript</title><p>
</head>
<body>

<script>
  var valor1, valor2; // São declaradas as variáveis valor1 e valor2.
```

```

valor1 = 5;           // valor1 assume o valor de 5.
valor2 = 7;           // valor2 assume o valor de 7.
document.write("valor1 é igual a: " + valor1 + "<br>");
document.write("valor2 é igual a: " + valor2 + "<br>");
document.write("valor1++ é igual a: " + valor1++ + "<br>");
document.write("valor2++ é igual a: " + valor2++ + "<br>");
document.write("++valor1 é igual a: " + ++valor1 + "<br>");
document.write("++valor2 é igual a: " + ++valor2 + "<br>");
document.write("valor1-- é igual a: " + valor1-- + "<br>");
document.write("valor2-- é igual a: " + valor2-- + "<br>");
document.write("--valor1 é igual a: " + --valor1 + "<br>");
document.write("--valor2 é igual a: " + --valor2 + "<br>");
</script>

</body>
</html>

```

2. Esse código após executado no browser exibirá o seguinte:

```

Tutorial JavaScript7 - Exemplo 2 - Como utilizar os operadores do JavaScript

valor1 é igual a: 5
valor2 é igual a: 7
valor1++ é igual a: 5
valor2++ é igual a: 7
++valor1 é igual a: 7
++valor2 é igual a: 9
valor1-- é igual a: 7
valor2-- é igual a: 9
--valor1 é igual a: 5
--valor2 é igual a: 7

```

Operadores de Atribuição

Esses operadores são utilizados para atribuir valores às variáveis em JavaScript. Esses operadores permitem a alocação de valores em memória. Para se atribuir um valor a uma variável, utiliza-se o sinal de igualdade (=).

Sintaxe:

```
var1 = valor;
```

Onde:

- **var1** – é a variável que irá receber um valor ou expressão.
- **valor** – qualquer valor ou expressão válida. É o valor a ser atribuído à variável.

Exemplos:

- `x = 100;` (expressão numérica)
- `y = 5.50;` (expressão numérica)
- `nome = "Joana";` (expressão literal ou string)
- `resultado = x - y;` (expressão numérica atribuída a uma variável)

São os seguintes os operadores de atribuição:

Considerando que **x** assumo o valor de 10 e **y** assumo o valor 5, veremos como fica nossa tabela:

Operador	Exemplo	O mesmo que	Resultado
=	x=y		x=5
+=	x+=y	x=x+y	x=15
-=	x-=y	x=x-y	x=5
=	x=y	x=x*y	x=50
/=	x/=y	x=x/y	x=2
%=	x%=y	x=x%y	x=0

Vejamos um exemplo prático:

Exemplo 3

Nesse exemplo vamos criar algumas variáveis e atribuir alguns valores a elas e em seguida calcular o produto das duas e armazenar na variável **produto**. Para isso:

3. Digite o código abaixo no seu editor de texto e salve-o como: **js7-exemplo3.html**.

```
<!DOCTYPE html>
<html>
Tutorial JavaScript7 - Exemplo 3 - Como utilizar os operadores do JavaScript
<head>
  <title>Tutorial JavaScript7 - Exemplo 3 - Como utilizar os operadores do JavaS-
cript</title><p>
</head>
<body>
<script>
  // São declaradas as variáveis valor1, valor2 e produto.
  var valor1, valor2, produto;
  valor1 = 5; // a variável valor1 assume o valor de 5.
  valor2 = 10; // a variável valor2 assume o valor de 10.
  produto = valor1 * valor2; // a variável produto assume o valor de 50.
  document.write("Valor1 --> " + valor1 + "<br>");
  document.write("Valor2 --> " + valor2 + "<br>");
  document.write("Produto --> " + produto);
</script>

</body>
</html>
```

4. Esse código após executado no browser exibirá o seguinte:

