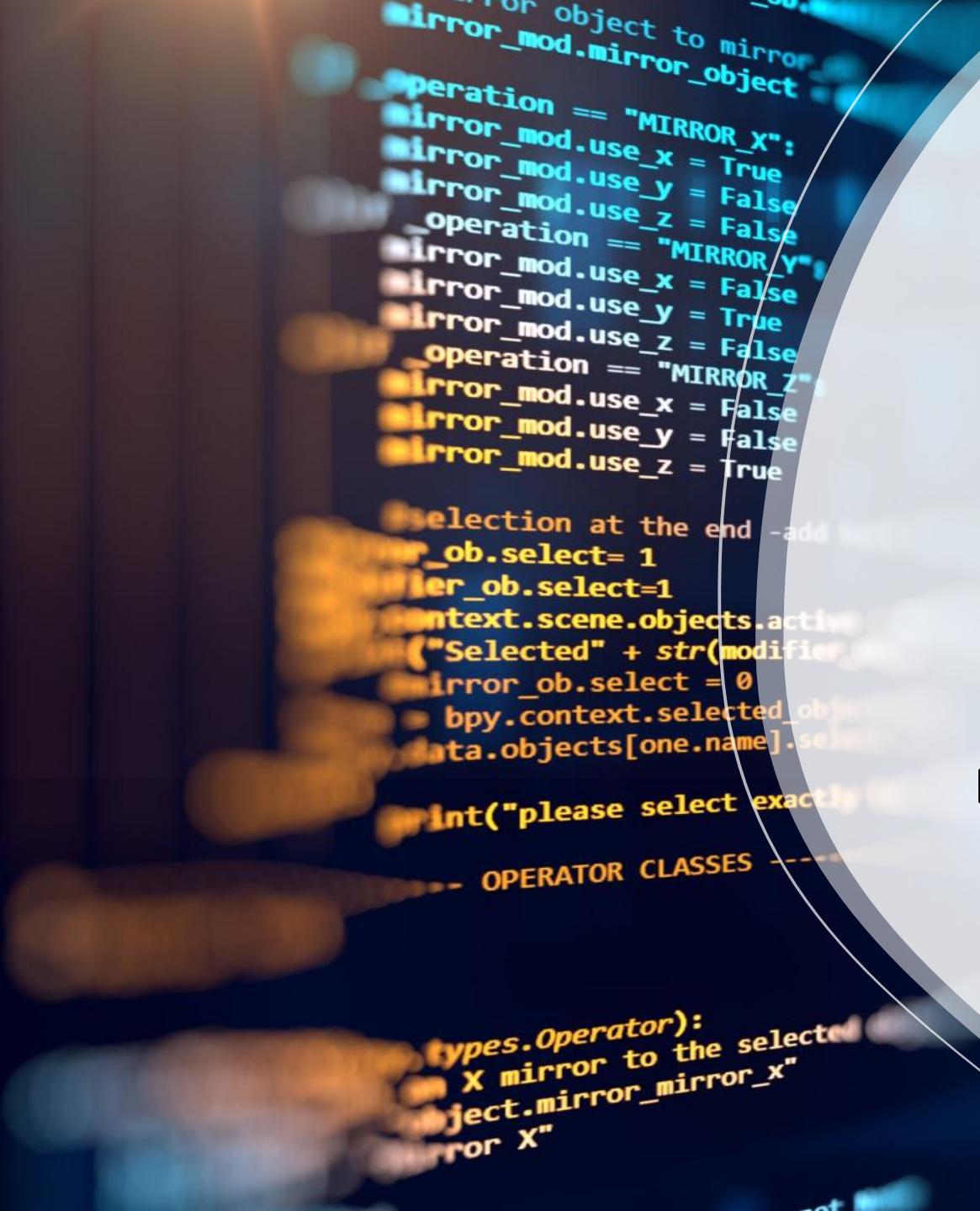


Introdução ao PHP - Documentos dinâmicos na web

Prof. Me. Hélio Esperidião

A close-up photograph of a person's hand with a ring on their ring finger, pointing its index finger towards a computer monitor. The monitor displays a dark-themed code editor with yellow and white text. The visible code includes conditional statements like 'if' and 'else if', and function definitions. The background is dark, making the bright screen stand out.

```
    types.Operator):
        X mirror to the selected object.mirror_mirror_x"
        "mirror X"
    else if(operation == "MIRROR_X"):
        mirror_mod.use_x = True
        mirror_mod.use_y = False
        mirror_mod.use_z = False
    else if(operation == "MIRROR_Y"):
        mirror_mod.use_x = False
        mirror_mod.use_y = True
        mirror_mod.use_z = False
    else if(operation == "MIRROR_Z"):
        mirror_mod.use_x = False
        mirror_mod.use_y = False
        mirror_mod.use_z = True

    selection at the end -add
    ob.select= 1
    mirr_ob.select=1
    context.scene.objects.active
    ("Selected" + str(modifier))
    mirror_ob.select = 0
    bpy.context.selected_objects
    data.objects[one.name].sele
int("please select exactly one object")
- OPERATOR CLASSES ---
```

O PHP

PHP é uma sigla recursiva que significa *PHP HyperText Preprocessor*.

O PHP é uma linguagem de código-fonte aberto, muito utilizada na Internet e especialmente criada para o desenvolvimento de aplicativos Web

Características do PHP

É Executado em um servidor web, não no navegador.

Tem pouca relação com layout

O resultado da execução do php pode ser um código html que é interpretado por um navegador.

Características do PHP

Permite criar páginas dinâmicas

Manipulação de banco de dados

Manipulação de arquivos

Manipulação de *cookies*.

Sintaxe parecida com C

Vantagens

É uma linguagem de fácil aprendizado;

Suporte a um grande número de bancos de dados como: dBase, Interbase, mSQL, mySQL, Oracle, PostgreSQL e vários outros.

É multiplataforma, tendo suporte aos sistemas Operacionais mais utilizados no mercado;

Não precisa ser compilado

Características

- Documentação muito bem elaborada com exemplos de todas as funções no site:
 - <http://www.php.net/>
- Amplamente utilizado

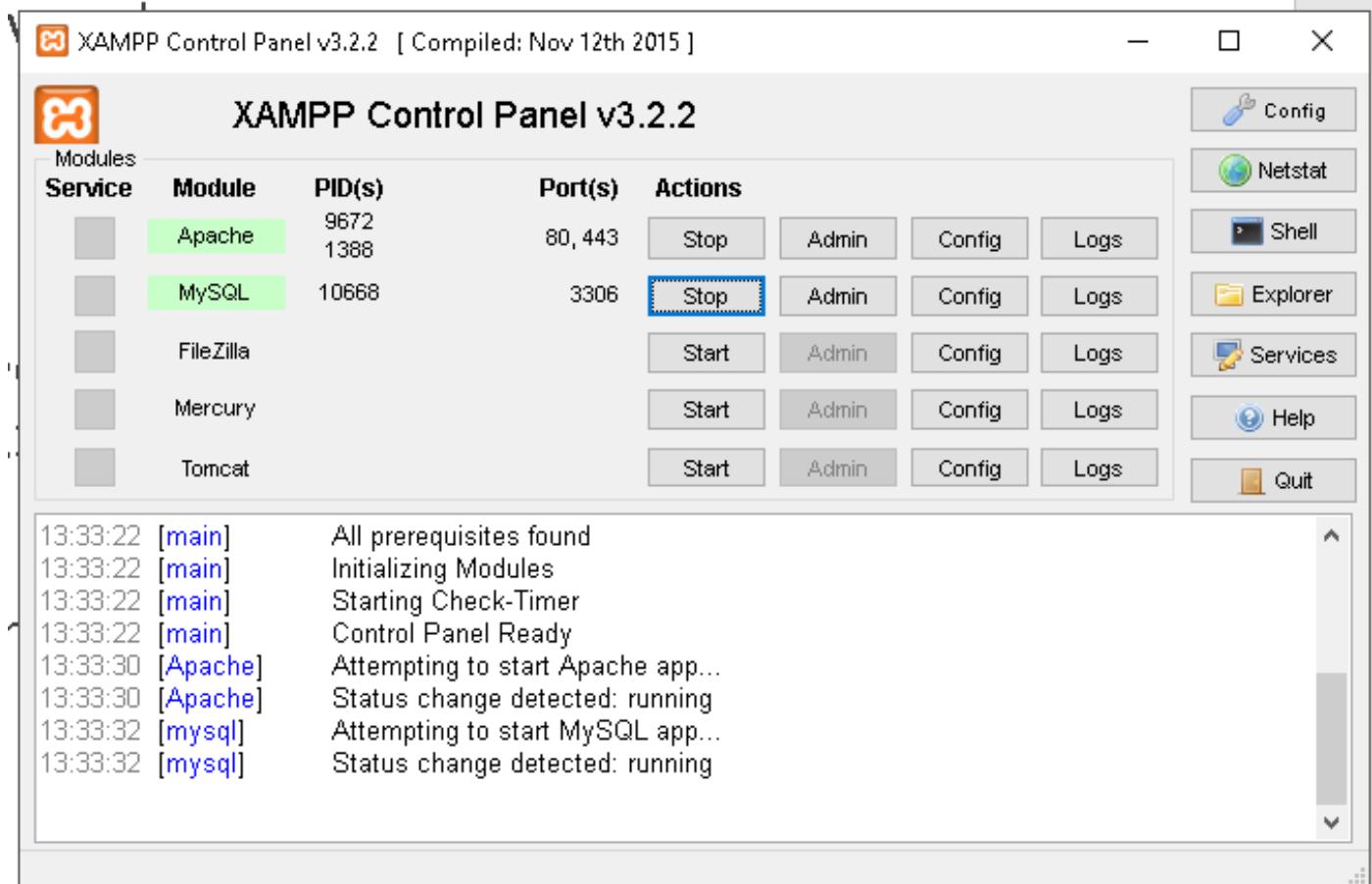




Configurando Servidor

- **Xampp** – Conjunto de ferramentas para desenvolvimento web
 - Apache - Servidor web
 - PHP
 - Mysql
 - PhpMyAdmin
- Acesso:
 - <https://www.apachefriends.org/>
- Após instalação abrir oxampp no INICIAR

Inicie os serviços



Configurando Servidor

- Endereço para acesso via navegador
 - `http://127.0.0.1/`
 - <http://localhost/>
- Onde encontro estes endereços no meu computador?
 - C:\xampp\htdocs
- Seus arquivos e pastas devem ser posicionados neste diretório.

