Expressões e Operadores

Introdução da seção

O que é uma expressão?

- Uma **instrução de código** que será avaliada **e resultará em um valor**;
- Uma **simples impressão de um texto** é uma expressão;
- **Uma soma ou operação matemática mais complexa** também;
- Na programação realizaremos **diversas expressões** durante nosso código, para formar nosso software;

O que é um operador?

- Operadores são **recursos que utilizamos para compor expressões mais complexas**;
- Alguns deles: +, -, **, /, ++, >, <, >=, <= e etc…
- Estas operações podem matemáticas ou até mesmo comparações;
- A ideia principal é que um **novo valor é gerado** ou também um **booleano pode ser retornado**;

Ordem dos operadores

- O PHP e as linguagens de programação **executam os operadores na mesma ordem que na matemática**;
- Ou seja em: 2 + 2 * 4, teremos o resultado de **10**;
- Pois **a multiplicação é avaliada antes da soma**;
- Mesmo que a primeira operação seja soma;
- Podemos utilizar **( )** para separar operações;

Exercício 10

- Crie um arquivo PHP;
- Crie uma operação que utiliza subtração (-), divisão (/) e multiplicação
- Armazene todos os valores em variáveis;
- Imprima o resultado final na tela;

Mudança de tipo implícito

- O PHP em certas operações **muda o tipo de dado** de forma implícita;
- Por exemplo 5 / 2 = 2.5 (gera um **float**)
- E 5 . 5 resulta em 55 (gera uma **string**, o . é o operador de concatenação)
- Por isso, temos que **tomar cuidado** com algumas expressões que podem gerar resultados indesejados;
- Este recurso é chamado de **auto cast**;

Exercício 11

- Crie um arquivo PHP;
- Teste a expressão “5” * 12;
- Utilize a função **gettype()** com o resultado como parâmetro para checar o tipo resultante da operação;

Operadores aritméticos

- Temos os **operadores básicos** da matemática em PHP;
- Soma: +
- Subtração: -
- Divisão: /
- Multiplicação: *

Exercício 12

- Crie um arquivo PHP;
- Crie uma operação com cada um dos operadores básicos;
- Cada operação deve estar em uma variável diferente;
- Imprima cada uma das etapas;
- Ex: soma -> multiplicação -> divisão -> subtração;

Operador de módulo

- O operador de módulo é inserido no código pelo símbolo de **%**
- Sua função é realizar **uma divisão**;
- Mas como resultado ele **apresenta apenas o resto** da mesma;

Exercício 13

- Crie um arquivo PHP;
- Teste o operador de resto em duas divisões;
- Uma não exata e outra exata;

Exponenciação

- Podemos realizar o cálculo de potência com o símbolo **;
- Exemplo: 5 ** 2;
- Desta maneira teremos o resultado de **5 elevado a 2**;

Operador de concatenação

- Em PHP podemos concatenar valores com **.** (ponto)
- Concatenar é o ato de **juntar vários textos e/ou números** em apenas uma string;
- **Não há limites** de quantas expressões podem ser concatenadas;

Exercício 14

- Crie um arquivo PHP;
- Crie uma variável saudação, nome e outra de sobrenome;
- Imprima com echo a concatenação de saudação, nome e sobrenome;

Auto incremento e auto decremento

- Podemos incrementar um valor ou decrementar com os operadores: **++** e **--**;
- Exemplo: $n++ ou $x--
- Onde n e x são variáveis, e **terão seus valores alterados com +1 e -1**;
- Estes operadores são muito utilizados em **estruturas de repetição**;

Operadores de comparação

- As operações com operadores de comparação resultarão em true or false;
- Igualdade: **==**
- Idêntico a: **===**
- Diferença: **!=**
- Não idêntico a: **!==**
- Maior e maior ou igual a: **> e >=**
- Menor e menor ou igual a: **< e <=**

Operador de igualdade

- Com o **operador de igualdade** verificamos se um valor é igual ao outro;
- O símbolo é: **==**
- Exemplo: 5 == 4 # false
- Exemplo: 3 == 3 # true

Exercício 15

- Crie uma operação que retorne falso com igualdade;
- Crie uma operação que retorne verdadeiro com igualdade;

Operador idêntico a

- Com o **operador idêntico a** verificamos se um valor é igual ao outro, avaliando o seu tipo também;
- O símbolo é: **===**
- Exemplo: 5 === 5 # true
- Exemplo: 3 === “3” # false

