



Web
PHP

Catarina Oliveira

DCT DEPARTAMENTO CIÊNCIA
E TECNOLOGIA

CONTEÚDO

1. Vantagens e desvantagens
2. Duas formas de utilizar
3. Variáveis
 1. Tipos de variáveis
 2. Âmbito das variáveis
4. Variáveis externas (formulários com GET e POST)
5. Operadores
 1. Aritméticos
 2. de atribuição
 3. de comparação
 4. de incremento / decremento
 5. lógicos
 6. de strings
 7. de arrays
 8. condicionais
6. Instruções condicionais
7. Instruções iterativas
8. Funções
9. Variáveis de sessão
10. Ligação a base de dados MySQL

Vantagens e desvantagens

Desvantagens

- É uma linguagem interpretada
 - Mais lenta que C ou C++
- Não é uma linguagem nativamente orientada a objetos

Vantagens

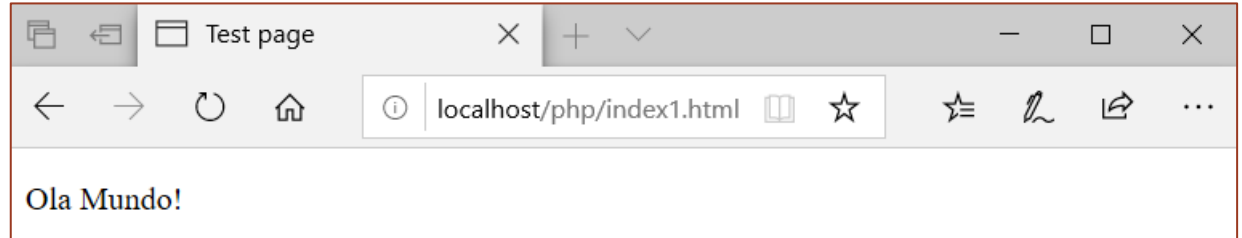
- Permite desenvolvimento rápido de aplicações
 - Como é uma linguagem interpretada, não necessita de compilação e ligação
- As aplicações são estáveis e não dependem do browser utilizado
 - Permite também escolha da plataforma servidora
- Tem ligação aos sistemas de bases de dados mais utilizados
- Tem suporte parcial para programação orientada a objetos

https://www.w3schools.com/php/php_oop_classes_objects.asp

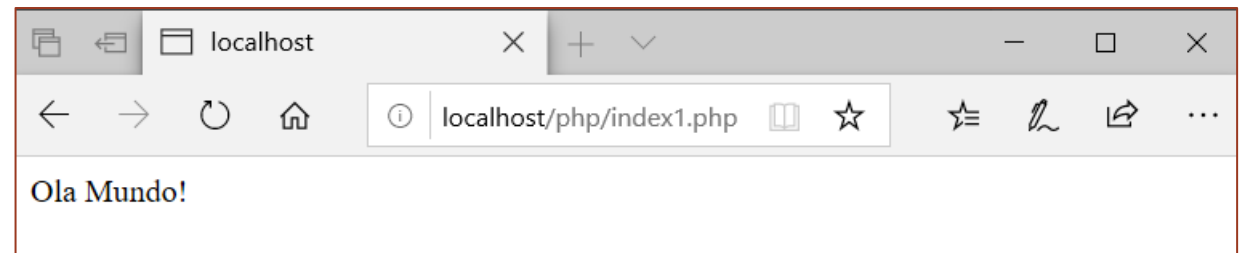
- Corre em muitas plataformas e é mais potente que ASP ou JSP
- Tem uma grande base de programadores e soluções já implementadas

Duas formas de utilizar

```
index1.html x
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title>Test page</title>
6  </head>
7  <body>
8      <p>
9          <?php echo "Ola Mundo!"; ?>
10     </p>
11 </body>
12 </html>
```



```
index1.php x
1  <?php
2      echo "Ola Mundo!";
3  ?>
```



