

TEMA 03 - Operadores

Habilidades

- Compreender a finalidade dos operadores aritméticos.
- Compreender o porquê do caractere empregado para declarar uma variável no PHP

Dentro de qualquer linguagem de programação temos os operadores, que têm a função que indica o tipo de operação matemática se é de texto, lógico, entre outros, aqui serão abordados os principais, estes operadores serão executados gerando novos valores a partir do seu chamado. E a linguagem de programação PHP, usa vários tipos de operadores que vamos ver neste tema.

3.1 Tipos de operadores

O PHP oferece vários tipos de operadores, vejamos alguns exemplos como uniário, binário, ternário, matemáticos e lógicos dentre outros.

3.2 Operadores aritméticos

Os operadores aritméticos são responsáveis por fazer os cálculos tradicionais usados na nossa matemática. Segue os exemplos:

Sintaxe	Operação
<code>\$v1+ \$v2</code>	Adição
<code>\$v1 - \$v2</code>	Subtração
<code>\$v1 * \$v2</code>	Multiplicação
<code>\$v1 / \$v2</code>	Divisão
<code>\$v1 % \$v2</code>	Resto da divisão
<code>-\$v2</code>	Negação

Exemplo: [ex1.php](#)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Operadores Aritméticos</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f4f4f4;
    }

    .container {
      width: 80%;
      margin: 0 auto;
      padding: 20px;
      background-color: #fff;
```

```

        border-radius: 5px;
        box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    }
</style>
</head>
<body>
    <div class="container">
        <?php
            // Definindo valores para operações
            $v1 = 10;
            $v2 = 3;

            // Adição
            $resultado_adicao = $v1 + $v2;
            echo "Adição: $v1 + $v2 = $resultado_adicao <br>";

            // Subtração
            $resultado_subtracao = $v1 - $v2;
            echo "Subtração: $v1 - $v2 = $resultado_subtracao <br>";

            // Multiplicação
            $resultado_multiplicacao = $v1 * $v2;
            echo "Multiplicação: $v1 * $v2 = $resultado_multiplicacao <br>";

            // Divisão
            $resultado_divisao = $v1 / $v2;
            echo "Divisão: $v1 / $v2 = $resultado_divisao <br>";

            // Resto da Divisão
            $resultado_resto_divisao = $v1 % $v2;
            echo "Resto da Divisão: $v1 % $v2 = $resultado_resto_divisao <br>";

            // Negação
            $resultado_negacao = -$v2;
            echo "Negação de $v2: $resultado_negacao <br>";
        ?>
    </div>
</body>
</html>

```

3.3 Operadores de comparação

Operadores de comparação ou também chamados de condicionais, tem a finalidade dentro da sintaxe do PHP de comparar valores e dados na programação.

Sintaxe	Operação
\$b == \$c	Igualdade
\$b === \$c	Idêntico
\$b != \$c	Não igual
\$b <> \$c	Diferente
\$b !== \$c	Não idêntico
\$b < \$c	Menor

\$b > \$c	Maior
\$b <= \$c	Menor ou igual
\$b >= \$c	Maior ou igual

Exemplo: [ex2.php](#)

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>Exemplo de Operadores de Comparação em PHP</title>
  <style>
    body { font-family: Arial, sans-serif; background: #f9f9f9; padding: 20px; }
    h1, h2 { color: #333; }
    table { border-collapse: collapse; width: 100%; max-width: 600px; background: white; margin-
bottom: 20px; }
    th, td { border: 1px solid #ccc; padding: 8px; text-align: left; }
    th { background: #eee; }
    .verdadeiro { color: green; font-weight: bold; }
    .falso { color: red; font-weight: bold; }
    .vars { font-family: monospace; background: #efefef; padding: 10px; border-radius: 5px; margin-
bottom: 20px; }
  </style>
</head>
<body>

<h1>Operadores de Comparação em PHP</h1>

<?php
$b = 10;
$c = "10";
$d = 15;

function resultado($condicao) {
  return $condicao ? "<span class='verdadeiro'>Verdadeiro</span>" : "<span class='falso'>
Falso</span>";
}

function tipo_valor($var) {
  return gettype($var) . " (" . var_export($var, true) . ")";
}
?>

<h2>Valores das Variáveis</h2>
<div class="vars">
  <strong>$b</strong>: <?= tipo_valor($b) ?><br>
  <strong>$c</strong>: <?= tipo_valor($c) ?><br>
  <strong>$d</strong>: <?= tipo_valor($d) ?><br>
</div>

<table>
```

