

# INTRODUÇÃO A DOCUMENTOS DINÂMICOS NA WEB EM PHP

Prof. Me. Hélio Esperidião

# 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 o xampp no INICIAR

# Inicie os serviços

XAMPP Control Panel v3.2.2 [ Compiled: Nov 12th 2015 ]

**XAMPP Control Panel v3.2.2**

Modules

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	9672 1388	80, 443	Stop Admin Config Logs
<input type="checkbox"/>	MySQL	10668	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

13:33:22 [main] All prerequisites found  
13:33:22 [main] Initializing Modules  
13:33:22 [main] Starting Check-Timer  
13:33:22 [main] Control Panel Ready  
13:33:30 [Apache] Attempting to start Apache app...  
13:33:30 [Apache] Status change detected: running  
13:33:32 [mysql] Attempting to start MySQL app...  
13:33:32 [mysql] Status change detected: running

Config  
Netstat  
Shell  
Explorer  
Services  
Help  
Quit



# Configurando Servidor

- Endereço para acesso via navegador
  - `http://127.0.0.1/pastaAluno`
  - <http://localhost/pastaAluno>
  - Na escola:
    - `http://localhost:8080/pastaAluno`
- Onde encontro estes endereços no meu computador?
  - `C:\xampp\htdocs\pastaAluno`
- 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");
?>
```

localhost/2023/php/p1.php

Olá mundo  
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.

Arquivo | C:/xampp/htdocs/2023/php/p1.php

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

# 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>";

?>
```

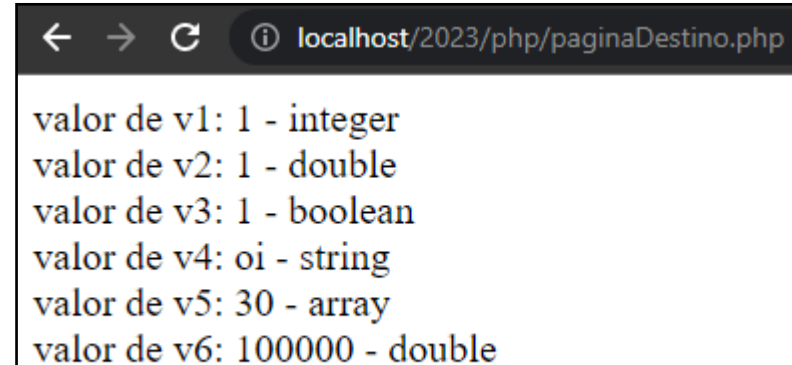
localhost/2023/php/paginaDestino.php

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)."<br>";
echo "valor de v2: $v2 - ". gettype($v2)."<br>";
echo "valor de v3: $v3 - ". gettype($v3)."<br>";
echo "valor de v4: $v4 - ". gettype($v4)."<br>";
echo "valor de v5: $v5[2] - ". gettype($v5)."<br>";
echo "valor de v6: $v6 - ". gettype($v6)."<br>";
?>
```



```
← → ↻ ⓘ localhost/2023/php/paginaDestino.php
valor de v1: 1 - integer
valor de v2: 1 - double
valor de v3: 1 - boolean
valor de v4: oi - string
valor de v5: 30 - array
valor de v6: 100000 - double
```

# Envio de dados HTTP

## Método POST

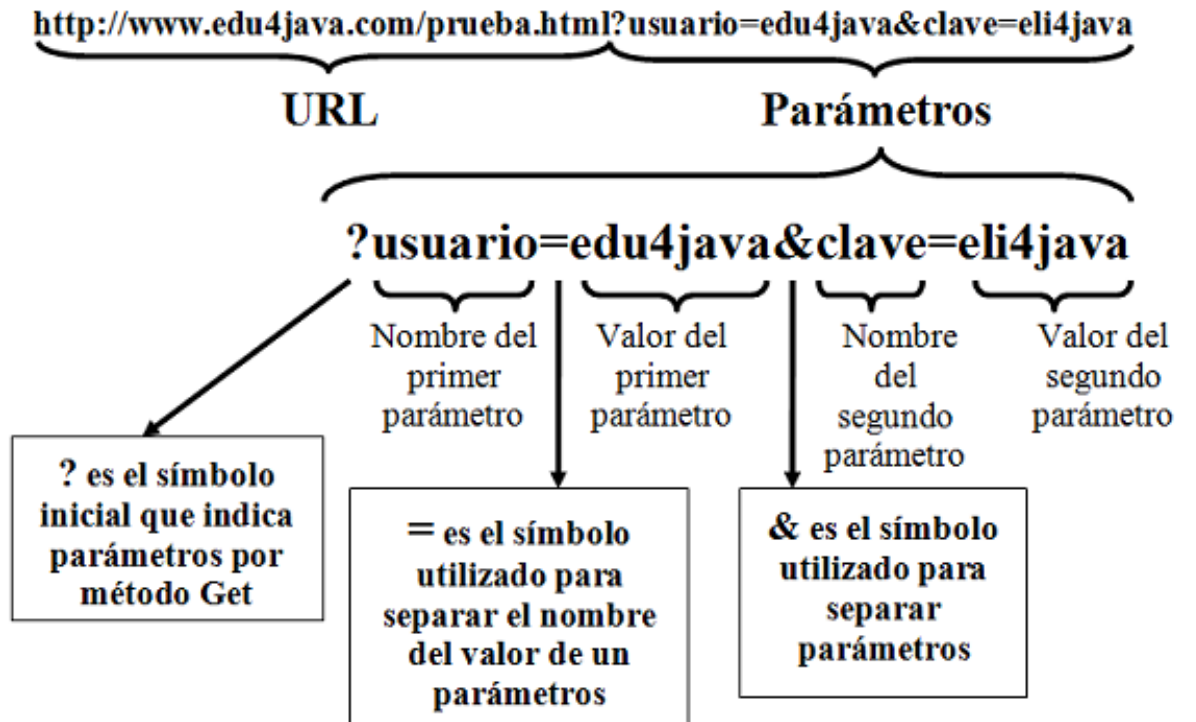
- Encapsula os dados e os envia junto do cabeçalho http.

## Método GET

- Os dados são enviados por meio da url.



# 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>

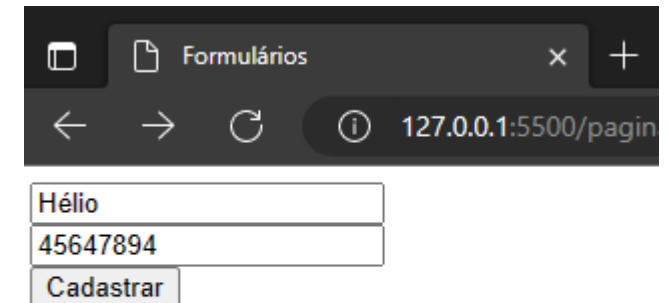
</body>
</html>
```

