

PHP/MYSQL
Semana Acadêmica de TI
Estácio JF - 2019
Prof. Evandro Bastos




Apresentação disponível em evandrobs.com/phpestacio19



Estácio

Resumo

- Conceitos básicos
 - Variáveis e tipos de dados
 - Arrays
 - Objetos
 - Operadores
 - Estruturas de Controle
 - Funções
- 
- A decorative graphic on the right side of the slide, consisting of several overlapping, curved, wavy shapes in shades of light blue, yellow, and a darker blue at the bottom right corner.

Conceitos Básicos

- Site oficial do php: www.php.net - Exemplos de códigos utilizados nessa apresentação!!!
- Versão atual: 7.4



- Site oficial do MySql: www.mysql.com
- Versão atual: 8.0

Conceitos Básicos

- O PHP (um acrônimo recursivo para *PHP: Hypertext Preprocessor*) é uma **linguagem** de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser **embutida** dentro do HTML. (php.net acesso em nov/19)
- No PHP o código é executado no **servidor (server side)**, gerando o HTML que é então enviado para o navegador. O navegador recebe os resultados da execução desse script, mas não sabe qual era o código fonte.(php.net acesso em nov/19)

Conceitos Básicos

- Simplicidade
- Alta disponibilidade - Para executar, só precisa de um navegador!!!
- Compatibilidade com SO (Linux, Mac, Windows e outros)
- Pode ser executado direto na linha de comando - Útil para agendamento de tarefas
- Utilizado desde sites pequenos até sistemas altamente distribuídos.

Conceitos Básicos

- PHP-GTK - Par aplicações *desktop*
- Programação estruturada e orientada a objeto
- Suporte a diversos bancos de dados (vamos abordar o MySql)
- Comunicação a diversos protocolos (IMAP, POP3, HTTP, etc...)
- Diversas funções para trabalho com PDF, XML e outros formatos.

Leia mais em: www.php.net

Conceitos Básicos

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Exemplo</title>
  </head>
  <body>

    <?php
      echo "Olá, eu sou um script PHP!";
    ?>

  </body>
</html>
```

Conceitos Básicos - Estruturação

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <head>
```

```
    <title>Exemplo</title>
```

```
  </head>
```

```
  <body>
```

```
    <?php
```

```
      echo "Olá, eu sou um script PHP!";
```

```
    ?>
```

```
  </body>
```

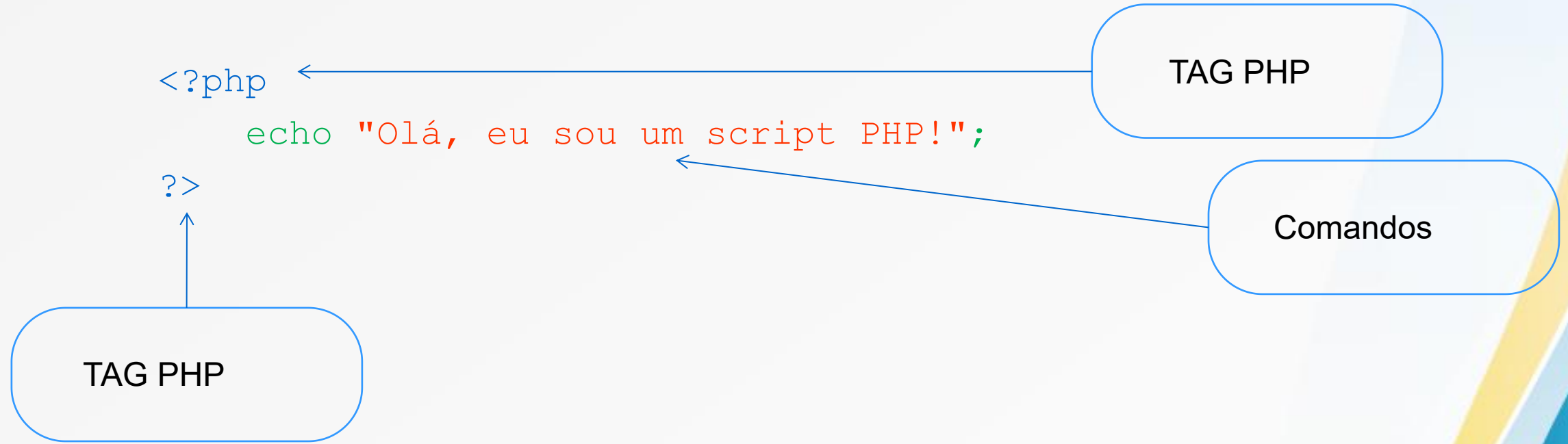
```
</html>
```

Estrutura HTML

Código PHP

Estrutura HTML

Conceitos Básicos - Estruturação



Conceitos Básicos - Comentários

```
<?php
```

```
    echo 'Isto é um teste'; // Estilo de  
comentário de uma linha em c++
```

```
    /* Este é um comentário de múltiplas  
linhas
```

```
        ainda outra linha de comentário */
```

```
    echo 'Isto é ainda outro teste';
```

```
    echo 'Um teste final'; # Este é um  
comentário de uma linha no estilo shell
```

```
?>
```

