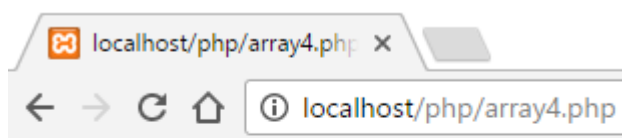


Administração Central

Cetec Capacitações



José
Pedro

Nome removido: Maria

10.4.2 Função array_pop()

Esta função remove e retorna o item encontrado no final do vetor.

```
<?php

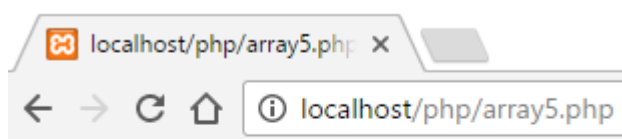
$nomes = array ("Maria", "José", "Pedro");
$excluido = array_pop ($nomes);

foreach ($nomes as $item) {
    echo "<br />".$item;
}

echo "<br /><br /> Nome removido: ".$excluido;

?>
```

Podemos notar, que o nome Pedro que constava na última posição do vetor, foi removido.



Maria
José

Nome removido: Pedro

Administração Central

Cetec Capacitações

10.5 Localizando elementos no vetor

10.5.1 Função `in_array()`

Esta função permite buscar um elemento em um vetor, retornando *TRUE* ou *FALSE*.

```
<?php

    $nomes = array ("Maria", "José", "Pedro", "Ana");

    in_array ("Pedro", $nomes){
        .....
        echo "<br /> Nome encontrado no vetor";
    }

?>
```

10.5.2 Função `array_key_exists()`

Esta função permite a busca de um elemento usando como parâmetro a chave de associação, sendo que retorna *TRUE* se a chave for localizada.

```
<?php

    $nomes = array ("MA"=>"Maria", "ZE"=>"José", "PE"=>"Pedro", "AN"=>"Ana");

    if(array_key_exists ("ZE", $nomes)){
        .....
        echo "<br /> Nome encontrado no vetor, através da chave";
    }

?>
```

10.5.3 Função `array_search()`

Esta função busca um valor no array e retorna a chave se for localizado este valor.

Administração Central

Cetec Capacitações

```
<?php

$nomes = array ("MA"=>"Maria", "ZE"=>"José", "PE"=>"Pedro", "AN"=>"Ana");
$chave = array_search("Ana", $nomes);

echo "<br /> Chave da busca realizada: ".$chave;

?>
```

10.5.4 Função array_keys()

Esta função retorna um array formado por todas as chaves localizadas de outro array.

```
<?php

$nomes = array ("MA"=>"Maria", "ZE"=>"José", "PE"=>"Pedro", "AN"=>"Ana");

$chaves = array_keys($nomes);

foreach ($chaves as $item){
    echo "<br />".$item;
}
echo "<br />";

//função para exibir dados de um vetor
print_r ($chaves);

?>
```

10.5.5 Função array_values()

Esta função retorna um array com todos os valores localizados de outro array (sem as chaves associadas).

```
<?php

$nomes = array ("MA"=>"Maria", "ZE"=>"José", "PE"=>"Pedro", "AN"=>"Ana");

$valores = array_values($nomes);

print_r ($valores);

?>
```

Administração Central

Cetec Capacitações

10.6 Movendo o ponteiro de um array

Existem algumas funções para tornar possível o deslocamento do ponteiro de um vetor.

As funções são:

- **next()** retorna o valor do próximo elemento do array na posição seguinte ao ponteiro.
- **prev()** retorna o valor anterior da atual posição do ponteiro.
- **reset()** volta ao começo do array.
- **end()** vai para o final do array.

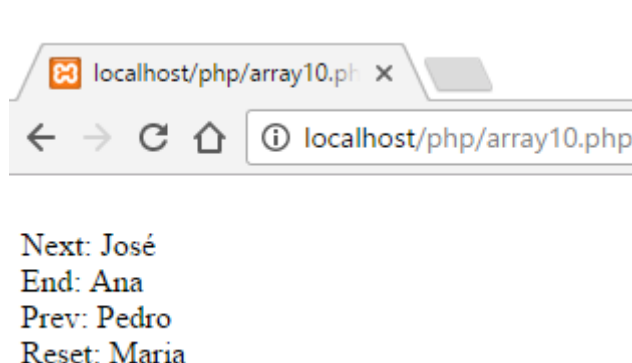
```
<?php

$nomes = array ("MA"=>"Maria", "ZE"=>"José", "PE"=>"Pedro", "AN"=>"Ana");

echo "<br />Next: ".next($nomes);
echo "<br />End: ".end($nomes);
echo "<br />Prev: ".prev($nomes);
echo "<br />Reset: ".reset($nomes);

?>
```

Resultado apresentado no navegador:



10.7 Recuperando a chave de um ponteiro

A função **key()** retorna a chave da posição atual do ponteiro.

Administração Central

Cetec Capacitações

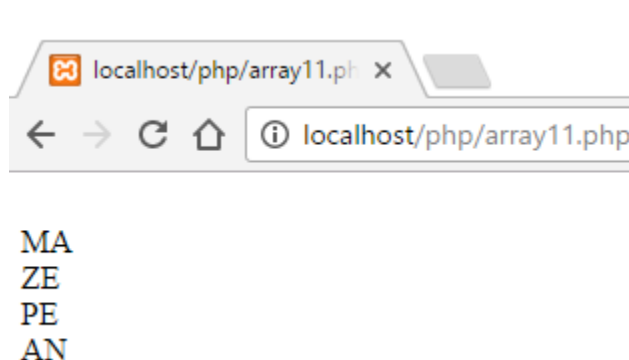
```
<?php

$nomes = array ("MA"=>"Maria", "ZE"=>"José", "PE"=>"Pedro", "AN"=>"Ana");

while ($chave = key($nomes)) {
    echo "<br />".$chave;
    next($nomes);
}

?>
```

O resultado apresentado no navegador, serão as chaves de cada elemento do vetor.



10.8 Classificando arrays

10.8.1 Função array_reverse()

Esta função reverte a ordem os elementos do vetor, ou seja, retorna o valor de forma invertida (chaves e valores).

```
<?php

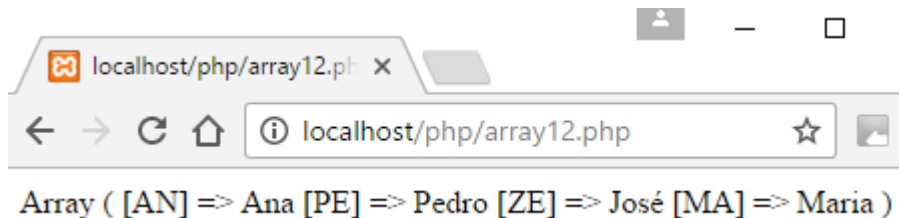
$nomes = array ("MA"=>"Maria", "ZE"=>"José", "PE"=>"Pedro", "AN"=>"Ana");

$reverso = array_reverse($nomes);

print_r ($reverso);

?>
```

Administração Central
Cetec Capacitações



10.8.2 A função sort()

Esta função classifica um array ordenando do menor para o maior.

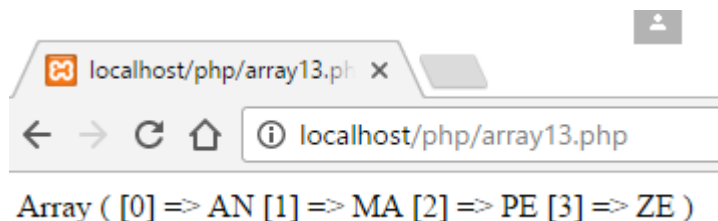
```
<?php

$nomes = array ("MA", "ZE", "PE", "AN");

sort($nomes);

print_r ($nomes);

?>
```



10.8.3 Função rsort()

Esta função classifica um array ordenando do maior para o menor.

```
<?php

$nomes = array ("MA", "ZE", "PE", "AN");

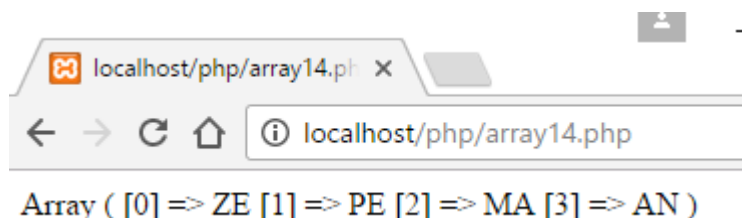
rsort($nomes);

print_r ($nomes);

?>
```