Expressão	Resultado	Explicação
<code>\$b == \$c</code>	<code><?= resultado(\$b == \$c) ?></code>	Igualdade de valores (10 == "10")
<code>\$b === \$c</code>	<code><?= resultado(\$b === \$c) ?></code>	Igualdade de valor e tipo
<code>\$b != \$c</code>	<code><?= resultado(\$b != \$c) ?></code>	Não igual
<code>\$b <> \$d</code>	<code><?= resultado(\$b <> \$d) ?></code>	Diferente
<code>\$b !== \$c</code>	<code><?= resultado(\$b !== \$c) ?></code>	Não idêntico (valor ou tipo diferentes)
<code>\$b < \$d</code>	<code><?= resultado(\$b < \$d) ?></code>	Menor
<code>\$d > \$b</code>	<code><?= resultado(\$d > \$b) ?></code>	Maior
<code>\$b <= \$c</code>	<code><?= resultado(\$b <= \$c) ?></code>	Menor ou igual
<code>\$d >= \$c</code>	<code><?= resultado(\$d >= \$c) ?></code>	Maior ou igual

```
</body>
</html>
```

3.4 Operadores String

Na linguagem de programação PHP, temos dois operadores de STRING, no caso junção de dois ou mais texto.

Sintaxe	Operação
\$nome . \$sobre	Concatenação “União”
\$nome .= \$sobre	Atribuição de concatenação

Exemplo: **ex3.php**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Operadores de String</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f4f4f4;
    }

    .container {
      width: 300px;
      margin: 50px auto;
      padding: 20px;
      background-color: #fff;
      border-radius: 5px;
      box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    }

    input[type="text"], input[type="submit"] {
      margin-bottom: 10px;
      width: 100%;
      padding: 8px;
      box-sizing: border-box;
    }
  </style>
</head>
<body>
  <div class="container">
    <h2>Operadores de String</h2>
    <form method="post" action="">
      <input type="text" name="nome" placeholder="Digite seu nome" required><br>
      <input type="text" name="sobrenome" placeholder="Digite seu sobrenome" required><br>
      <input type="submit" value="Concatenar">
```

```

</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nome = $_POST['nome'];
    $sobrenome = $_POST['sobrenome'];

    // Concatenação
    $nome_completo = $nome . ' ' . $sobrenome;
    echo "Nome Completo (Concatenação): $nome_completo <br>";

    // Atribuição de Concatenação
    $nome .= ' ' . $sobrenome;
    echo "Nome Completo (Atribuição de Concatenação): $nome";
}
?>
</div>
</body>
</html>

```

3.5 Operadores incremento e decremento

No PHP temos dois operadores de aumentar e diminuir respectivos valores, incremental e decrementar valores dentro de variáveis. Eles podem ser pré-incremento e pós-decremento.

Exemplos:

Incremental (++)

Pré e pós incremental

Decremental (--)

Pré e pós decremental

Significa que se o sinal for colocado antes da variável, primeiro será executado subtraído ou acrescentado sendo pré, logo o sinal sendo depois fica sendo pós decrementa ou incremental, veja um exemplo na sintaxe do PHP.

Exemplo: **ex4.php**

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Diferença entre Pré e Pós-incremento</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
        }

        .container {
            width: 300px;
            margin: 50px auto;
            padding: 20px;