Salvando documentos

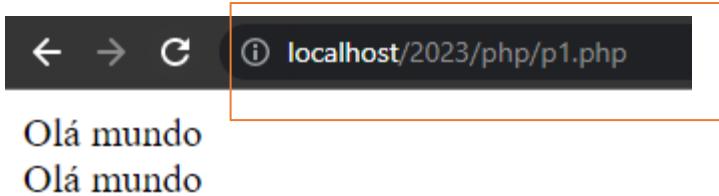
- Salve o arquivo em : C:\xampp\htdocs**nomeAluno**
- Salve os arquivos de seus programas sempre com a extensão .php
- Não utilize caracteres especiais e espaços nos nomes das pastas e arquivos.
- Caso salve o arquivo com uma extensão diferente de .php o código php não será executado.

Comandos ou instruções

- Comandos são responsáveis por realizar tarefas específicas(atribuir valor, imprimir na tela, etc).
- Todo comando em **PHP** é separado por :
;

Olá mundo (p1.php)

```
<?php  
echo "Olá mundo";  
echo "<br>";  
printf("Olá mundo");  
?>
```



- Para execução de script php é necessário que o mesmo esteja entre as tags `<?php` e `?>`
- Fora das tags o script não funciona. É um texto comum.
- Acesse o script apenas por meio de um servidor web. Caso contrário não será executado.

A screenshot of a code editor window titled "Arquivo | C:/xampp/htdocs/2023/php/p1.php". The code editor displays the PHP script `<?php echo "Olá mundo"; echo "
"; printf("Olá mundo"); ?>`. The entire code block is highlighted with an orange rectangle, and the status bar at the bottom of the editor window also has an orange rectangle around it.

Variáveis em php

- No PHP, uma variável começa com o **\$** seguido do nome da variável:
- *O tipo é definido em tempo de execução.*

```
$x = 5;
```

```
<?php  
    $x=1;  
    echo "valor da variável = $x."  
?>
```

- 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 de 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 das variáveis diferenciam maiúsculas de minúsculas (\$a e \$A são duas variáveis diferentes)

String

- O tipo de dados STRING é uma cadeia de caracteres alfanuméricos (letras, números e caracteres especiais). O tipo STRING pode ser utilizado de duas maneiras:
- Utilizando aspas simples o valor da variável será exatamente o texto contido entre as aspas, com exceção de: \\\ e \\'.
- Utilizando aspas duplas qualquer variável ou caractere de escape será expandido antes de ser atribuído.

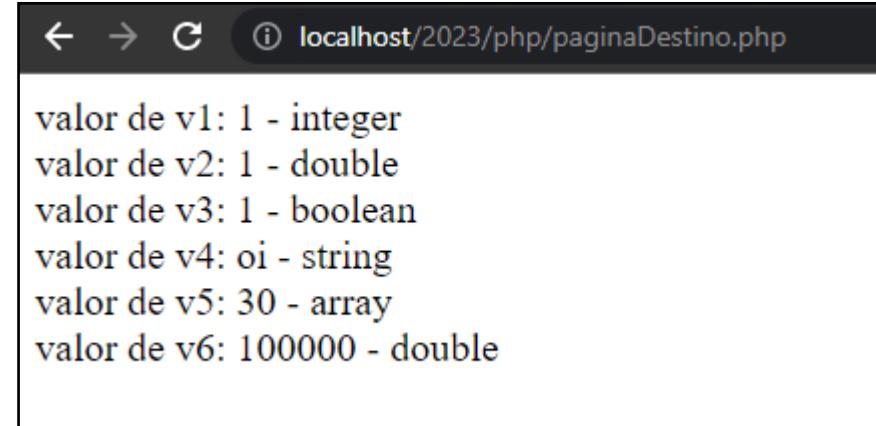
```
<?php  
    $x="helio";  
    $y="esperidião";  
  
    $v1 = "$x $y";  
    $v2 ='$x $y';  
  
    echo $v1."<br>";  
    echo $v2."<br>";  
?>
```

← → C i localhost/2023/php/paginaDestino.php

helio esperidião
\$x \$y

Exemplos de tipos

```
<?php  
$v1=1;  
$v2=1.0;  
$v3=TRUE;  
$v4="oi";  
$v5=[10, 20, 30];  
$v6 = 10e4; // (equivale a 100.000)  
  
echo "valor de v1: $v1 - ". gettype($v1)."  
echo "valor de v2: $v2 - ". gettype($v2)."  
echo "valor de v3: $v3 - ". gettype($v3)."  
echo "valor de v4: $v4 - ". gettype($v4)."  
echo "valor de v5: $v5[2] - ". gettype($v5)."  
echo "valor de v6: $v6 - ". gettype($v6)."  
?>
```



Envio de dados HTTP

Método POST

- Encapsula os dados e os envia junto do cabeçalho http.

Método GET

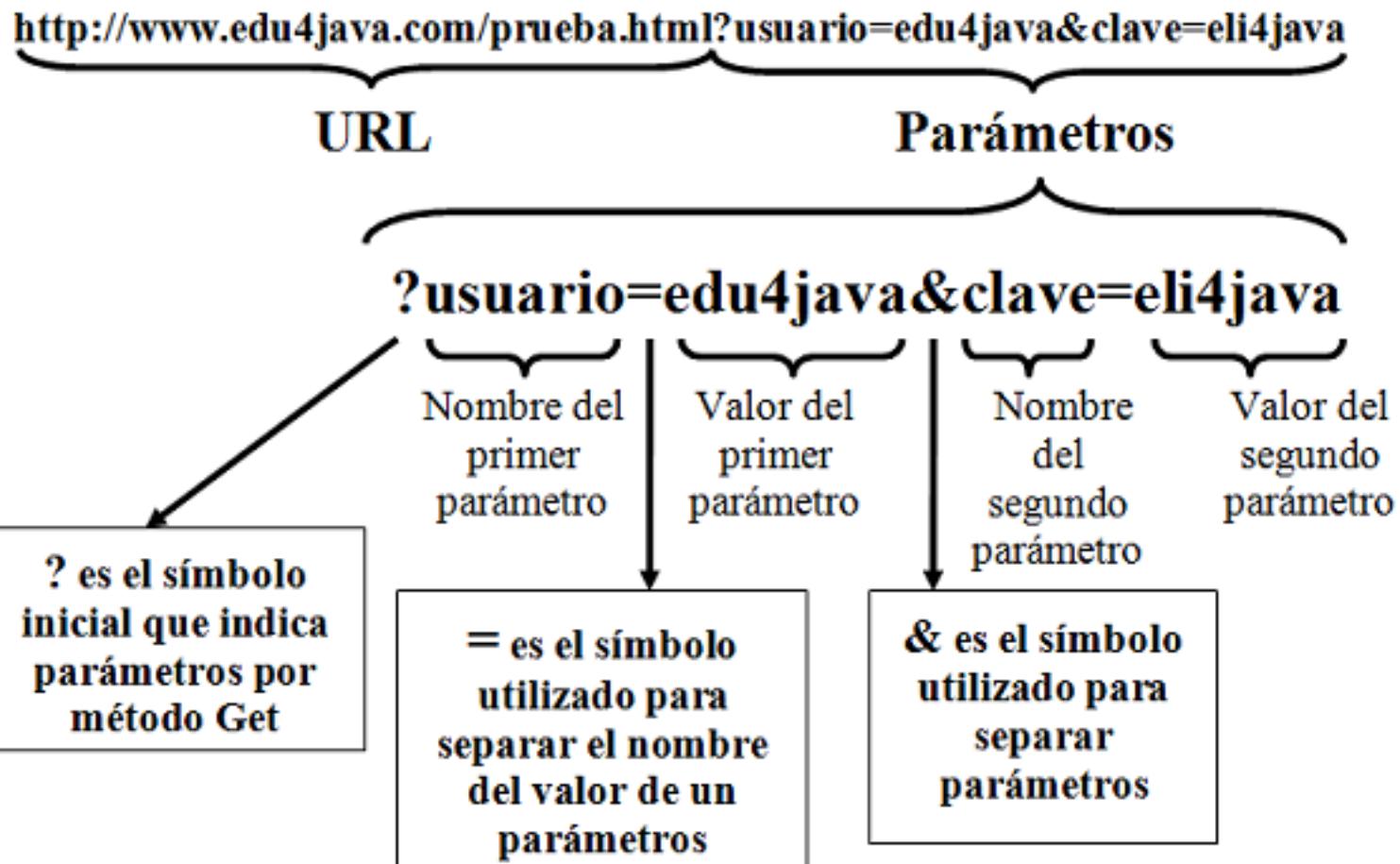
- Os dados são enviados por meio da url.

Entrada e saída dados.

O php pode receber dados via get ou via post.

O `$_GET` e o `$_POST` são dois arrays associativos que contem os dados enviados para o script (servidor)

Método GET



Exemplo google

- Observe os links abaixo, perceba que é enviada a variável **q** para “search”. Search é o serviço web da google que faz a busca.
- O dado é enviado pelo método **GET**
- <https://www.google.com/search?q=helioesperidiao>
- <https://www.google.com/search?q=batataFrita>
- <https://www.google.com/search?q=Queijo>

Formulários

GET

- Permitem o envio de dados para servidores.
- Permitem o envio de dados para scripts dentro da mesma página

```
<!DOCTYPE html>
<html>
<head>
  <title>Formulários</title>
</head>
<body>
  <form action="paginaDestino.php" method="get">

  </form>

</body>
</html>
```

Caixa de texto

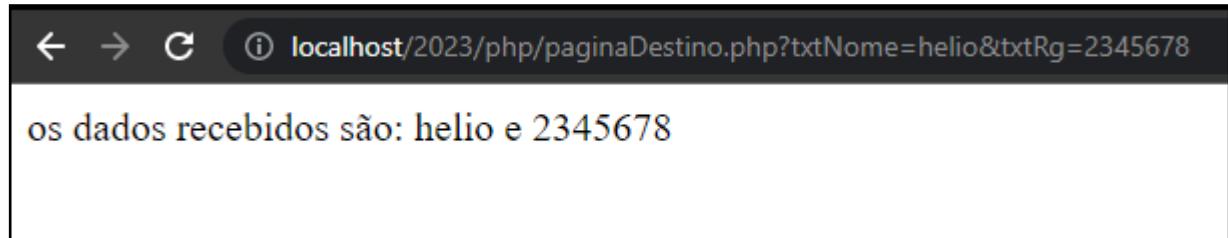
Formulario.html

```
<!DOCTYPE html>
<html>
<head>
<title>Formulários</title>
</head>
<body>
<form action="paginaDestino.php" method="get">
<input type="text" name="txtNome" placeholder="Nome"><br>
<input type="text" name="txtRg" placeholder="RG"><br>
<input type="submit" value="Cadastrar">
</form>

</body>
</html>
```

paginaDestino.php

```
<?php
$nome = $_GET['txtNome'];
$rg  = $_GET['txtRg'];
echo "os dados recebidos são: $nome e $rg";
?>
```