Administração Central

Cetec Capacitações



10.8.4 Função natsort()

Esta função organiza o array de forma natural, testando também sequência numéricas em string.

```
<?php
$arqs = array ("arq1", "arq10", "arq9", "arq11", "arq20", "arq18");
natsort($arqs);
print_r ($arqs);
?>
```



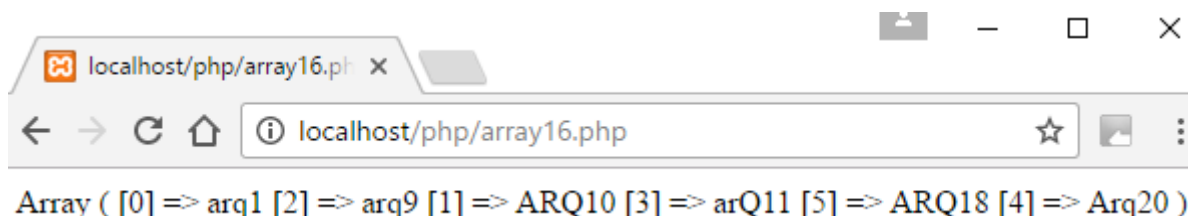
10.8.5 Função natcasesort()

Esta função oferece outro recurso de ordem natural sem fazer a diferenciação entre as letras maiúsculas e minúsculas.

```
<?php
$arqs = array ("arq1", "ARQ10", "arq9", "arQ11", "Arq20", "ARQ18");
natcasesort($arqs);
print_r ($arqs);
?>
```

Administração Central

Cetec Capacitações



11 Operações com array

O PHP tem funções que são capazes de realizar algumas operações um pouco mais complexas com arrays, veja algumas delas.

11.1 Função array_merge()

Esta função permite unir arrays, retornando um único array.

```
<?php

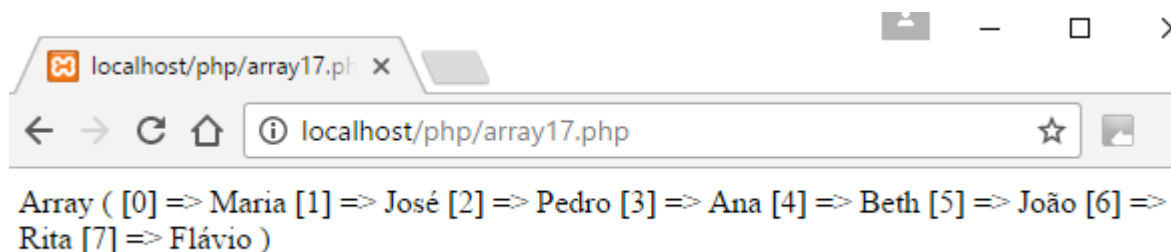
$nomes1 = array ("Maria","José","Pedro","Ana");
$nomes2 = array ("Beth","João","Rita","Flávio");

$nomes_geral = array_merge($nomes1, $nomes2);

print_r ($nomes_geral);

?>
```

Como resultado no navegador, teremos a exibição de todos os nomes que foram unidos em um único vetor.



Administração Central

Cetec Capacitações

11.2 Função array_combine()

Esta função permite criar um novo array realizando uma associação entre chaves e valores.

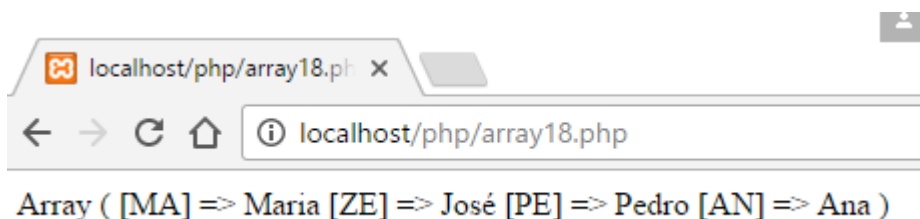
```
<?php

$nomes = array ("Maria", "José", "Pedro", "Ana");
$chaves = array ("MA", "ZE", "PE", "AN");

$geral = array_combine($chaves, $nomes);

print_r ($geral);

?>
```



11.3 Função array_slice()

Esta função permite dividir um array a partir de um *offset* inicial e final, ou seja, extrai um intervalo de um vetor.

```
<?php

$nomes = array ("Maria", "José", "Pedro", "Ana", "Beth", "João", "Rita");

//retorna todos os valores a partir da posição 4 do vetor
$intervalo1 = array_slice($nomes, 4);

//retorna os 2 próximos valores a partir da posição 3 do vetor
$intervalo2 = array_slice($nomes, 3, 2);

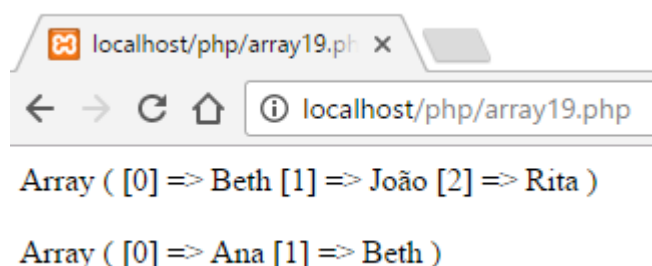
print_r ($intervalo1);

echo "<br /><br />";
print_r ($intervalo2);

?>
```

Administração Central

Cetec Capacitações



11.4 Função array_intersect()

Esta função mantém os valores iguais presentes nos arrays, considerando o primeiro array.

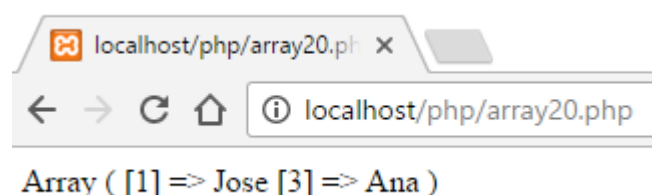
```
<?php

$nomes1 = array ("Maria", "Jose", "Pedro", "Ana");
$nomes2 = array ("Beth", "Ana", "Jose", "Flavio");
$nomes3 = array ("Ana", "Caio", "Jose", "Carla");

//retorna um array contendo todos os valores de nomes1
//que estão presentes nos outros arrays
$interseccao = array_intersect($nomes1, $nomes2, $nomes3);

print_r ($interseccao);

?>
```



11.5 Função array_diff()

Esta função retorna os valores localizados no primeiro array que não existe nos demais.

Administração Central
Cetec Capacitações

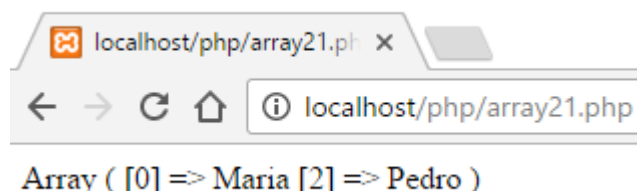
```
<?php

$nomes1 = array ("Maria","Jose","Pedro","Ana");
$nomes2 = array ("Beth","Ana","Jose","Flavio");
$nomes3 = array ("Ana","Caio","Jose","Carla");

//retorna um array contendo todos os valores de nomes1
//que estão presentes nos outros arrays
$diferentes = array_diff($nomes1, $nomes2, $nomes3);

print_r ($diferentes);

?>
```



localhost/php/array21.php X

localhost/php/array21.php

Array ([0] => Maria [2] => Pedro)

11.6 Função shuffle()

Esta função reordena um array aleatoriamente.

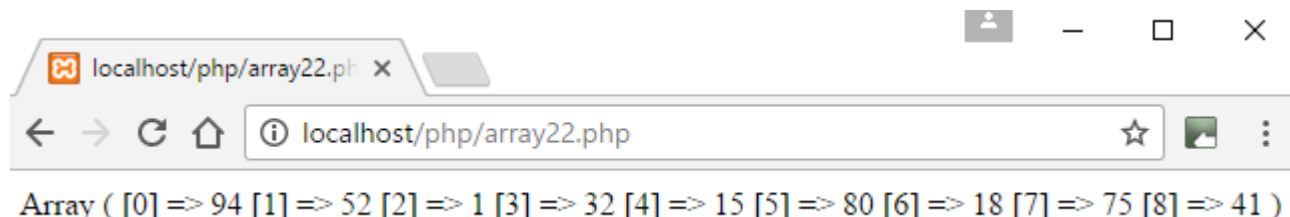
```
<?php

$num = array (1,15,18,32,41,52,75,80,94,);

shuffle($num);

print_r ($num);

?>
```



localhost/php/array22.php X

localhost/php/array22.php

Array ([0] => 94 [1] => 52 [2] => 1 [3] => 32 [4] => 15 [5] => 80 [6] => 18 [7] => 75 [8] => 41)

Administração Central

Cetec Capacitações

11.7 Função array_sum()

Esta função realiza a soma de todos os valores numéricos do array. Caso existe algum elemento do tipo string, o mesmo será desconsiderado.