```

```

        background-color: #fff;
        border-radius: 5px;
        box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
        text-align: center;
    }

    .result {
        margin-top: 20px;
    }
</style>
</head>
<body>
<div class="container">
    <h2>Diferença entre Pré e Pós-incremento</h2>
    <div class="result">
        <?php
            $i = 0;
            echo "<h3>Pré-incremento (++\${i}):</h3>";
            echo "Valor inicial de \${i}: \${i} <br>";
            echo "Valor após o pré-incremento: " . ++\${i} . " <br>";
            echo "Valor final de \${i}: \${i} <br>";

            $j = 0;
            echo "<h3>Pós-incremento (\${j++}):</h3>";
            echo "Valor inicial de \${j}: \${j} <br>";
            echo "Valor antes do pós-incremento: " . \${j++} . " <br>";
            echo "Valor final de \${j}: \${j} <br>";
        ?>
    </div>
</div>
</body>
</html>

```

Ambos ++\$i e \$i++ são operadores de incremento em PHP, mas há uma diferença fundamental entre eles: ++\$i é chamado de "pré-incremento" e \$i++ é chamado de "pós-incremento".

Neste exemplo, usamos o operador de pré-incremento (++\$i) e o operador de pós-incremento (\$i++) em duas variáveis diferentes, \$i e \$j, respectivamente. Após a exibição dos valores iniciais, mostramos o resultado após o incremento e o valor final da variável em ambos os casos.

A diferença chave é que o pré-incremento (++\$i) incrementa o valor da variável antes de utilizá-lo em uma expressão, enquanto o pós-incremento (\$i++) utiliza o valor atual da variável em uma expressão e então a incrementa. Isso resulta em um comportamento diferente quando você quer utilizar o valor da variável imediatamente após o incremento.

3.6 Operadores lógicos

Operadores lógicos são usados para a parte lógica de entrada de fluxo de dados, das operações de comparações. Veja os exemplos:

Sintaxe	Operação
\$a && \$b	E - o valor será verdadeiro quando as duas variáveis são verdadeiras

\$a \$b	OU – O valor será verdadeiro quando umas das variáveis é verdadeiras
\$a xor \$b	Exclusivo – quando ambas forem verdadeiras
! \$a	Não – A variável será verdadeira quando o valor de \$a não for verdadeiro

Exemplo: **ex5.php**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Operadores Lógicos</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f4f4f4;
    }

    h2 {
      text-align: center;
    }

    form {
      text-align: center;
      margin-top: 20px;
    }

    label {
      display: block;
      margin-bottom: 5px;
    }

    select {
      padding: 5px;
      border-radius: 4px;
      border: 1px solid #ccc;
    }

    input[type="submit"] {
      margin-top: 10px;
      padding: 8px 16px;
      cursor: pointer;
      background-color: #007bff;
      color: #fff;
      border: none;
      border-radius: 4px;
      transition: background-color 0.3s ease;
    }

    input[type="submit"]:hover {
```



```

        background-color: #0056b3;
    }

    .result {
        margin-top: 20px;
        text-align: center;
    }
</style>
</head>
<body>
    <h2>Operadores Lógicos</h2>
    <form method="post" action="">
        <label for="a">Valor A (true/false):</label>
        <select name="a" id="a">
            <option value="true">True</option>
            <option value="false">False</option>
        </select><br><br>
        <label for="b">Valor B (true/false):</label>
        <select name="b" id="b">
            <option value="true">True</option>
            <option value="false">False</option>
        </select><br><br>
        <input type="submit" value="Calcular">
    </form>

