

VARIÁVEIS E TIPOS DE DADOS EM PHP

PROF. FÁBIO KRAVETZ

DESENVOLVIMENTO DE SISTEMAS

1º SEMESTRE/2025



Sintaxe básica - PHP

- Assim como o HTML precisa de uma TAG de inicialização, no PHP existem duas TAG's de inicialização:

```
<?php //tag de inicialização convencional      ?><!-- Tag de fechamento -->
<? //tag de inicialização curta                ?><!-- Tag de fechamento -->
```

- TAG de inicialização curta precisa ser configurada no servidor Apache para funcionar corretamente;
- Existem dois tipos de comentários no PHP, o comentário de linha e o comentário de bloco:

```
// Comentário em linha
/*
    Comentário em bloco
*/
```



echo e var_dump

- O comando **echo** gera uma saída na tela. Segundo a documentação da linguagem PHP ***“echo não é uma função mas sim uma construção da linguagem PHP que envia uma ou mais expressões para a saída, sem novas linhas ou espaços adicionais”***;
- A função **echo()** é um pouco mais rápida que o `print()`;
- A função **var_dump()** mostra o conteúdo de uma variável e, é muito utilizada na depuração de códigos;
- A informação fornecida pelo **var_dump()** contém o tipo e o valor da(s) variável(is).



Declaração e uso de variáveis

- Não é necessário iniciar a variável com o tipo (diferente da linguagem C ou Java, por exemplo);
- **Variáveis se iniciam com o caractere \$;**
 \$nome = "José";
 \$sobrenome = "Silva";
 \$idade = 33;
 \$peso = 90;
 \$casado = false;
- Para saber qual o tipo da variável é possível utilizar a função **gettype()**; para poder visualizar é necessário utilizar o **echo**;
- Não é recomendável iniciar o nome da variável com números ou caracteres especiais, exceto se iniciado com _;



Tipos de dados em PHP

- **Existem os tipos primitivos: escalares, compostos e especiais;**
- **ESCALARES**
 - **String:** Sequência de letras, números e símbolos, sempre representados entre aspas. Em linguagem não se tem este tipo de variável;
 - **Int ou Integer:** Um valor numérico ou inteiro;
 - **Float ou Double (Real):** Um valor numérico real, que vem com a parte decimal;
 - **Bool ou Boolean:** Um valor lógico ou Booleano que aceita apenas os valores verdadeiro ou falso;



Declaração e uso de variáveis

- **Array (vetor)**

- É uma variável do tipo lista, onde pode-se armazenar informações dentro de uma mesma chave;
- Utiliza-se o construtor de linguagem array();
- Passagem de valores por parâmetro;
- `$array = array(1, 2 , 3);`

- **Array multidimensional (matriz)**

- Um array pode ser utilizado como valor para outro array;

```
$linguagens = array(  
    array("PHP", "PHP: Hypertext Preprocessor"),  
    array("SQL", "Structured Query Language")  
);
```



Casting

- Permite converter um valor de um tipo de dado para outro;

Operadores	Converte para
(array)	Array
(bool) ou (boolean)	Boolean
(int) ou (integer)	Integer
(object)	Object
(double) ou (float)	Float
(string)	String

```
<?php
    echo (int)12.5 . '<br>';
    echo (float)'13' . '<br>';
    echo (int)200.4 . '<br>';
    echo (int)'de texto para int' .
'<br>';
?>
```



Operadores aritméticos

- Os operadores permitem a obtenção de novos valores a partir de uma ou mais variáveis

Operadores Aritméticos

Exemplo	Nome	Resultado
+\$a	Identidade	Conversão de \$a para int ou float conforme apropriado.
-\$a	Negação	Oposto de \$a.
\$a + \$b	Adição	Soma de \$a e \$b.
\$a - \$b	Subtração	Diferença entre \$a e \$b.
\$a * \$b	Multiplicação	Produto de \$a e \$b.
\$a / \$b	Divisão	Quociente de \$a e \$b.
\$a % \$b	Módulo	Resto de \$a dividido por \$b.
\$a ** \$b	Exponencial	Resultado de \$a elevado a \$b.



