

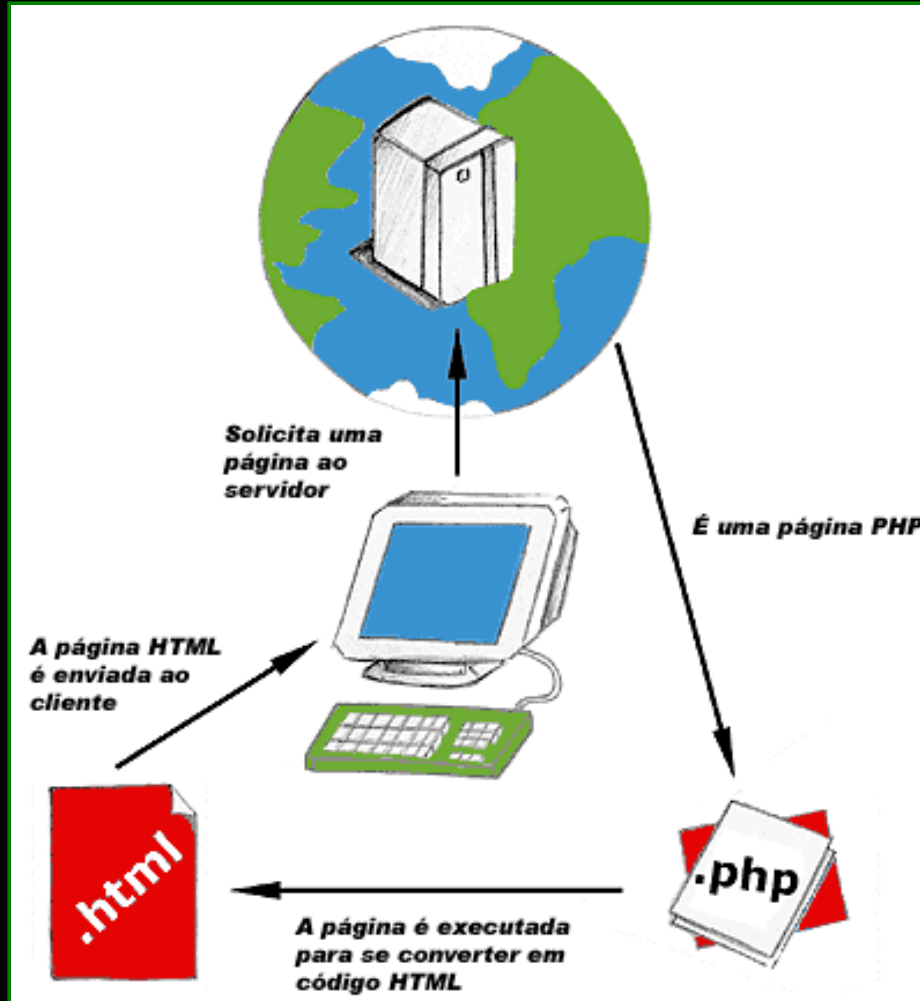
Desenvolvimento Web com



Introdução

- PHP: Hypertext Preprocessor
- Desenvolvimento de aplicações WEB
- Desenvolvimento de aplicações Desktop
 - PHP-GTK
- Linguagem interpretada
- Frameworks: CakePHP , XAJAX

Como funciona ?



Estrutura do código

```
<?  
    // código php  
?>
```

```
<html>  
<body>  
<?  
    // código php  
?>  
</html>  
</body>
```

Estrutura do código

```
<html>  
<body>  
<?  
    // código php  
>  
<center> Alguma coisa escrita</center>  
<?  
    // código php  
>  
</html>  
</body>
```

Variáveis

- Começam com \$
 - \$string = "Daniel"; ou \$string='Daniel';
 - \$inteiro = 1;
 - \$boolean = true;
 - \$double = 2,312
- Case sensitive
- Concatenação de variáveis
 - \$concatenado = \$string.\$inteiro.\$double

Constantes

- `define(string $name, mixed $value [, bool $case_sensitive]);`

```
<?  
define("CONSTANTE1", "Esta é uma constante");  
define("CONSTANTE2", 2008, false);  
define("CONSTANTE3", 2,32);  
define("CONSTANTE4", false);  
  
echo CONSTANTE1."\\n";  
echo constante2;  
?>
```

```
Esta é uma constante  
2008
```

Array

```
<?
$nomes = array ('Daniel', 'Karina', 'José');
$nomes[] = "Livia";

$idades = array ( "Daniel" => 22,
                  "Karina" => 29,
                  "José" => 54);
$idades["Livia"] = 21;

$caracteristicas = array ( "Daniel" => array ( "Altura" => '1,82',
                                                "Olhos" => "Castanhos"),
                           "Karina" => array ( "Altura" => '1,65',
                                                "Olhos" => "Castanhos"),
                           "José" => array ( "Altura" => '1,72',
                                                "Olhos" => "Verdes")
                           );

print_r($nomes);
print_r($idades);
print_r($caracteristicas);
?>
```