    <div class="result">
        <?php
            if ($_SERVER["REQUEST_METHOD"] == "POST") {
                $a = $_POST['a'] == 'true';
                $b = $_POST['b'] == 'true';

                echo "<h3>Resultados:</h3>";
                echo "A = " . ($a ? 'True' : 'False') . "<br>";
                echo "B = " . ($b ? 'True' : 'False') . "<br><br>";

                echo "A && B (E) = " . ($a && $b ? 'True' : 'False') . " - O valor será verdadeiro quando as
duas variáveis são verdadeiras<br>";
                echo "A || B (OU) = " . ($a || $b ? 'True' : 'False') . " - O valor será verdadeiro quando uma
das variáveis é verdadeira<br>";
                echo "A xor B (Exclusivo) = " . (($a xor $b) ? 'True' : 'False') . " - Exclusivo, quando
exatamente uma das variáveis é verdadeira<br>";
                /*xor retorna verdadeiro se os dois operandos forem diferentes, e falso se os dois forem
iguais, tanto verdadeiro como falso.*/
                echo "!A (Negação) = " . (!$a ? 'True' : 'False') . " - A variável será verdadeira quando o valor
de A não for verdadeiro<br>";
            }
        ?>
    </div>
</body>
</html>

```

ATIVIDADES

1. Qual é o caractere usado para definir uma variável no PHP?
2. Qual é a função dos operadores?
3. Qual é o nome do operador que usamos para cálculos matemáticos?
4. Qual é a diferença do operador && e ||?
5. Por que é importante usar o símbolo "\$" antes do nome de uma variável no PHP? Explique o conceito por trás dessa prática.
6. Escolha um operador aritmético e um operador de comparação e crie uma situação hipotética em que ambos seriam usados em um código PHP.
7. Explique a finalidade dos operadores aritméticos em PHP e forneça exemplos de uso para cada um deles, incluindo adição, subtração, multiplicação e divisão
8. Qual é a diferença no PHP, usando um sinal de igual e dois sinal de igual?
9. Explique a diferença entre os operadores de pré-incremento e pós-incremento em PHP, utilizando exemplos para ilustrar suas explicações.
10. O que significa "concatenação" quando se trata de operadores de String em PHP? Dê um exemplo de como você usaria a concatenação para unir duas strings.

Respostas para as Atividades:

1. O caractere usado para definir uma variável no PHP é o símbolo \$.
2. Os operadores têm a função de executar operações específicas, como cálculos matemáticos, comparações, concatenação de strings, entre outros.
3. O operador usado para cálculos matemáticos em PHP é o sinal de adição (+), subtração (-), multiplicação (*), divisão (/), resto da divisão (%), entre outros.
4. A diferença entre os operadores && e || é que && é o operador lógico "E" (AND), que retorna verdadeiro somente se ambas as expressões forem verdadeiras, enquanto || é o operador lógico "OU" (OR), que retorna verdadeiro se pelo menos uma das expressões for verdadeira.
5. O símbolo \$ é importante antes do nome de uma variável no PHP porque indica ao interpretador que aquilo ali é uma variável e não uma palavra-chave ou uma função.
6. Exemplo de uso de operadores aritméticos e de comparação:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Verificação de Maioridade</title>
</head>
<body>
  <h2>Verificação de Maioridade</h2>
  <?php
    // Definindo a idade da pessoa e a idade mínima para ser considerada maior de idade
    $idade = 25;
    $maioridade = 18;

    // Operadores aritméticos e de comparação
    // Adicionamos 5 à idade e multiplicamos por 2 e então comparamos se o resultado é maior ou
```

igual à idade mínima de maioridade

```
$resultado = ($idade + 5) * 2 >= $maioridade;
```

```
// Exibindo o resultado com base na avaliação da condição
```

```
/* Se o resultado for verdadeiro, a pessoa é considerada maior de idade; caso contrário, é considerada menor de idade*/
```

```
echo $resultado ? "A pessoa é maior de idade." : "A pessoa é menor de idade.";
```

```
?>
```

```
</body>
```

```
</html>
```

7. Finalidade dos operadores aritméticos em PHP e exemplos:

Adição (+): Soma dois valores.

Subtração (-): Subtrai um valor de outro.