```
Tutorial JavaScript7 - Exemplo 3 - Como utilizar os operadores do JavaScript
Valor1 --> 5
Valor2 --> 10
Produto --> 50
```

Precedência dos Operadores Aritméticos

Os operadores aritméticos em JavaScript obedecem às mesmas regras de precedência utilizadas pela álgebra. Veja como devem ser utilizados esses operadores de acordo com a prioridade de cada um.

Operadores	Operação realizada	Precedência
()	Parênteses	Os primeiros que deverão ser avaliados. Se existirem vários parênteses aninhados, o par mais interno deverá ser avaliado primeiro. Se não estiverem aninhados, deverão ser avaliados da esquerda para a direita.
*, / ou %	Multiplicação, divisão ou módulo	Esses operadores são avaliados em segundo plano, da mesma forma que o anterior, da esquerda para a direita.
+ ou -	Adição ou subtração	São os últimos a serem avaliados.

Em qualquer expressão algébrica os parênteses possuem maior precedência, ou seja, os operadores contidos dentro deles deverão ser executados primeiro. Os parênteses servem para quebrar a ordem de precedência dos operadores e dar outro destino ao resultado de uma expressão aritmética. Veja no exemplo abaixo, os vários resultados obtidos de uma mesma expressão, utilizando-se os parênteses em posições diferentes.

Exemplo

$$\begin{aligned}
 7 + (5 * 4 - 8) / 4 &= 10 \\
 (7 + 5) * 4 - 8 / 4 &= 46 \\
 7 + 5 * (4 - 8 / 4) &= 17
 \end{aligned}$$

Em seguida virão os operadores de multiplicação, divisão e de módulo num mesmo nível de precedência. Esses operadores deverão ser executados a partir da esquerda numa expressão. Se nenhum parêntese for utilizado, essa sequência deverá ser obedecida. Veja alguns exemplos abaixo.

Exemplo 1

$$\begin{aligned}
 5 * 2 + 4 * 1 - 6 / 2 \\
 10 + 4 - 3 = 11
 \end{aligned}$$

Exemplo 2

$$\begin{aligned}
 28 - 4 + 3 * 2 * 3 - 8 / 2 \\
 28 - 4 + 18 - 4 \\
 24 + 14 = 38
 \end{aligned}$$

Exemplo 3

$$\begin{aligned}
 18 - 4 + 3 \% 2 * 3 - 6 / 2 \\
 18 - 4 + 3 - 3 \\
 14 + 0 = 14
 \end{aligned}$$

Vejamos um exemplo prático:

Exemplo 4

Nesse exemplo vamos criar algumas variáveis e atribuir algumas expressões a elas para demonstrar a precedência dos operadores aritméticos. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js7-exemplo4.html**.

```

<!DOCTYPE html>
<html>
Tutorial JavaScript 7 - Exemplo 4 - Como utilizar os operadores do JavaScript

```

```

<head>
  <title>Tutorial JavaScript7 - Exemplo 4 - Como utilizar os operadores do JavaScript</title><p>
</head>
<body>

<script>
  // São declaradas as variáveis expressao1, expressao2 e expressao3.
  var expressao1, expressao2, expressao3;
  expressao1 = 5*2+4*1-6/2;
  expressao2 = 28-4+3*2*3-8/2;
  expressao3 = 18-4+3%2*3-6/2;
  document.write("A expressao1 é igual a: " + expressao1 + "<br>");
  document.write("A expressao2 é igual a: " + expressao2 + "<br>");
  document.write("A expressao3 é igual a: " + expressao3);
</script>