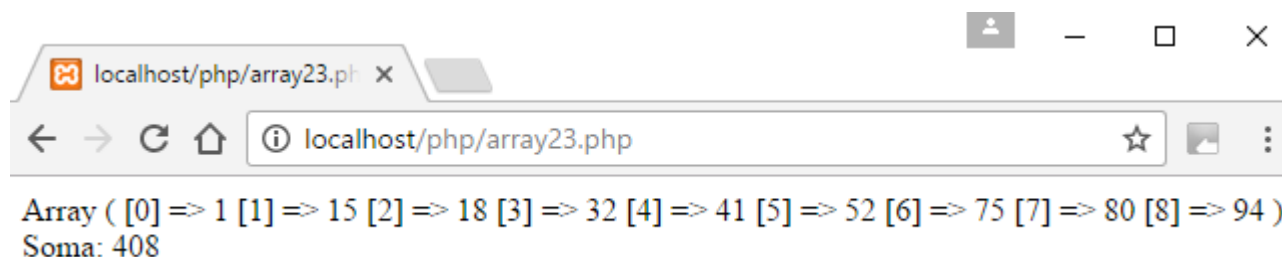
```
<?php

$num = array (1,15,18,32,41,52,75,80,94,);

$soma = array_sum($num);

print_r ($num);
echo "<br />Soma: ".$soma;

?>
```



12 Array Multidimensional

Quando usamos arrays multidimensionais ou matrizes, podemos dizer que temos arrays que contém arrays dentro deles. Quando utilizamos arrays multidimensionais (matrizes) existem mais índices numa mesma variável, ou seja:

- **Array comum:** \$vetor []
- **Array bidimensional:** \$matriz [][]
- **Array tridimensional:** \$matriz [][][]

A seguir temos alguns exemplos com alternativas para criação de matrizes.

Administração Central**Cetec Capacitações**

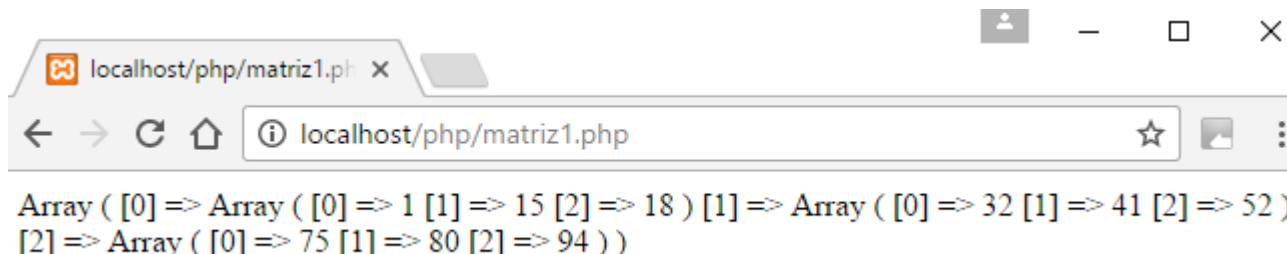
```
<?php

$num1 = array (1,15,18);
$num2 = array (32,41,52);
$num3 = array (75,80,94);

$matriz = array($num1, $num2, $num3);

print_r ($matriz);

?>
```



Outra forma para a construção da mesma matriz anterior.

```
<?php

$matriz = array(array (1,15,18), array (32,41,52),array (75,80,94));

print_r ($matriz);

?>
```

Uma outra forma mais tradicional é atribuir os valores indicando as respectivas posições (linha x coluna);

Administração Central

Cetec Capacitações

```
<?php

$matriz[0][0] = 1;
$matriz[0][1] = 15;
$matriz[0][2] = 18;

$matriz[1][0] = 32;
$matriz[1][1] = 41;
$matriz[1][2] = 52;

$matriz[2][0] = 75;
$matriz[2][1] = 80;
$matriz[2][2] = 94;

print_r ($matriz);

?>
```

13 Funções com strings

O PHP possui inúmeras funções para manipulação de **strings**, mas serão destacadas as mais comuns, que são:

strlen() retorna o tamanho de uma string.

strtoupper() converte uma string para letras maiúsculas.

strtolower() converte uma string para letras minúsculas.

ucfirst() converte para maiúsculo a primeira letra da string.

ucwords() converte para maiúsculo a primeira letra de cada palavra da string.

substr() retorna parte de uma string.

str_replace() substitui um termo pelo outro levando em consideração letras maiúsculas e minúsculas.

Administração Central
Cetec Capacitações

```
<?php

$texto = "micro Computador";

$tamanho = strlen($texto); //tamanho da string
echo "<br />".$texto." possui ".$tamanho." caracteres";

echo "<br />".strtoupper($texto);

echo "<br />".strtolower($texto);

echo "<br />".ucfirst($texto);

echo "<br />".ucwords($texto);

//3 caracteres a partir da posição 13
echo "<br />".substr($texto, 13, 3);

echo "<br />".str_replace("micro", "Nano", $texto);

?>
```

14 Funções de Data e Hora

A função **date()** de acordo com os caracteres de formatações permite inúmeras possibilidades de retorno. Um exemplo de manipulação com data e hora:

```
<?php

$dataAtual = date("d/m/Y");
echo "<br />Hoje é: ".$dataAtual;

$hora = date("h:i:s");
echo "<br />Agora são: ".$hora;

$dataHora = date("d/m/Y h:i:s");
echo "<br />Data e Hora: ".$dataHora;

?>
```

A seguir temos uma tabela contendo os caracteres de formatação, descrição e exemplo de cada formatação envolvendo data e hora:

Administração Central
Cetec Capacitações

Caracter	Descrição	Exemplo de valores retornados
Dia		
d	Dia do mês, 2 dígitos com preenchimento de zero	01 até 31
D	Uma representação textual de um dia, três letras	Mon até Sun
j	Dia do mês sem preenchimento de zero	1 até 31
l ('L' minúsculo)	A representação textual completa do dia da semana	Sunday até Saturday
N	Representação numérica ISO-8601 do dia da semana (adicionado no PHP 5.1.0)	1 (para Segunda) até 7 (para Domingo)
S	Sufixo ordinal inglês para o dia do mês, 2 caracteres	st, nd, rd ou th. Funciona bem com j
w	Representação numérica do dia da semana	0 (para domingo) até 6 (para sábado)
z	O dia do ano (começando do 0)	0 through 365
Semana		
W	Número do ano da semana ISO-8601, semanas começam na Segunda (adicionado no PHP 4.1.0)	Exemplo: 42 (the 42nd week in the year)
Mês		
F	Um representação completa de um mês, como January ou March	January até December
m	Representação numérica de um mês, com leading zeros	01 a 12
M	Uma representação textual curta de um mês, três letras	Jan a Dec
n	Representação numérica de um mês, sem leading zeros	1 a 12
t	Número de dias de um dado mês	28 through 31
Ano		
L	Se está em um ano bissexto	1 se está em ano bissexto, 0 caso contrário.
o	Número do ano ISO-8601. Este tem o mesmo valor como Y, exceto que se o número da semana ISO (W) pertence ao anterior ou próximo ano, o ano é usado ao invés. (adicionado no PHP 5.1.0)	Exemplos: 1999 ou 2003
Y	Uma representação de ano completa, 4 dígitos	Exemplos: 1999 ou 2003

Administração Central
Cetec Capacitações

y	Uma representação do ano com dois dígitos	Exemplos: 99 ou 03
Tempo		
a	Antes/Depois de meio-dia em minúsculo	am or pm
A	Antes/Depois de meio-dia em maiúsculo	AM or PM
B	Swatch Internet time	000 até 999
g	Formato 12-horas de uma hora sem preenchimento de zero	1 até 12
G	Formato 24-horas de uma hora sem preenchimento de zero	0 até 23
h	Formato 12-horas de uma hora com zero preenchendo à esquerda	01 até 12
H	Formato 24-horas de uma hora com zero preenchendo à esquerda	00 até 23
i	Minutos com zero preenchendo à esquerda	00 até 59
s	Segundos, com zero preenchendo à esquerda	00 até 59
u	Milisegundos (adicionado no PHP 5.2.2)	Exemplo: 54321
Timezone		
e	Identificador de Timezone (adicionado no PHP 5.1.0)	Exemplos: UTC,GMT,Atlantic/Azores
I (capital i)	Se a data está ou não no horário de verão	1 se horário de verão, 0 caso contrário.
O	Diferença para Greenwich time (GMT) em horas	Exemplo: +0200
P	Diferença para Greenwich time (GMT) com dois pontos entre horas e minutos (adicionado no PHP 5.1.3)	Exemplo: +02:00
T	Abreviação de Timezone	Exemplos: EST,MDT ...
Z	Timezone offset in seconds. The offset for timezones west of UTC is always negative, and for those east of UTC is always positive.	-43200 até 50400
Full Date/Time		
c	ISO 8601 date (adicionado no PHP 5)	2004-02-12T15:19:21+00:00
r	» RFC 2822 formatted date	Exemplo: Thu, 21 Dec 2000 16:01:07 +0200
U	Segundos desde a Época Unix (January 1 1970 00:00:00 GMT)	<u>Veja também time()</u>