Tipos de variáveis

Grupo	Tipo	Exemplo	Conversão
Escalares	boolean	<code>\$b1 = True;</code> <code>\$b2 = False;</code>	<code>\$b3 = (bool)\$i1; // True</code>
	integer	<code>\$i1 = 123;</code> <code>\$i2 = -123;</code> <code>\$i3 = 0123; // octal (83)</code> <code>\$i4 = 0x1A; // hexadecimal (26)</code>	<code>\$i5 = (int)\$b1; // 1</code> <code>\$i6 = (int)\$b2; // 0</code> <code>\$i7 = (int)\$f1; // 1</code> <code>\$i8 = (int)\$f2; // 1</code> <code>\$i9 = 1+ "123" // 124</code>
	float	<code>\$f1 = 1.234;</code> <code>\$f2 = 1.734;</code> <code>\$f3 = 1.2e3; // 1200</code> <code>\$f4 = 7E-10; //</code> <code>0.0000000007</code>	<code>\$f5 = 1 + "1.23" // 2.23</code>
	string	<code>\$s1 = 'ab\nc'; // ab\nc</code> <code>\$s2 = "ab\nc"; // ab</code> <code>// c</code> <code>\$s3 = <<<FIM // ab</code> <code>ab // c</code> <code>c</code> <code>FIM;</code>	<code>\$s4 = "Preco: \$i1"</code> <code>// Preco: 123</code> <code>\$s5 = \$s1{0} // a</code> <code>\$s6 = \$s1{1} // b</code>

Valores considerados FALSE:

- Booleano FALSE
- Inteiro 0
- Real 0.0
- String vazia ""
- String "0"
- Vetor com 0 elementos
- Tipo especial NULL

Carateres especiais:

- \n: nova linha
- \r: retorno
- \t: tabulação
- \\: barra invertida
- \\$: cifrão
- \": aspa
- \[0-7]{1,3}: um octal
- \x[0-9A-Fa-f]{1,2}: um hexadecimal

Tipos de variáveis

Grupo	Tipo	Exemplo
Compostos	array	<pre>\$a = array {1 => 'um', 2=> 'dois', 3 => 'tres'}; \$a[3] = 30; // array {1 => 'um', 2=> 'dois', 3 => 30} \$a[] = 'quinze'; // array {1 => 'um', 2=> 'dois', 3 => 'tres', 4 => 'quinze'} unset(\$a[2]); // array {1 => 'um', 3 => 'tres' , 4 => 'quinze'}</pre> <pre>\$b = array {'cor' => 'azul', 'forma' => 'quadrada'} \$b['cor'] = 'azul'; \$b['forma'] = 'quadrada';</pre> <pre>\$c = array{ 'cores' => array {'a'=>'azul','b'=>'amarelo','c'=>'branco'}, 'numeros' => array (1,2,3,4) }</pre>
	object	<pre>\$o = new teste(); \$o->testar();</pre>
Especiais	resource	<pre>\$ligacao = mysql_connect(\$server,\$user,\$password);</pre>
	NULL	<pre>\$n = NULL // variável sem valor</pre>

Variáveis

- Nome
 - Começa sempre por um \$.
 - Depois é obrigatório ter uma letra ou um _
 - Restantes podem ser letras ou números
- Referência (&)


```
$a = 'Ana';
$b = &$a; // $b aponta para $a
$b = 'Joana'; // altera também $a
```
- Âmbito
 - Globais (a seguir)
 - Estáticas: mantêm o seu valor mesmo depois de sair do âmbito
- Variáveis de variáveis


```
$a = 'Ola';
$$a = 'Mundo';
echo "$a ${$a}";
echo "$a $Ola";
```

- Variáveis vetoriais predefinidas

Vetor	Descrição
\$GLOBALS	Referencia para cada variável global
\$_SERVER	Variáveis da responsabilidade do servidor (IP, porta)
\$_GET	Variáveis obtidas por GET
\$_POST	Variáveis obtidas por POST
\$_COOKIE	Variáveis obtidas por <i>cookies</i>
\$_FILES	Variáveis respeitantes ao carregamento de ficheiros
\$_ENV	Variáveis de ambiente
\$_REQUEST	Variáveis fornecidas pelo utilizador
\$_SESSION	Variáveis associadas à sessão