</body>
</html>

```

- Esse código após executado no browser exibirá o seguinte:

Tutorial JavaScript7 - Exemplo 4 - Como utilizar os operadores do JavaScript

A expressao1 é igual a: 11
A expressao2 é igual a: 38
A expressao3 é igual a: 14

Operadores Short-Cut Aritméticos

Os operadores Short-Cut são também conhecidos como atalhos, ou seja, alternativas para efetuar operações mais rapidamente. A tabela abaixo exemplifica melhor a utilização desses operadores.

Operadores Short-Cut Aritméticos		
Normal	Short-Cut	Descrição
$x = x + y$	$x += y$	Adiciona 2 números e armazena o resultado ao primeiro.
$x = x - y$	$x -= y$	Subtrai 2 números e armazena o resultado ao primeiro.
$x = x * y$	$x *= y$	Multiplica 2 números e armazena o resultado ao primeiro.
$x = x / y$	$x /= y$	Divide 2 números e armazena o resultado ao primeiro.
$x = x \% y$	$x \% = y$	Calcula os módulos de 2 números e armazena o resultado ao primeiro.

Vejamos um exemplo prático utilizando esses operadores:

Exemplo 5

Nesse exemplo vamos utilizar os operadores aritméticos para fazer algumas operações utilizando os operadores short-cut. Para isso:

- Digite o código abaixo no seu editor de texto e salve-o como: **js7-exemplo5.html**.

```

<!DOCTYPE html>
<html>
  Tutorial JavaScript7 - Exemplo 5 - Como utilizar os operadores do JavaScript
</head>

```

```

<title>Tutorial JavaScript7 - Exemplo 5 - Como utilizar os operadores do JavaS-
cript</title><p>
</head>
<body>

<script>
    var x, y, z;    // São declaradas as variáveis x, y e z.
    x = 2;
    y =3;
    z = 5;
    document.write("A variável x é igual a: " + x + "<br>");
    document.write("A variável y é igual a: " + y + "<br>");
    document.write("A variável z é igual a: " + z + "<br>");
    document.write("A expressão: x += x (x =x + x), é igual a: " + (x += x) +
"<br>");
    document.write("A expressão: y += x (y =y + x), é igual a: " + (y += x) +
"<br>");
    document.write("A expressão: z += y (z =z + y), é igual a: " + (z += y) +
"<br>");

    document.write("A expressão: x /= x (x =x / x), é igual a: " + (x /= x) +
"<br>");
    document.write("A expressão: y -= x (y =y - x), é igual a: " + (y -= x) +
"<br>");
    document.write("A expressão: z *= y (z =z * y), é igual a: " + (z *= y) +
"<br>");

    document.write("A expressão: x *= 5 (x =x * 5), é igual a: " + (x *= 5) +
"<br>");
    document.write("A expressão: y %= x (y =y % x), é igual a: " + (y %= x) +
"<br>");
    document.write("A expressão: z %= y (z =z % y), é igual a: " + (z %= y) );
</script>

</body>
</html>

```

2. Esse código após executado no browser exibirá o seguinte:

```

Tutorial JavaScript7 - Exemplo 5 - Como utilizar os operadores do JavaScript

A variável x é igual a: 2
A variável y é igual a: 3
A variável z é igual a: 5
A expressão: x += x (x =x + x), é igual a: 4
A expressão: y += x (y =y + x), é igual a: 7
A expressão: z += y (z =z + y), é igual a: 12
A expressão: x /= x (x =x / x), é igual a: 1
A expressão: y -= x (y =y - x), é igual a: 6
A expressão: z *= y (z =z * y), é igual a: 72
A expressão: x *= 5 (x =x * 5), é igual a: 5
A expressão: y %= x (y =y % x), é igual a: 1
A expressão: z %= y (z =z % y), é igual a: 0

```

É bom observar que neste exemplo, as variáveis x, y e z vão se modificando à medida que os cálculos vão sendo realizados. Portanto, muita atenção nos valores já assumidos por essas variáveis.

O Operador + usado em Strings

Esse operador pode ser usado também para concatenar variáveis strings com textos. Veja um exemplo:

Para adicionar duas ou mais variáveis strings usa-se o operador +:

```
texto1 = "JavaScript não";
texto2 = "é o mesmo";
texto3 = "Java";
texto = texto1 + texto2 + texto3;
```

O resultado de **texto** seria:

```
JavaScript nãoé o mesmoque Java
```

Para evitar que as palavras saiam juntas é necessário adicionar espaços no início ou no final de algumas delas. Veja como ficaria o mesmo exemplo:

```
texto1 = "JavaScript não ";
texto2 = "é o mesmo ";
texto3 = "Java";
texto = texto1 + texto2 + texto3;
```

O resultado de **texto** agora será:

```
JavaScript não é o mesmo que Java
```

Ou então inserir espaços entre as expressões sem alterar os valores das variáveis, conforme mostra o exemplo a seguir:

```
texto1 = "JavaScript não";
texto2 = "é o mesmo";
texto3 = "Java";
texto = texto1 + " " + texto2 + " " + texto3;
```

Adicionando Strings e Números

Se você usar o sinal de adição (+) para adicionar dois números, o resultado será a soma desses números. Mas, se você adicionar um número a uma string, o resultado será uma string. Veja um exemplo:

```
x = 5 + 5;
y = "5" + 5;
z = "Olá" + 5;
```

O resultado desse código seria:

```
10
55
Olá5
```

Operadores de Comparação

Esses operadores são utilizados em declarações lógicas para testar se uma situação é verdadeira ou falsa. São os seguintes os operadores de comparação:

Considerando que **x** assumo o valor de 5, vejamos como fica nossa tabela:

Operador	Descrição	Comparação	Retorno
==	É igual a	x == 8	false
		x == 5	true
===	É exatamente igual a (valor e tipo)	x === "5"	false
		x === 5	true
!=	Não é igual (diferente)	x != 8	true
!==	Não é igual (diferente), nem valor, nem tipo	x !== "5"	true
		x !== 5	false
>	Maior do que	x > 8	false
<	Menor do que	x < 8	true
>=	Maior do que ou igual a	x >= 8	false
<=	Menor do que ou igual a	x <= 8	true

O Operador == (Igual)

O operador de igualdade é utilizado para igualar duas expressões. Não deve ser confundido com o operador de atribuição (=). O operador de igualdade (==) é lido como "é igual a" ou ainda, "iguais iguais". Não utilize espaços em branco entre os dois sinais de igual (por exemplo: = =). Isso provocará erro durante a execução. Veja como utilizar esse operador:

Uso:

```
var1 == var2
```

Onde:

var1 e **var2** são dois valores ou expressões quaisquer.

Resultado:

Os valores de ambos os operandos são avaliados, e retorna verdadeiro somente se ambos os operandos forem iguais.

Exemplo:

```
se 100 == 100; retorna verdadeiro  
se 200 == 100; retorna falso
```

O Operador != (Não igual ou diferente)

O operador de desigualdade ou diferente é utilizado para diferenciar duas expressões. Ele deve ser lido como "não é igual a" ou "diferente de". Utilizar espaços em branco entre os sinais (por exemplo: !=) ou invertê-los (por exemplo: =!), constitui erro de sintaxe. Veja abaixo como utilizar esse operador em seus aplicativos.

Uso:

```
var1 != var2
```

Onde:

var1 e **var2** são dois valores ou expressões quaisquer.

Resultado:

Os valores de ambos os operandos são avaliados, e retorna verdadeiro somente se ambos os operandos não forem iguais, ou seja, é o inverso do operador de igualdade (==).

Exemplo:

se 100 != 100; retorna falso
se 200 != 100; retorna verdadeiro

Vejamos um exemplo prático utilizando esses operadores:

Exemplo 6

Nesse exemplo vamos utilizar algumas variáveis e fazer algumas operações com os operadores de comparação estudados. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js7-exemplo6.html**.

```
<!DOCTYPE html>
<html>
Tutorial JavaScript7 - Exemplo 6 - Como utilizar os operadores do JavaScript
<head>
  <title>Tutorial JavaScript7 - Exemplo 6 - Como utilizar os operadores do JavaScript</title><p>
</head>
<body>

<script>
  // São declaradas as variáveis valor1 e valor2.
  var valor1, valor2;
  valor1 = 5;
  valor2 = 12;
  document.write("valor1 é igual a --> " + valor1 + "<br>");
  document.write("valor2 é igual a --> " + valor2 + "<br>");
  if ( valor1 + 7 == valor2 ) // Avalia se valor1 mais 7 é igual a valor2.
    document.write("valor1 + 7 é igual a valor2" + "<br>");

  if ( valor1 == valor2 ) // Avalia se valor1 é igual a valor2.
    document.write("valor1 é igual a valor2" + "<br>");

  if ( valor1 != valor2 ) // Avalia se valor1 é diferente de valor2.
    document.write("valor1 é diferente de valor2" + "<br>");
</script>