# Caixa de texto

```
<!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>
```

- <http://127.0.0.1:5500/paginaDestino.php?txtNome=Helio&txtRg=45647894>

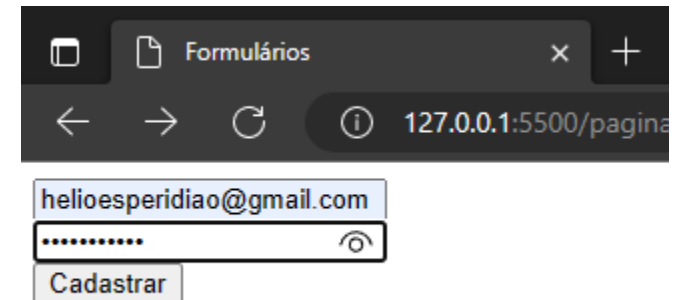


# Password

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

</body>
</html>
```

- <http://127.0.0.1:5500/paginaDestino.php?txtEmail=helioesperidiao@gmail.com&txtSenha=batataFrita>

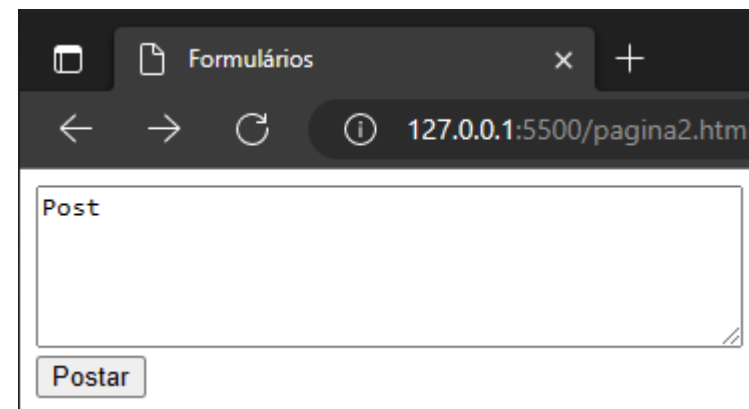


# textarea

```
<!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>
```

- <http://127.0.0.1:5500/paginaDestino.php?txtPost=Post>

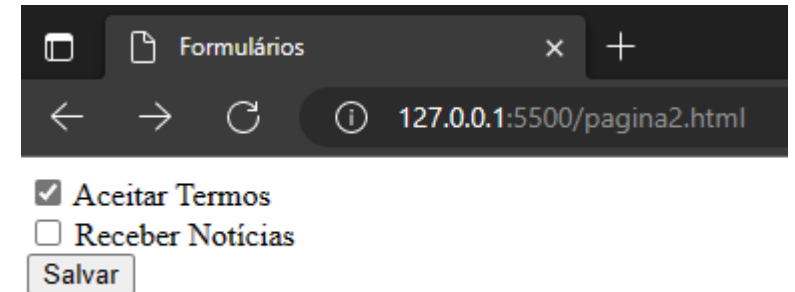


# checkbox

```
<!DOCTYPE html>
<html>
<head>
  <title>Formulários</title>
</head>
<body>
  <form action="paginaDestino.php" method="get">
    <input type="checkbox" name="cboTermos" > Aceitar Termos<br>
    <input type="checkbox" name="cboNoticias" > Receber Notícias<br>
    <input type="submit" value="Salvar">
  </form>

</body>
</html>
```

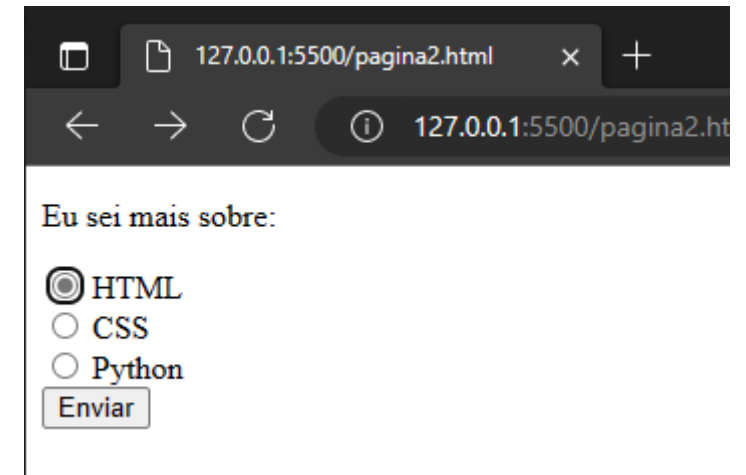
- <http://127.0.0.1:5500/paginaDestino.php?cboTermos=on>



# radio

<http://127.0.0.1:5500/paginaDestino.php?rdoFavorito=HTML>

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

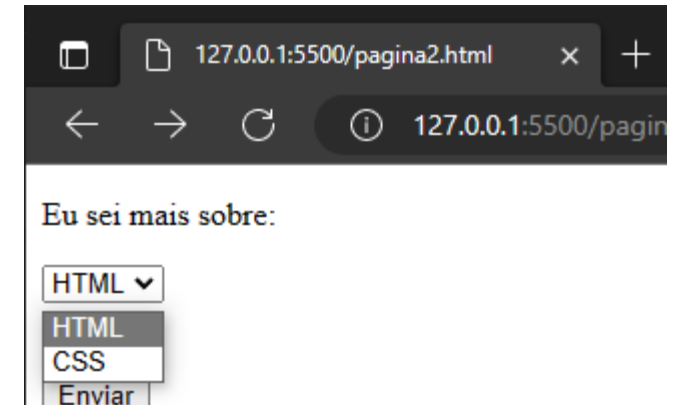




# Caixa de seleção

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

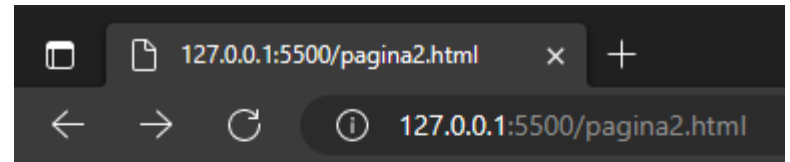
http://127.0.0.1:5500/paginaDestino.php?cboLinguagem=HTML



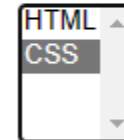
# Caixa de lista

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

- <http://127.0.0.1:5500/paginaDestino.php?IstLinguagem=CSS>



Eu sei mais sobre:



Enviar

# Entrada e saída dados.

---

O php pode receber dados via get ou via post.

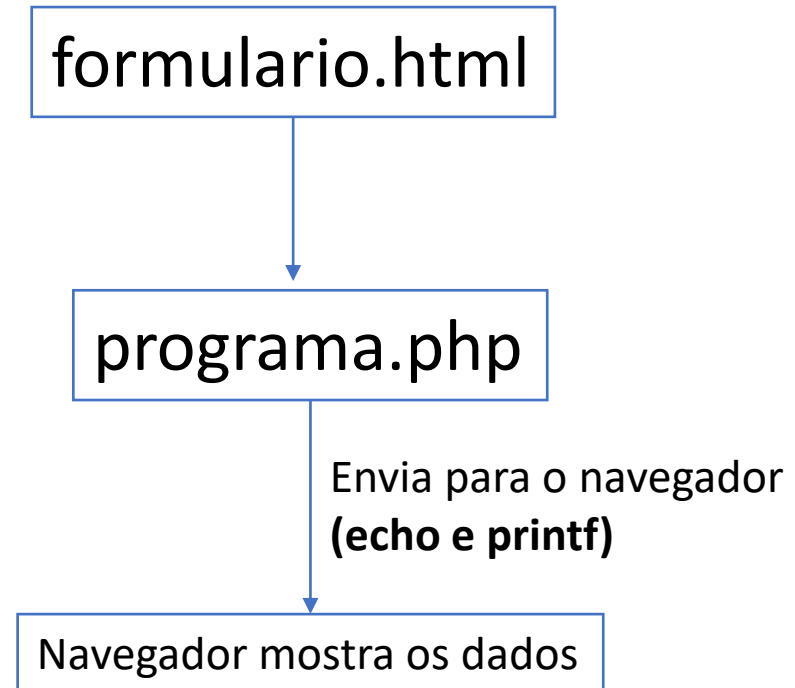
O `$_GET` e o `$_POST` são dois arrays associativos que contem os dados enviados para o script (servidor)

# 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>

# Arquitetura de arquivos

- Durante as aulas de c e durante as aulas de lógica para programação os programas trabalhavam com um único arquivo.
- Sites e aplicações para web são constituídos por muitos arquivos. Veja ao lado a arquitetura base para uma aplicação web simples.



# Caixa de texto

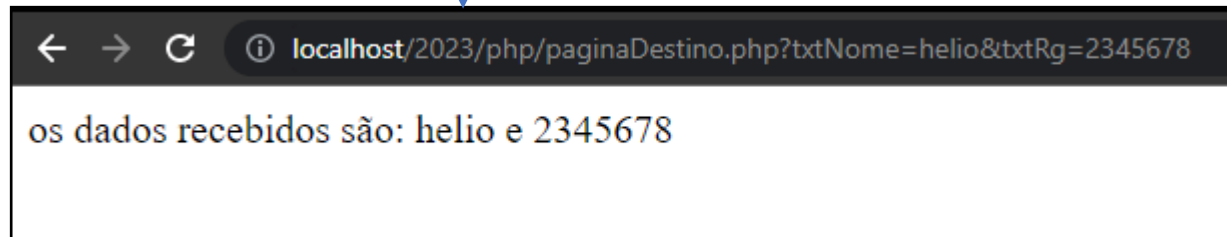
## Formulario.html