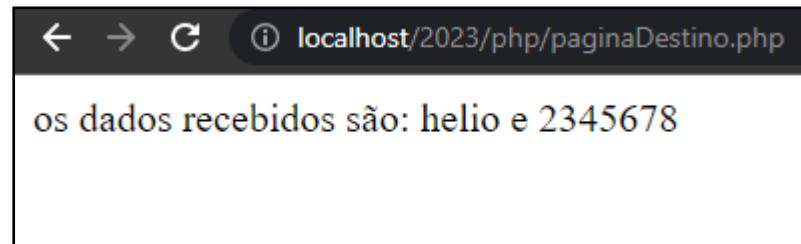
Caixa de texto

Formulario.html

```
<!DOCTYPE html>
<html>
<head>
<title>Formulários</title>
</head>
<body>
<form action="paginaDestino.php" method="post">
<input type="text" name="txtNome" placeholder="Nome"><br>
<input type="text" name="txtRg" placeholder="RG"><br>
<input type="submit" value="Cadastrar">
</form>
</body>
</html>
```

paginaDestino.php

```
<?php
$nome = $_POST['txtNome'];
$rg    = $_POST['txtRg'];
echo "os dados recebidos são: $nome e $rg";
?>
```





Operadores Matemáticos

Operador	Função
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo (resto da Divisão)
++	Adiciona 1 em uma variável numérica
--	Subtrai 1 em uma variável numérica
+=	adiciona o valor à direita à variável ou propriedade à esquerda e atribui o resultado à variável ou à propriedade à esquerda
-=	subtrai o valor à direita à variável ou propriedade à esquerda e atribui o resultado à variável ou à propriedade à esquerda
.=	multiplica o valor à direita à variável ou propriedade à esquerda e atribui o resultado à variável ou à propriedade à esquerda



Operadores relacionais

<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 de <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> .

Controle de fluxo: if

```
if (condition) {  
    //Código;  
}
```

```
if (condition) {  
    //código  
} else {  
    //código  
}
```

```
if (condition) {  
    //Código;  
} elseif (condition) {  
    //Código;  
} else {  
    //Código;  
}
```

Receber dados numéricos

calcularMedia.html

```
<!DOCTYPE html>
<html>
<head>
<title>Formulários</title>
</head>
<body>
<form action="calcularMedia.php" method="post">
<input type="text" name="txtNota1"><br>
<input type="text" name="txtNota2"><br>
<input type="submit" value="Calcular Media">
</form>
</body>
</html>
```

calcularMedia.php

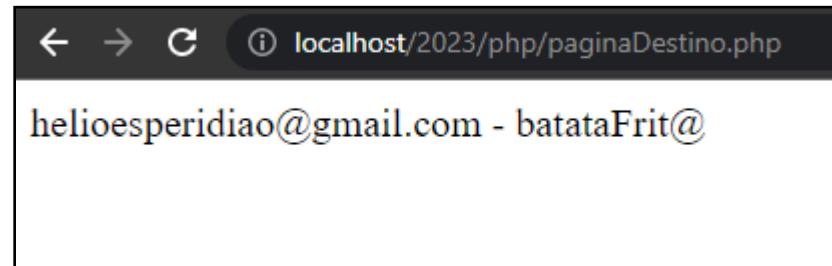
```
<?php
$n1  = $_POST['txtNota1'];
$n2  = $_POST['txtNota2'];
$media = ($n1+$n2)/2;
printf("nota final: %.2f ",$media);
if($media>=6){
    echo "aprovado";
} else{
    echo "reprovado";
}
?>
```

Password

```
<!DOCTYPE html>
<html>
<head>
<title>Formulários</title>
</head>
<body>
<form action="paginaDestino.php" method="post">
<input type="email" name="txtEmail"><br>
<input type="password" name="txtSenha"><br>
<input type="submit" value="Cadastrar">
</form>

</body>
</html>
```

```
<?php
$senha = $_POST['txtSenha'];
$email = $_POST['txtEmail'];
echo "$email - $senha";
?>
```

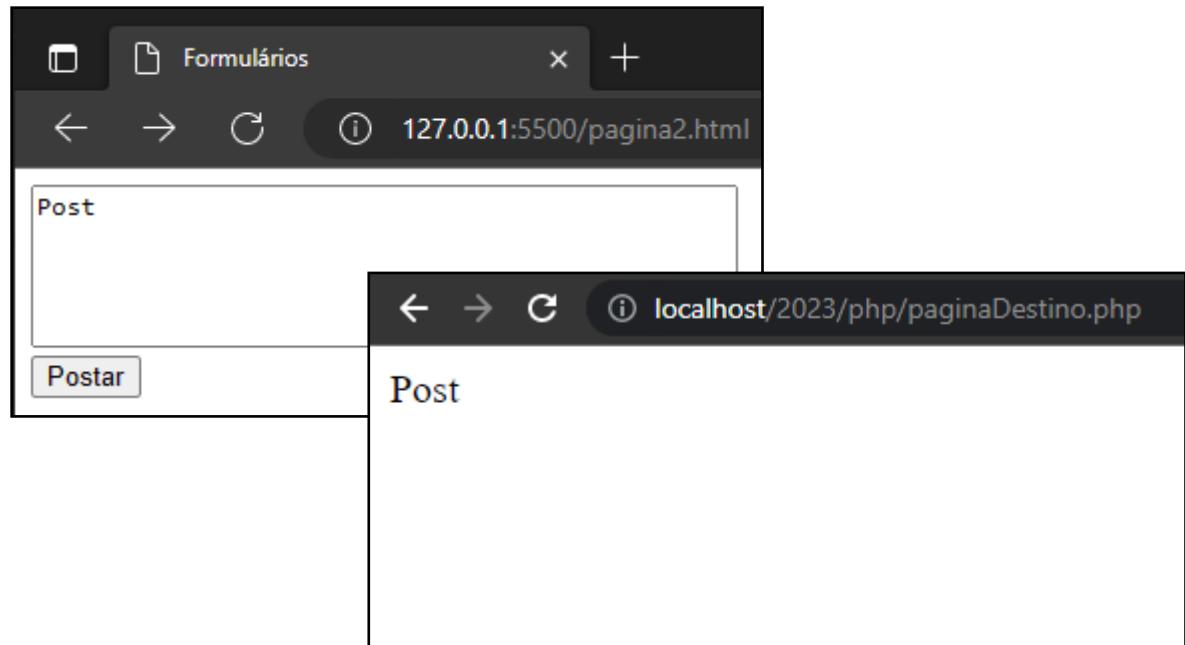


textarea

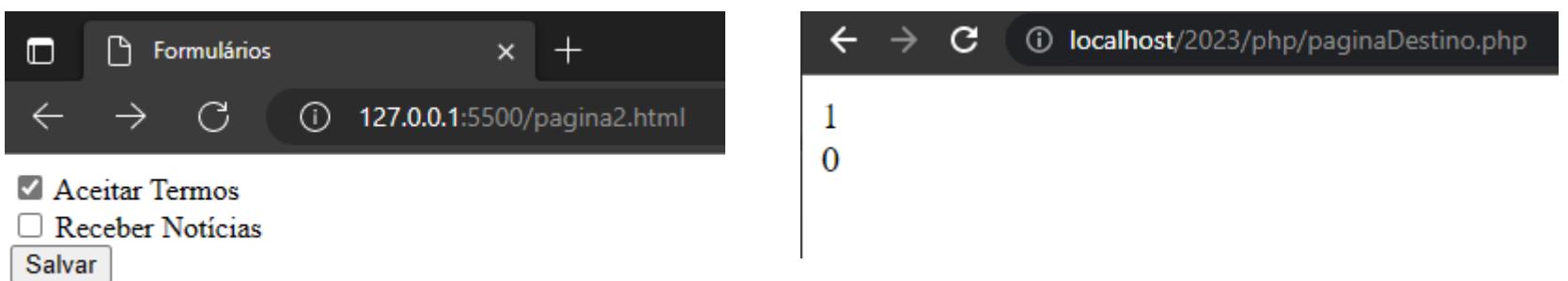
```
<!DOCTYPE html>
<html>
<head>
<title>Formulários</title>
</head>
<body>
<form action="paginaDestino.php" method="get">
<textarea cols="45" rows="5" name="txtPost">
</textarea><br>
<input type="submit" value="Postar">
</form>

</body>
</html>
```

```
<?php
$postDigitado = $_POST['txtPost'];
echo $postDigitado;
?>
```



checkbox



Formulários

127.0.0.1:5500/pagina2.html

Aceitar Termos
 Receber Notícias
Salvar

```
<html>
<head>
<title>Formulários</title>
</head>
<body>
<form action="paginaDestino.php" method="post">
<input type="checkbox" name="chkTermos" >
Aceitar Termos<br>
<input type="checkbox" name="chkNoticias" >
Receber Notícias
<br>
<input type="submit" value="Salvar">
</form>

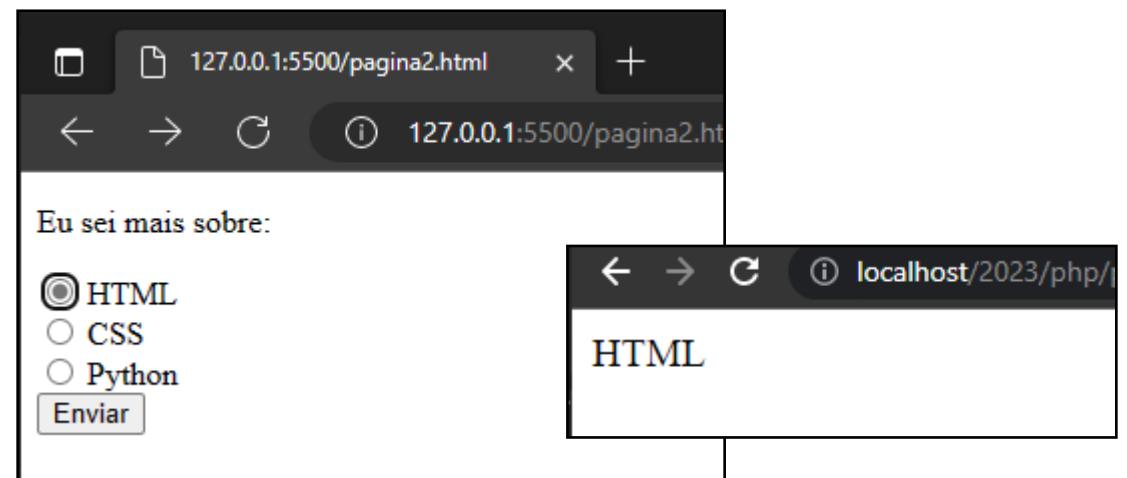
</body>
</html>
```

```
<?php
if(isset($_POST['chkTermos'])){
    $termos=1;
} else{
    $termos=0;
}
if(isset($_POST['chkNoticias'])){
    $noticias=1;
} else{
    $noticias=0;
}
echo $termos;
echo "<br>";
echo $noticias;
?>
```

radio

```
<!DOCTYPE html>
<html>
<body>
<form action="paginaDestino.php" method="post">
<p>Eu sei mais sobre:</p>
<input type="radio" name="rdoFavorito" value="HTML">
HTML<br>
<input type="radio" name="rdoFavorito" value="CSS">
CSS<br>
<input type="radio" name="rdoFavorito" value="Python">
Python<br>
<input type="submit" value="Enviar">
</form>
</body>
</html>
```