Variáveis

- Escalares: boolean, integer, float (e double), string
- Compostos: array, object, callable, iterable
- Especiais: resource, *NULL*

No PHP o tipo de uma variável geralmente não é definido pelo programador: isto é **decidido em tempo de execução** pelo PHP, dependendo do contexto no qual a variável é usada.

Variáveis e tipos de dados

```
<?php
```

```
    $nome="Evandro Bastos";// String
```

```
    $nota=9.5;// float
```

```
    $idade=31;// integer
```

```
    echo "O aluno $nome com idade $idade  
anos obteve nota $nota";
```

```
?>
```

As regras de atribuição aos nomes das variáveis seguem as principais regras de outras linguagens!

Os nomes de variáveis não são *case sensitive*!!!!

Variáveis e tipos de dados

```
<?php
    echo $_GET['nome'];
    echo $_POST['nome'];
?>
```

Tipos de Dados

```
<?php
$a_bool = TRUE;    // um booleano
$a_str  = "foo";    // uma string
$a_str2 = 'foo';    // uma string
$an_int = 12;       // um inteiro

var_dump($a_str);

?>
```

Como a tipagem do php muda com a atribuição,
recomenda-se muita atenção durante esse processo!!!!

Arrays

```
<?php
$array = array(
    "foo" => "bar",
    "bar" => "foo",
);
```

```
// a partir do PHP 5.4
$array = [
    "foo" => "bar",
    "bar" => "foo",
];
```

```
<?php
$array["foo"] = "bar";
$array[1] = 100;
$array[1]["foo"]
= "3.4";

$nomes=
["Evandro","Galba","Jos
é"]; // Posições 0,1 e 2
```

Objetos

```
<?php
```

```
class Cliente{
```

```
    public $nome; //Atributo
```

```
    public function novoNome ($nome) { //método
```

```
        $this->$nome=$nome;
```

```
    }
```

```
}
```

```
$cliente=new Cliente();
```

```
$cliente->novoNome ("Evandro");
```


Operadores - Concatenação de Strings

```
<?php
```

```
$nome='Evandro Bastos';
```

```
$hoje=date("d/m/Y");
```

```
echo "Hoje é $hoje e meu nome é $nome";
```

```
echo 'Hoje é $hoje e meu nome é $nome';
```

```
//As saidas serão diferentes
```

```
echo 'Hoje é '.$hoje.'e meu nome é '.$nome;
```

Operadores Aritiméticos

```
<?php
```

```
$nota["Pedro"]=7.8;
```

```
$nota["Evandro"]=9.8;
```

```
$nota["Galba"]=8.5;
```

```
$quantasNotas=3;
```

```
$somaNotas=$nota["Pedro"]+$nota["Evandro"]+$  
nota["Galba"];
```

```
$mediaNotas=$somaNotas/$quantasNotas;
```

Operadores Aritméticos

```
<?php
```

```
$nota["Evandro"]["P1"]=9.8;
```

```
$nota["Evandro"]["P2"]=4.8;
```

```
$peso["P1"]=1;
```

```
$peso["P2"]=2;
```

```
$somaPesos=$peso["P1"]+$peso["P2"];
```

```
$notaFinal=($nota["Evandro"]["P1"]*$peso["P1"]+$nota["Evandro"]["P2"]*$peso["P2"])/$somaPesos;
```

```
//Ainda há o operador exponencial **
```

```
//Módulo %
```

Operadores - Incrementos e Atribuição

```
<?php
```

```
$numero=1;
```

```
$numero++; // $numero agora é 2
```

```
$numero--; // $numero agora é 1
```

```
$numero+=2; // $numero agora é 3
```

```
$numero*=2; // $numero agora é 6
```

```
$nome="Evandro";
```

```
$nome.="Bastos"; // $nome agora é EvandroBastos
```

