

PROGRAMAÇÃO WEB II

Prof. Ricardo e Prof. André

2º ETIM DS



Operadores PHP

Operadores são usados para realizar operações em variáveis e valores.

O PHP divide os operadores nos seguintes grupos:

- Operadores aritméticos
- Operadores de atribuição
- Operadores de comparação
- Operadores de incremento / decremento
- Operadores lógicos
- Operadores de string
- Operadores de matriz
- Operadores de atribuição condicional

Operadores aritméticos PHP

Os operadores aritméticos PHP são usados com valores numéricos para realizar operações aritméticas comuns, como adição, subtração, multiplicação etc.

Operator	Name	Example	Result
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power

Operadores de atribuição de PHP

Os operadores de atribuição PHP são usados com valores numéricos para gravar um valor em uma variável.

O operador de atribuição básico em PHP é '='. Isso significa que o operando esquerdo é definido com o valor da expressão de atribuição à direita.

Assignment	Same as...	Description
<code>x = y</code>	<code>x = y</code>	The left operand gets set to the value of the expression on the right
<code>x += y</code>	<code>x = x + y</code>	Addition
<code>x -= y</code>	<code>x = x - y</code>	Subtraction
<code>x *= y</code>	<code>x = x * y</code>	Multiplication
<code>x /= y</code>	<code>x = x / y</code>	Division
<code>x %= y</code>	<code>x = x % y</code>	Modulus

Operadores de comparação de PHP

Os operadores de comparação PHP são usados para comparar dois valores (número ou string):

Operator	Name	Example	Result
<code>==</code>	Equal	<code>\$x == \$y</code>	Returns true if <code>\$x</code> is equal to <code>\$y</code>
<code>===</code>	Identical	<code>\$x === \$y</code>	Returns true if <code>\$x</code> is equal to <code>\$y</code> , and they are of the same type
<code>!=</code>	Not equal	<code>\$x != \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
<code><></code>	Not equal	<code>\$x <> \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
<code>!==</code>	Not identical	<code>\$x !== \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code> , or they are not of the same type
<code>></code>	Greater than	<code>\$x > \$y</code>	Returns true if <code>\$x</code> is greater than <code>\$y</code>
<code><</code>	Less than	<code>\$x < \$y</code>	Returns true if <code>\$x</code> is less than <code>\$y</code>
<code>>=</code>	Greater than or equal to	<code>\$x >= \$y</code>	Returns true if <code>\$x</code> is greater than or equal to <code>\$y</code>
<code><=</code>	Less than or equal to	<code>\$x <= \$y</code>	Returns true if <code>\$x</code> is less than or equal to <code>\$y</code>
<code><=></code>	Spaceship	<code>\$x <=> \$y</code>	Returns an integer less than, equal to, or greater than zero, depending on if <code>\$x</code> is less than, equal to, or greater than <code>\$y</code> . Introduced in PHP 7.

O operador `<=>` é utilizado para fazer comparações combinadas.

Retorna 0 se os valores de ambos os lados são iguais.

Retorna 1 se o valor à esquerda é maior.

Retorna -1 se o valor à direita é maior.

Exemplo:

```
echo 1 <=> 1; // 0
echo 3 <=> 4; // -1
echo 4 <=> 3; // 1
```

Operadores de incremento / decremento de PHP

Os operadores de incremento do PHP são usados para incrementar o valor de uma variável.

Os operadores de decremento do PHP são usados para diminuir o valor de uma variável.

Operator	Name	Description
<code>++\$x</code>	Pre-increment	Increments <code>\$x</code> by one, then returns <code>\$x</code>
<code>\$x++</code>	Post-increment	Returns <code>\$x</code> , then increments <code>\$x</code> by one
<code>--\$x</code>	Pre-decrement	Decrements <code>\$x</code> by one, then returns <code>\$x</code>
<code>\$x--</code>	Post-decrement	Returns <code>\$x</code> , then decrements <code>\$x</code> by one

Operadores lógicos PHP

Os operadores lógicos do PHP são usados para combinar declarações condicionais.

Operator	Name	Example	Result
and	And	<code>\$x and \$y</code>	True if both <code>\$x</code> and <code>\$y</code> are true
or	Or	<code>\$x or \$y</code>	True if either <code>\$x</code> or <code>\$y</code> is true
xor	Xor	<code>\$x xor \$y</code>	True if either <code>\$x</code> or <code>\$y</code> is true, but not both
&&	And	<code>\$x && \$y</code>	True if both <code>\$x</code> and <code>\$y</code> are true
	Or	<code>\$x \$y</code>	True if either <code>\$x</code> or <code>\$y</code> is true
!	Not	<code>!\$x</code>	True if <code>\$x</code> is not true

Operadores de string PHP

O PHP possui dois operadores que são especialmente projetados para strings.

Operator	Name	Example	Result
.	Concatenation	<code>\$txt1 . \$txt2</code>	Concatenation of <code>\$txt1</code> and <code>\$txt2</code>
<code>.=</code>	Concatenation assignment	<code>\$txt1 .= \$txt2</code>	Appends <code>\$txt2</code> to <code>\$txt1</code>

Operadores de array PHP

Os operadores de array PHP são usados para comparar arrays.