Multiplicação (*): Multiplica dois valores.

Divisão (/): Divide um valor pelo outro.

8. No PHP, um sinal de igual (=) é usado para atribuição de valores a variáveis, enquanto dois sinais de igual (==) são usados para comparação de valores. O operador == verifica se os valores dos dois lados da expressão são iguais, enquanto = atribui um valor à variável.

9. Diferença entre os operadores de pré-incremento e pós-incremento em PHP:

Pré-incremento (++\$i): Incrementa o valor da variável antes de usá-la em uma expressão.

Pós-incremento (\$i++): Utiliza o valor atual da variável em uma expressão e então a incrementa.

10. Concatenação em PHP:

Concatenação é a união de duas ou mais strings em uma única string. Em PHP, o operador de concatenação é o ponto (.).

```
<!DOCTYPE html>
```

```
<html lang="pt-BR">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Exemplo de Concatenação em PHP</title>
```

```
</head>
```

```
<body>
```

```
  <h2>Exemplo de Concatenação em PHP</h2>
```

```
  <form method="post" action="">
```

```
    <label for="nome">Nome:</label>
```

```
    <input type="text" name="nome" id="nome" required><br>
```

```
    <label for="sobrenome">Sobrenome:</label>
```

```
    <input type="text" name="sobrenome" id="sobrenome" required><br>
```

```
    <input type="submit" value="Enviar">
```

```
  </form>
```

```
<?php
```

```
// Verifica se os dados do formulário foram enviados
```

```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Obtém os valores dos campos do formulário
    $nome = $_POST['nome'];
    $sobrenome = $_POST['sobrenome'];

    // Concatena os valores para formar o nome completo
    $nome_completo = $nome . " " . $sobrenome;

    // Exibe o nome completo na página
    echo "<p>O nome completo é: $nome_completo</p>";
}
?>

```

```

</body>
</html>

```

Desafios

- 01) Faça um algoritmo no PHP, que solicita três valores e soma os mesmos, usando os elementos echo.
- 02) Faça um algoritmo no PHP, que solicita três notas de um aluno e mostre na tela a média dele e o nome, usando os elementos echo e ternário.

Desafio 01:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Soma de Três Valores</title>
</head>
<body>
    <h2>Soma de Três Valores</h2>
    <form method="post" action="">
        <input type="number" name="valor1" placeholder="Digite o primeiro valor" required><br>
        <input type="number" name="valor2" placeholder="Digite o segundo valor" required><br>
        <input type="number" name="valor3" placeholder="Digite o terceiro valor" required><br>
        <input type="submit" value="Calcular">
    </form>

    <?php
    if ($_SERVER["REQUEST_METHOD"] == "POST") {

        //Código aqui !!!
    }
    ?>
</body>
</html>

```

Desafio 02:

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Verificação de Média</title>
</head>
<body>
  <h2>Verificação de Média</h2>
  <form method="post" action="">
    <label for="nota1">Nota 1:</label>
    <input type="number" name="nota1" id="nota1" required><br>
    <label for="nota2">Nota 2:</label>
    <input type="number" name="nota2" id="nota2" required><br>
    <input type="submit" value="Calcular Média">
  </form>

  <?php
  if ($_SERVER["REQUEST_METHOD"] == "POST") {

      //Código aqui !!!
  }
  ?>
</body>
</html>

```

O operador ternário em PHP é uma forma abreviada de expressar uma estrutura condicional simples. Ele permite que você avalie uma expressão e retorne um valor com base nessa avaliação em uma única linha de código. A sintaxe do operador ternário é a seguinte:

condição ? valor_se_verdadeiro : valor_se_falso;

Desafio 03:

Desafio: Cálculo da Média de Três Valores e Verificação

O usuário deverá inserir três valores e o sistema calculará a média. Se a média for maior ou igual a 7, o sistema indicará que o resultado está "acima do esperado". Caso contrário, o resultado será "abaixo do esperado".