Âmbito das variáveis

Variáveis globais

- O âmbito estende-se aos ficheiros incluídos

```
$a = 2;  
include "fich.php"; // $a está disponível em fich.php. fich.php pode emitir warnings  
// require tem comportamento semelhante. fich.php pode emitir erros fatais  
// também existe include_once e require_once (para evitar repetições)
```

- Não se estende pelas funções

```
$b = 3;  
function func(){  
    echo $b; // referência a uma variável local (indefinida)  
}
```

- A não ser que se defina explicitamente

```
$a = 3;  
$b = 2;  
function mult(){  
    global $a, $b;  
    $b = $a*$b;  
}  
mult();  
echo $b; // 6
```


Variáveis externas (formulários com GET e POST)

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <title>Formulario</title>
6 </head>
7 <body>
8 <form action="index2.php" method="post">
9 <div>
10 <div>Nome: <input type="text" name="nome"></div>
11 <div>Email: <input type="email" name="email"></div>
12 <div><input type="submit" value="Enviar"></div>
13 </div>
14 </form>
15 </body>
16 </html>

```

```

1 <?php
2 echo "Nome introduzido: " . $_POST['nome'] . "<br>";
3 echo "Email introduzido: " . $_POST['email'];
4 ?>

```

Formulario

localhost/php/index2.html

Nome: abc

Email: abc@de.fg

Enviar

localhost

localhost/php/index2.php

Nome introduzido: abc

Email introduzido: abc@de.fg

```
setcookie("meucookie", "teste", time()+3600);
```

```
// define uma cookie chamada meucookie com o valor "teste" e válida por 1h (3600s)
```

```
echo $_COOKIE['meucookie'];
```

```
// acede à cookie
```

Operadores Aritméticos

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

Fonte: https://www.w3schools.com/php/php_operators.asp

Operadores de atribuição

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

Fonte: https://www.w3schools.com/php/php_operators.asp

Operadores de comparação

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

Fonte: https://www.w3schools.com/php/php_operators.asp

Operadores de incremento / decremento

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

Fonte: https://www.w3schools.com/php/php_operators.asp

Operadores lógicos

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

Fonte: https://www.w3schools.com/php/php_operators.asp

Operadores de strings

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

Fonte: https://www.w3schools.com/php/php_operators.asp

Operadores de arrays

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>

Fonte: https://www.w3schools.com/php/php_operators.asp

Operadores condicionais

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

Fonte: https://www.w3schools.com/php/php_operators.asp

Instruções condicionais

```
if($a == ) {  
    print "nenhum";  
} elseif ($a == 1) {  
    print "um";  
} else {  
    print "muitos";  
}
```

```
switch($a) {  
    case 0:  
        print "nenhum";  
        break;  
    case 1:  
        print "um";  
        break;  
    default:  
        print "muitos";  
}
```

Instruções iterativas

```
$i = 0;
while($i < 10){
    print $i;
    i++;
}
```

```
$i = 0;
do {
    print $i;
    i++;
} while($i < 10)
```

```
for($i=0; $i<10; $i++){
    print $i;
}
```

```
$a = array (1,2,3,4);
foreach($a as $v){
    print $v;
}
```

```
$i = 0;
while(++$i){
    switch($i){
        case 3:
            print "tres";
            break 1; // sai só do switch
        case 6:
            print "seis";
            break 2; // sai do switch e do
while
        default:
            break;
    }
}
```

```
$a = array (1,2,3,4);
foreach($a as $v){
    if(($v % 2)==0) {
        continue;
    }
    impares($v);
}
```

Funções