```
<!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";
?>
```

**echo** escreve os dados no documento que será enviado para o navegador



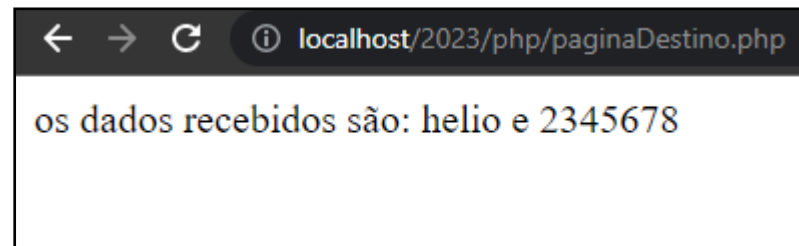
# Caixa de texto

## 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
.=	Concatena o valor à direita à variável à esquerda e atribui o resultado à variável à esquerda



# Operadores relacionais

$\$a == \$b$	Igual	Verdadeiro ( <b>true</b> ) se $\$a$ é igual a $\$b$ .
$\$a === \$b$	Idêntico	Verdadeiro ( <b>true</b> ) se $\$a$ é igual a $\$b$ , e eles são do mesmo tipo.
$\$a != \$b$	Diferente	Verdadeiro se $\$a$ não é igual a $\$b$ .
$\$a <> \$b$	Diferente	Verdadeiro se $\$a$ não é igual a $\$b$ .
$\$a !== \$b$	Não idêntico	Verdadeiro se $\$a$ não é igual a $\$b$ , ou eles não são do mesmo tipo (introduzido no PHP4).
$\$a < \$b$	Menor que	Verdadeiro se $\$a$ é estritamente menor que $\$b$ .
$\$a > \$b$	Maior que	Verdadeiro se $\$a$ é estritamente maior que $\$b$ .
$\$a <= \$b$	Menor ou igual	Verdadeiro se $\$a$ é menor ou igual a $\$b$ .
$\$a >= \$b$	Maior ou igual	Verdadeiro se $\$a$ é maior ou igual a $\$b$ .

# 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

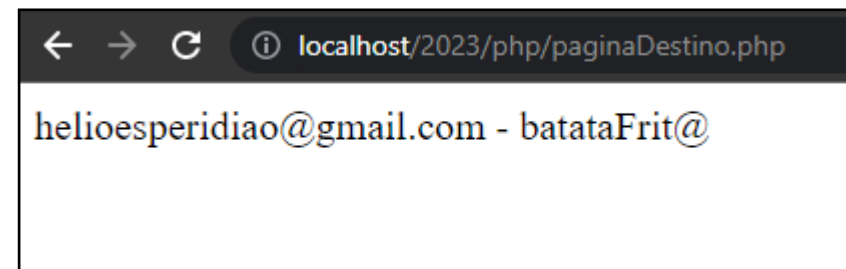
## Front.html

```
<!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>
```

## paginaDestino.php

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



# textarea

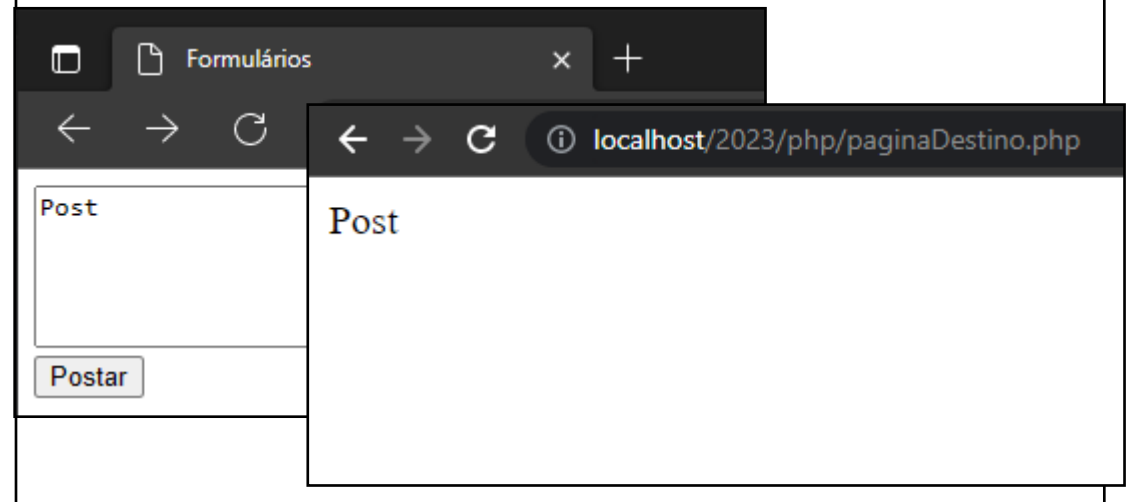
## Exemplo03.html

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

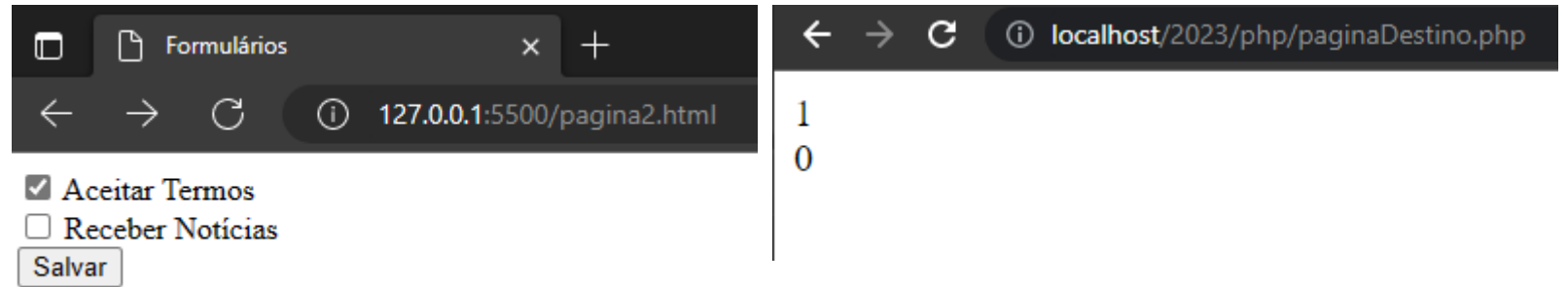
</body>
</html>
```