Administração Central

Cetec Capacitações

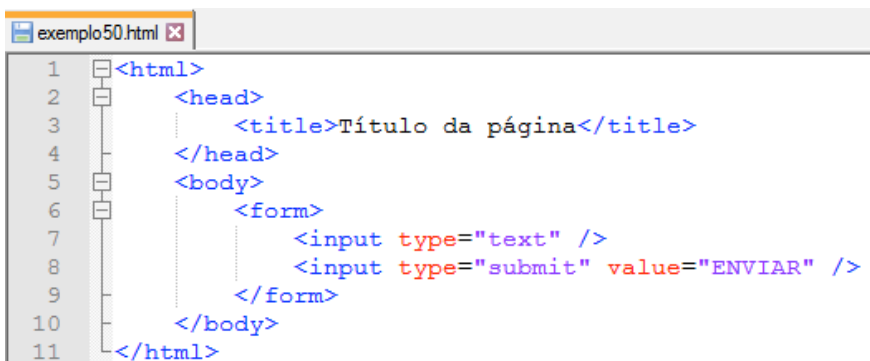
Fonte: http://php.net/manual/pt_BR/function.date.php

15 Formulário

É o elemento utilizado para oferecer recursos de interatividade com os usuários, disponibilizando recursos em páginas web, permitindo dar origem ao preenchimento e o envio de dados para outras páginas e banco de dados.

O elemento **<form>** identifica os limites de um formulário dentro de uma página. Cada *form* representa uma estrutura única. Podemos ter vários *forms* em uma única página, desde que devidamente agrupados.

Encontramos este elemento em páginas de contato de empresas, formulários de pesquisa, telas de cadastro em sistemas comerciais, enfim, é um recurso essencial para a comunicação com o mundo exterior.



```
1 <html>
2 <head>
3   <title>Título da página</title>
4 </head>
5 <body>
6   <form>
7     <input type="text" />
8     <input type="submit" value="ENVIAR" />
9   </form>
10 </body>
11 </html>
```

15.1 Propriedade action

Para que o conteúdo de um formulário seja processado, as informações devem ser repassadas para outra página, que geralmente é escrita em PHP, Java, ASP entre outras linguagens. Na chamada do **action**, deverá ser identificado o nome do arquivo que receberá estes dados, quando não se sabe para qual arquivo seja enviado, esta propriedade pode ficar sem preenchimento.

Administração Central

Cetec Capacitações

```
exemplo51.html x
1  <html>
2  <head>
3      <title>Título da página</title>
4  </head>
5  <body>
6      <form action="outrapagina.php">
7          <input type="text" />
8          <input type="submit" value="ENVIAR" />
9      </form>
10 </body>
11 </html>
```

Ao clicarmos no botão “Enviar” teremos um erro de página caso o arquivo **cadastro.php** não exista.

15.2 Propriedade *name*

A propriedade *name* é de suma importância para validação e controle dos dados. Esta propriedade é imprescindível constar praticamente em todos os elementos de um formulário, pois é através dela que identificamos os campos para posteriormente realizarmos a leitura utilizando alguma linguagem de programação, inclusive o formulário.

```
exemplo52.html x
1  <html>
2  <head>
3      <title>Título da página</title>
4  </head>
5  <body>
6      <form name="cadastro" action="outrapagina.php">
7          <input type="text" />
8          <input type="submit" value="ENVIAR" />
9      </form>
10 </body>
11 </html>
```

1.3 Propriedade *method*

Para envio do conteúdo é preciso definir o método *http* que será utilizado para a transação dos dados. A propriedade *method* aceita dois valores: **get** e **post**:

get: utilizado quando os dados do formulário não tem a necessidade de ficar oculto ao usuário, pois as informações são visualizadas na URL quando o *submit* é acionado. É bastante utilizado

Administração Central

Cetec Capacitações

quando o usuário deseja recuperar URL com informações que serão tratadas por outras páginas, um bom exemplo são os sites de compras, botões de pesquisa, etc.

```
exemplo53.html x
1 <html>
2   <head>
3     <title>Título da página</title>
4   </head>
5   <body>
6     <form name="cadastro" method="get" action="outrapagina.php">
7       <input type="text" />
8       <input type="submit" value="ENVIAR" />
9     </form>
10  </body>
11 </html>
```

➤ **post**: usado quando o formulário remete dados que devem ser ocultos e principalmente enviados para banco de dados, sendo assim, as informações não serão visualizadas na URL.

```
exemplo54.html x
1 <html>
2   <head>
3     <title>Título da página</title>
4   </head>
5   <body>
6     <form name="cadastro" method="get" action="outrapagina.php">
7       <input type="text" />
8       <input type="submit" value="ENVIAR" />
9     </form>
10  </body>
11 </html>
```

15.3 Elemento *input* e a propriedade *type*

O elemento **input** é utilizado para inserir controles/campos no formulário. Por meio da propriedade *type* definimos qual o tipo de controle será utilizado, são eles: **text**, **password**, **checkbox**, **radio**, **submit**, **reset**, **hidden**, **image** e **button**. Todos os elementos também devem utilizar a propriedade *name* para identificação.

➤ **text** permite a entrada de dados em uma única linha (caixa de texto)

Administração Central

Cetec Capacitações

- **password** permite a entradas do tipo senha, permanecendo os dados oculto na visualização
- **checkbox** permite criar caixas de seleção, ou seja, permite deixar selecionar vários itens dentre vários.
- **radio** permite que o usuário selecione apenas uma opção dentre várias, desde que a propriedade *name* possua o mesmo identificador em todas as opções.
- **submit** permite que os dados do formulário sejam enviados
- **reset** permite limpar os campos ou retorná-los aos respectivos valores padrões
- **hidden** permite campos ocultos, ou seja, permitem a passagem de valores para a página destino sem a necessidade de apresentá-los ao usuário. Um exemplo é passar o ID de um elemento do banco de dados para recuperá-lo posteriormente.
- **image** permite utilização de imagens como botões
- **button** permite personalizar botões em um formulário

Administração Central

Cetec Capacitações

```
exemplo56.html
1 <html>
2   <head>
3     <title>Título da página</title>
4   </head>
5   <body>
6     <h2>Exemplo de Formulário</h2>
7     <form name="cadastro" method="get" action="outrapagina.php">
8
9       Nome: <input type="text" name="nome" size="40" maxlength="20" />
10      <br /> <br />
11
12      <input type="submit" value="ENVIAR" />
13    </form>
14  </body>
15</html>
```

Fig. 1 - Limitando a entrada de dados

Neste caso o campo de texto **nome** possui 40 caracteres de largura e deverá receber no máximo 20 caracteres. Caso o campo de texto seja menor que a quantidade de caracteres a serem digitados, o rolamento de caracteres deverá ocorrer normalmente.