</body>
</html>
```

2. Esse código após executado no browser exibirá o seguinte:

Tutorial JavaScript7 - Exemplo 6 - Como utilizar os operadores do JavaScript

```
valor1 é igual a --> 5  
valor2 é igual a --> 12  
valor1 + 7 é igual a valor2  
valor1 é diferente de valor2
```

Observe que utilizamos o comando **if** para que pudéssemos explicar a utilização dos operadores de igualdade. O comando **if**, que é utilizado para tomada de decisões, será explicado mais detalhadamente no tutorial apropriado.

O Operador > (Maior do que)

Esse operador é utilizado para avaliar dois valores ou expressões. Uma decisão deverá ser tomada a partir do resultado obtido. Veja como utilizar esse operador:

Uso:

var1 > var2

Onde:

var1 e **var2** são dois valores ou expressões quaisquer.

Resultado:

Os valores de ambos os operandos são avaliados, e retorna verdadeiro somente se o operando da esquerda for maior que o operando da direita.

Exemplo:

Se $100 > 10$; retorna verdadeiro
Se $20 > 100$; retorna falso

O Operador >= (Maior do que ou igual)

Semelhante ao operador anterior, esse operador é utilizado para avaliar dois valores ou expressões, mas utilizando duas situações ('*maior do que*' ou '*igual a*'). Uma decisão deverá ser tomada a partir do resultado obtido. Utilizar espaços em branco entre os sinais (por exemplo: $> =$) ou invertê-los (por exemplo: $=>$), constitui erro de sintaxe. Veja abaixo o uso correto desse operador.

Uso:

var1 >= var2

Onde:

var1 e **var2** são dois valores ou expressões quaisquer.

Resultado:

Os valores de ambos os operandos são avaliados, e retorna verdadeiro somente se o operando da esquerda for maior ou igual ao operando da direita.

Exemplo:

Se $100 \geq 10$; retorna verdadeiro
Se $20 \geq 100$; retorna falso

O Operador < (Menor do que)

Esse operador é utilizado para avaliar dois valores ou expressões. Uma decisão deverá ser tomada a partir do resultado obtido. Veja como utilizar esse operador nos seus programas.

Uso:

var1 < var2

Onde:

var1 e **var2** são dois valores ou expressões quaisquer.

Resultado:

Os valores de ambos os operandos são avaliados, e retorna verdadeiro somente se o operando da esquerda for menor que o operando da direita.

Exemplo:

Se $10 < 100$; retorna verdadeiro
Se $200 < 10$; retorna falso

O Operador <= (Menor do que ou igual)

Semelhante ao operador anterior, esse operador é utilizado para avaliar dois valores ou expressões, mas utilizando duas situações ('*menor do que*' ou '*igual a*'). Uma decisão deverá ser tomada a partir do resultado obtido. Utilizar espaços em branco entre os sinais (por exemplo: $< =$) ou invertê-los (por exemplo: $= <$), constitui erro de sintaxe. Veja abaixo o uso correto desse operador.

Uso:

var1 <= var2

Onde:

var1 e **var2** são dois valores ou expressões quaisquer.

Resultado:

Os valores de ambos os operandos são avaliados, e retorna verdadeiro somente se o operando da esquerda for menor ou igual ao operando da direita.

Exemplo:

Se $10 \leq 100$; retorna verdadeiro
Se $200 \leq 10$; retorna falso

Vejamos um exemplo prático utilizando esses operadores:

Exemplo 7

Nesse exemplo vamos utilizar algumas variáveis e fazer algumas operações com os operadores de comparação estudados. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js7-exemplo7.html**.