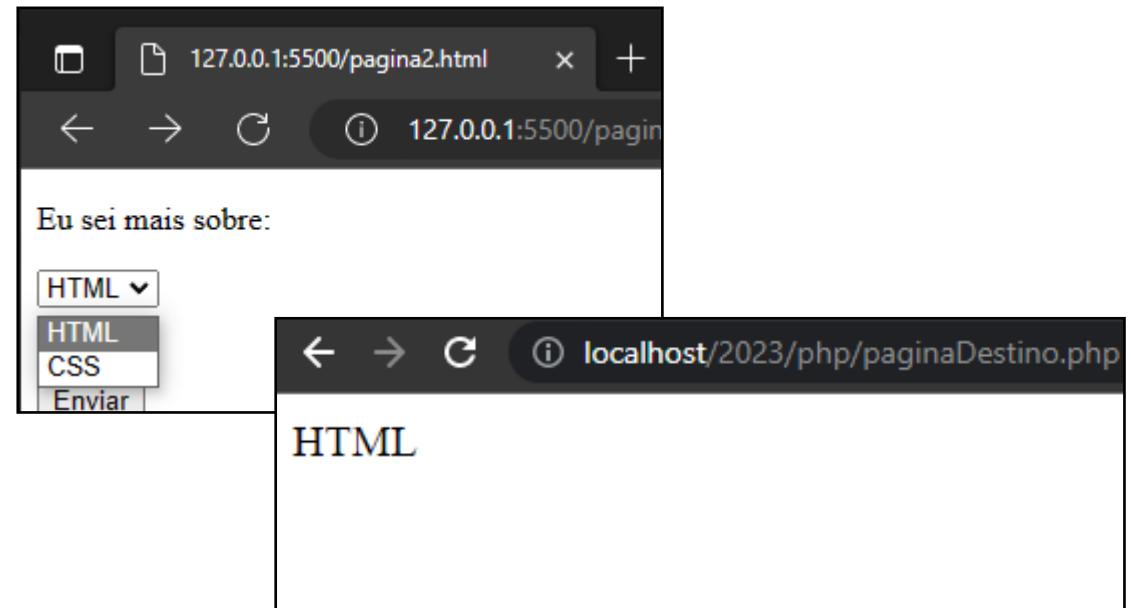
```
<?php
if(isset($_POST['rdoFavorito'])){
    $favorito=$_POST['rdoFavorito'];
}else{
    $favorito='não selecionado';
}
echo $favorito;
?>
```



Caixa de seleção

```
<!DOCTYPE html>
<html>
<body>
<form action="paginaDestino.php" method="post">
<p>Eu sei mais sobre:</p>
<select name="cboLinguagem">
    <option value="HTML">HTML</option>
    <option value="CSS">CSS</option>
</select>
<br><br><br>
<input type="submit" value="Enviar">
</form>
</body>
</html>
```

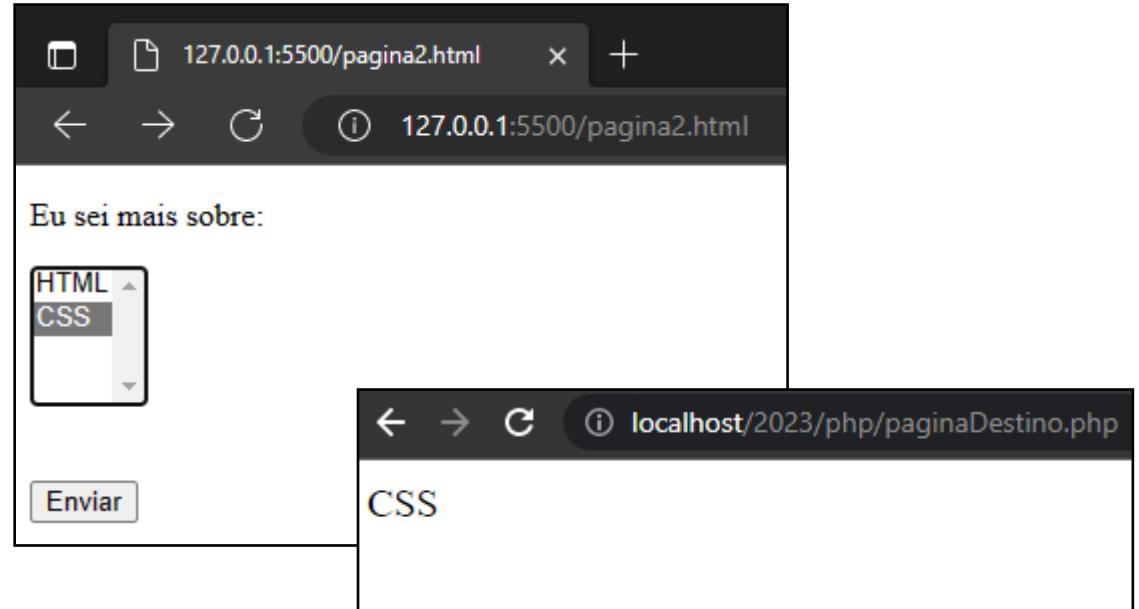
```
<?php
$linguagem = $_POST['cboLinguagem'];
echo $linguagem ;
?>
```



Caixa de lista

```
<!DOCTYPE html>
<html>
<body>
<form action="paginaDestino.php" method="post">
<p>Eu sei mais sobre:</p>
<select name="lstLinguagem" size="4">
<option value="HTML">HTML</option>
<option value="CSS">CSS</option>
</select>
<br><br><br>
<input type="submit" value="Enviar">
</form>
</body>
</html>
```

```
<?php
$linguagem = $_POST['lstLinguagem'];
echo $linguagem ;
?>
```

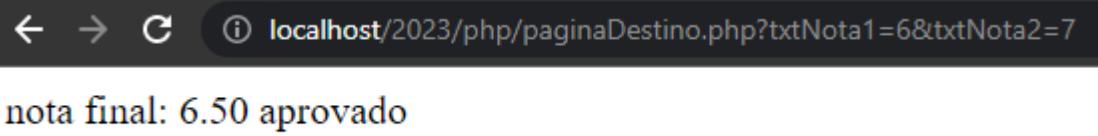


Recebendo dados pela URL – Método GET

```
http://localhost/2023/php/paginaDestino.php?txtNota1=6&txtNota2=7
```

```
<!DOCTYPE html>
<html>
<head>
<title>Formulários</title>
</head>
<body>
<form action="paginaDestino.php" method="get">
<input type="text" name="txtNota1"><br>
<input type="text" name="txtNota2"><br>
<input type="submit" value="Calcular Media">
</form>
</body>
</html>
```

```
<?php
$n1 = $_GET['txtNota1'];
$n2 = $_GET['txtNota2'];
$media = ($n1+$n2)/2;
printf("nota final: %.2f ",$media);
if($media>=6){
    echo "aprovado";
} else{
    echo "reprovado";
}
?>
```



A screenshot of a web browser window. The address bar shows the URL: `localhost/2023/php/paginaDestino.php?txtNota1=6&txtNota2=7`. The main content area of the browser displays the output of the PHP script: `nota final: 6.50 aprovado`.

Não é necessário um formulário para trabalhar com o método get

Altere direto na url do navegador:

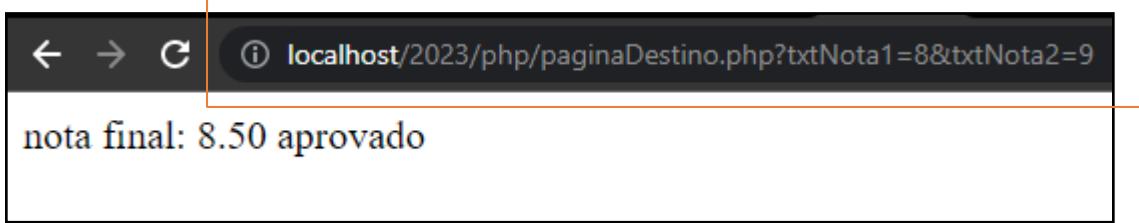
http://localhost/2023/php/paginaDestino.php?txtNota1=6&txtNota2=7

http://localhost/2023/php/paginaDestino.php?txtNota1=8&txtNota2=9



localhost/2023/php/paginaDestino.php?txtNota1=6&txtNota2=7

nota final: 6.50 aprovado



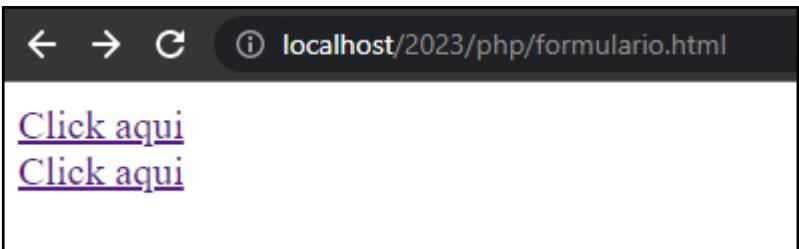
localhost/2023/php/paginaDestino.php?txtNota1=8&txtNota2=9

nota final: 8.50 aprovado

```
<?php
    $n1 = $_GET['txtNota1'];
    $n2 = $_GET['txtNota2'];
    $media = ($n1+$n2)/2;
    printf("nota final: %.2f ",$media);
    if($media>=6){
        echo "aprovado";
    }else{
        echo "reprovado";
    }
?>
```

