

Escola EB2,3/S de Mondim de Basto

PHP



Trabalho realizado por:
Renato Brizida Nº12
Disciplina: Rede de computadores
Módulo 5
Professora: Anabela Leite
2019/2020

Índice

Índice de Imagens.....	Erro! Marcador não definido.
Introdução	4
PHP	5
O que eu preciso?	6
Sintaxe básica do PHP	7
Comentários em PHP	8
Variáveis PHP	9
Escopo de variáveis do PHP.....	10
Escopo global e local	10
PHP A palavra-chave global	11
PHP A palavra-chave estática	12
Instruções de eco e impressão do PHP	13
A instrução PHP echo.....	13
A declaração de impressão PHP	14
Tipos de dados PHP	15
Funções de String PHP	16
Constantes PHP	16
Matrizes constantes do PHP	17
.....	18
Operadores PHP	18
Operadores de incremento / redução de PHP.....	20
Operadores lógicos PHP	21
Operadores de string PHP	21
Operadores de matriz PHP	22
Operadores de atribuição condicional do PHP.....	22
PHP if ... else ... elseif Instruções	23
Instruções condicionais do PHP	23
PHP - A instrução if.....	23
Cofinanciado por:	

Sintaxe	23
Exemplo	24
A declaração if ... elseif ... else	24
Sintaxe	24
Conclusão	26
Bibliografia	27

Introdução

A temática deste trabalho consiste na apresentação de PHP que é uma linguagem de script de servidor e uma ferramenta poderosa para criar páginas da Web dinâmicas e interativas, vou aprofundar tudo o que está relacionado com PHP, como escrever um sintaxe básica, como declarar variáveis, exemplo de tipo de dados, exemplos e funções de strings, PHP Constants e PHP Operators.

PHP

PHP é uma linguagem de script de servidor e uma ferramenta poderosa para criar páginas da Web dinâmicas e interativas, é uma alternativa amplamente usada, gratuita e eficiente para concorrentes como o ASP da Microsoft.

O PHP 7 é a última versão estável.



O que eu preciso?

Para começar a usar o PHP, você pode:

- Encontre um host com suporte PHP e MySQL
- Instale um servidor web em seu próprio PC e instale PHP e MySQL

Use um host da Web com suporte a PHP

Se o seu servidor ativou o suporte ao PHP, você não precisa fazer nada.

Basta criar alguns `.php` arquivos, colocá-los no diretório da web e o servidor os analisará automaticamente para você.

Você não precisa compilar nada ou instalar nenhuma ferramenta extra.

Como o PHP é gratuito, a maioria dos hosts da Web oferece suporte ao PHP.

Configure o PHP no seu próprio PC

No entanto, se o seu servidor não suportar PHP, você deverá:

- instalar um servidor web
- instalar PHP
- instalar um banco de dados, como MySQL

Sintaxe básica do PHP

Um script PHP pode ser colocado em qualquer lugar do documento.

Um script PHP começa com `<?php` e termina com `?>`:

```
<?php
// PHP code goes here
?>
```

A extensão de arquivo padrão para arquivos PHP é ".php".

Um arquivo PHP normalmente contém tags HTML e algum código de script PHP.

Abaixo, temos um exemplo de um arquivo PHP simples, com um script PHP que usa uma função PHP embutida "echo" para gerar o texto "Hello World!" em uma página da web:

Exemplo:

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

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

Comentários em PHP

Um comentário no código PHP é uma linha que não é executada como parte do programa. Seu único objetivo é ser lido por alguém que esteja visualizando o código.

Os comentários podem ser usados para:

- Permita que outras pessoas entendam seu código
- Lembre-se do que você fez - a maioria dos programadores experimentou voltar ao seu trabalho um ou dois anos depois e ter que repensar o que eles fizeram. Os comentários podem lembrá-lo do que você estava pensando quando escreveu o código

Exemplo:

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

<?php
// You can also use comments to leave out parts of a code line
$x = 5 /* + 15 */ + 5;
echo $x;
?>

</body>
</html>
```

Variáveis PHP

No PHP, uma variável começa com o **\$**sinal, seguido pelo nome da variável:

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

Após a execução das instruções acima, a variável **\$txt** reterá o valor **Hello world!**, a variável **\$x** reterá o valor **5** e a variável **\$y** reterá o valor **10.5**.

Pense nas variáveis como contêineres para armazenar dados.

Uma variável pode ter um nome abreviado (como xey) ou um nome mais descritivo (idade, carname, volume total).

Regras para variáveis PHP:

- Uma variável começa com o **\$**sinal, seguido pelo nome da variável
- Um nome de variável deve começar com uma letra ou o caractere sublinhado
- Um nome de variável não pode começar com um número
- Um nome de variável pode conter apenas caracteres alfanuméricos e sublinhados (Az, 0-9 e _)
- Os nomes de variáveis diferenciam maiúsculas de minúsculas (**\$age** e **\$AGE** são duas variáveis diferentes)

Lembre-se de que os nomes de variáveis do PHP diferenciam maiúsculas de minúsculas!

A **echo** instrução PHP é frequentemente usada para gerar dados na tela.