## paginaDestino.php

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



# checkbox



## Formulario.html

```
<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>
```

## paginaDestino.php

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

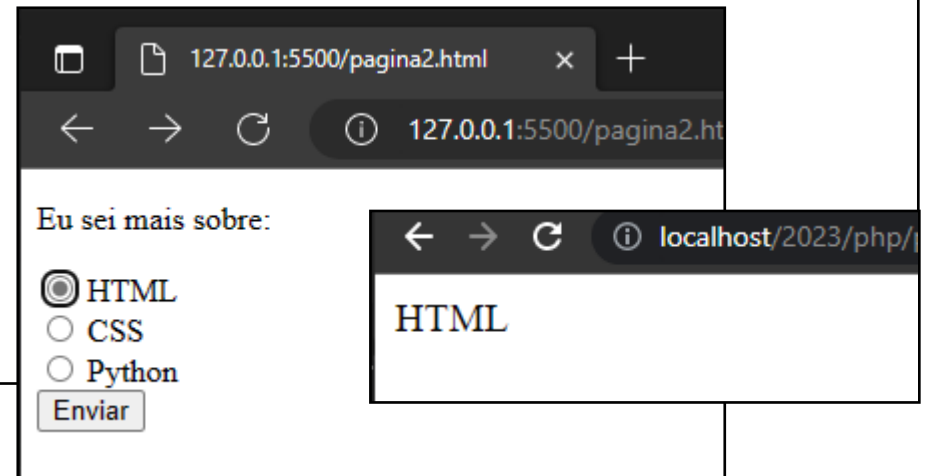
# radio

## front.html

```
<!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>
```

## paginaDestino.php

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



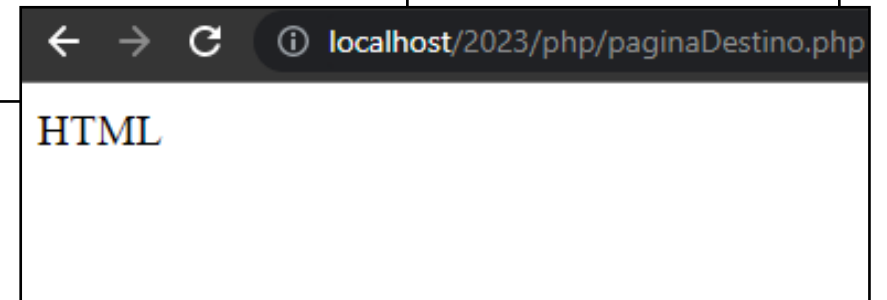
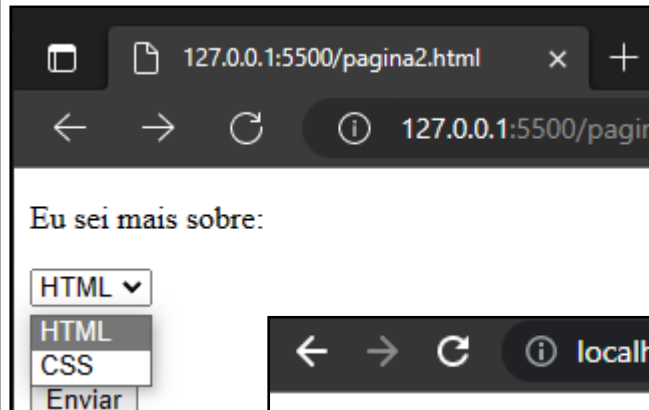
# Caixa de seleção

## Front.html

```
<!DOCTYPE html>
<html>
<body>
<form action="back.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>
```