Envie dados pela URL

```
<!DOCTYPE html>
<html>
<head>
<title>Formulários</title>
</head>
<body>
<a href="paginaDestino.php?txtNota1=8&txtNota2=9">Click aqui</a><br>
<a href="paginaDestino.php?txtNota1=6&txtNota2=7">Click aqui</a>
</body>
</html>
```



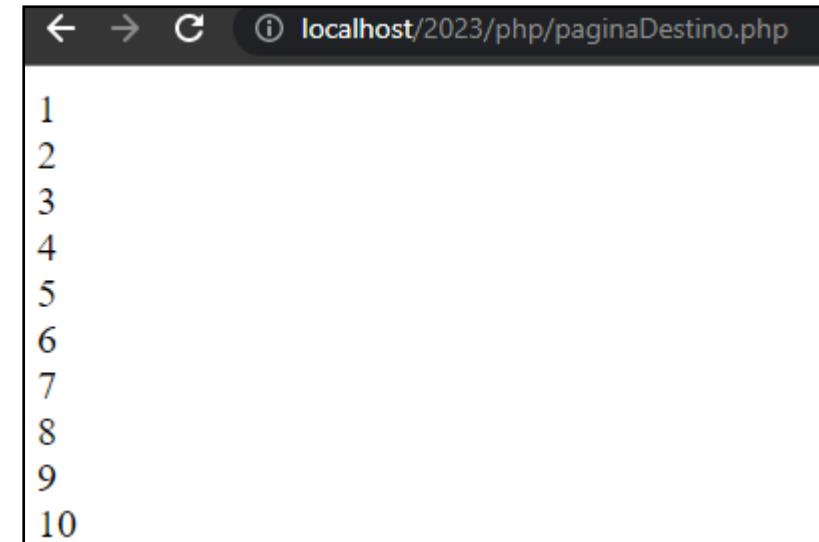
Observe que os hiperlinks enviam dados no formato get para “paginaDestino.php”.

```
<?php
$n1 = $_GET['txtNota1'];
$n2 = $_GET['txtNota2'];
$media = ($n1+$n2)/2;
printf("nota final: %.2f ",$media);
if($media>=6){
    echo "aprovado";
} else{
    echo "reprovado";
}
?>
```

Estruturas de repetição em php (while)

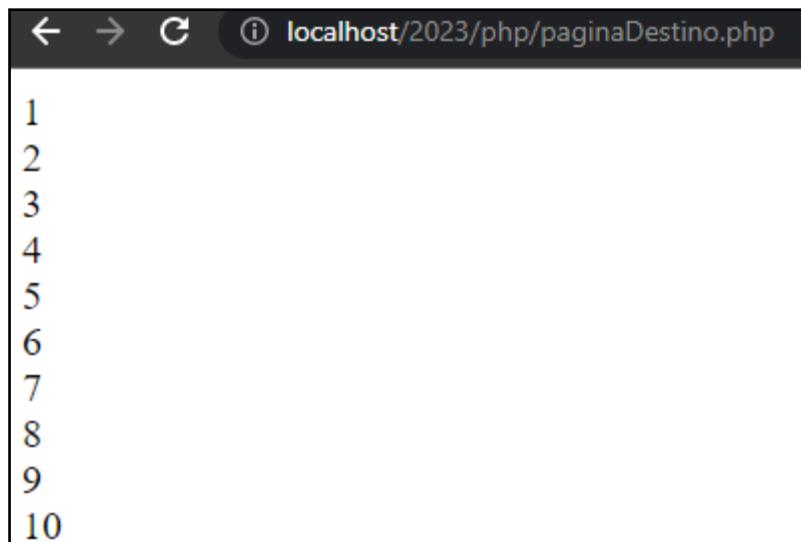
```
while (<<condição Verdadeira>>) {  
    //código;  
}
```

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



Estruturas de repetição em php (do-while)

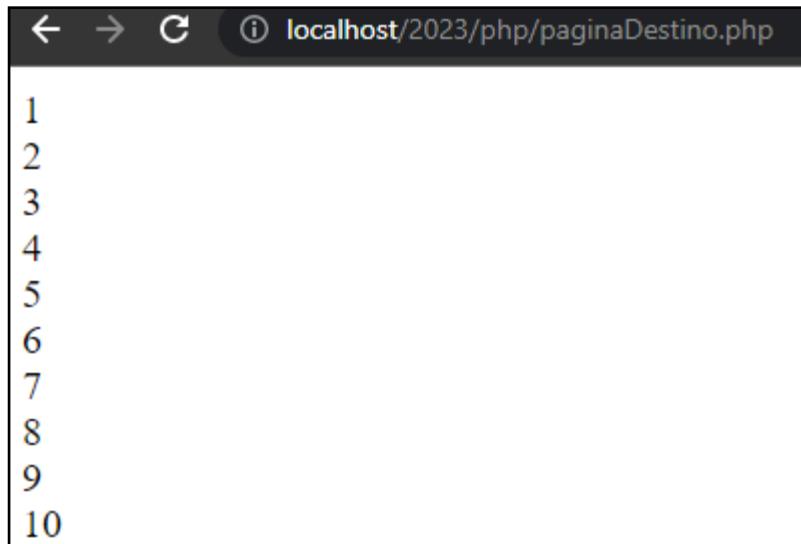
```
do {  
    //Código;  
} while (<<condição Verdadeira>>);
```



```
<?php  
$num = 1;  
do{  
    echo "$num <br>";  
    $num++;  
  
}while ($num <= 10);  
?>
```

Estruturas de repetição em php (for)

```
for (inic contador; teste contador; icnremento contador) {  
    //repetir;  
}
```



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

Break

Ao encontrar o break a repetição é interrompida.

```
<?php  
for ($x = 0; $x < 10; $x++) {  
    if ($x == 4) {  
        break;  
    }  
    echo "o número é: $x <br>";  
}  
?>
```



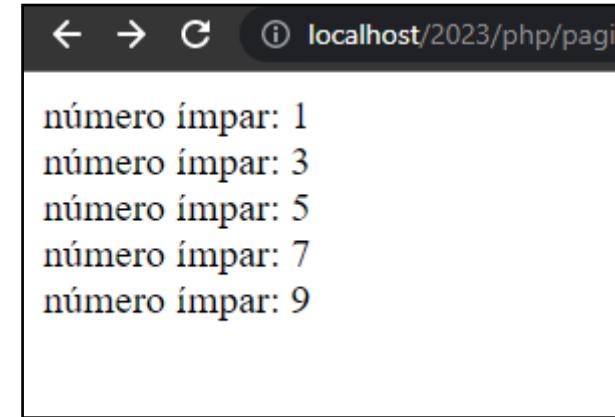
← → C ⓘ localhost/2023/php/paginaDestino.php

o número é: 0
o número é: 1
o número é: 2
o número é: 3

Continue

Quando ele é executado, o laço volta imediatamente para o teste de condição do laço de repetição.

```
<?php
for ($x = 0; $x <= 10; $x++) {
    if ($x%2==0) {
        continue;
    }
    echo "número ímpar: $x <br>";
}
?>
```



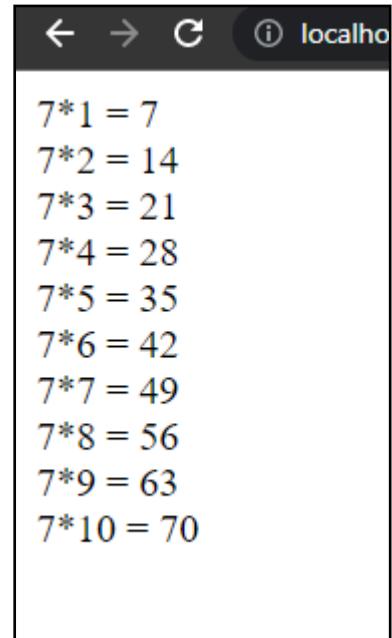
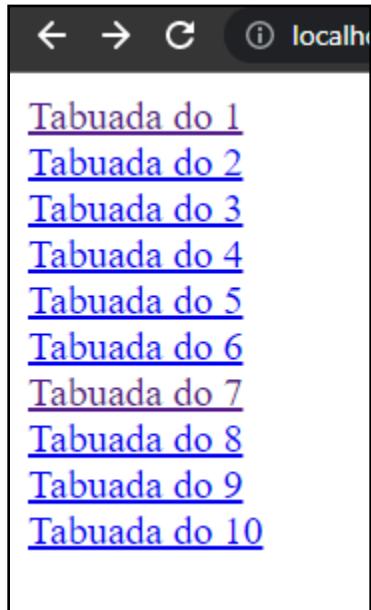
Utilizando get sem forms

```
<!DOCTYPE html>
<html>
<body>
<a href="paginaDestino.php?v1=1">Tabuada do 1</a><br>
<a href="paginaDestino.php?v1=2">Tabuada do 2</a><br>
<a href="paginaDestino.php?v1=3">Tabuada do 3</a><br>
<a href="paginaDestino.php?v1=4">Tabuada do 4</a><br>
<a href="paginaDestino.php?v1=5">Tabuada do 5</a><br>
<a href="paginaDestino.php?v1=6">Tabuada do 6</a><br>
<a href="paginaDestino.php?v1=7">Tabuada do 7</a><br>
<a href="paginaDestino.php?v1=8">Tabuada do 8</a><br>
<a href="paginaDestino.php?v1=9">Tabuada do 9</a><br>
<a href="paginaDestino.php?v1=10">Tabuada do 10</a><br>
</body>
</html>
```

```
<?php
$valor = $_GET['v1'];
$num = 1;
while ($num <= 10) {
    $tabuada = $valor*$num;
    echo "$valor*$num = $tabuada<br>";
    $num++;
}
?>
```

Cada url envia um valor diferente de **v1** para paginaDestino.php

- <http://localhost/2023/php/paginaDestino.php?v1=1>
- <http://localhost/2023/php/paginaDestino.php?v1=7>

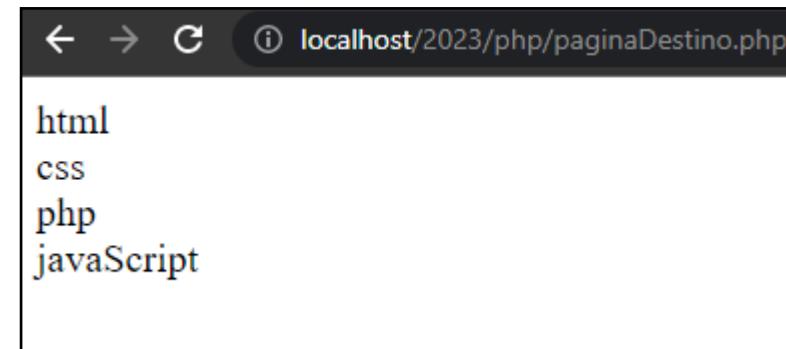


```
<?php
$valor = $_GET['v1'];
$num = 1;
while ($num <= 10) {
    $tabuada = $valor*$num;
    echo "$valor*$num = $tabuada<br>";
    $num++;
}
?>
```

Estruturas de repetição em php (foreach)

```
foreach ($array as $posicaoArray) {  
    //repetir;  
}
```

```
<?php  
$vetor = array("html", "css", "php", "javaScript");  
  
foreach ($vetor as $posicao) {  
    echo "$posicao <br>";  
}  
?>
```

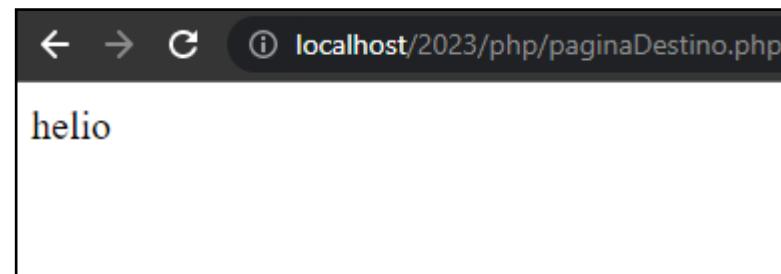


Array associativo

É um tipo de array onde os índices podem ser utilizados como textos.

