

Treinamento Fundamentos de PHP

Turma 2023/01 – Aula 2/8



1

Semana 1



Aula 2 – Operadores e Estruturas de Controle

- Operadores aritméticos, de comparação e lógicos
- Estrutura de controle: if, else, elseif, switch
- Laço de repetição: while, do-while, for, foreach

2

Recapitulando...



Na última aula vimos sobre variáveis e tipos de dados

- Tipo de dados escalares (integer, float, boolean, string)
 - Formas de especificar uma string (aspas simples, aspas duplas, nowdoc, heredoc)
- Tipo de dados complexos
 - Array
 - Array associativo
 - Array multidimensional
- Concatenação de string
- Interpolação
- Acessando elemento do array pelo índice
- Algumas funções de array
- PHPDoc / DocComment / DocBlock
 - <https://docs.phpdoc.org/guide/getting-started/what-is-a-docblock.html>
 - <https://github.com/php-fig/fig-standards/blob/master/proposed/phpdoc.md>
 - <https://github.com/php-fig/fig-standards/blob/master/proposed/phpdoc-tags.md>

Copyright© 2023 Accenture All Rights Reserved

3

3

Operadores aritméticos



Lembram da aritmética básica da escola? Estes operadores funcionam exatamente como aqueles.

- **Adição (+):** usado para somar dois valores;
- **Subtração (-):** usado para subtrair um valor de outro;
- **Multiplicação (*):** usado para multiplicar dois valores;
- **Divisão (/):** usado para dividir um valor pelo outro;
- **Módulo (%):** usado para obter o resto da divisão de um valor pelo outro;

Copyright© 2023 Accenture All Rights Reserved

4

4

Operadores de comparação



- Os operadores de comparação no PHP retornam um valor booleano (**true** ou **false**) indicando se a comparação é verdadeira ou falsa.
- É importante notar que ao comparar valores de tipos diferentes, o PHP fará a conversão de tipos automaticamente, o que pode levar a resultados inesperados

```

1 <?php
2
3 var_dump(2 == '2'); // Igual - Saída: true
4 var_dump(2 === 2); // Idêntico - Saída: true
5 var_dump(2 != 3); // Diferente - Saída: true
6 var_dump(2 < 3); // Diferente - Saída: true
7 var_dump(2 !== '2'); // Não idêntico - Saída: true
8 var_dump(2 > 3); // Maior que - Saída: false
9 var_dump(2 >= 3); // Maior ou igual que - Saída: false
10 var_dump(2 < 3); // Menor que - Saída: true
11 var_dump(2 <= 3); // Menor ou igual que - Saída: true
12

```

Copyright© 2023 Accenture All Rights Reserved

5

5

Operadores de comparação



Curiosidade

- Antes do PHP 8, se uma string fosse comparada com um número, ela era convertida.

<https://onlinephp.io/>

PHP Sandbox	
<pre> 1 <?php 2 echo 'Versão do PHP: ' . PHP_VERSION . PHP_EOL . PHP_EOL; 3 4 echo '0 == "a" => '; 5 var_dump(0 == 'a'); 6 7 echo PHP_EOL . '2 == 2 => '; 8 var_dump(2 == '2'); 9 </pre>	<p>Result for 7.4.33:</p> <p>Versão do PHP: 7.4.33</p> <p>0 == "a" => bool(true)</p> <p>2 == 2 => bool(true)</p>
PHP Sandbox	
<pre> 1 <?php 2 echo 'Versão do PHP: ' . PHP_VERSION . PHP_EOL . PHP_EOL; 3 4 echo '0 == "a" => '; 5 var_dump(0 == 'a'); 6 7 echo PHP_EOL . '2 == 2 => '; 8 var_dump(2 == '2'); 9 </pre>	<p>Result for 8.2.5:</p> <p>Versão do PHP: 8.2.5</p> <p>0 == "a" => bool(false)</p> <p>2 == 2 => bool(true)</p>

Copyright© 2023 Accenture All Rights Reserved

6

6

Operadores lógicos



São usados para avaliar expressões booleanas e retornar um valor booleano (verdadeiro ou falso) como resultado.

Existem quatro operadores lógicos no PHP:

Exemplo	Nome	Resultado
\$a and \$b	E	Verdadeiro (true) se tanto \$a quanto \$b são verdadeiros.
\$a or \$b	OU	Verdadeiro se \$a ou \$b são verdadeiros.
\$a xor \$b	XOR	Verdadeiro se \$a ou \$b são verdadeiros, mas não ambos.
! \$a	NÃO	Verdadeiro se \$a não é verdadeiro.
\$a & \$b	E	Verdadeiro se tanto \$a quanto \$b são verdadeiros.
\$a \$b	OU	Verdadeiro se \$a ou \$b são verdadeiros.

https://www.php.net/manual/pt_BR/language.operators.logical.php

Copyright© 2023 Accenture All Rights Reserved

7

7

Estrutura de controle



IF / ELSE / ELSEIF / ELSE IF

- O construtor **if** permite a execução condicional de fragmentos de código
- Se uma expressão for avaliada como **true**, o PHP executará a declaração, e se avaliá-la como **false**, irá ignorar (caso não tenha outra estrutura de controle – else/elseif/ else if – para testar).

```
if ($idade == 18) {
    echo 'Idade igual a 18';
} else {
    echo 'Idade não é igual a 18';
}
```

https://www.php.net/manual/pt_BR/control-structures.if.php

Copyright© 2023 Accenture All Rights Reserved

8

8

Estrutura de controle



SWITCH

- A declaração switch é similar a uma série de declarações IF na mesma expressão.
- O **switch/case** faz comparação *frouxa*. Isso é diferente da comparação "estrita" ou "rigorosa", que é feita usando o operador `===`.