Array

Array

```
(  
  [0] => Daniel  
  [1] => Karina  
  [2] => José  
  [3] => Livia  
)
```

Array

```
(  
  [Daniel] => 22  
  [Karina] => 29  
  [José] => 54  
  [Livia] => 21  
)
```

Array

```
Array
(  
  [Daniel] => Array  
    (  
      [Altura] => 1,82  
      [Olhos] => Castanhos  
    )  
  
  [Karina] => Array  
    (  
      [Altura] => 1,65  
      [Olhos] => Castanhos  
    )  
  
  [José] => Array  
    (  
      [Altura] => 1,72  
      [Olhos] => Verdes  
    )  
)
```

Array

```
<?
$idades = array ( "Daniel" => 22,
                  "Karina" => 29,
                  "José" => 54);
//echo "A idade do Daniel é $idades['Daniel']"; // Erro de sintaxe
echo "A idade do Daniel é {$idades['Daniel']}\n";

$quantidade = count($idades);
echo "O array tem $quantidade itens";
?>
```

```
A idade do Daniel é 22
O array tem 3 itens
```

Operadores

- Operação

- `$variavel = 1;` → recebe o valor de
- `$variavel ++;` → incrementa em 1 a variável
- `$variavel--;` → decrementa em 1 a variável
- `$variavel += 2;` → soma 2 ao valor da variável
- `$variavel -= 2;` → subtrai 2 ao valor da variável
- `$variavel .= " é o numero da sorte";` → concatena o texto à variável
- `$variavel = 3%2` → A variável recebe o resto da divisão de 3 por 2

Operadores

- Comparação
 - == → igualdade
 - != → desigualdade
 - >, <, >=, <= → maior que, menor que, maior ou igual a, menor ou igual a
- Lógicos
 - ! → negação
 - && ou and → E
 - || ou or → OU

Estruturas de controle

```
<?  
  // Exemplo de if  
  if ($nome == "Daniel") {  
    echo "Olá $nome";  
  } else if ($nome == "Karina") {  
    echo "$nome, seu acesso é restrito.";  
  } else {  
    echo "Você não tem permissão nenhuma";  
  }  
>
```

Estruturas de controle

```
<?  
  // Exemplo de switch  
  switch ($nome) {  
    case "Daniel":  
      echo "Seja bem vindo!";  
      break;  
    case "Karina":  
      echo "Você tem acesso restrito";  
      break;  
    default:  
      echo "Você não tem acesso";  
      break;  
  }  
?>
```

Estruturas de controle

```
<?  
  $numero = 1;  
  while( $numero < 5 ) {  
    echo $numero."\\n";  
    $numero++;  
  }  
?>
```

```
1  
2  
3  
4
```

Estruturas de controle

```
<?  
$numero = 5;  
do {  
    echo $numero."\\n";  
    $numero++;  
} while( $numero < 5 );  
?>
```

Estruturas de controle

```
<?  
  for ($i = 1; $i < 5; $i++) {  
    echo $i."\\n";  
  }  
?>
```

```
1  
2  
3  
4
```

Estruturas de controle

```
<?
  $idades = array( "Daniel" => 22,
                  "Karina" => 29,
                  "Livia" => 21);
  foreach ($idades as $chave => $valor) {
    echo "A idade do(a) $chave é $valor \n";
  }
?>
```

```
A idade do(a) Daniel é 22
A idade do(a) Karina é 29
A idade do(a) Livia é 21
```

Exercício 1

- Faça um programa que construa um vetor de 6 elementos numéricos aleatórios inteiros e mostre:
 - Quantidade e quais são os números pares
 - Quantidade e quais são os números ímpares

Dica: Use a função `rand([int $minimo, int $maximo])` para retornar um número aleatório.

String

```
<?  
$nome = "Daniel";  
$nota = 10;  
  
$texto = "A nota do ".$nome." é igual a ".$nota."\n";  
$texto2 = "A nota do $nome é igual a $nota\n";  
$texto3 = 'A nota do $nome é igual a $nota\n';  
  
echo $texto;  
echo $texto2;  
echo $texto3;  
?>
```

```
A nota do Daniel é igual a 10  
A nota do Daniel é igual a 10  
A nota do $nome é igual a $nota\n
```

String

- `trim(string $str [, string $charlist]);`
 - “ ” – espaço normal
 - “\t” – tabulação
 - “\n” – quebra de linha
 - “\r” – retorno de carro
 - “\0” – byte NULL
 - “\X0B”, tabulação vertical

String

- `trim(string $str [, string $charlist]);`

```
<?
$texto = "      Ola, este é um texto de teste ...      ";
$resultado1 = trim($texto);
$resultado2 = trim($texto, '\t. ');

echo $resultado1."\\n".$resultado2;
?>
```

```
Ola, este é um texto de teste ...
Ola, este é um texto de teste
```