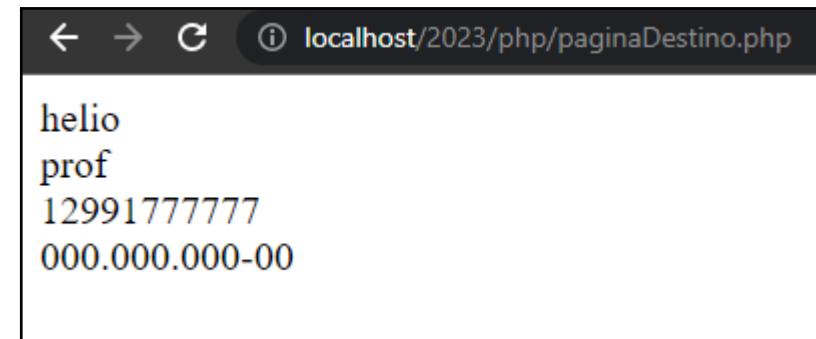
Observe que existe o “índice” nome,cargo,telefone e cpf.

```
<?php  
$cliente['nome'] = "helio";  
$cliente['cargo'] = "prof";  
$cliente['telefone'] = "12991777777";  
$cliente['cpf'] = "000.000.000-00";  
  
echo $cliente['nome'];  
  
?>
```



Foreach – Array associativo

```
<?php  
$cliente['nome'] = "helio";  
$cliente['cargo'] = "prof";  
$cliente['telefone'] = "12991777777";  
$cliente['cpf'] = "000.000.000-00";  
  
foreach ($cliente as $posicao) {  
    echo "$posicao <br>";  
}  
?>
```



localhost/2023/php/paginaDestino.php

helio
prof
12991777777
000.000.000-00

Observe que o `$_GET` E `$_POST` são vetores associativos

```
<?php  
    $n1 = $_POST['txtNota1'];  
    $n2 = $_POST['txtNota2'];  
?  
?
```

- `$_POST` é um array, existem as posições '**txtNota1**', '**txtNota2**', etc.

`$_GET` E `$_POST` são vetores associativos

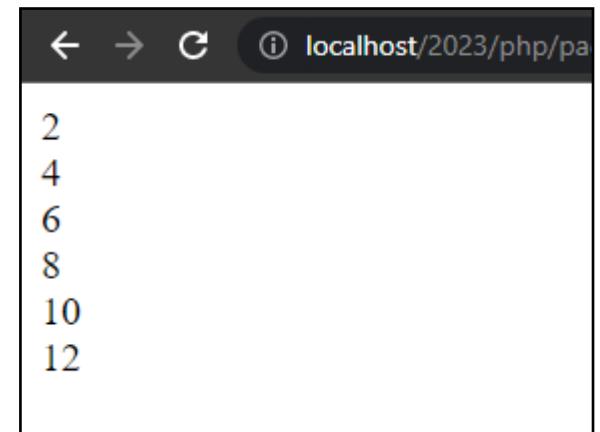
```
<!DOCTYPE html>
<html>
<body>
<form action="paginaDestino.php" method="POST">
<input type="text" name="txtN1"><br>
<input type="text" name="txtN2"><br>
<input type="text" name="txtN3"><br>
<input type="text" name="txtN4"><br>
<input type="text" name="txtN5"><br>
<input type="text" name="txtN6"><br>
<input type="submit" value="Cadastrar">
</form>
</body>
</html>
```

```
<?php
foreach ($_POST as $posicao) {
    echo "$posicao <br>";
}
?>
```

localhost/2023/paginaDestino.php

2
4
6
8
10
12

Cadastrar

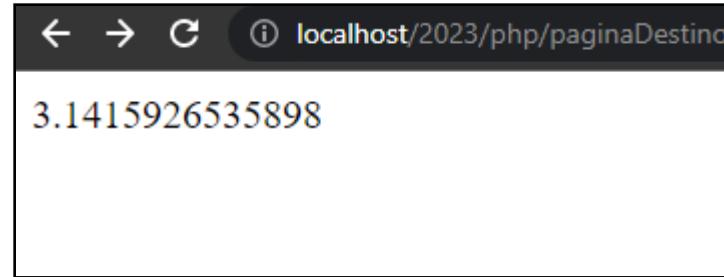


Constantes matemáticas.

Constant	Value	Description
M_E	2.7182818284590452354	Returns e
M_EULER	0.57721566490153286061	Returns Euler constant
M_LNPI	1.14472988584940017414	Returns the natural logarithm of PI: log_e(pi)
M_LN2	0.69314718055994530942	Returns the natural logarithm of 2: log_e 2
M_LN10	2.30258509299404568402	Returns the natural logarithm of 10: log_e 10
M_LOG2E	1.4426950408889634074	Returns the base-2 logarithm of E: log_2 e
M_LOG10E	0.43429448190325182765	Returns the base-10 logarithm of E: log_10 e
M_PI	3.14159265358979323846	Returns Pi
M_PI_2	1.57079632679489661923	Returns Pi/2
M_PI_4	0.78539816339744830962	Returns Pi/4
M_1_PI	0.31830988618379067154	Returns 1/Pi
M_2_PI	0.63661977236758134308	Returns 2/Pi
M_SQRTPI	1.77245385090551602729	Returns the square root of PI: sqrt(pi)
M_2_SQRTPI	1.12837916709551257390	Returns 2/square root of PI: 2/sqrt(pi)
M_SQRT1_2	0.70710678118654752440	Returns the square root of 1/2: 1/sqrt(2)
M_SQRT2	1.41421356237309504880	Returns the square root of 2: sqrt(2)
M_SQRT3	1.73205080756887729352	Returns the square root of 3: sqrt(3)

Exemplos de uso de constantes matemáticas:

```
<?php  
$x= M_PI;  
echo $x;  
?>
```



Funções matemáticas

abs()	Retorna o absoluto (positive) valor de um número
acos()	Retorna o arco co-seno de um número
acosh()	Retorna o cosseno hiperbólico inverso de um número
asin()	Retorna o arco seno de um número
asinh()	Retorna o seno hiperbólico inverso de um número
atan()	Retorna o arco tangente de um número em radianos
atan2()	Retorna o arco tangente de duas variáveis x e y
atanh()	Retorna a tangente hiperbólica inversa de um número
base_convert()	Converte um número a partir de uma base numérica para outra
bindec()	Converte um número binário em um número decimal

Funções matemáticas

ceil()	Arredonda um número para o número inteiro mais próximo
cos()	Retorna o co-seno de um número
cosh()	Retorna o cosseno hiperbólico de um número
decbin()	Converte um número decimal para um número binário
dechex()	Converte um número decimal para um número hexadecimal
decoct()	Converte um número decimal para um número octal
deg2rad()	Converte um valor de grau para um valor radiano
exp()	Calcula o expoente e
expm1()	Retorna $\exp(x) - 1$
floor()	Arredonda um número para baixo para o número inteiro mais próximo

Funções matemáticas

fmod()	Devolve o resto de x / y
getrandmax()	Retorna o maior valor possível retornado por rand()
hexdec()	Converte um número hexadecimal para um número decimal
hypot()	Calcula a hipotenusa de um triângulo de ângulo reto
is_finite()	Verifica se um valor é finito ou não
is_infinite()	Verifica se um valor é infinito ou não
is_nan()	Verifica se um valor é 'não-um-número'
lcg_value()	Retorna um número pseudo aleatório num intervalo entre 0 e 1
log()	Retorna o logaritmo natural de um número
log10()	Retorna o logaritmo de base 10 de um número

Funções matemáticas

log1p()	Retornos log(1+number)
max()	Devolve o valor mais elevado em uma matriz, ou o mais alto valor de vários valores especificados
min()	Devolve o valor mais baixo em uma matriz, ou o menor valor de vários valores especificados
mt_getrandmax()	Retorna o maior valor possível retornado por mt_rand()
mt_rand()	Gera um número inteiro aleatório usando o algoritmo de Mersenne Twister
mt_srand()	Semeia o gerador de número aleatório Mersenne tornado
octdec()	Converte um número octal em um número decimal
pi()	Retorna o valor de PI
pow()	Retorna x elevado à potência de y
rad2deg()	Converte um valor em radianos para um valor de grau

Funções matemáticas

rand()	Gera um número aleatório
round()	Rondas um número de ponto flutuante
sin()	Retorna o seno de um número
sinh()	Retorna o seno hiperbólico de um número
sqrt()	Retorna a raiz quadrada de um número
srand()	Semeia o gerador de número aleatório
tan()	Retorna a tangente de um número
tanh()	Retorna a tangente hiperbólica de um número

Exemplo de uso de funções matemáticas.

```
<?php  
    $x= sqrt(4);  
    echo $x;  
?>
```

Uso de funções de strings

- Durante o desenvolvimento de uma aplicação utilizamos o tipo de dado string para uma cadeia de caracteres ou texto.
- Diferente de outros tipos como int que trabalham com valores simples, uma string é um objeto composto de uma coleção de caracteres.
- Uma string pode utilizar métodos para manipular sua coleção de caracteres.
- Esses métodos permitem algumas tarefas, dentre elas:
 - Obter o seu tamanho ou quantidade de caracteres
 - Obter um trecho específico do texto
 - Substituir um trecho específico do texto
 - Verificar a ocorrência de um trecho específico de texto