Operator	Name	Example	Result
+	Union	<code>\$x + \$y</code>	Union of <code>\$x</code> and <code>\$y</code>
<code>==</code>	Equality	<code>\$x == \$y</code>	Returns true if <code>\$x</code> and <code>\$y</code> have the same key/value pairs
<code>===</code>	Identity	<code>\$x === \$y</code>	Returns true if <code>\$x</code> and <code>\$y</code> have the same key/value pairs in the same order and of the same types
<code>!=</code>	Inequality	<code>\$x != \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
<code><></code>	Inequality	<code>\$x <> \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
<code>!==</code>	Non-identity	<code>\$x !== \$y</code>	Returns true if <code>\$x</code> is not identical to <code>\$y</code>

Declarações condicionais de PHP

Muitas vezes, ao escrever código, você deseja executar ações diferentes para condições diferentes. Você pode usar instruções condicionais em seu código para fazer isso.

No PHP, temos as seguintes instruções condicionais:

- **if** declaração - executa algum código se uma condição for verdadeira
- **if...else** instrução - executa algum código se uma condição for verdadeira e outro código se essa condição for falsa
- **if...elseif...else** instrução - executa códigos diferentes para mais de duas condições
- **switch** declaração - seleciona um de muitos blocos de código a ser executado

PHP - a declaração if

A **if** instrução executa algum código se uma condição for verdadeira.

Exemplo

Saída 'Tenha um bom dia!' se a hora atual (HORA) for inferior a 20:

```
<?php  
$t = date('H');  
  
if ($t < '20') {  
    echo 'Tenha um bom dia!';  
}  
?>
```

PHP - A instrução if ... Else

A instrução **if...else** executa algum código se uma condição for verdadeira e outro código se essa condição for falsa.

```
<?php
$t = date('H');

if ($t < '20') {
    echo 'Tenha um bom dia!';
} else {
    echo 'Tenha uma boa noite!';
}
?>
```

PHP - A instrução if ... elseif ... else

A instrução `if...elseif...else` executa códigos diferentes para mais de duas condições.

Exemplo

Saída 'Tenha um bom dia!' se a hora atual for inferior a 10 e 'Tenha um bom dia!' se a hora atual for inferior a 20. Caso contrário, será exibido 'Tenha uma boa noite!':

```
<?php
$t = date('H');

if ($t < '10') {
    echo 'Tenha uma boa manhã!';
} elseif ($t < '20') {
    echo 'Tenha um bom dia!';
} else {
    echo 'Tenha uma boa noite!';
}
?>
```

Exercício:

Escreva 'Olá Mundo' se \$a for maior que \$b.

```
<?php  
  
$ a = 50;  
$ b = 10;  
if ($a > $b)  
{  
    echo 'Olá, mundo';  
}  
?>
```

A instrução PHP switch

Use a instrução **switch** para **selecionar um dos muitos blocos de código a serem executados** .

```
switch (n) {  
    case Label1:  
        code to be executed if n=Label1;  
        break;  
    case Label2:  
        code to be executed if n=Label2;  
        break;  
    case Label3:  
        code to be executed if n=Label3;  
        break;  
    ...  
    default:  
        code to be executed if n is different from  
        all labels;  
}
```

Funciona assim: primeiro, temos uma única expressão *n* (geralmente uma variável), que é avaliada uma vez. O valor da expressão é então comparado com os valores para cada caso na estrutura. Se houver uma correspondência, o bloco de código associado a esse caso é executado. Use **break** para evitar que o código seja executado no próximo caso automaticamente. A instrução **default** é usada se nenhuma correspondência for encontrada.

```
<?php
$cor = 'vermelho';

switch ($cor) {
    case 'vermelho':
        echo 'Sua cor favorita é vermelho!';
        break;
    case 'azul':
        echo Sua cor favorita é azul!;
        break;
    case 'verde':
        echo ' Sua cor favorita é verde!';
        break;
    default:
        echo 'Sua cor favorita é outra!';
}
?>
```

Correção do Exercício

Crie um programa em PHP que receba um número digitado pelo usuário e verifique se esse valor é positivo, negativo ou igual a zero. A saída deve ser: "Valor Positivo", "Valor Negativo" ou "Igual a Zero"

C:\xampp\htdocs\pw2etim\03correcao\correcao1.php - Notepad ++

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas

index.php index2.php index4.php index5.php correcao1.php

```
1 <?php
2 $num = 5;
3
4 if ($num > 0) {
5     echo 'Valor positivo';
6 } elseif ($num < 0) {
7     echo 'Valor negativo';
8 } else {
9     echo 'Igual a zero';
10 }
11 ?>
```

Correção do Exercício

Crie um programa em que o usuário escolha uma operação (soma, subtração, multiplicação ou divisão). Crie duas caixas de texto para receber 2 números. Realize a operação escolhida em cada um dos números. */
//1 soma+ 2 sub- 3 div/ 4 mult* 5 resto%