15.5 Evento *onclick*

Esta propriedade permite aplicar ações para o formulário quando clicado em algum botão, como por exemplo, incluir ações para os botões **enviar** e **limpar** deste formulário de exemplo.

```
exemplo57.html
1 <html>
2   <head>
3     <title>Título da página</title>
4   </head>
5   <body>
6     <h2>Exemplo de Formulário</h2>
7     <form name="cadastro" method="get" action="outrapagina.php">
8
9       Nome: <input type="text" name="nome" size="40" maxlength="20" />
10      <br /> <br />
11
12      <input type="submit" value="ENVIAR" />
13      <br /> <br />
14
15      <a href="#" onclick="document.cadastro.reset()"> LIMPAR </a>
16    </form>
17  </body>
18</html>
```

Fig. 2 - Evento onClick

Administração Central

Cetec Capacitações

Podemos identificar alguns recursos que estão sendo utilizados na propriedade onclick, como por exemplo:

- **document** representa um documento html
- **cadastro** representa o nome do formulário
- **reset()** representa o evento de limpar os campos do formulário

15.6 Botões fora do formulário

O elemento *button*, permite diferentes tipos de aplicações fora do formulário, mas não podemos deixar de programá-los.

```
<button onclick="window.location.href='http://www.google.com.br' "> Google </button>
```

```
<button onclick="window.location.reload()"> Atualizar página </button>
```

```
<button onclick="window.print()" > Imprimir </button>
```

```
<button onclick="window.history.back()" > Voltar</button>
```

15.7 Caixa de seleção (select)

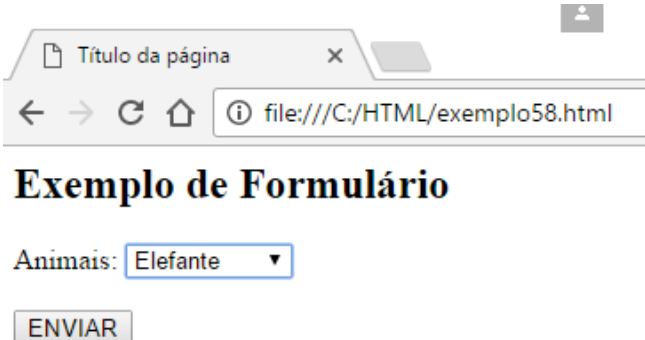
Este elemento permite criar uma caixa de seleção, podendo ter várias opções.

Administração Central
Cetec Capacitações

```
1 <html>
2 <head>
3   <title>Título da página</title>
4 </head>
5 <body>
6   <h2>Exemplo de Formulário</h2>
7   <form name="cadastro" method="get" action="outrapagina.php">
8
9       Animais:
10      <select name="animais">
11          <option> Elefante </option>
12          <option> Hipopótamo </option>
13          <option> Rinoceronte </option>
14          <option> Dinossauro </option>
15      </select>
16      <br /> <br />
17      <input type="submit" value="ENVIAR" />
18
19   </form>
20 </body>
21 </html>
```

Fig. 3 - Exemplo elemento select

Resultado apresentado no navegador:

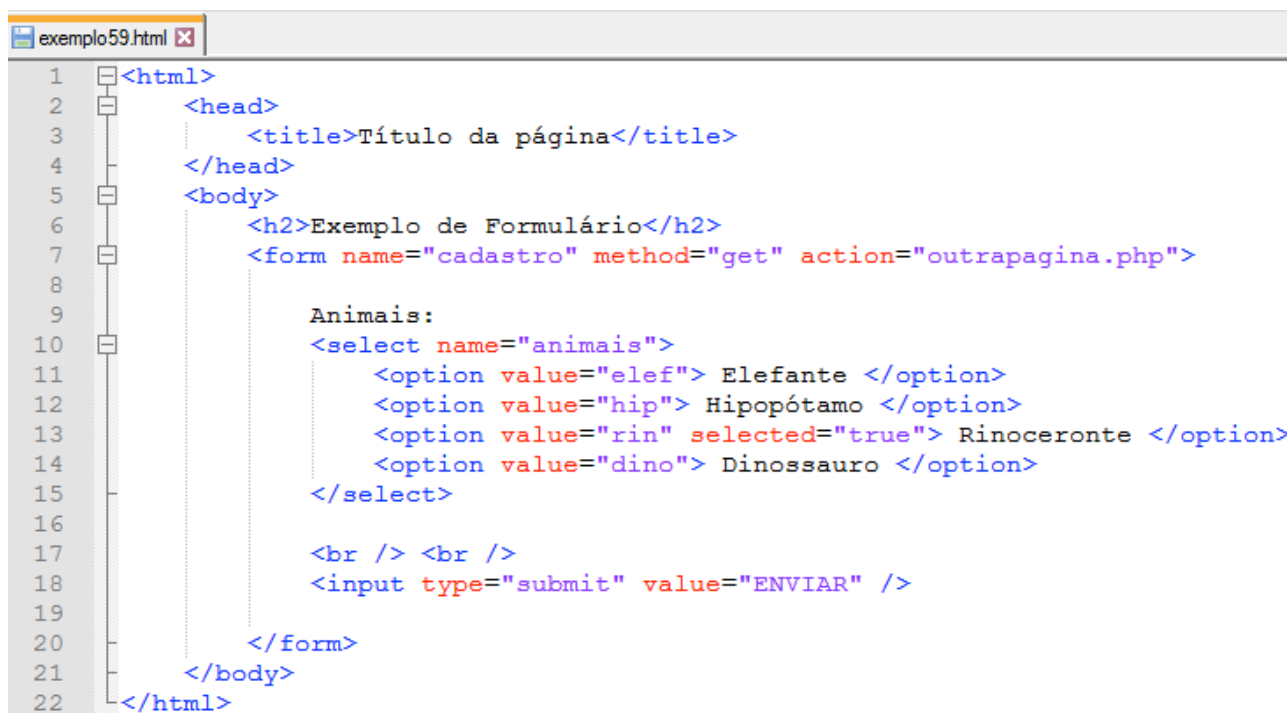


Caso seja necessário, podemos indicar um elemento pré-selecionado, usando a propriedade *selected*.

O importante é controlar os valores de retorno para as opções (*option*) indicadas pelo usuário, para isso fazemos o uso da propriedade *value*.

Administração Central

Cetec Capacitações



```
1 <html>
2   <head>
3     <title>Título da página</title>
4   </head>
5   <body>
6     <h2>Exemplo de Formulário</h2>
7     <form name="cadastro" method="get" action="outrapagina.php">
8
9       Animais:
10      <select name="animais">
11        <option value="elef"> Elefante </option>
12        <option value="hip"> Hipopótamo </option>
13        <option value="rin" selected="true"> Rinoceronte </option>
14        <option value="dino"> Dinossauro </option>
15      </select>
16
17      <br /> <br />
18      <input type="submit" value="ENVIAR" />
19
20    </form>
21  </body>
22</html>
```

Fig. 4 - Outras propriedades do select

15.8 Elemento *textarea*

O elemento *textarea* permite a criação de caixas de texto de mais de uma linha, sua criação é bem simples. Podemos ajustar o tamanho do elemento, utilizando as propriedades *col* e *rows*.

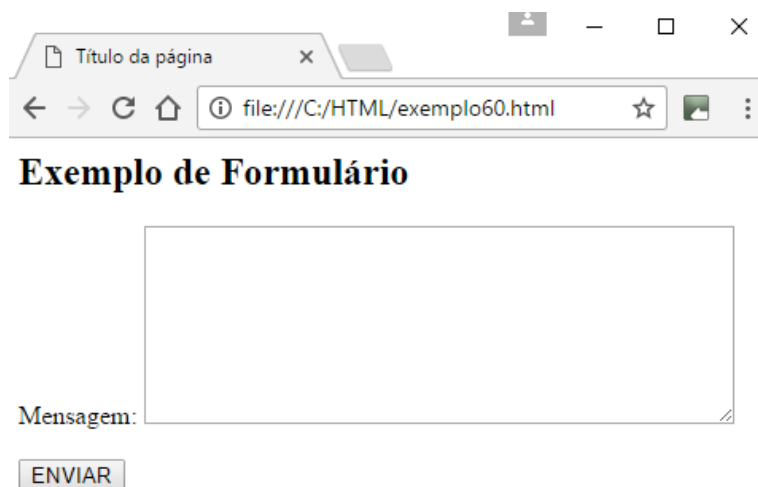
Administração Central

Cetec Capacitações

```
1 <html>
2   <head>
3     <title>Título da página</title>
4   </head>
5   <body>
6     <h2>Exemplo de Formulário</h2>
7     <form name="cadastro" method="get" action="outrapagina.php">
8
9       Mensagem:
10      <textarea name="msg" cols="50" rows="8">
11      </textarea>
12
13      <br /> <br />
14      <input type="submit" value="ENVIAR" />
15
16    </form>
17  </body>
18 </html>
```

Fig. 5 - Elemento textarea

Resultado apresentado no navegador:



16 Formulário - Recuperando dados

Existem diferentes maneiras de recuperar os dados pelo PHP, os quais deverão ser aplicados de acordo com a origem dos dados. Vamos trabalhar com a coleção de dados **`$_POST[]`**, **`$_GET[]`** e **`$_REQUEST[]`**.

Administração Central

Cetec Capacitações

16.1 Uso do \$_POST[]

O método **post** (formulário), passa suas variáveis codificadas dentro da própria submissão, desta forma não são visíveis pelo usuário, apenas o servidor e o navegador tem acesso a estas informações.