Exemplos de funções de strings

```
<?php  
$x=chr(65);  
echo $x;  
?>
```

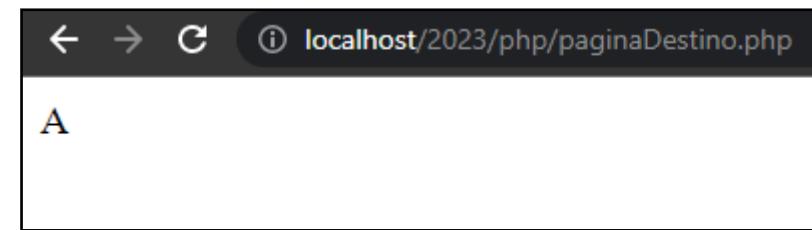


Tabela ascii

- O ASCII é um código que foi proposto por Robert W. Bemer como uma solução para unificar a representação de caracteres alfanuméricos em computadores.
- Antes de 1960, cada computador utilizava uma regra diferente para representar estes caracteres e o código ASCII nasceu para se tornar comum entre todas as máquinas.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0 000	000	NUL (null)	32	20	040	 	Space	64	40	100	@	Ø	96	60	140	`	~
1	1 001	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2 002	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3 003	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4 004	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5 005	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6 006	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7 007	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8 010	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9 011	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A 012	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B 013	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C 014	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D 015	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E 016	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F 017	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10 020	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11 021	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12 022	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13 023	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14 024	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15 025	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16 026	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17 027	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18 030	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19 031	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A 032	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B 033	033	ESC (escape)	59	3B	073	;	:	91	5B	133	[[123	7B	173	{	{
28	1C 034	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D 035	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E 036	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F 037	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

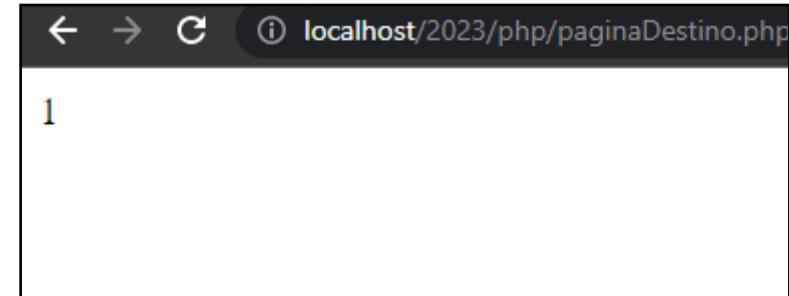
exemplo

```
<?php  
$i=0;  
while($i<255){  
    $x=chr($i);  
    echo "[\$i] =\$x<br>";  
    $i++;  
}  
?>
```

[33] =!
[34] ="
[35] =#
[36] =\$
[37] =%
[38] =&
[39] ='
[40] =(
[41] =)
[42] =*
[43] =+
[44] =,
[45] =-
[46] =.
[47] =/
[48] =0
[49] =1
[50] =2
[51] =3
[52] =4
[53] =5
[54] =6
[55] =7
[56] =8
[57] =9

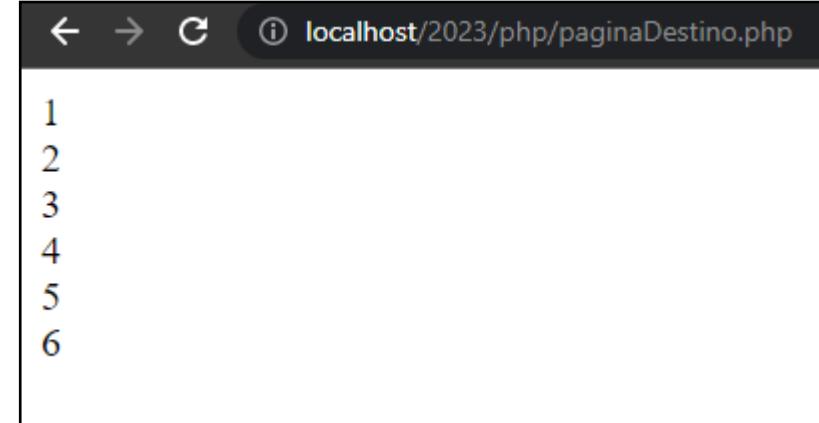
explode - Divide uma String em um vetor

```
<?php  
$string = "1;2;3;4;5;6";  
$vetor = explode(";", $string);  
echo $vetor[0];  
?>
```



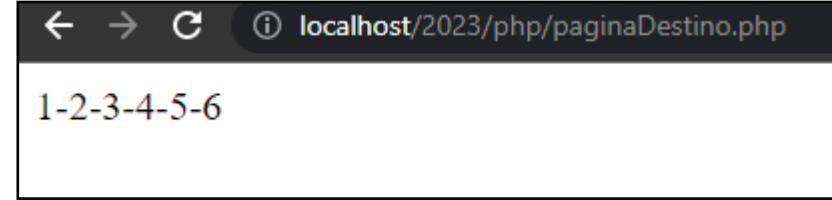
explode - Divide uma String em um vetor

```
<?php  
  
$string = "1;2;3;4;5;6";  
  
$vetor = explode(";", $string);  
  
foreach($vetor as $posicao){  
    echo "$posicao<br>";  
}  
  
?>
```



Join – junta as strings dentro de um vetor em uma string única.

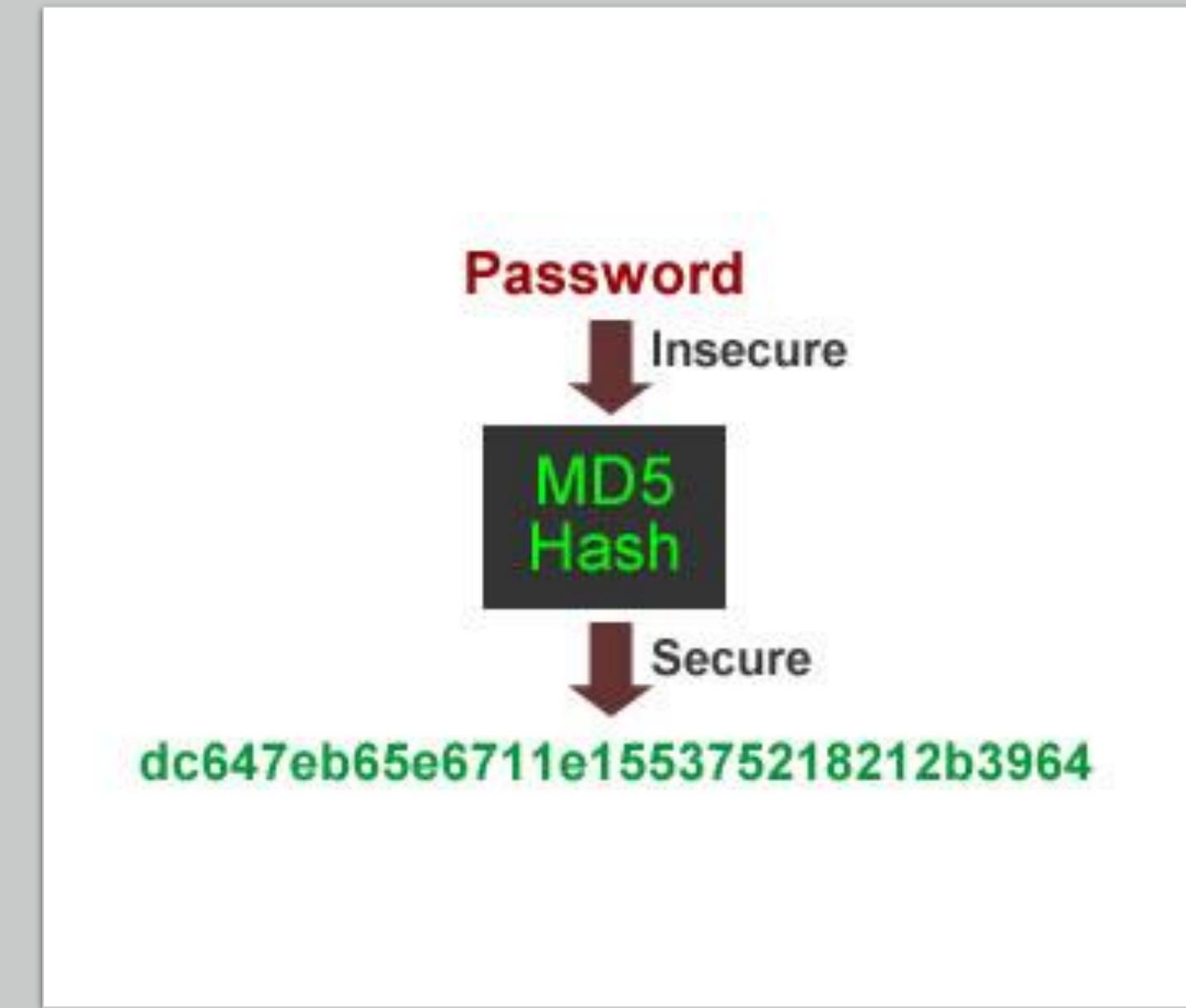
```
<?php  
$string = "1;2;3;4;5;6";  
$vetor = explode(";", $string);  
  
$String2 = join("-", $vetor) ;  
echo $String2;  
?>
```



MD5

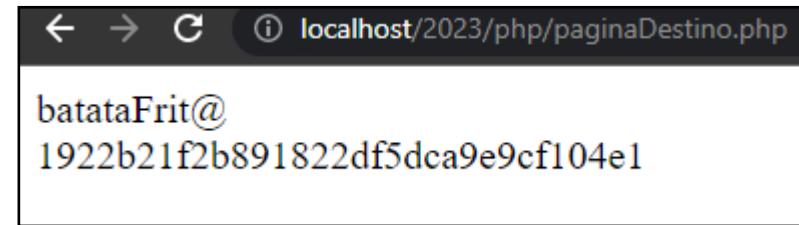
Md5: Retorna um hash em uma string de **32 caracteres**. Não pode ser decodificada.

.