## back.php

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





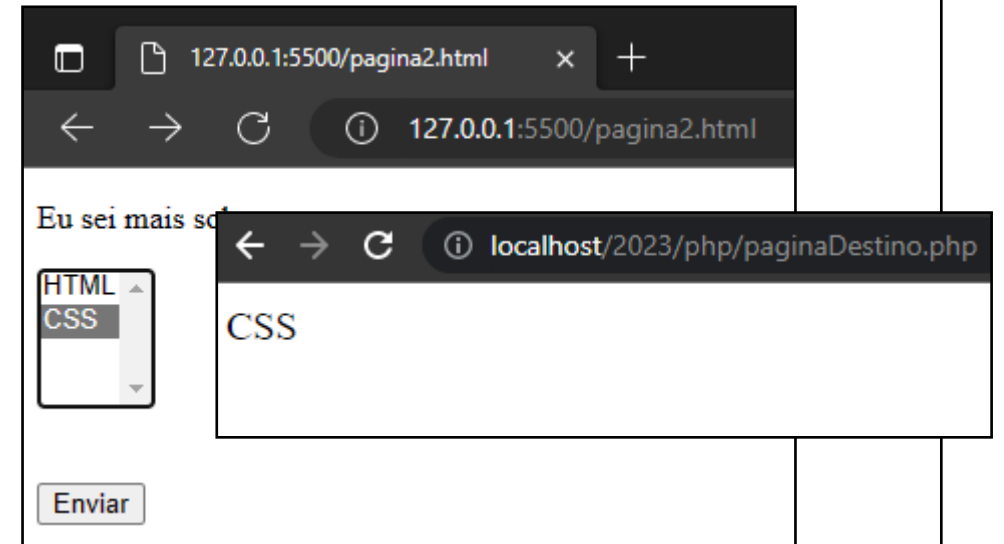
# Caixa de lista

## Front.html

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

## paginaDestino.php

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



# Enviado dados pela URL – Método GET

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

# 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";
  }
?>
```

```
← → ↻ ⓘ localhost/2023/php/paginaDestino.php?txtNota1=6&txtNota2=7
```

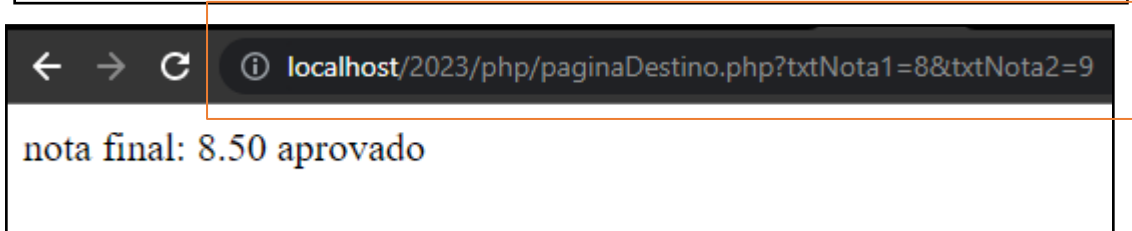
```
nota final: 6.50 aprovado
```

# 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>

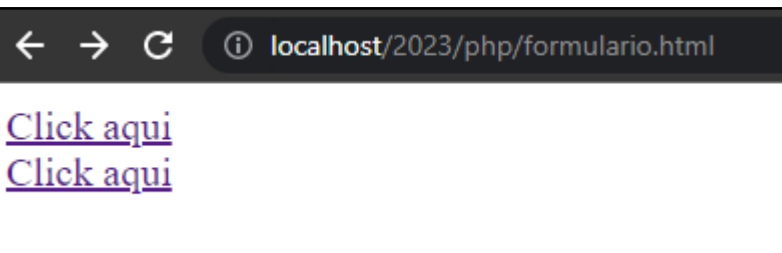
<http://localhost/2023/php/paginaDestino.php?txtNota1=8&txtNota2=9>



```
<?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>
```



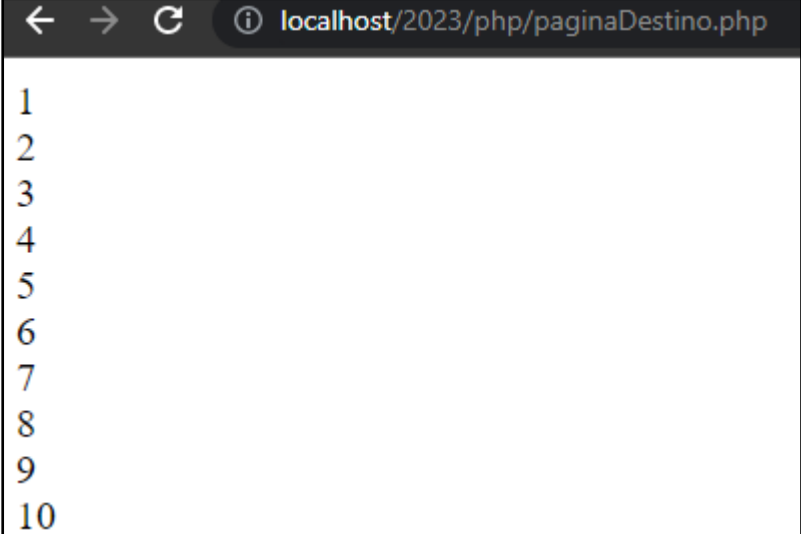
Observe que os hiperlinks enviam dados no formato get para “paginaDestino.php”.

```
<?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;  
}
```

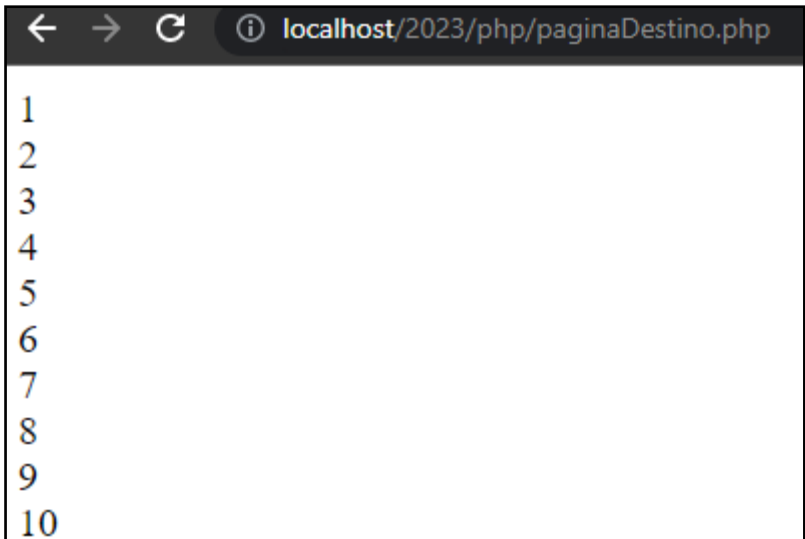
```
<?php  
    $num = 1;  
    while ($num <= 10) {  
        echo "$num <br>";  
        $num++;  
    }  
?>
```



```
localhost/2023/php/paginaDestino.php  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

# Estruturas de repetição em php (do-while)

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



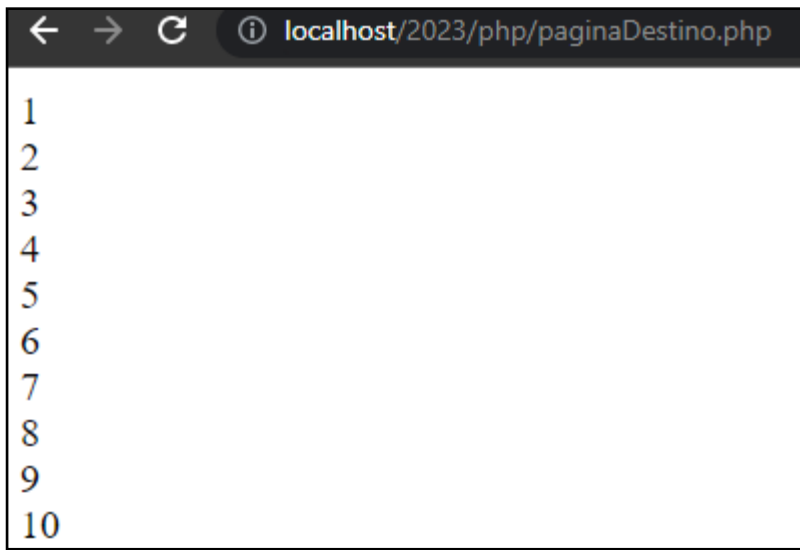
A screenshot of a web browser window. The address bar shows the URL 'localhost/2023/php/paginaDestino.php'. The main content area of the browser displays the numbers 1 through 10, each on a new line, representing the output of a PHP do-while loop.

```
<?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>";  
}  
?>
```



localhost/2023/php/paginaDestino.php

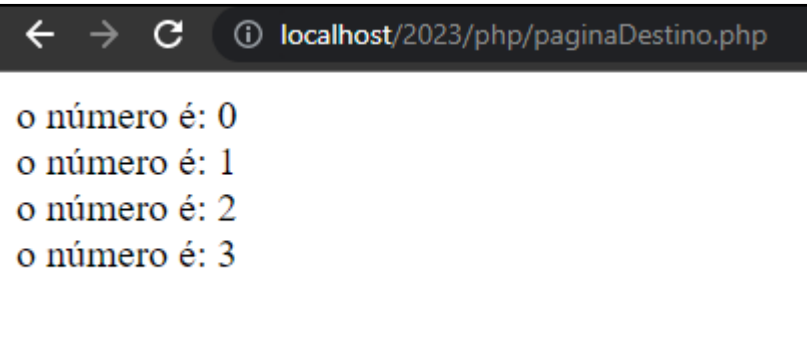
1  
2  
3  
4  
5  
6  
7  
8  
9  
10



# 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>";
}
?>
```

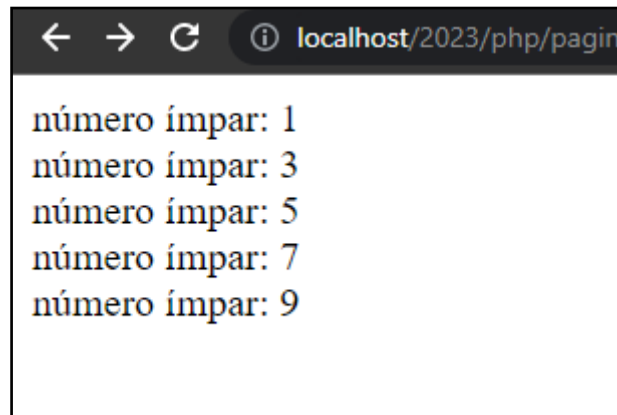


```
o número é: 0
o número é: 1
o número é: 2
o número é: 3
```

# 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>";
}
?>
```