Para recuperar os dados do formulário o processo é bem simples, mas devemos levar em consideração as indicações da propriedade **name** dos elementos do HTML.

```

1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6      <form name="cadastro" method="post" action="exemplo01.php">
7          Login:
8          <input type="text" name="nome" /> <br><br>
9          Senha:
10         <input type="password" name="senha" /> <br><br>
11         <input type="submit" value="Enviar" />
12     </form>
13 </body>
14 </html>

```

```

1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      echo "Nome: " . $_POST['nome'];
4      echo "<br>";
5      echo "Senha: " . $_POST['senha'];
6  ?>

```

Usando variáveis para receber os dados

Administração Central

Cetec Capacitações

```

1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6      <form name="cadastro" method="post" action="exemplo02.php">
7          Login:
8          <input type="text" name="nome"/> <br><br>
9          Senha:
10         <input type="password" name="email"/> <br><br>
11         <input type="submit" value="Enviar" />
12     </form>
13 </body>
14 </html>

```

```

1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      $nome = $_POST['nome'];
4      $senha = $_POST['email'];
5      echo "<b>Nome: </b>". $nome;
6      echo "<br>";
7      echo "<b>Senha: </b>". $senha;
8  ?>

```

Este recurso pode ser utilizado nos demais elementos.

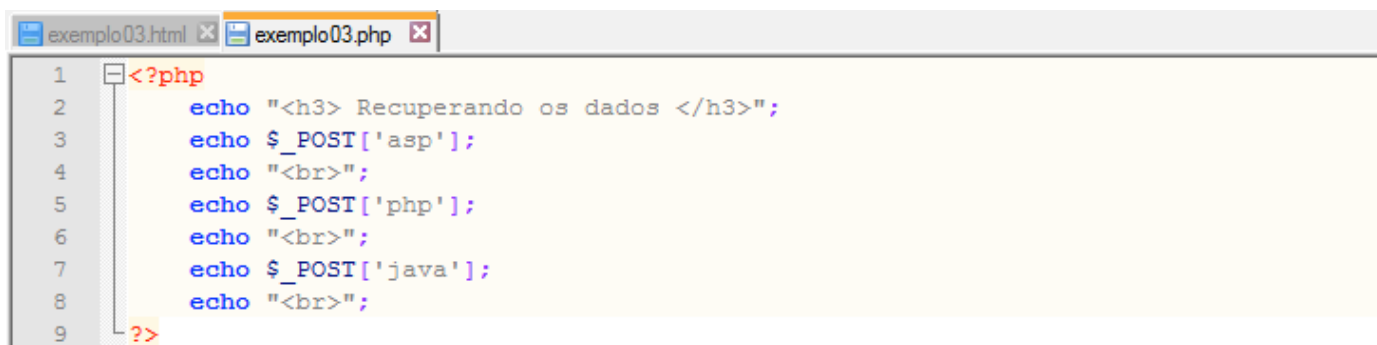
Devemos tomar cuidado ao recuperar dados do elemento *checkbox*, pois somente serão considerados os elementos com valores atribuídos.

```

1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6      <form name="cadastro" method="post" action="exemplo03.php">
7          Qual linguagem de programação você conhece?<br>
8          <input type="checkbox" name="asp"> ASP <br>
9          <input type="checkbox" name="php"> PHP <br>
10         <input type="checkbox" name="java"> JAVA <br>
11         <br><input type="submit" value="Enviar">
12     </form>
13 </body>
14 </html>

```

Administração Central
Cetec Capacitações



```
1 <?php
2     echo "<h3> Recuperando os dados </h3>";
3     echo $_POST['asp'];
4     echo "<br>";
5     echo $_POST['php'];
6     echo "<br>";
7     echo $_POST['java'];
8     echo "<br>";
9 ?>
```

Caso não tenha nenhum valor selecionado, teremos uma mensagem de erro, pois o PHP entenderá que a variável não existe.

Recuperando os dados

Notice: Undefined index: asp in C:\xampp\htdocs\CPS - PHP\aula11\exemplo03.php on line 3

Notice: Undefined index: php in C:\xampp\htdocs\CPS - PHP\aula11\exemplo03.php on line 5

Notice: Undefined index: java in C:\xampp\htdocs\CPS - PHP\aula11\exemplo03.php on line 7

Para resolver este problema, a solução é utilizar a função **isset()**. Esta por sua vez verifica a existência da variável.

Administração Central

Cetec Capacitações

```

1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3
4      if (isset($_POST['asp'])) {
5          echo $_POST['asp'];
6          echo "<br>";
7      }
8      if (isset($_POST['php'])) {
9          echo $_POST['php'];
10         echo "<br>";
11     }
12     if (isset($_POST['java'])) {
13         echo $_POST['java'];
14         echo "<br>";
15     }
16     ?>
    
```

Verifique que a resposta apresenta somente a expressão **on**, informando que a opção foi selecionada. Para melhorar a recuperação dos dados utilize o *value* no elemento *checkbox* do seu formulário.

```

1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6      <form name="cadastro" method="post" action="exemplo05.php">
7          Qual linguagem de programação você conhece?<br>
8          <input type="checkbox" name="asp" value="Sim, você escolheu ASP"> ASP <br>
9          <input type="checkbox" name="php" value="Selecionado"> PHP <br>
10         <input type="checkbox" name="java" value="Java"> JAVA <br>
11         <br><input type="submit" value="Enviar">
12     </form>
13 </body>
14 </html>
    
```

Administração Central

Cetec Capacitações

```

1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3
4      if (isset($_POST['asp'])) {
5          echo "ASP: " . $_POST['asp'];
6          echo "<br>";
7      }
8      if (isset($_POST['php'])) {
9          echo "PHP: " . $_POST['php'];
10         echo "<br>";
11     }
12     if (isset($_POST['java'])) {
13         echo "JAVA: " . $_POST['java'];
14         echo "<br>";
15     }
16     ?>
    
```

Semelhante à estrutura de recuperação do *checkbox*, utilize o **isset()** para evitar erros de definição.

```

1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6      <form name="cadastro" method="post" action="exemplo06.php">
7          Estado Civil<br>
8          <input type="radio" name="estadocivil"> Casado <br>
9          <input type="radio" name="estadocivil"> Solteiro <br>
10         <br><input type="submit" value="Enviar">
11     </form>
12 </body>
13 </html>
    
```

```

1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      if(isset($_POST['estadocivil'])) {
4          echo $_POST['estadocivil'];
5      }
6      ?>
    
```


Administração Central

Cetec Capacitações

Note que novamente a resposta é a expressão **on**, para qualquer uma das opções selecionadas, desta forma, utilize o *value* no elemento *radio* do seu formulário para atribuir um valor.

```
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="post" action="exemplo07.php">
7     Estado Civil<br>
8     <input type="radio" name="estadocivil" value="Casado"> Casado <br>
9     <input type="radio" name="estadocivil" value="Solteiro"> Solteiro <br>
10    <br><input type="submit" value="Enviar">
11  </form>
12 </body>
13 </html>
```

```
1 <?php
2   echo "<h3> Recuperando os dados </h3>";
3   if(isset($_POST['estadocivil'])){
4     echo $_POST['estadocivil'];
5   }
6 >?
```

Para recuperar elementos ocultos (*hidden*) o processo é o mesmo que o elemento do tipo *text*.

Administração Central

Cetec Capacitações

```

exemplo08.html x exemplo08.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="post" action="exemplo08.php">
7     Enviando dados ocultos.
8     <input type="hidden" name="user" value="admin">
9     <input type="hidden" name="nivel" value="2">
10    <input type="submit" value="OK">
11  </form>
12 </body>
13 </html>

exemplo08.html x exemplo08.php x
1 <?php
2   echo "<h3> Recuperando os dados </h3>";
3
4   echo "Olá, usuário <b>".$_POST['user']. "</b> você possui o "
5     .$_POST['nivel']. "º nível de acesso."
6
7   ?>

```

Indiferente da quantidade de opções (*option*) o valor de recuperação, estará indicado no atributo *name*.

```

exemplo09.html x exemplo09.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="post" action="exemplo09.php">
7     Estados:
8     <select name="estados">
9       <option> São Paulo </option>
10      <option> Rio de Janeiro </option>
11      <option> Minas Gerais </option>
12      <option> Bahia </option>
13    </select><br><br>
14    <input type="submit" value="Enviar">
15  </form>
16 </body>
17 </html>

```

Administração Central

Cetec Capacitações

```
exemplo09.html x exemplo09.php x
1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      echo "<b>Estado: </b>".$_POST['estados'];
4  ?>
```

Melhorando o resultado da recuperação com *value*.

```
exemplo10.html x exemplo10.php x
1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6      <form name="cadastro" method="post" action="exemplo09.php">
7          Estados:
8          <select name="estados">
9              <option value="SP"> São Paulo </option>
10             <option value="RJ"> Rio de Janeiro </option>
11             <option value="MG"> Minas Gerais </option>
12             <option value="BA"> Bahia </option>
13         </select><br><br>
14         <input type="submit" value="Enviar">
15     </form>
16 </body>
17 </html>

exemplo10.html x exemplo10.php x
1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      echo "<b>Estado (sigla): </b>".$_POST['estados'];
4  ?>
```

Recuperação semelhante ao *text*.