Operador de diferença

- Com o **operador de diferença** verificamos se um valor é diferente de outro;
- O símbolo é: **!=**
- Exemplo: 5 != 5 # false
- Exemplo: 10 != 5 # true

Operador não idêntico a

- Com o **operador não idêntico a** verificamos se um valor é diferente de outro, avaliando o seu tipo também;
- O símbolo é: **!==**
- Exemplo: 5 !== 4 # false
- Exemplo: 3 !== “3” # true

Exercício 16

- Insira o valor 5 em uma variável, e o valor 3 em outra;
- Teste os operadores de: igualdade, diferença, idêntico e não idêntico;

Operador maior e maior ou igual

- Com o **operador maior que** verificamos se um valor é maior que outro;
- O símbolo é: **>**
- Exemplo: 5 > 4 # true
- Com o **operador maior ou igual a** verificamos se um valor é maior ou igual a outro;
- O símbolo é: **>=**
- Exemplo: 5 >= 5 # true

Operador menor e menor ou igual

- Com o **operador menor que** verificamos se um valor é menor que outro;
- O símbolo é: **<**
- Exemplo: 5 < 4 # false
- Com o **operador menor ou igual a** verificamos se um valor é menor ou igual a outro;
- O símbolo é: **<=**
- Exemplo: 11 <= 12 # true

Operadores lógicos

- Com os operadores lógicos podemos **encadear várias comparações**;
- Operador AND: **&&**
- Operador OR: **||**
- Operador NOT: **!**

Tabela verdade

- Com a tabela verdade, temos um resumo dos operadores lógicos:

NOT

x	x'
0	1
1	0

AND

x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

fonte: https://introcs.cs.princeton.edu/java/home/

Operador lógico AND

- Os operadores lógicos em conjunto dos de comparação **também retornam uma booleano** (true ou false);
- No caso de **AND** temos **true** apenas quando **as duas comparações são verdadeiras**;
- Símbolo: &&
- Ex: 5 > 2 && 10 < 100 # true

Exercício 17

- Verifique as seguintes operações com AND;
- 15 > 5 AND “João” === “João”
- “teste” > 5 AND 1
- 2 == 3 AND 5 >= 3

Operador lógico OR

- O operador lógico OR resulta em **verdadeiro** caso **qualquer um dos lados da operação seja verdadeiro**;
- E só resulta em **falso** caso os **dois lados sejam falsos**;
- Símbolo: ||
- Exemplo: 5 > 15 || “teste” == “teste” # true

Exercício 18

- Verifique as seguintes operações com OR;
- 12 < 5 OR “João” === “João”
- 1 > 5 OR 1
- 20 === “20” AND 51 >= 31

Operador lógico NOT

- O operador lógico **NOT** apenas **inverte o resultado booleano** de uma operação, se é true vira false e se é false vira true;
- Símbolo: **!**
- Exemplo: !true # false
- Exemplo: !(5 > 2) # false

Operadores de conversão (cast)

- Com os **operadores de conversão** podemos **forçar uma variável ser de um determinado tipo**;
- **Nem todos são úteis**, os mais utilizados são para converter uma string em número;
- Operadores: int, bool, float, string, array, object e unset;
- Exemplo: $a = (float) “5.34243” # string é convertida para float

Exercício 19

- Converta os seguintes dados para int com o operador de cast;
- “testando”
- 12.9
- true
- [1, 2, 3]
- E veja os resultados

Operadores de atribuição

- Com estes operadores podemos **atribuir valor a uma variável**;
- O mais conhecido é o **=**, porém temos algumas variações do mesmo;
- Operadores: **+=, -=, /=, *= e %=**;
- Cada um destes fará uma **operação antes da atribuição**;

Operador ternário

- Este operador constitui uma **estrutura de condição resumida**;
- **Na maioria dos casos** vamos optar por if/else;
- Porém em situações simples podemos utilizar o ternário;
- Exemplo: 5 > 2 **?** echo “5 é maior que dois” **:** echo “5 é menor que 2”
- A primeira interrogação vem **antes da comparação**;
- E o : é utilizado para uma segunda situação, caso a primeira seja falsa;

Exercício 20

- Atribua dois números a variáveis distintas;
- Faça uma comparação de menor ou igual com o operador ternário;
- Imprima resultados para ambas as possibilidades;

Expressões e Operadores

Conclusão da seção