Operadores de atribuição

Exemplo	Equivalente	Operação
\$a += \$b	\$a = \$a + \$b	Adição
\$a -= \$b	\$a = \$a - \$b	Subtração
\$a *= \$b	\$a = \$a * \$b	Multiplificação
\$a /= \$b	\$a = \$a / \$b	Divisão
\$a %= \$b	\$a = \$a % \$b	Módulo
\$a **= \$b	\$a = \$a ** \$b	Exponentiation

Exemplo	Equivalente	Operação
\$a .= \$b	\$a = \$a . \$b	Concatenação de strings
\$a ??= \$b	\$a = \$a ?? \$b	Aglutinação de nulls (null coalesce)

Operadores de atribuição

- Para concatenar, ou combinar, duas strings utiliza-se o operador (.);

```
$x = "Hello";  
$y = "World";  
$z = $x . $y;  
echo $z;
```

```
$x = "Hello";  
$y = "World";  
$z = $x . " " . $y;  
echo $z;
```



- Operador de coalescência nula (??):** Retorna o primeiro operando se este existir e não for *null*; caso contrário retorna o segundo operando

```
$username = $_GET['user'] ?? 'nobody';
```

Operadores de comparação

Operadores de comparação

Exemplo	Nome	Resultado
\$a == \$b	Igual	<u>true</u> se \$a é igual a \$b após equalização de tipos.
\$a === \$b	Idêntico	<u>true</u> se \$a é igual a \$b, e eles são do mesmo tipo.
\$a != \$b	Diferente	<u>true</u> se \$a não é igual a \$b depois de equalização de ativos.
\$a <> \$b	Diferente	<u>true</u> se \$a não é igual a \$b depois de equalização de ativos.
\$a !== \$b	Não idêntico	<u>true</u> se \$a não é igual a \$b, ou eles não são do mesmo tipo.
\$a < \$b	Menor que	<u>true</u> se \$a é estritamente menor que \$b.
\$a > \$b	Maior que	<u>true</u> se \$a é estritamente maior que \$b.
\$a <= \$b	Menor ou igual	<u>true</u> se \$a é menor ou igual a \$b.
\$a >= \$b	Maior ou igual	<u>true</u> se \$a é maior ou igual a \$b.
\$a <=> \$b	Spaceship (nave espacial)	Um <u>int</u> menor que, igual a ou maior que zero quando \$a é, respectivamente, menor que, igual a ou maior que \$b.



Operadores aritméticos/comparação

- Os operadores aritméticos permitem a obtenção de novos valores a partir de uma ou mais variáveis;
- O operador **exponencial** pode ser utilizado de duas formas:
 - `$resultado_00 = $a ** $b;`
 - `$resultado_01 = pow($a, $b);`
- Os operadores de comparação são utilizados para comparar dois valores (número ou string);
- O operador **idêntico** e **não idêntico** são apresentados abaixo:
 - `$x === $y` – Retorna verdadeiro se `$x` é igual a `$y`, e eles são do mesmo tipo;
 - `$x !== $y` – Retorna verdadeiro se `$x` não é igual a `$y`, ou eles não são do mesmo tipo.



Operadores comparação

- O operador *spaceship* (<=>) é utilizado para comparar duas expressões. Ele retorna -1, 0 ou 1 quando uma variável (\$a) for respectivamente menor que, igual a ou maior que outra variável (\$b);
- Tal lógica pode ser utilizada para salvar status/ condições em um banco de dados.

```
<?php
// Inteiros
echo 1 <=> 1; // 0
echo 1 <=> 2; // -1
echo 2 <=> 1; // 1

// Números de ponto flutuante
echo 1.5 <=> 1.5; // 0
echo 1.5 <=> 2.5; // -1
echo 2.5 <=> 1.5; // 1

// Strings
echo "a" <=> "a"; // 0
echo "a" <=> "b"; // -1
echo "b" <=> "a"; // 1
?>
```



Operadores lógicos

Operadores Lógicos		
Exemplo	Nome	Resultado
\$a and \$b	E	<u>true</u> se ambos \$a e \$b é <u>true</u> .
\$a or \$b	OU	<u>true</u> se \$a ou \$b é <u>true</u> .
\$a xor \$b	XOR	<u>true</u> se \$a ou \$b é <u>true</u> , mas não ambos ao mesmo tempo.
! \$a	Não	<u>true</u> se \$a não é <u>true</u> .
\$a && \$b	E	<u>true</u> se ambos \$a e \$b são <u>true</u> .
\$a \$b	OU	<u>true</u> se \$a ou \$b é <u>true</u> .

Operador lógico XOR

- Operador lógicos são utilizados para combinar instruções condicionais;
- O valor de retorno só será verdadeiro se uma das instruções for verdadeira e a outra for falsa

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

```
<!DOCTYPE html>
<html>
<body>

<?php
if(5 < 3 xor 5 < 10) {
    echo "Apenas uma das expressões é verdadeira";
}
else
{
    echo "As duas setenças são verdadeiras";
}
?>

</body>
</html>
```



Dúvidas?