```
<!DOCTYPE html>
<html>
Tutorial JavaScript7 - Exemplo 7 - Como utilizar os operadores do JavaScript
<head>
  <title>Tutorial JavaScript7 - Exemplo 7 - Como utilizar os operadores do JavaScript</title><p>
</head>
<body>

<script>
  // São declaradas as variáveis valor1 e valor2.
  var valor1, valor2;
  valor1 = 25;
  valor2 = 12;
  document.write("valor1 é --> " + valor1 + "<br>");
  document.write("valor2 é --> " + valor2 + "<br>");
  if (valor1 > valor2) {
    document.write("valor1 é maior que valor2" + "<br>");
  } else {
    document.write("valor1 é menor que valor2" + "<br>");
  }
  if (valor1 >= valor2) {
    document.write("valor1 é maior ou igual que valor2" + "<br>");
  } else {
    document.write("valor1 é menor que valor2" + "<br>");
  }
  if (valor2 < valor1) {
    document.write("valor2 é menor que valor1" + "<br>");
  } else {
    document.write("valor1 é maior que valor2" + "<br>");
  }
  if (valor1 <= valor2) {
    document.write("valor1 é menor ou igual que valor2" + "<br>");
  } else {
    document.write("valor1 é maior ou igual que valor2" + "<br>");
  }
</script>

</body>
</html>
```

2. Esse código após executado no browser exibirá o seguinte:

```
Tutorial JavaScript7 - Exemplo 7 - Como utilizar os operadores do JavaScript
valor1 é --> 25
valor2 é --> 12
valor1 é maior que valor2
valor1 é maior ou igual que valor2
valor2 é menor que valor1
valor1 é maior ou igual que valor2
```

Da mesma forma que o exemplo anterior, foi necessária a utilização do comando **if** para que pudéssemos explicar o uso dos operadores de comparação. Nesse caso, além do comando **if**, que é utilizado para tomada de decisões, utilizamos um outro comando que é

vinculado ao comando **if**, o **else**, que será usado caso a condição do comando **if** não seja satisfeita.

Note que para cada bloco de comando **if/else**, apenas uma expressão é verdadeira. Sendo assim, apenas uma será impressa, de acordo com o resultado obtido.

Dica 1: Procure recuar cada bloco de instrução nos seus programas. Isso facilita a leitura e melhora a legibilidade deles.

Dica 2: Não confunda em hipótese alguma, o operador de igualdade (**==**) com o operador de atribuição (**=**). Eles são totalmente diferentes.

Dica 3: Procure colocar apenas uma instrução por linha, e procure documentá-la.

Operadores Lógicos ou booleanos

Os operadores lógicos avaliam os valores Booleanos (lógicos) dos operandos de duas expressões, e retornam um valor verdadeiro (**true**) ou falso (**false**).

Sintaxe:

expr1 operador lógico **expr2**

Onde:

expr1 e **expr2** são duas expressões quaisquer (numéricas ou não).

Os operadores lógicos são os seguintes:

Considerando que **x** seja igual a **6** e **y** seja igual a **3**, vejamos como fica nossa tabela:

Operador	Descrição	Exemplo
&&	and	(x < 10 && y > 1) é verdadeira
	or	(x == 5 y == 5) é falsa
!	not	!(x == y) é verdadeira

Operador Lógico AND (&&)

Retorna um resultado verdadeiro somente se ambas as expressões forem verdadeiras.

Sintaxe:

expr1 && **expr2**

Onde:

expr1 e **expr2** são duas expressões quaisquer (numéricas ou não).

Exemplo:

Se (4 > 3) && (10 < 50); retorna verdadeiro, pois ambas as expressões são verdadeiras.

Se (2 >= 4) && (10 < 50); retorna falso, pois a primeira expressão é falsa.

Operador Lógico OU (||)

Retorna um resultado verdadeiro se uma das expressões for verdadeira.

Sintaxe:

expr1 || expr2

Onde:

expr1 e **expr2** são duas expressões quaisquer (numéricas ou não).

Exemplo:

Se (4 > 3) || (10 > 50); retorna verdadeiro, pois a primeira expressão é verdadeira.
Se (2 >= 4) || (10 > 50); retorna falso, pois nenhuma expressão é verdadeira.

Operador Lógico NOT (!)

Esse operador é o oposto do operador && (AND). Ele retorna um resultado verdadeiro se a expressão for falsa, e vice-versa.

Sintaxe:

!expr

Onde:

expr é uma expressão qualquer (numérica ou não).

Exemplo:

Se !(10 > 50); retorna verdadeiro, pois a expressão é falsa.
Se !(8 > 4); retorna falso, pois a expressão é verdadeira.

Vejamos um exemplo prático utilizando esses operadores:

Exemplo 8

Nesse exemplo vamos utilizar algumas variáveis e fazer algumas operações com os operadores de comparação estudados. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js7-exemplo8.html**.