A screenshot of a web browser window. The address bar shows 'localhost/2023/php/pagin'. The main content area displays the output of the PHP code: 'número ímpar: 1', 'número ímpar: 3', 'número ímpar: 5', 'número ímpar: 7', and 'número ímpar: 9', each on a new line.

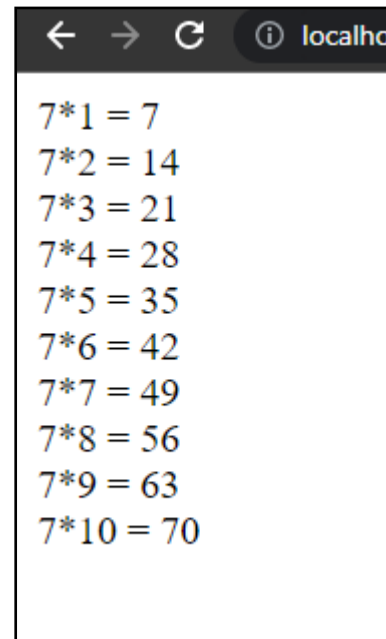
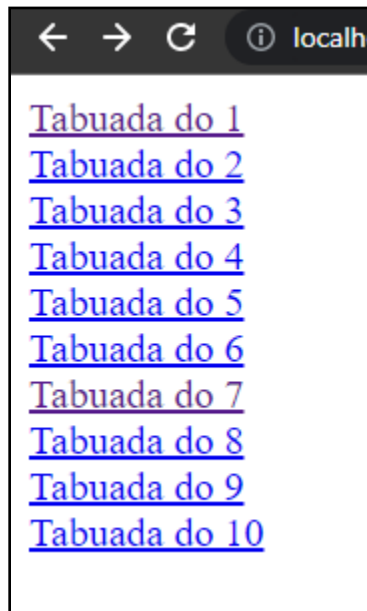
# Utilizando get sem forms

```
<!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>

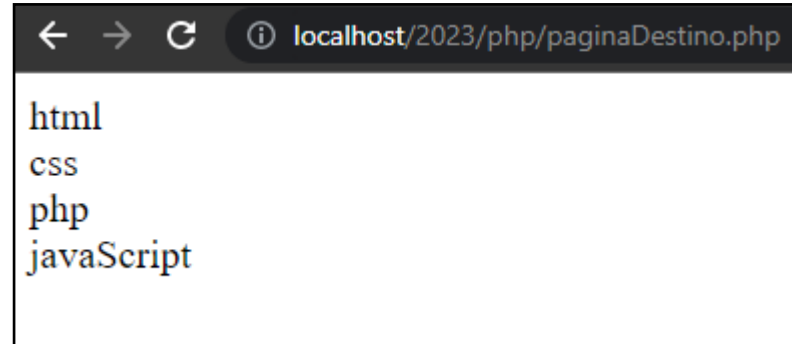


```
<?php
$valor = $_GET['v1'];
$num = 1;
while ($num <= 10) {
    $tabuada = $valor*$num;
    echo "$valor*$num = $tabuada<br>";
    $num++;
}
?>
```

# 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>";  
}  
?>
```



A screenshot of a web browser window. The address bar shows the URL 'localhost/2023/php/paginaDestino.php'. The main content area of the browser displays the output of the PHP code: 'html', 'css', 'php', and 'javaScript', each on a new line.

# Array associativo

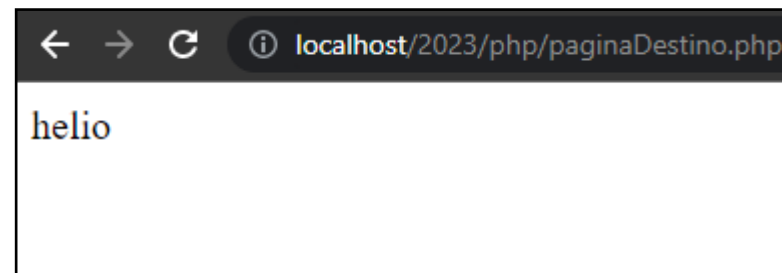
É um tipo de array onde os índices podem ser utilizados como textos.

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'];

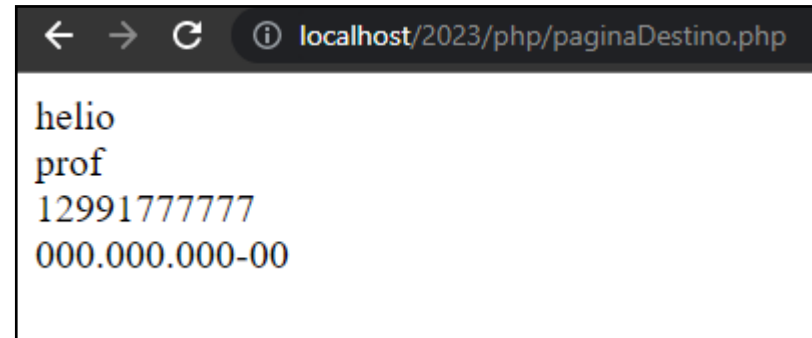
?>
```



# 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'];
?>
```

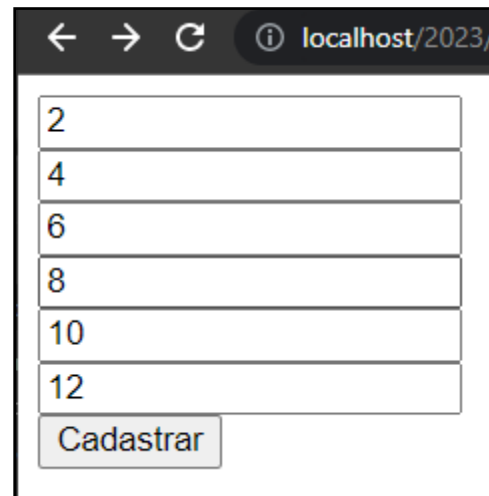
- `$_POST` é um array, existem as posições **'txtNota1', 'txtNota2', etc.**