```
<?php
$txt = "W3Schools.com";
echo "I love $txt!";
?>
```

Escopo de variáveis do PHP

No PHP, variáveis podem ser declaradas em qualquer lugar do script.

O escopo de uma variável é a parte do script em que a variável pode ser referenciada / usada.

O PHP possui três escopos de variáveis diferentes:

- local
- global
- estático

Escopo global e local

Uma variável declarada **fora de** uma função possui um GLOBAL SCOPE e só pode ser acessada fora de uma função:

Exemplo:

Variável com escopo global:

```
<?php
$x = 5; // global scope

function myTest() {
    // using x inside this function will generate an error
    echo "<p>Variable x inside function is: $x</p>";
}
myTest();

echo "<p>Variable x outside function is: $x</p>";
?>
```

Uma variável declarada **dentro de** uma função possui um LOCAL SCOPE e só pode ser acessada nessa função:

Exemplo:

Variável com escopo local

```
<?php
function myTest() {
    $x = 5; // local scope
    echo "<p>Variable x inside function is: $x</p>";
}
myTest();

// using x outside the function will generate an error
echo "<p>Variable x outside function is: $x</p>";
?>
```

PHP A palavra-chave global

A **global** palavra-chave é usada para acessar uma variável global de dentro de uma função.

Para fazer isso, use a **global** palavra - chave antes das variáveis (dentro da função):

Exemplo:

```
<?php
$x = 5;
$y = 10;

function myTest() {
    global $x, $y;
    $y = $x + $y;
}

myTest();
echo $y; // outputs 15
?>
```

Cofinanciado por:

O PHP também armazena todas as variáveis globais em uma matriz chamada `$GLOBALS`. O conteúdo contém o nome da variável. Essa matriz também é acessível a partir de funções e pode ser usada para atualizar variáveis globais diretamente. `$GLOBALS[index] index`

O exemplo acima pode ser reescrito assim:

Exemplo:

```
<?php
$x = 5;
$y = 10;

function myTest() {
    $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
}

myTest();
echo $y; // outputs 15
?>
```

PHP A palavra-chave estática

Normalmente, quando uma função é concluída / executada, todas as suas variáveis são excluídas. No entanto, às vezes queremos que uma variável local NÃO seja excluída. Precisamos disso para outro emprego.

Para fazer isso, use a `static` palavra-chave quando declarar a variável:

Exemplo:

```
<?php
function myTest() {
    static $x = 0;
    echo $x;
    $x++;
}

myTest();
myTest();
myTest();
?>
```

Instruções de eco e impressão do PHP

`Echo` e `print` são mais ou menos iguais. Ambos são usados para enviar dados para a tela.

As diferenças são pequenas: `echo` não tem valor de retorno, enquanto `print` tem um valor de retorno 1, para que possa ser usado em expressões. `Echo` pode usar vários parâmetros (embora esse uso seja raro) e `print` pode usar um argumento. `Echo` é marginalmente mais rápido que `print`.

A instrução PHP echo

A `echo` declaração pode ser usada com ou sem parênteses: `echo` ou `echo()`.

Texto de exibição

O exemplo a seguir mostra como gerar texto com o `echo` comando (observe que o texto pode conter marcação HTML):

Exemplo:

```
<?php
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
?>
```

Variáveis de exibição

O exemplo a seguir mostra como gerar texto e variáveis com a **echo** instrução:

Exemplo:

```
<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>
```

A declaração de impressão PHP

A **print** declaração pode ser usada com ou sem parênteses: **printou print()**.

Texto de exibição

O exemplo a seguir mostra como gerar texto com o **print** comando (observe que o texto pode conter marcação HTML):

Exemplo:

```
<?php
print "<h2>PHP is Fun!</h2>";
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>
```

Variáveis de exibição

O exemplo a seguir mostra como gerar texto e variáveis com a **print** instrução:

Exemplo:

```
<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

print "<h2>" . $txt1 . "</h2>";
print "Study PHP at " . $txt2 . "<br>";
print $x + $y;
?>
```

Tipos de dados PHP

As variáveis podem armazenar dados de diferentes tipos e diferentes tipos de dados podem fazer coisas diferentes.

O PHP suporta os seguintes tipos de dados:

String

Integer

Float

Boolean

Array

Object

NULL

Cofinanciado por:



UNIÃO EUROPEIA
Fundo Social Europeu

Value

Resource

Funções de String PHP

`strlen ()` - Retorna o comprimento de uma string

`str_word_count ()` - conta palavras em uma string

`strrev ()` - reverte uma string

`strpos ()` - procure um texto dentro de uma string

`str_replace ()` - Substituir texto dentro de uma string

Constantes PHP

Uma constante é um identificador (nome) para um valor simples. O valor não pode ser alterado durante o script.

Um nome de constante válido começa com uma letra ou um sublinhado (nenhum sinal \$ antes do nome da constante).

Nota: Diferentemente das variáveis, as constantes são automaticamente globais em todo o script.

Para criar uma constante, use a `define ()` função

Sintaxe