Operadores de Comparação

```
<?php
```

```
if (1=='1') { //Igualdade
```

```
echo "São iguais";
```

```
}else{
```

```
echo "Não são iguais";
```

```
}
```

```
if (1<'1') { //Menor que
```

```
echo "Menor";
```

```
}else{
```

```
echo "Maior";
```

```
}
```

```
<?php
```

```
if (1==='1') { //Identidade
```

```
echo "São idênticos";
```

```
}else{
```

```
echo "Não são idênticos";
```

```
}
```

```
<?php
```

```
if (1!== '1') { //Identidade
```

```
echo "São idênticos";
```

```
}else{
```

```
echo "Não são idênticos";
```

```
}
```

Operadores de Comparação

Exemplo	Nome	Resultado
<code>\$a == \$b</code>	Igual	Verdadeiro (TRUE) se <code>\$a</code> é igual a <code>\$b</code> .
<code>\$a === \$b</code>	Idêntico	Verdadeiro (TRUE) se <code>\$a</code> é igual a <code>\$b</code> , e eles são do mesmo tipo.
<code>\$a != \$b</code>	Diferente	Verdadeiro se <code>\$a</code> não é igual a <code>\$b</code> .
<code>\$a <> \$b</code>	Diferente	Verdadeiro se <code>\$a</code> não é igual a <code>\$b</code> .
<code>\$a !== \$b</code>	Não idêntico	Verdadeiro se <code>\$a</code> não é igual a <code>\$b</code> , ou eles não são do mesmo tipo (introduzido no PHP4).
<code>\$a < \$b</code>	Menor que	Verdadeiro se <code>\$a</code> é estritamente menor que <code>\$b</code> .
<code>\$a > \$b</code>	Maior que	Verdadeiro se <code>\$a</code> é estritamente maior que <code>\$b</code> .
<code>\$a <= \$b</code>	Menor ou igual	Verdadeiro se <code>\$a</code> é menor ou igual a <code>\$b</code> .
<code>\$a >= \$b</code>	Maior ou igual	Verdadeiro se <code>\$a</code> é maior ou igual a <code>\$b</code> .
<code>\$a <=> \$b</code>	Spaceship (nave espacial)	Um integer menor que, igual a ou maior que zero quando <code>\$a</code> é, respectivamente, menor que, igual a ou maior que <code>\$b</code> . Disponível a partir do PHP 7.

Operadores Lógicos e Estruturas Condicionais

```
<?php
```

```
$nome="Evandro"; $nota=9.8;
```

```
if ($nome=="Evandro" and $nota==9.8) { //equivalente &&
```

```
echo "Verdadeiro";
```

```
}else{
```

```
echo "Falso";
```

```
}
```

```
if ($nome=="Evandro" or $nota>=7) { //equivalente ||
```

```
echo "Verdadeiro";
```

```
}else{
```

```
echo "Falso";
```

```
}
```

Operadores Lógicos e Estruturas Condicionais

```
<?php
```

```
$b=1;
```

```
$a=3;
```

```
if ($a > $b) {  
    echo "$a é maior que $b";  
} elseif ($a == $b) {  
    echo "$a é igual a $b";  
} else {  
    echo "$a é menor que $b";  
}
```

```
?>
```


Operadores Lógicos e Estruturas Condicionais

```
<?php
```

```
$a=3;
```

```
if ($a) {
```

```
    echo "$a é o valor de a";
```

```
} else {
```

```
    echo "a não tem valor atribuído";
```

```
}
```

```
?>
```

Operadores Lógicos e Estruturas Condicionais

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.

Repetição - while

```
<?php
$i = 1;
while ($i <= 10) {
    echo $i++;
}
?>
```

```
<?php
$i = 1;
while ($i <= 10) :
    echo $i;
    $i++;
endwhile;
?>
```

Repetição - for

```
<?php  
for ($i = 1; $i <= 10; $i++) {  
    echo $i;  
}  
?>
```

Há outras formas equivalentes, mas não serão abordadas nesse minicurso.

Repetição - foreach (para cada)

```
<?php
$arr = array(1, 2, 3, 4);
foreach ($arr as &$novoValor) {
    $novoValor = $novoValor* 2;
} //será atribuído a $novoValor um array com duas
vezes o valor de cada elemento de $arr nas mesmas
posições (2, 4, 6, 8);
?>
```

Repetição - foreach (para cada)

```
<?php
$cores = array('preto', 'azul', 'verde', 'rosa');
foreach ($cores as $chave => $novoValor) {
    // $novoValor será atualizado com cada valor de
    $cores...
    echo "{$chave} => {$novoValor}<br>";
}
?>
```

Outras estruturas de controle (bastante utilizadas)

`break`

`continue`

`switch`

`return`

`require`

`include`

`require_once`

`include_once`

Funções - Definidas pelo Usuário

```
<?php
```

```
function soma($a,$b) {  
    $valorSoma=$a+$b;  
    return $valorSoma;  
}
```

```
echo soma(1,2);
```