String

- `strpos(string $texto, mixed $parte [, int $offset]);`

```
<?
  $texto = "abcdefg abcdefg";
  $posicao1 = strpos($texto,"a");
  $posicao2 = strpos($texto,"a",1);

  echo $posicao1."\n";
  echo $posicao2;
?>
```

```
0
8
```

String

- `substr(string $str, int $start [,int $length]);`

```
<?  
$texto = "Meu nome é Daniel";  
$trecho1 = substr($texto, 0);  
$trecho2 = substr($texto,4);  
$trecho3 = substr($texto,4, strpos($texto, "é"));  
  
echo $trecho1. "\n";  
echo $trecho2. "\n";  
echo $trecho3;  
?>
```

```
Meu nome é Daniel  
nome é Daniel  
nome é Da
```

String

- `strlen(string $texto);`

```
<?  
  $texto = "Meu nome é Daniel";  
  $tamanho = strlen($texto);  
  echo $tamanho;  
?>
```

String

- `str_replace(mixed $search, mixed $replace, mixed $subject [, int $count]);`

```
<?
  $texto = "Meu nome é Daniel e o da minha mãe é Regina";
  $textoAlterado = str_replace("Meu", "Seu", $texto);
  $textoAlterado2 = str_replace("é", "não é", $texto, $qtde);

  echo $textoAlterado."
";
  echo $textoAlterado2."
";
  echo "Foram alterado $qtde vezes";
?>
```

```
Seu nome é Daniel e o da minha mãe é Regina
Meu nome não é Daniel e o da minha mãe não é Regina
Foram alterado 2 vezes
```

String

- `explode(string $delimiter, string $string [,int $limit]);`
- `list (mixed $varname [, mixed $...]);`

```
<?  
$texto = "Daniel Antonio Fernandes Bojczuk";  
$array1 = explode(" ", $texto);  
list($primeiroNome, $resto) = explode(" ", $texto, 2);  
  
print_r($array1);  
echo "\n". $primeiroNome. "\n";  
echo $resto;  
  
?>
```

String

- `explode(string $delimiter, string $string [,int $limit]);`
- `list (mixed $varname [, mixed $...]);`

Array

```
(  
    [0] => Daniel  
    [1] => Antonio  
    [2] => Fernandes  
    [3] => Bojczuk  
)
```

Daniel

Antonio Fernandes Bojczuk

String

- `implode(string $glue, array $pieces);`

```
<?  
  $pedacos = array ("Daniel", "Antonio", "Fernandes", "Bojczuk");  
  
  $nome = implode (" ", $pedacos);  
  
  echo $nome;  
?>
```

Daniel Antonio Fernandes Bojczuk

Exercício 2

- Faça um script que inverta a ordem das palavras de uma frase.

Dica: Utiliza a função `rsort($array)` para inverter a ordem dos elementos de um array.

Exercício 3

- Faça um script que conte o número de vogais (maiúsculas ou minúsculas) existentes em uma frase. Desconsidere os acentos.

Dica: Use a função `in_array($elemento, $array)` para procurar um elemento em um array.

Dica: Use `strtoupper($string)` caso precise transformar uma string para letras maiúsculas.

Formulários com PHP

- POST
 - Envio pelo cabeçalho da página

```
<form action="formulario.php" method="POST">
```

- GET
 - Envio pela URL da página

```
<form action="formulario.php" method="GET">
```

Formulários com PHP

```
<html>
<body>
<form action="formulario.php" method="POST">
Nome: <input type="text" name="usuario" size="20" maxlength="50"><br>
Senha:<input type="password" name="senha" size="8" maxlength="8"><br>
<input type="submit" value="Entrar">
</body>
</html>
```

```
<?
    print_r($_POST);

    echo "O nome do usuário é {$_POST['usuario']}\n";
    echo "A senha é {$_POST['senha']}";

?>
```

Formulários com PHP

```
<?
  if (empty($_POST)) {
?>
<html>
<body>
<form action="<?=htmlentities($_SERVER['PHP_SELF']) ?>" method="POST">
Nome: <input type="text" name="usuario" size="20" maxlength="50"><br>
Senha:<input type="password" name="senha" size="8" maxlength="8"><br>
<input type="submit" value="Entrar">
</body>
</html>
<?
  } else {
      print_r($_POST);

      echo "O nome do usuário é {$_POST['usuario']}\n";
      echo "A senha é {$_POST['senha']}";
  }
?>
```

Exercício 4

- Valide um formulário de usuário e senha da seguinte forma:
 - Username: Não pode estar em branco
 - Senha: Deve ter no mínimo 4 caracteres, tendo no mínimo um dos caracteres abaixo:
! @ # \$ % & *

Caso não satisfaça a condição, exiba uma mensagem de erro.

Variáveis de Sessão

- Iniciar sessão – gera um SID da máquina cliente.
 - `session_start();`
- Estabelecer e usar variáveis de sessão
 - `$_SESSION['nome'] = “Daniel”;`
 - `echo $_SESSION['nome'];`
- Encerrar a sessão.
 - `session_destroy();`

Conexão com Banco de Dados

Orientação a objetos com PHP