```
<!DOCTYPE html>
<html>
Tutorial JavaScript7 - Exemplo 8 - Como utilizar os operadores do JavaScript
<head>
  <title>Tutorial JavaScript7 - Exemplo 8 - Como utilizar os operadores do JavaS-
cript</title><p>
</head>
<body>

<script>
  // São declaradas as variáveis valor1, valor2 e resultado.
  var valor1, valor2, resultado;
  valor1 = 5;
  valor2 = 10;
  document.write("valor1 é igual a: " + valor1 + "<br>");
  document.write("valor2 é igual a: " + valor2 + "<p>");
```

```

        if ((valor1 > 3) && (valor2 < 15)) {
            resultado = "VERDADEIRA, pois ambas as expressões são verdadei-
ras.";
            document.write("A expressão (valor1 > 3) && (valor2 < 15) é " + re-
sultado + "<br>");
        } else {
            resultado = "FALSA, pois uma das expressões é falsa.";
            document.write("A expressão (valor1 > 3) && (valor2 < 15) é " + re-
sultado + "<br>");
        }

        if ((valor1 > 2) || (valor2 < 5)) {
            resultado = "VERDADEIRA, pois a primeira expressão é verdadeira.";
            document.write("A expressão (valor1 >2) || (valor2 < 5) é: " + resul-
tado + "<br>");
        } else {
            resultado = "FALSA, pois nenhuma das expressões é verdadeira.";
            document.write("A expressão (valor1 >2) || (valor2 < 5) é: " + resul-
tado + "<br>");
        }

        if (!(valor1 < 2)) {
            resultado = "VERDADEIRA, pois a expressão é falsa.";
            document.write("A expressão (!(valor1 < 2) é: " + resultado +
"<br>");
        } else {
            resultado = "FALSA, pois a expressão é verdadeira.";
            document.write("A expressão (!(valor1 < 2) é: " + resultado +
"<br>");
        }
    }
</script>
</body>
</html>

```

2. Esse código após executado no browser exibirá o seguinte:

Tutorial JavaScript7 - Exemplo 8 - Como utilizar os operadores do JavaScript

valor1 é igual a: 5
valor2 é igual a: 10

A expressão (valor1 > 3) && (valor2 < 15) é VERDADEIRA, pois ambas as expressões são verdadeiras.
A expressão (valor1 >2) || (valor2 < 5) é: VERDADEIRA, pois a primeira expressão é verdadeira.
A expressão (!(valor1 < 2) é: VERDADEIRA, pois a expressão é falsa.

Neste exemplo também foi necessária a utilização do comando **if/else** para que pudéssemos explicar o uso dos operadores lógicos.

Operador Condicional

Esse operador atribui um valor a uma variável baseada em alguma condição. Este é o único operador em JavaScript que possui três operandos. Ele é geralmente usado como um short-cut (atalho) para a declaração **"if"**. Veja a seguir como utilizá-lo. Sua sintaxe é a seguinte:

```
nome_da_variável = (condição) ? valor1 : valor2
```

Sintaxe:

condição ? expr1 : expr2

Onde:

condição – é qualquer expressão que possa ser avaliada como **falsa** ou **verdadeira**.

expr1 e **expr2** – são expressões de quaisquer valores (numéricas ou strings).

Descrição:

Se **condição** for verdadeira, a expressão terá como resultado o valor de **expr1**, caso contrário, retorna o valor de **expr2**.

Exemplos:

```
10 > 5 ? "10 é maior que 5" : "10 é menor que 5"  
A == B ? "A é igual a B" : "A é diferente de B"
```

Vejamos um exemplo prático utilizando esses operadores:

Exemplo 9

Nesse exemplo vamos utilizar algumas variáveis e fazer uma operação com o operador estudado. A partir desse exemplo você poderá criar outras situações com esse operador. Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js7-exemplo9.html**.

