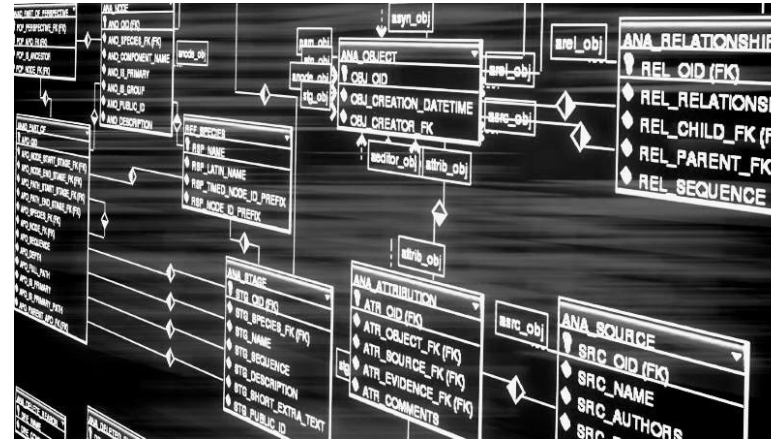


PHP



Trabalho realizado por Francisco Miranda N3 12°C

Disciplina: PSI

Módulo 19

Curso: Técnico de Gestão e Programação de Sistemas Informáticos

Cofinanciado por:



UNIÃO EUROPEIA
Fundo Social Europeu

1. O que é o PHP?

- ▶ PHP é uma linguagem de script de servidor e uma ferramenta poderosa para criar páginas da Web dinâmicas e interativas.
- ▶ O PHP é uma alternativa amplamente usada, gratuita e eficiente para concorrentes como o ASP da Microsoft.
- ▶ O PHP 7 é a última versão estável.

2. O que é necessário instalar para começar a utilizar o PHP?

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

O site oficial do PHP (PHP.net) possui instruções de instalação para o PHP: <http://php.net/manual/en/install.php>

3. Escreva uma sintaxe básica em 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 "echo" embutida para gerar o texto "Hello World!" em uma página da web:

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

<h1>My first PHP page</h1>

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

</body>
</html>
```

4. Exemplo de um comentário em PHP

- ▶ Um comentário no código PHP é uma linha que não é executada como parte do programa. O seu único objetivo é ser lido por alguém que esteja visualizando o código.
- ▶ Os comentários podem ser usados para permitir 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 fez. Os comentários podem lembrá-lo do que você estava pensando quando escreveu o código
- ▶ O PHP suporta várias maneiras de comentar:

```
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// This is a single-line comment  
  
# This is also a single-line comment  
>  
  
</body>  
</html>
```

5. Exemplo de uma declaração de variáveis

- ▶ Uma variável pode ter um nome abreviado (como x e y) ou um nome mais descritivo (idade, carname, volume total).

Regras para variáveis PHP:

- ▶ Uma variável começa com o \$, 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)

Exemplo

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

6. Exemplo de echo e print

- ▶ 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, 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):

```
<?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:

```
<?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: print ou print().
- ▶ **Texto de exibição**

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

```
<?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:

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

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

7. Exemplos de tipos de dados

- ▶ 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
- ▶ Resource

String

- ▶ Uma sequência é uma sequência de caracteres, como "Olá, mundo!".
- ▶ Uma string pode ser qualquer texto entre aspas. Pode ser usado com aspas simples ou duplas:

```
<?php
$x = "Hello world!";
$y = 'Hello world!';

echo $x;
echo "<br>";
echo $y;
?>
```

Integer

- ▶ Um tipo de dados inteiro é um número não decimal entre -2.147.483.648 e 2.147.483.647.

Regras para números inteiros:

- ▶ Um número inteiro deve ter pelo menos um dígito
- ▶ Um número inteiro não deve ter um ponto decimal
- ▶ Um número inteiro pode ser positivo ou negativo

Os números inteiros podem ser especificados em: notação decimal (base 10), hexadecimal (base 16), octal (base 8) ou binária (base 2)

No exemplo a seguir, \$x é um número inteiro. A função PHP var_dump () retorna o tipo e valor dos dados:

```
<?php
$x = 5985;
var_dump($x);
?>
```

Float

- ▶ Um número flutuante (número de ponto flutuante) é um número com um ponto decimal ou um número na forma exponencial.
- ▶ No exemplo a seguir, \$ x é um float. A função PHP var_dump () retorna o tipo e valor dos dados:

```
<?php  
$x = 10.365;  
var_dump($x);  
?>
```

Boolean

- ▶ Um boolean representa dois estados possíveis: VERDADEIRO ou FALSO.

```
$x = true;  
$y = false;
```

- ▶ Boolean são frequentemente usados em testes condicionais.

Array

- ▶ Um array armazena vários valores numa única variável.
- ▶ No exemplo a seguir, \$cars é uma matriz. A função PHP var_dump () retorna o tipo e valor dos dados:

```
<?php  
$cars = array("Volvo", "BMW", "Toyota");  
var_dump($cars);  
?>
```

Object

- ▶ Um object é um tipo de dados que armazena dados e informações sobre como processá-los.
- ▶ No PHP, um object deve ser declarado explicitamente.