```
define(name, value, case-insensitive)
```

Parâmetros:

nome : especifica o nome da constante

valor : especifica o valor da constante

case-insensitive : Especifica se o nome da constante deve fazer distinção entre maiúsculas e minúsculas. O padrão é falso

Exemplo:

Crie uma constante com um nome que **diferencia maiúsculas de minúsculas**:

```
<?php
define("GREETING", "Welcome to W3Schools.com!");
echo GREETING;
?>
```

Crie uma constante com um nome que não **diferencia maiúsculas de minúsculas**:

```
<?php
define("GREETING", "Welcome to W3Schools.com!", true);
echo greeting;
?>
```

Matrizes constantes do PHP

No PHP7, você pode criar uma constante Array usando a `define ()` função

Exemplo:

```
<?php
define("cars", [
    "Alfa Romeo",
    "BMW",
    "Toyota"
]);
echo cars[0];
?>
```

Cofinanciado por:

As constantes são automaticamente globais e podem ser usadas em todo o script.

Exemplo:

Este exemplo usa uma constante dentro de uma função, mesmo que seja definida fora da função:

```
<?php
define("GREETING", "Welcome to W3Schools.com!");

function myTest() {
    echo GREETING;
}

myTest();
?>
```

Operadores PHP

Operadores são usados para executar operações em variáveis e valores.

O PHP divide os operadores nos seguintes grupos:

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

Cofinanciado por:

Operadores aritméticos PHP

Os operadores aritméticos do PHP são usados com valores numéricos para executar 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 PHP

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

O operador de atribuição básica no PHP é "=". Isso significa que o operando esquerdo é definido como o valor da expressão de atribuição à direita.

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

Operadores de comparação PHP

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

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

Operadores de incremento / redução 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 \$x by one, then returns \$x
<code>\$x++</code>	Post-increment	Returns \$x, then increments \$x by one
<code>--\$x</code>	Pre-decrement	Decrements \$x by one, then returns \$x
<code>\$x--</code>	Post-decrement	Returns \$x, then decrements \$x by one

Operadores lógicos PHP

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

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

Operadores de string PHP

O PHP possui dois operadores especialmente projetados para strings.

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

Operadores de matriz PHP

Os operadores de matriz PHP são usados para comparar matrizes.

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

Operadores de atribuição condicional do PHP

Os operadores de atribuição condicional do PHP são usados para definir um valor dependendo das condições:

Operator	Name	Example	Result
?:	Ternary	<code>\$x = expr1 ? expr2 : expr3</code>	Returns the value of <code>\$x</code> . The value of <code>\$x</code> is <code>expr2</code> if <code>expr1</code> = TRUE. The value of <code>\$x</code> is <code>expr3</code> if <code>expr1</code> = FALSE
??	Null coalescing	<code>\$x = expr1 ?? expr2</code>	Returns the value of <code>\$x</code> . The value of <code>\$x</code> is <code>expr1</code> if <code>expr1</code> exists, and is not NULL. If <code>expr1</code> does not exist, or is NULL, the value of <code>\$x</code> is <code>expr2</code> . Introduced in PHP 7

PHP if ... else ... elseif Instruções

Instruções condicionais do PHP

Muitas vezes, ao escrever código, você deseja executar ações diferentes para diferentes condições. Você pode usar instruções condicionais no seu código para fazer isso.

No PHP, temos as seguintes instruções condicionais:

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

PHP - A instrução if

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

Sintaxe

```
if (condition) {
    code to be executed if condition is true;
}
```

Exemplo

Saída "Tenha um bom dia!" se o horário atual (HOUR) for menor que 20:

```
<?php
$t = date("H");

if ($t < "20") {
    echo "Have a good day!";
}
?>
```

A declaração if ... elseif ... else

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

Sintaxe

```
if (condition) {
    code to be executed if this condition is true;
} elseif (condition) {
    code to be executed if first condition is false and this condition is true;
} else {
    code to be executed if all conditions are false;
}
```

Exemplo

Saída "Tenha um bom dia!" se o horário atual for menor que 10 e "Tenha um bom dia!" se o horário atual for menor que 20. Caso contrário, ele exibirá "Tenha uma boa noite"

```
<?php
$t = date("H");

if ($t < "10") {
    echo "Have a good morning!";
} elseif ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

Conclusão

Neste trabalho falei de tudo relacionado com o PHP, o que é, também falei de variáveis, tipo de dados e mais aprofundadamente das strings dando exemplos e delas e quais as suas funções.

Este trabalho foi importante para o meu conhecimento pois aprendi um bocado do que é e como se trabalha no PHP.

Bibliografia

<http://11pesp1011.blogspot.com/2010/12/exemplos-de-sistemas-operativos-open.html>

<https://es.wikipedia.org/wiki/Knoppix>

<https://pt.wikipedia.org/wiki/FreeNAS>

Youtube



Cofinanciado por:



UNIÃO EUROPEIA
Fundo Social Europeu