# \$\_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>
```

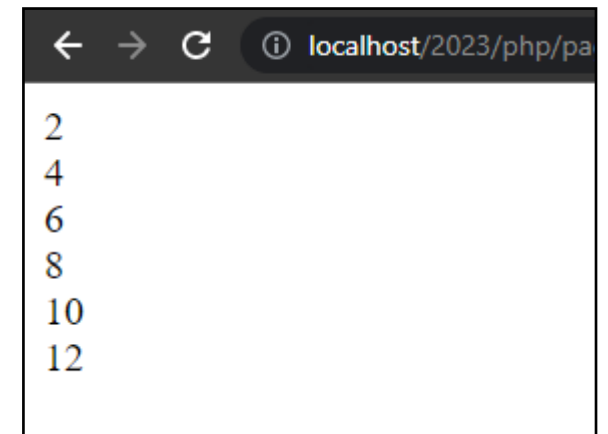
```
<?php
foreach ($_POST as $posicao) {
  echo "$posicao <br>";
}
?>
```



localhost/2023/p

2
4
6
8
10
12

Cadastrar



localhost/2023/php/pa

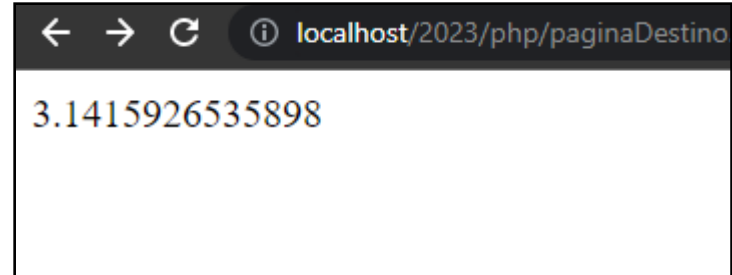
2  
4  
6  
8  
10  
12

# 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;
?>
```



A screenshot of a web browser window. The address bar shows the URL `localhost/2023/php/paginaDestino`. The main content area of the browser displays the numerical value `3.1415926535898`, which is the value of the mathematical constant  $\pi$ .

# Funções matemáticas

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

# Funções matemáticas

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

# Funções matemáticas

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

# Funções matemáticas

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

# Funções matemáticas

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



# Exemplo de uso de funções matemáticas.

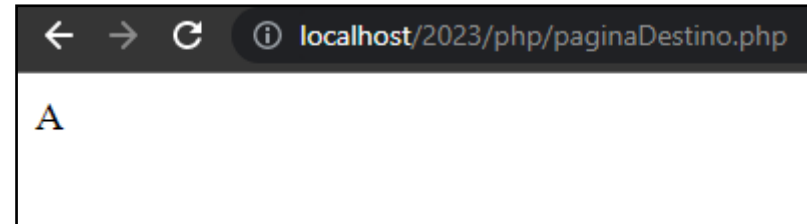
```
<?php  
    $x= sqrt(4);  
    echo $x;  
?>
```

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

Source: [www.LookupTables.com](http://www.LookupTables.com)

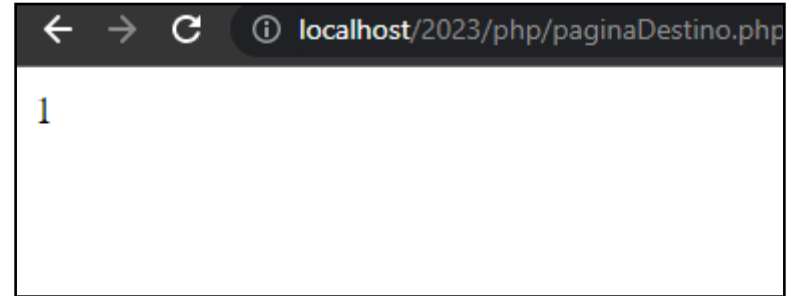
# exemplo

```
<?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
```

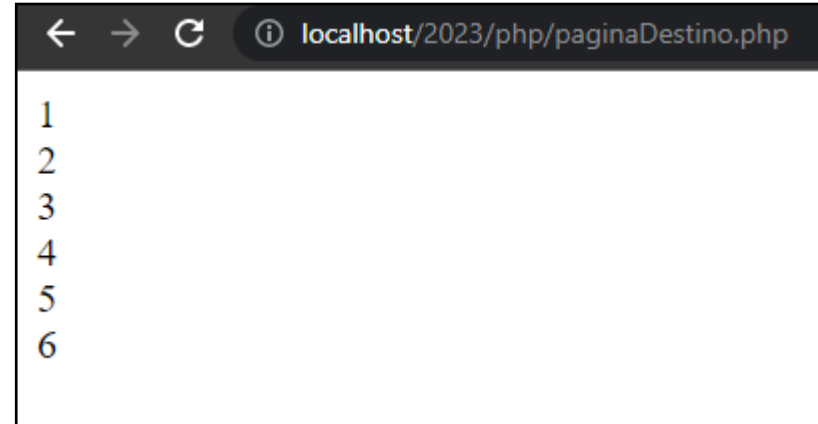
# explode - Divide uma String em um vetor

```
<?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>";
    }
?>
```

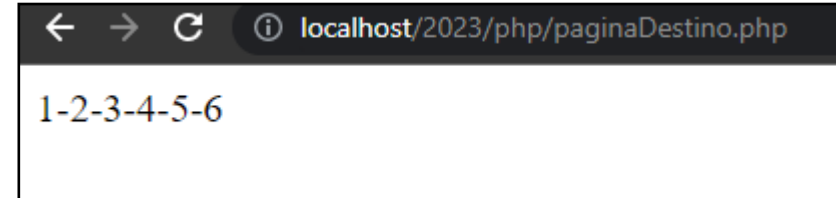


A screenshot of a web browser window. The address bar shows the URL `localhost/2023/php/paginaDestino.php`. The main content area of the browser displays a vertical list of numbers from 1 to 6, each on a new line, representing the output of the PHP code shown in the adjacent code block.

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.

**Password**



Insecure

**MD5  
Hash**



Secure

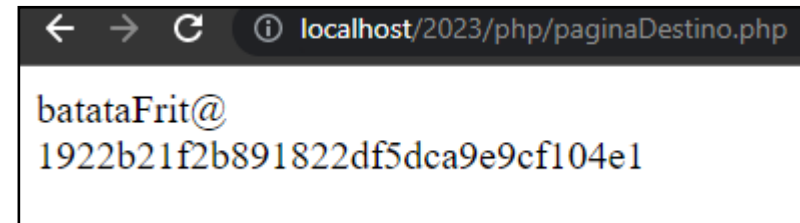
**dc647eb65e6711e155375218212b3964**

# MD5

```
<?php
$email = $_POST['txtEmail'];
$senha = $_POST['txtSenha'];

$senhaMd5 = md5($senha);
echo "$senha <BR>$senhaMd5" ;

?>
```