Administração Central

Cetec Capacitações

```

exemplo11.html x exemplo11.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="post" action="exemplo11.php">
7     Mensagem:<br>
8     <textarea name="mensagem" cols="80" rows="7"></textarea>
9     <br><br>
10    <input type="submit" value="Enviar">
11  </form>
12 </body>
13 </html>

exemplo11.html x exemplo11.php x
1 <?php
2   echo "<h3> Recuperando os dados </h3>";
3   echo "<b>Mensagem: </b><br>".$_POST['mensagem'];
4   ?>

```

16.2 Uso do \$_GET[]

O método **get** (formulário), passa suas variáveis via URL deixando as informações visíveis para futura recuperação. Cuidado ao expor as informações e principalmente com o limite de caracteres de uma URL.

```

exemplo01.html x exemplo01.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="get" action="exemplo01.php">
7     Login:
8     <input type="text" name="nome"/> <br><br>
9     E-mail:
10    <input type="password" name="email"/> <br><br>
11    <input type="submit" value="Enviar" />
12  </form>
13 </body>
14 </html>

```

Administração Central

Cetec Capacitações

```

1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      echo "Nome: " . $_GET['nome'];
4      echo "<br>";
5      echo "E-mail: " . $_GET['email'];
6  ?>

```

16.3 Uso do \$_REQUEST[]

Semelhante ao \$_POST[] e \$_GET[], o \$_REQUEST é uma variável de coleção genérica, sendo assim não há diferença se o valor passado for pelo método **get** ou **post**. Geralmente a busca segue uma ordem: **get**, **post**, **cookies** e **sessão** (ver *variables_order* do php.ini).

Usando método **get**.

```

1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6      <form name="cadastro" method="get" action="exemplo02.php">
7          Login:
8          <input type="text" name="nome"/> <br><br>
9          E-mail:
10         <input type="password" name="email"/> <br><br>
11         <input type="submit" value="Enviar" />
12     </form>
13 </body>
14 </html>

```

```

1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      echo "Nome: " . $_REQUEST['nome'];
4      echo "<br>";
5      echo "E-mail: " . $_REQUEST['email'];
6  ?>

```

Administração Central

Cetec Capacitações

Usando método **post**.

```
exemplo03.html x exemplo03.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="post" action="exemplo03.php">
7     Login:
8     <input type="text" name="nome" /> <br><br>
9     E-mail:
10    <input type="password" name="email" /> <br><br>
11    <input type="submit" value="Enviar" />
12  </form>
13 </body>
14 </html>
```

```
exemplo03.html x exemplo03.php x
1 <?php
2   echo "<h3> Recuperando os dados </h3>";
3   echo "Nome: " . $_REQUEST['nome'];
4   echo "<br>";
5   echo "E-mail: " . $_REQUEST['email'];
6 >?
```

17 Uso do \$_GET[] com URL

Muitas vezes utilizamos os links ou formamos links dinâmicos que permitem a passagem de dados via URL, desta forma podemos utilizar o `$_GET[]` ou `$_REQUEST[]`, para recuperar estes dados.

As informações (**variável e conteúdo**), devem ser passadas depois no nome do arquivo, precedido do sinal de "?".

Administração Central

Cetec Capacitações

```

exemplo04.html x exemplo04.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6
7   <a href="exemplo04.php?id=10"> Passando dados via URL </a>
8
9 </body>
10</html>

exemplo04.html x exemplo04.php x
1 <?php
2   echo "<h3> Recuperando os dados </h3>";
3   echo "Id: " . $_GET['id'];
4 >?
  
```

Para passar mais de um valor via URL, utilize o símbolo "&" para separar os elementos.

```

exemplo05.html x exemplo05.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6
7   <a href="exemplo05.php?id=10&user=admin"> Passando dados via URL </a>
8
9 </body>
10</html>

exemplo05.html x exemplo05.php x
1 <?php
2   echo "<h3> Recuperando os dados </h3>";
3   echo "Id: " . $_GET['id'];
4   echo "<br>";
5   echo "User: " . $_GET['user'];
6 >?
  
```

Usando o mesmo exemplo com `$_REQUEST[]`.

Administração Central

Cetec Capacitações



The screenshot shows two code editors. The top editor, titled 'exemplo06.html', contains the following HTML code:

```

1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6
7   <a href="exemplo06.php?id=10&user=admin"> Passando dados via URL </a>
8
9 </body>
10 </html>

```

The bottom editor, titled 'exemplo06.php', contains the following PHP code:

```

1 <?php
2   echo "<h3> Recuperando os dados </h3>";
3   echo "Id: " . $_REQUEST['id'];
4   echo "<br>";
5   echo "User: " . $_REQUEST['user'];
6 ?>

```

17.1 Uso de variáveis para compor a URL

Para criar links dinâmicos, podemos estruturá-los a partir de variáveis do PHP.



The screenshot shows two code editors. The top editor, titled 'exemplo07.php', contains the following HTML code:

```

1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6
7   <?php
8     $idBanco = 10;
9     $nomeUser = "admin"
10
11   <a href="exemplo07_dados.php?id=?php echo $idBanco;?>&user=?php echo $nomeUser;?> ">
12     Passando dados via URL </a>
13
14 </body>
15 </html>

```

The bottom editor, titled 'exemplo07_dados.php', is empty.

Administração Central

Cetec Capacitações

```

1 <?php
2     echo "<h3> Recuperando os dados </h3>";
3     echo "Id: " . $_REQUEST['id'];
4     echo "<br>";
5     echo "User: " . $_REQUEST['user'];
6 ?>

```

Ou podemos indicar do seguinte formato.

```

1 <html>
2 <head>
3     <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6     <?php
7         $idBanco = 10;
8         $nomeUser = "admin";
9
10        echo "<a href='exemplo08_dados.php?id=";
11        echo $idBanco;
12        echo "&user=";
13        echo $nomeUser;
14        echo "'>Passando dados via URL </a>";
15    ?>
16 </body>
17 </html>

```

Simplificando.

```

1 <html>
2 <head>
3     <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6     <?php
7         $idBanco = 10;
8         $nomeUser = "admin";
9         echo "<a href='exemplo09_dados.php?id=" . $idBanco . "&user=" . $nomeUser . "'>Passando dados via URL </a>";
10    ?>
11 </body>
12 </html>

```

Ou ainda:

Administração Central

Cetec Capacitações

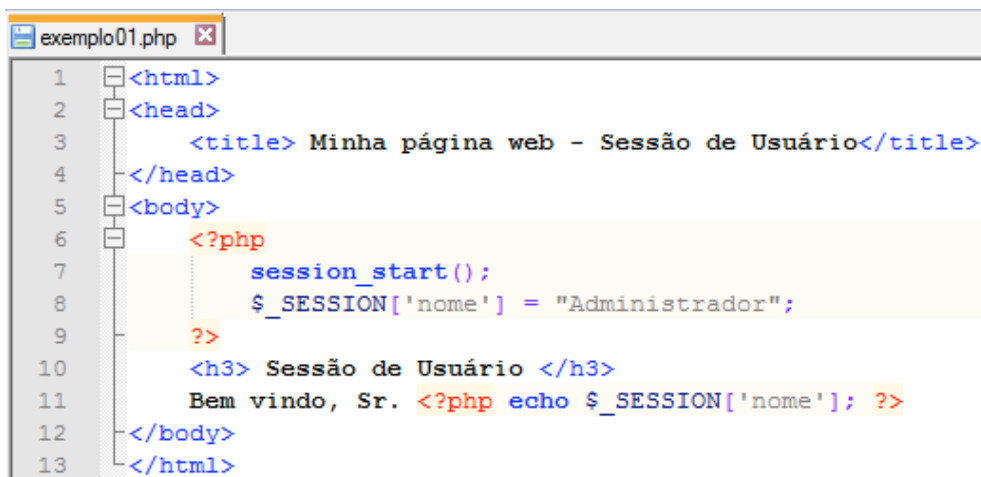
```
<?php
    $idBanco = 10;
    $nomeUser = "admin";
    echo "<a href='exemplo10_dados.php?id=$idBanco&user=$nomeUser'>Passando dados via URL </a>";
?>
```

18 Sessão

O uso e gerenciamento de sessão é essencialmente uma solução para ajustar preferências e comportamentos de usuário, permitindo assim a identificação dos usuários ou visitantes através da Sessão ID (SID).