```
<!DOCTYPE html>  
<html>  
Tutorial JavaScript7 - Exemplo 9 - Como utilizar os operadores do JavaScript  
<head>  
  <title>Tutorial JavaScript7 - Exemplo 9 - Como utilizar os operadores do JavaScript</title> <p>  
</head>  
<body>  
  
<script>  
  // São declaradas as variáveis valor1 e valor2.  
  var valor1, valor2;  
  valor1 = 5;           // É atribuído o valor 5 à variável valor1.  
  valor2 = 10;         // É atribuído o valor 10 à variável 10.  
  document.write("valor1 é igual a --> " + valor1 + "<br>");  
  document.write("valor2 é igual a --> " + valor2 + "<br>");  
  document.write( valor1 > valor2 ? "valor1 é maior que valor2" : "valor1 é  
menor que valor2" + "<br>");  
</script>  
  
</body>  
</html>
```

2. Esse código após executado no browser exibirá o seguinte:

Tutorial JavaScript7 - Exemplo 9 - Como utilizar os operadores do JavaScript

valor1 é igual a --> 5
valor2 é igual a --> 10
valor1 é menor que valor2

JORGE ELDER

Exercícios de fixação

- Qual o operador utilizado para extrair o resto de uma divisão?
 - divisão
 - concatenação
 - multiplicação
 - módulo
 - atribuição
- O resultado da expressão: $8 * 2 - 4 / 1 + 5$ é:
 - 17
 - 15
 - 12
 - 27
 - 7
- Qual dos operadores abaixo não é um operador de comparação?
 - =
 - <
 - >=
 - >
 - <=
- Qual dos operadores abaixo está escrito de forma errada?
 - ==
 - =>
 - >=
 - <=
 - !=
- Que operadores aritméticos possuem a mesma precedência que a multiplicação?
 - + e /
 - / e %
 - % e -
 - / e ()
 - + e ()
- A expressão: $2 * (3 + 8 - 4) * 1 + 4 / 4 * 2$ tem como resultado.
 - 16
 - 14
 - 12
 - 10
 - 8
- O operador de igualdade é "==" e o operador de atribuição é "=". Essa declaração é:
 - Verdadeira
 - Falsa
- O operador módulo é utilizado para extrair o resto de uma divisão. Para isso é usado o sinal de porcentagem (|). Essa declaração é:

- a) Verdadeira
- b) Falsa

9. Suponha o código abaixo:

```
var x = 10;  
document.write(++x);  
document.write(--x);  
document.write(x--);
```

Qual o resultado da terceira linha após ele ser executado?

- a) 8
- b) 9
- c) 10
- d) 11
- e) 12

10. Qual o resultado da operação abaixo?

$17 \% 3 = ?$

- a) 0
- b) 1
- c) 2
- d) 3
- e) 4

11. O operador lógico _____ retorna um resultado verdadeiro somente se ambas as expressões forem verdadeiras.

- a) ||
- b) &&
- c) !
- d) !=

12. Qual o resultado de x no código abaixo depois de executado?

```
var x = 5;  
x += 10;  
++x;
```

- a) 14
- b) 15
- c) 16
- d) 20

13. Quais os valores de x e y no código abaixo depois de executado?

```
var x=0, y=5;  
++x;  
--y;  
x--;  
x+=y;  
y-=y++;  
x+=y--;  
x%y;  
document.write(x + "<br>");  
document.write(x + "<br>");
```

- a) 1 e 2
- b) 2 e 2
- c) 3 e 3
- d) 4 e 4
- e) 5 e 4

JORGE ELDER