```
function cinco(){  
    return 5;  
}  
$x = cinco(); // $x fica com o valor 5
```

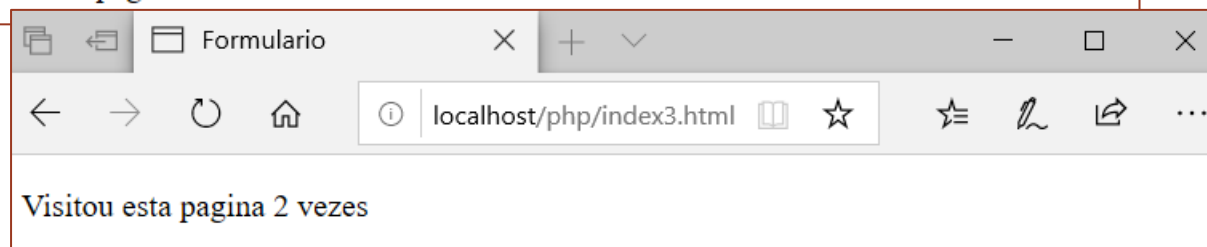
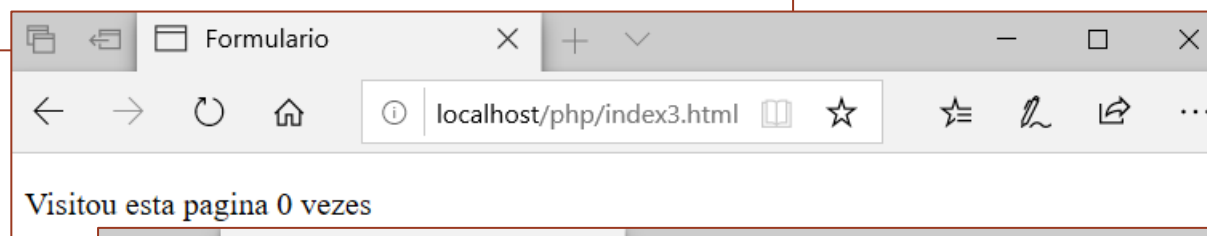
```
function soma($p, $s = 1){ // argumentos por omissão  
    return $p + $s;  
}  
$a = soma(2,3); // 5  
$b = soma(2); // 2
```

```
function conc(&$var){ // argumentos por referência  
    $var .= ' extra';  
}  
$h = 'horas';  
conc($h);  
echo $h; // 'horas extra'
```

```
function vec(){  
    return array (1,2,3);  
}  
$y = vec(); // $x = array (1,2,3)
```

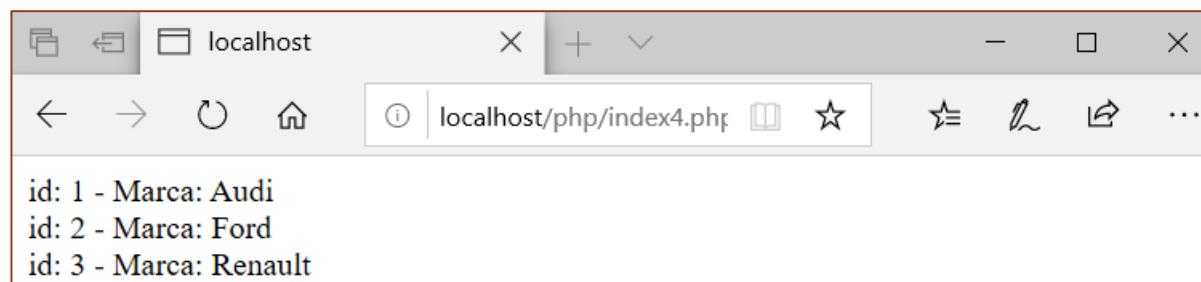
Variáveis de sessão

```
index3.html x
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title>Formulario</title>
6   </head>
7   <body>
8     <?php
9       session_start();
10      if(!isset($_SESSION['contador'])) {
11        $_SESSION['contador'] = 0;
12      } else {
13        $_SESSION['contador']++;
14      }
15    ?>
16    <p>Visitou esta pagina <?php echo $_SESSION['contador'];?> vezes</p>
17  </body>
18 </html>
```



Ligação a base de dados MySQL

```
index4.php x
1 <?php
2 $servidor = "localhost";
3 $user = "root";
4
5 $ligacao = mysqli_connect($servidor,$user) or die("Sem ligação");
6 mysqli_select_db($ligacao, "carros") or die("Sem DB");
7 $consulta = "SELECT * FROM marca";
8 $resultado = mysqli_query($ligacao,$consulta);
9 if($resultado->num_rows > 0){
10     while($row = $resultado->fetch_assoc()){
11         echo "id: " . $row["mc_id"] . " - Marca: " . $row["mc_marca"] . "<br>";
12     }
13 } else {
14     echo "0 results";
15 }
16 ?>
```





UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.