https://www.php.net/manual/pt_BR/control-structures.switch.php

• Sintaxe:

```

1  <?php
2
3  $valor = 20;
4
5  switch ($valor) {
6      case 10:
7          echo 'Igual a 10' . PHP_EOL;
8          break;
9      case 15:
10         echo 'Igual a 15' . PHP_EOL;
11         break;
12     case 20:
13         echo 'Igual a 20' . PHP_EOL;
14         break;
15     default:
16         echo 'Todos os outros valores' . PHP_EOL;
17         break;
18 }
19

```

Estrutura de repetição



Os laços (ou estruturas) de repetição permitem que o código seja executado repetidamente, permitindo a automatização de tarefas e o processamento de grandes quantidades de dados.

No PHP, existem quatro tipos principais de laços de repetição:

- While
- Do...while
- For
- Foreach

Estrutura de repetição



WHILE

- É usado quando o número de iterações não é conhecido e o bloco de código deve ser executado enquanto a condição for verdadeira

```
1 <?php
2
3 $numero = 0;
4 while ($numero < 10) {
5     $numero = rand(0, 20); // Gera um número aleatório entre 0 e 20
6     echo "Número gerado: $numero <br>";
7 }
8 echo "O loop foi encerrado.";
9
```

Copyright© 2023 Accenture All Rights Reserved

11

11

Estrutura de repetição



DO-WHILE

- É semelhante ao **while**, mas executa o bloco de código pelo menos uma vez, antes de verificar a condição.

```
1 <?php
2
3 $contador = 1;
4
5 do {
6     echo "O contador é: $contador" . PHP_EOL;
7     $contador++;
8 } while ($contador <= 5);
9
```

Copyright© 2023 Accenture All Rights Reserved

12

12

Estrutura de repetição



FOR

- É usado quando o número de iterações é conhecido. Ele é composto por uma inicialização, uma condição de continuação e uma expressão de incremento. de verificar a condição.

```
1  <?php
2
3  // Imprime números de 1 a 12
4  for ($i = 1; $i <= 12; $i++) {
5      echo $i . PHP_EOL;
6  }
7
```

Copyright© 2023 Accenture All Rights Reserved

13

13

Estrutura de repetição



FOREACH

- É usado para iterar sobre os elementos de um array ou objeto. Ele percorre todos os elementos do array ou objeto e executa o bloco de código para cada um deles.

```
1  <?php
2
3  $frutas = array('maçã', 'banana', 'laranja', 'morango');
4
5  foreach ($frutas as $fruta) {
6      echo $fruta . PHP_EOL;
7  }
8
```

Copyright© 2023 Accenture All Rights Reserved

14

14

Estrutura de repetição



BREAK e CONTINUE

- O **break** e o **continue** são usados em laços de repetição (como for, while, do-while, e foreach) para controlar o fluxo de execução do código.
- O **break** interrompe a execução do laço de repetição imediatamente quando a condição especificada é atendida.
- Já o **continue** interrompe a execução atual do laço de repetição e passa para a próxima iteração.

Copyright© 2023 Accenture All Rights Reserved

15

15

Estrutura de repetição



BREAK e CONTINUE

```

1  <?php
2
3  for ($i = 1; $i <= 10; $i++) {
4      // verifica se $i é múltiplo de 3
5      if ($i % 3 == 0) {
6          // se for, ignora o restante do código dentro do laço
7          continue;
8      }
9
10     // imprime o valor de $i
11     echo $i . ' ';
12
13     // verifica se $i é igual a 7
14     if ($i == 7) {
15         // se for, encerra o laço completamente
16         break;
17     }
18 }
19
20 // o código continua aqui após o laço for
21

```

Copyright© 2023 Accenture All Rights Reserved

16

16

Atividade



Calculadora no console (CLI)

- Utilizando entrada de dados pelo prompt de comando, escreva um código em PHP que leia a operação matemática (adição, subtração, multiplicação ou divisão), o primeiro número e o segundo número, e exiba o resultado da operação.

Copyright© 2023 Accenture All Rights Reserved

17

17

Desafio



Mega-Sena (CLI)

- Escreva um código em PHP que leia o nome do apostador e 6 números inteiros, realize o sorteio de 6 números e compare se o apostador acertou todos.

```
$ php aula2/loteria-mega-sena.php
*****
*      Mega-Sena PHP      *
*****
Para sair, pressione Ctrl+C

Informe seu nome: Fulano

Olá Fulano, Informe 6 números inteiros entre 1 e 60
Separe-os ou com . , ; ou espaço
Números: 6 54 15 10 3 21
```

- Caso tenha acertado os 6 números, exibir a mensagem abaixo:

```
Caro(a) [apostador],

Você foi contemplado na mega-sena, acertando os 6 números.
Confira seus números da sorte:
[numero1], [numero2], [numero3], [numero4], [numero5], [numero6]

Parabéns!
```

- Caso não tenha acertado, informe quantos números ele acertou e lhe desejo sorte da próxima vez.

```
Caro(a) [apostador],

Não foi desta vez. Você acertou apenas [x] números.
São eles:
[numero1], [numero2], [numero3], [numero4], [numero5], [numero6]

Boa sorte na próxima!
```

Copyright© 2023 Accenture All Rights Reserved

18

18