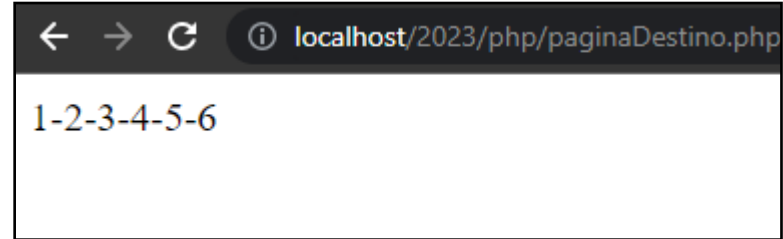
# str\_replace – Substitui um conjunto de caracteres por outro.

```
<?php
    $string = "1;2;3;4;5;6";

    $str2 = str_replace(";", "-", $string);

    echo $str2 ;

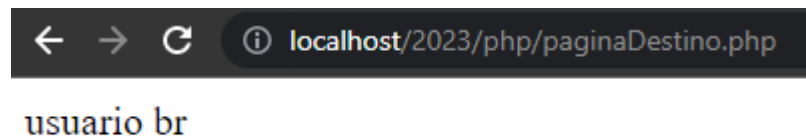
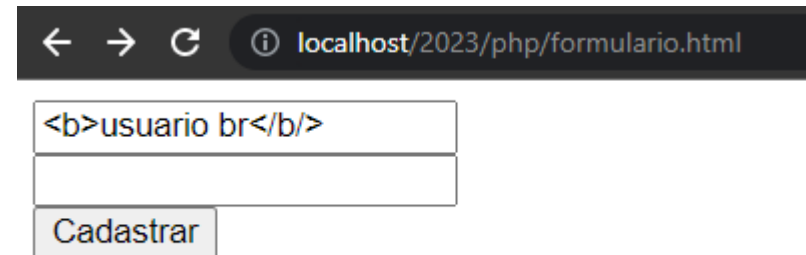
?>
```



# 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 <b>



# 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

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

# Outras funções de String

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

# Outras funções de String

<code>strrev</code>	Reverte uma string
<code>strrpos</code>	Encontra a posição da última ocorrência de uma string case-insensitive em uma string
<code>strrpos</code>	Encontra a posição da última ocorrência de um caractere em uma string
<code>strstr</code>	Encontra a primeira ocorrência de uma string
<code>strtolower</code>	Converte uma string para minúsculas
<code>strtoupper</code>	Converte uma string para maiúsculas
<code>substr</code>	Retorna uma parte de uma string
<code>substr_compare</code>	A comparação binária entre duas strings de um offset até o limite do comprimento de caracteres
<code>substr_count</code>	Conta o número de ocorrências de uma substring
<code>substr_replace</code>	Substitui o texto dentro de uma parte de uma string
<code>trim</code>	Retira espaço no início e final de uma string
<code>ucfirst</code>	Converte para maiúscula o primeiro caractere de uma string
<code>ucwords</code>	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]; // exhibe 5  
echo $notas[1]; // exhibe 6  
echo $notas[2]; // exhibe 4  
echo $notas[3]; // exhibe 8  
echo $notas[4]; // exhibe 2  
echo $notas[5]; // exhibe 9
```

# Criando arrays associativos no PHP

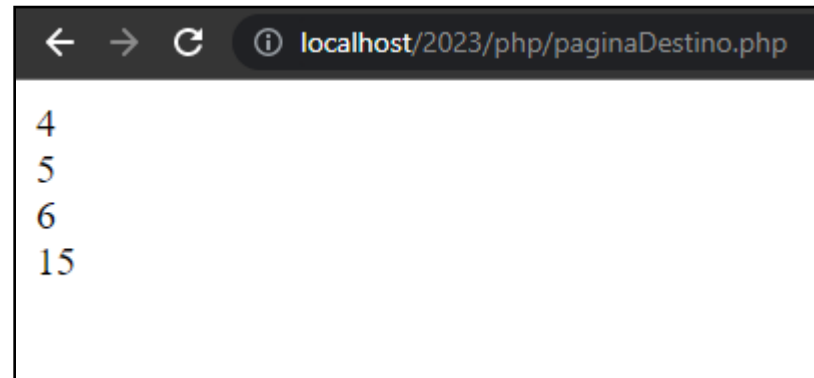
```
<?php
$notas = [
    "joana" => 5,
    "maria" => 6,
    "ana" => 4,
    "mariana" => 8,
    "luciana" => 2,
    "tatiana" => 9
];

echo $notas["maria"];

?>
```

# Trabalhando com Array multidimensional do PHP

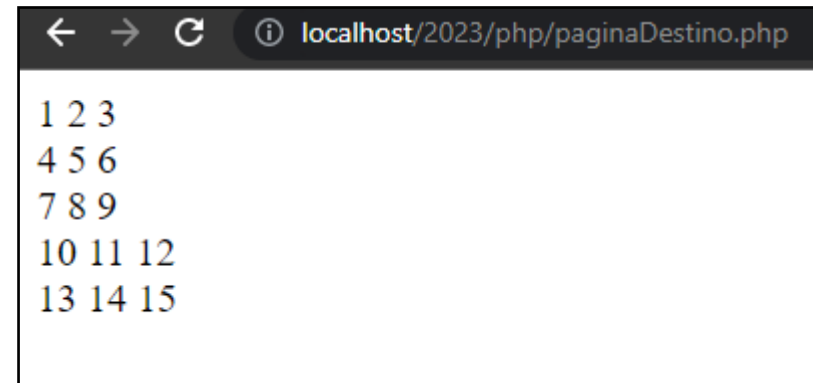
```
<?php
$alunos = [
    'Ana' => [
        'nota-provas' => [4, 5, 6],
        'nota-final' => 5,
        'faltas' => 15 ],
    'Mariana' => [
        'nota-provas' => [7, 5, 6],
        'nota-final' => 6,
        'faltas' => 8 ]
];
echo $alunos['Ana']['nota-provas'][0]."<br>";
echo $alunos['Ana']['nota-provas'][1]."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



A screenshot of a web browser window. The address bar shows the URL `localhost/2023/php/paginaDestino.php`. The main content area of the browser displays the output of the PHP code: the numbers 4, 5, 6, and 15, each on a new line.

# 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>";
}
?>
```

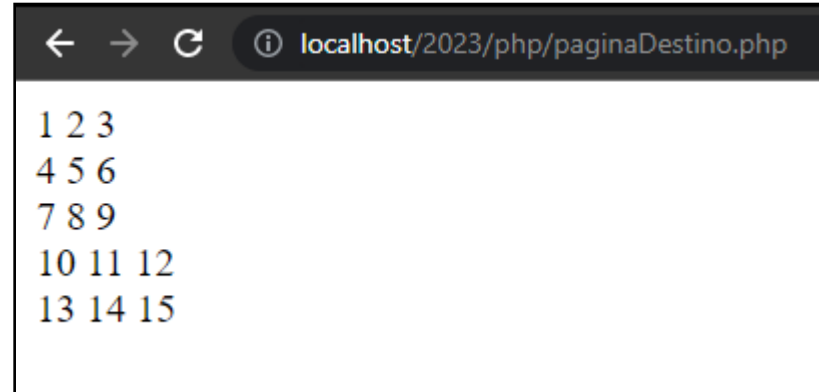


A screenshot of a web browser window. The address bar shows "localhost/2023/php/paginaDestino.php". The main content area displays the output of the PHP code: a 5x3 matrix of numbers. The numbers are arranged in five rows and three columns, with spaces between them. The rows are: 1 2 3, 4 5 6, 7 8 9, 10 11 12, and 13 14 15.

```
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>";
}
?>
```



```
localhost/2023/php/paginaDestino.php
1 2 3
4 5 6
7 8 9
10 11 12
13 14 15
```