18.1 Trabalhando com sessão

Uma sessão é iniciada ao ser executado o **`session_start()`**, permitindo assim as operações com as variáveis de sessão. Para criá-las, utiliza-se a variável superglobal **`$_SESSION`**, veja o exemplo.



```
exemplo01.php
1  <html>
2  <head>
3      <title> Minha página web - Sessão de Usuário</title>
4  </head>
5  <body>
6      <?php
7          session_start();
8          $_SESSION['nome'] = "Administrador";
9      ?>
10     <h3> Sessão de Usuário </h3>
11     Bem vindo, Sr. <?php echo $_SESSION['nome']; ?>
12 </body>
13 </html>
```

Para eliminar os dados de uma sessão, podemos utilizar:

`session_destroy()` – elimina toda a sessão;

`session_unset()` – elimina todas as variáveis de sessão mas, mantém a sessão ativa.

Administração Central

Cetec Capacitações

```
exemplo02.php
1 <html>
2 <head>
3   <title> Minha página web - Sessão de Usuário</title>
4 </head>
5 <body>
6   <?php
7     session_start();
8     $_SESSION['nome'] = "Administrador";
9     session_unset();
10  >
11   <h3> Sessão de Usuário </h3>
12   bem vindo, Sr. <?php echo $_SESSION['nome'];?>
13 </body>
14 </html>
```

Teremos um erro ao tentar imprimir a variável de sessão **nome**.

Outra forma de encerrar uma sessão e fechar o navegador que está sendo utilizado.

18.2 ID de uma sessão

Podemos recuperar o número de controle da sessão (SID) através do ***session_id()***.

```
exemplo03.php
1 <html>
2 <head>
3   <title> Minha página web - Sessão de Usuário</title>
4 </head>
5 <body>
6   <?php
7     session_start();
8   >
9   <h3> Sessão de Usuário </h3>
10   SID desta sessão é: <?php echo session_id();?>
11 </body>
12 </html>
```

18.3 Bloqueando páginas por sessão

Um artifício bastante utilizado é a proteção de páginas, por **login** de usuários, o qual poderá ser tratado através das sessões.

Administração Central

Cetec Capacitações

```
exemplo04.php x exemplo04_sessao.php x exemplo04_pag.php x
1 <html>
2 <head>
3 <title>Trabalhando com Sessão - PHP</title>
4 </head>
5 <body>
6 <form name="logar" method="post" action="exemplo04_sessao.php">
7     Nome: <br>
8     <input type="text" name="nome"><br>
9     Senha: <br>
10    <input type="password" name="senha"><br>
11    <input type="submit" name="enviar" value="enviar">
12 </form>
13 </body>
14 </html>
```

Na próxima sequência de código, duas variáveis de sessão são criadas **nome** e **senha** a partir das informações do **formulário**.

```
exemplo04.php x exemplo04_sessao.php x exemplo04_pag.php x
1 <html>
2 <head>
3 <title>Trabalhando com Sessão - PHP</title>
4 <?php
5     session_start();
6     $_SESSION["nome"] = $_POST["nome"];
7     $_SESSION["senha"] = $_POST["senha"];
8 >?>
9 </head>
10 <body>
11     <b>Funcionário: </b><?php echo $_SESSION["nome"] ?>, logado com sucesso.<br>
12     <b>Data de conexão: </b><?php echo date("d/m/Y") ?> <br>
13     <b>Hora de conexão: </b><?php echo date("H:i:s") ?> <br>
14     <br>
15     <a href="exemplo04_pag.php"> Área de administrador </a><br>
16 </body>
17 </html>
```

A página **exemplo04_pag.php**, verifica se o conteúdo da sessão **nome** é **"master"**. Somente em caso positivo esta página será aberta.

Administração Central

Cetec Capacitações

```
exemplo04.php x exemplo04_sessao.php x exemplo04_pag.php x
1 <html>
2 <head>
3 <title>Trabalhando com Sessão - PHP</title>
4 <?php
5 session_start();
6 if ($_SESSION["nome"] != "master"){
7     session_destroy();
8     header("location:exemplo04.php");
9 }
10 ?>
11 </head>
12 <body>
13     <b>Página autorizada</b>
14 </body>
15 </html>
```

Realize o seguinte teste:

1. Feche o navegador para que a sessão se encerre.
2. Tente executar o arquivo **exemplo04_pag.php**, como não existe mais a sessão o navegador irá redirecionar para a página **exemplo04.php**.

18.4 Identificando uma sessão

Se preferir, uma sessão de usuário poderá ser identificada (nomeada).

```
exemplo05.php x
1 <html>
2 <head>
3 <title> Minha página web - Sessão de Usuário</title>
4 </head>
5 <body>
6 <?php
7     session_start("user");
8     $_SESSION['nome'] = "usuário";
9 ?>
10 </body>
11 </html>
```

Administração Central

Cetec Capacitações

19 Implementando os formulários

Uma alternativa para se trabalhar com formulários é realizar na mesma página a entrada e recuperação de dados. Para entender melhor este conceito vamos aplicar em um formulário (login e senha) a recuperação dos dados e criação de sessão.

Este recurso pode ser utilizado para os modelos de formulários vistos anteriormente, mas deve-se tomar cuidado com valores armazenados em *cache*.

```
exemplo06.php exemplo06_pag.php exemplo06_sessao.php
1 <html>
2 <head>
3 <title>Trabalhando com Sessão - PHP</title>
4 <?php
5     if($_POST){
6         session_start();
7         $_SESSION["nome"] = $_POST["nome"];
8         $_SESSION["senha"] = $_POST["senha"];
9         header("location:exemplo06_sessao.php");
10    }
11 >
12 </head>
13 <body>
14 <form name="logar" method="post" action="exemplo06.php">
15     Nome:<br>
16     <input type="text" name="nome"><br>
17     Senha:<br>
18     <input type="password" name="senha"><br>
19     <input type="submit" name="enviar" value="enviar">
20 </form>
21 </body>
22 </html>
```

Note que o nome do arquivo indicado na cláusula **action** é o mesmo da página:
exemplo06.php

Administração Central

Cetec Capacitações

```
exemplo06.php x exemplo06_pag.php x exemplo06_sessao.php x
1 <html>
2 <head>
3 <title>Trabalhando com Sessão - PHP</title>
4 <?php session_start(); ?>
5 </head>
6 <body>
7 <b>Funcionário: </b><?php echo $_SESSION["nome"] ?>, logado com sucesso.<br>
8 <b>Data de conexão: </b><?php echo date("d/m/Y") ?> <br>
9 <b>Hora de conexão: </b><?php echo date("H:i:s") ?> <br>
10 <br>
11 <a href="exemplo06_pag.php"> Área de administrador </a><br>
12 </body>
13 </html>

exemplo06.php x exemplo06_pag.php x exemplo06_sessao.php x
1 <html>
2 <head>
3 <title>Trabalhando com Sessão - PHP</title>
4 <?php
5 session_start();
6 if ($_SESSION["nome"] != "master"){
7 session_destroy();
8 header("location:exemplo06.php");
9 }
10 ?>
11 </head>
12 <body>
13 <b>Página autorizada</b>
14 </body>
15 </html>
```

“Graduação em Tecnologia em Processamento de Dados pelo Centro Universitário de Rio Preto, Pós-Graduação em Tecnologia na Gestão da Informação pela Anhanguera Educacional, Licenciatura Plena pelo Programa Especial de Formação Pedagógica do Centro Paula Souza e curso Técnico de Administração pelo Centro Paula Souza. Nas instituições de ensino superior, atualmente é docente e coordenador de cursos de TI das Faculdades IPEP Campinas. Nas instituições de ensino técnico, atualmente é tutor EAD / Presencial, docente e coordenador de cursos da ETEC Bento Quirino, e docente no Colégio Politécnico Bento Quirino. Já atuou como Supervisor de TI, Coordenador de TI, Analista Programador, DBA, Analista de Redes, Analista de Suporte, Supervisor Administrativo e Gerente de Filial. Possui bons conhecimentos nas áreas de TI, Administrativa, Operacional, Comercial e Gestão”