- ▶ Primeiro, devemos declarar uma classe de objeto. Para isso, usamos a palavra-chave class. Uma classe é uma estrutura que pode conter propriedades e métodos:

```
<?php
class Car {
    function Car() {
        $this->model = "VW";
    }
}

// create an object
$herbie = new Car();

// show object properties
echo $herbie->model;
?>
```

Null

- ▶ Null é um tipo de dados especial que pode ter apenas um valor: NULL.
- ▶ Uma variável do tipo de dados NULL é uma variável que não possui valor atribuído a ela.
- ▶ **Dica:** Se uma variável for criada sem um valor, será atribuído automaticamente um valor NULL.
- ▶ As variáveis também podem ser esvaziadas definindo o valor como NULL:

```
<?php
$x = "Hello world!";
$x = null;
var_dump($x);
?>
```

Resource

- ▶ O tipo de recurso especial não é um tipo de dados real. É o armazenamento de uma referência a funções e recursos externos ao PHP.
- ▶ Um exemplo comum de uso do tipo de dados do recurso é uma chamada à base de dados.

8. Exemplo de Strings

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

```
<?php  
echo strlen("Hello world!"); // outputs 12  
?>
```

- ▶ `str_word_count ()` - conta palavras em uma string

```
<?php  
echo str_word_count( "Hello world!" ); // outputs 2  
?>
```

- ▶ `strrev ()` - reverte uma string

```
<?php  
echo strrev("Hello world!"); // outputs !dlrow olleH  
?>
```

- ▶ strpos () - procura um texto dentro de uma string

```
<?php  
echo strpos("Hello world!", "world"); // outputs 6  
?>
```

- ▶ `str_replace()` – Substitui um texto dentro de uma string

```
<?php  
echo str_replace("world", "Dolly", "Hello world!"); // outputs Hello Dolly!  
?>
```

10. PHP constants

- ▶ 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.

Create PHP constants

- ▶ Para criar uma constante, use a define ()

```
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;
?>
```

Exemplo

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

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

- ▶ No PHP7, você pode criar uma constante Array usando a define()

Exemplo

Crie uma constante de matriz:

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

11. PHP Operators - Lista de todas as tabelas

▶ Operators são usados para executar operações em variáveis e valores.

O PHP divide os operators 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

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

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	Show it
<code>x = y</code>	<code>x = y</code>	The left operand gets set to the value of the expression on the right	Try it »
<code>x += y</code>	<code>x = x + y</code>	Addition	Try it »
<code>x -= y</code>	<code>x = x - y</code>	Subtraction	Try it »
<code>x *= y</code>	<code>x = x * y</code>	Multiplication	Try it »
<code>x /= y</code>	<code>x = x / y</code>	Division	Try it »
<code>x %= y</code>	<code>x = x % y</code>	Modulus	Try it »

Operadores de comparação PHP

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

Operator	Name	Example	Result	Show it
==	Equal	<code>\$x == \$y</code>	Returns true if \$x is equal to \$y	Try it »
===	Identical	<code>\$x === \$y</code>	Returns true if \$x is equal to \$y, and they are of the same type	Try it »
!=	Not equal	<code>\$x != \$y</code>	Returns true if \$x is not equal to \$y	Try it »
<>	Not equal	<code>\$x <> \$y</code>	Returns true if \$x is not equal to \$y	Try it »
!==	Not identical	<code>\$x !== \$y</code>	Returns true if \$x is not equal to \$y, or they are not of the same type	Try it »
>	Greater than	<code>\$x > \$y</code>	Returns true if \$x is greater than \$y	Try it »
<	Less than	<code>\$x < \$y</code>	Returns true if \$x is less than \$y	Try it »
>=	Greater than or equal to	<code>\$x >= \$y</code>	Returns true if \$x is greater than or equal to \$y	Try it »
<=	Less than or equal to	<code>\$x <= \$y</code>	Returns true if \$x is less than or equal to \$y	Try it »
<=>	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.	Try it »

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	Show it
<code>++\$x</code>	Pre-increment	Increments \$x by one, then returns \$x	Try it »
<code>\$x++</code>	Post-increment	Returns \$x, then increments \$x by one	Try it »
<code>--\$x</code>	Pre-decrement	Decrements \$x by one, then returns \$x	Try it »
<code>\$x--</code>	Post-decrement	Returns \$x, then decrements \$x by one	Try it »

Operadores lógicos PHP

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

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

Operadores de string PHP

O PHP possui dois operadores especialmente projetados para strings.

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

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	Show it
?:	Ternary	$\$x = \text{expr1} ? \text{expr2} : \text{expr3}$	Returns the value of $\$x$. The value of $\$x$ is expr2 if $\text{expr1} = \text{TRUE}$. The value of $\$x$ is expr3 if $\text{expr1} = \text{FALSE}$	Try it »
??	Null coalescing	$\$x = \text{expr1} ?? \text{expr2}$	Returns the value of $\$x$. The value of $\$x$ is expr1 if expr1 exists, and is not NULL. If expr1 does not exist, or is NULL, the value of $\$x$ is expr2 . Introduced in PHP 7	Try it »

12. PHP If... Else... Elseif

- ▶ Muitas vezes, ao escrever o código, você pode desejar executar ações diferentes para diferentes condições. 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!";  
}  
?>
```

If else elseif

- ▶ 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!";  
}  
?>
```