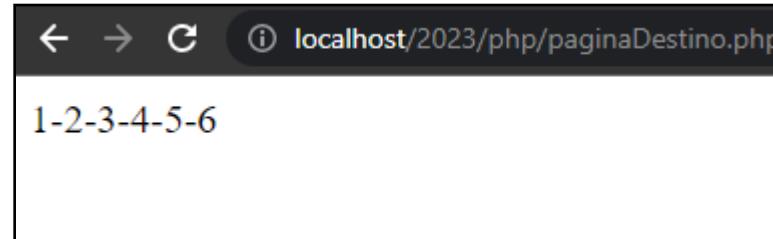
MD5

```
<?php  
$email = $_POST['txtEmail'];  
$senha = $_POST['txtSenha'];  
  
$senhaMd5 = md5($senha);  
echo "$senha <BR>$senhaMd5" ;  
  
?>
```



`str_replace` – Substitui um conjunto de caracteres por outro.

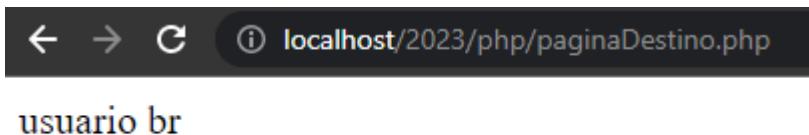
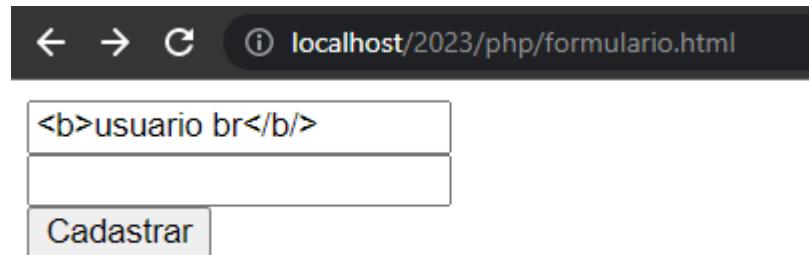
```
<?php  
$string = "1;2;3;4;5;6";  
  
$str2 = str_replace(";", "-", $string);  
  
echo $str2 ;  
?>
```



strip_tags – remove html de uma string

```
<?php  
$email = $_POST['txtEmail'];  
$senha = $_POST['txtSenha'];  
$email = strip_tags($email);  
echo $email;  
?>
```

- Os caracteres foram removidos e não há mais a tag



Outras funções de String

chr	Retorna um caracter específico
echo	Exibe uma ou mais strings
explode	Divide uma string em strings
fprintf	Escreve uma string formatada para um stream
join	Sinônimo de implode
lcfirst	Torna minúsculo o primeiro caractere de uma string
ltrim	Retira espaços em branco (ou outros caracteres) do início da string
md5	Calcula o "hash MD5" de uma string
money_format	Formata um número como uma string de moeda
nl2br	Insere quebras de linha HTML antes de todas newlines em uma string

Outras funções de String

number_format	Formata um número com os milhares agrupados
ord	Retorna o valor ASCII do caractere
print	Mostra uma string
printf	Mostra uma string formatada
rtrim	Retira espaço em branco (ou outros caracteres) do final da string
str_contains	Determina se uma string contém uma substring fornecida
str_ends_with	Verifica se uma string termina com uma substring fornecida
str_getcsv	Analisa uma string CSV e retorna os dados em um array
str_ireplace	Versão que não diferencia maiúsculas e minúsculas de str_replace.
str_replace	Substitui todas as ocorrências da string de procura com a string de substituição

Outras funções de String

str_split	Converte uma string para um array
str_starts_with	Verifica se uma string começa com uma substring fornecida
str_word_count	Retorna informação sobre as palavras usadas em uma string
strcasecmp	Comparação de strings sem diferenciar maiúsculas e minúsculas segura para binário
strip_tags	Retira as tags HTML e PHP de uma string
stripos	Encontra a primeira ocorrência de uma string sem diferenciar maiúsculas e minúsculas
stristr	stristr sem diferenciar maiúsculas e minúsculas
strlen	Retorna o tamanho de uma string
strpos	Encontra a posição da primeira ocorrência de uma string
strrchr	Encontra a ultima ocorrência de um caractere em uma string

Outras funções de String

strrev	Reverte uma string
strripos	Encontra a posição da última ocorrência de uma string case-insensitive em uma string
strrpos	Encontra a posição da última ocorrência de um caractere em uma string
strstr	Encontra a primeira ocorrência de uma string
strtolower	Converte uma string para minúsculas
strtoupper	Converte uma string para maiúsculas
substr	Retorna uma parte de uma string
substr_compare	A comparação binária entre duas strings de um offset até o limite do comprimento de caracteres
substr_count	Conta o número de ocorrências de uma substring
substr_replace	Substitui o texto dentro de uma parte de uma string
trim	Retira espaço no início e final de uma string
ucfirst	Converte para maiúscula o primeiro caractere de uma string
ucwords	Converte para maiúsculas o primeiro caractere de cada palavra

Mais sobre arrays e matrizes

- \$notas = [7, 5, 6, 4, 8, 2, 9];
- \$notas = array(7, 5, 6, 4, 8, 2, 9);

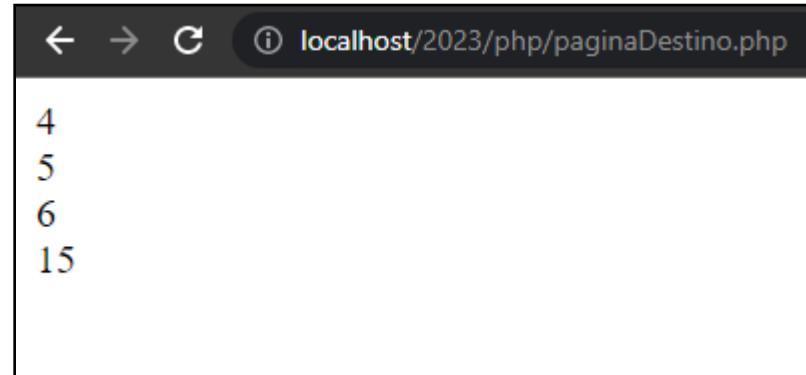
```
echo $notas[0]; // exibe 5  
echo $notas[1]; // exibe 6  
echo $notas[2]; // exibe 4  
echo $notas[3]; // exibe 8  
echo $notas[4]; // exibe 2  
echo $notas[5]; // exibe 9
```

Criando arrays associativos no PHP

```
<?php
$notas = [
    "joana" => 5,
    "maria" => 6,
    "ana" => 4,
    "mariana" => 8,
    "luciana" => 2,
    "tatiana" => 9
];
echo $notas["maria"];
?>
```

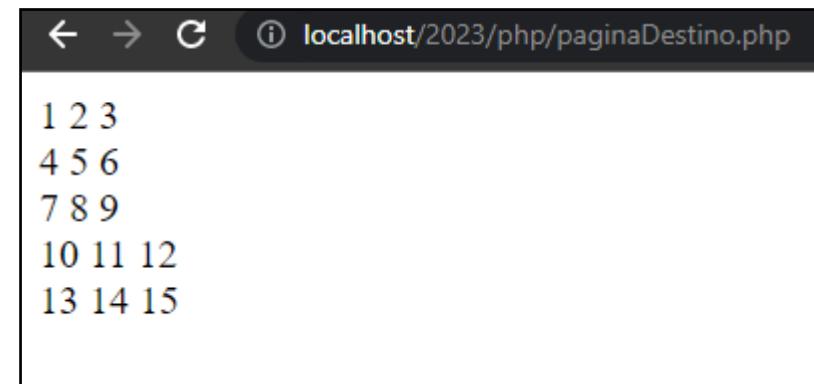
Trabalhando com Array multidimensional do PHP

```
<?php
$alunos = [
    'Ana' => [
        'nota-provas' => [4, 5, 6],
        'nota-final' => 5,
        'faltas' => 15 ],
    'Mariana' => [
        'nota-provas' => [7, 5, 6],
        'nota-final' => 6,
        'faltas' => 8 ]
];
echo $alunos['Ana']['nota-provas'][0]."<br>";
echo $alunos['Ana']['nota-provas'][1]."<br>";
echo $alunos['Ana']['nota-provas'][2]."<br>";
echo $alunos['Ana']['faltas'];
?>
```



Matrizes em php

```
<?php  
$matriz = array(  
    array(1, 2, 3),  
    array(4, 5, 6),  
    array(7, 8, 9),  
    array(10, 11, 12),  
    array(13, 14, 15)  
);  
  
for ($i = 0; $i < 5; $i++) {  
    for ($j = 0; $j < 3; $j++) {  
        echo $matriz[$i][$j] . " ";  
    }  
    echo "<br>";  
}  
?>
```

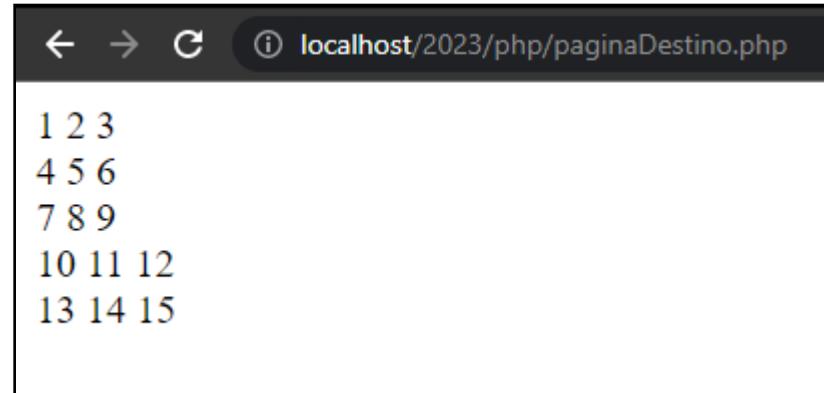


localhost/2023/php/paginaDestino.php

1	2	3
4	5	6
7	8	9
10	11	12
13	14	15

Número de linhas e colunas

```
<?php  
$matriz = array(  
    array(1, 2, 3),  
    array(4, 5, 6),  
    array(7, 8, 9),  
    array(10, 11, 12),  
    array(13, 14, 15)  
);  
$num_linhas = count($matriz);  
$num_colunas = count($matriz[0]);  
for ($i = 0; $i < $num_linhas; $i++) {  
    for ($j = 0; $j < $num_colunas; $j++) {  
        echo $matriz[$i][$j] . " ";  
    }  
    echo "<br>";  
}  
?>
```



A screenshot of a web browser window displaying the output of a PHP script. The address bar shows 'localhost/2023/php/paginaDestino.php'. The page content consists of a 3x3 grid of numbers:

1	2	3
4	5	6
7	8	9

Below the first two rows, there is additional output: '10 11 12' and '13 14 15', which likely represents the remaining elements of the matrix after the first three rows have been printed.