



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Birigui

Parte VIII



Vetores e Matrizes



Vetores

As variáveis comuns podem armazenar apenas um valor por vez. Um vetor pode armazenar vários valores ao mesmo tempo. Assim como as variáveis, possui um identificador, mas além disso há um índice associado (que pode ser um número ou um texto), e cada índice indica uma posição de memória em que fica armazenado um elemento do vetor.

O índice deve aparecer entre colchetes, logo após o identificador do vetor.



Vetores

Exemplos:

```
$vetor[0] = 30;
```

```
$vetor[1] = 40;
```

Se não colocarmos o índice do vetor entre colchetes, o PHP procurará o último índice utilizado e irá incrementá-lo, armazenando assim o valor na posição seguinte do array:

```
$vet[] = 1
```

```
$vet[] = 2
```



Vetores

O índice também pode ser um texto, e nesses casos o texto é chamado de chave associativa:

- `$vetor["turma"] = "Informática";`
- `$vetor["ano"] = 2001;`



Matrizes

As matrizes são vetores multidimensionais. Possuem um único identificador, mas possuem dois ou mais índices para referenciar uma posição de memória.

Imagine que queremos armazenar na memória os nomes dos melhores clubes do futebol brasileiro, separando-os por estados e cidades. Podemos fazer isso utilizando um vetor bidimensional:





Matrizes

Exemplo:

\$clube ["RS"] ["PortoAlegre"] = "Gremio";

\$clube ["RS"] ["Caxias"] = "Juventude";

\$clube ["MG"] ["BeloHorizonte"] = "Atlético";

\$clube ["SP"] ["São Paulo"] = "Corinthians";



Matrizes

Outra forma de criar um vetor é por meio da função array do PHP:

```
<?php  
$vetor = array(10,50,100,150,200);  
echo $vetor[2]."<br>";  
?>
```



Matrizes

Funções básicas aplicadas aos vetores:

- **Count()** – Informa quantos elementos o array possui. `echo count($vetor);`
- **Sizeof()** – idêntica à count, retornará um valor inteiro contendo o número de elementos de um array. `echo sizeof($vetor);`
- **Reset()** – Coloca o índice para o primeiro elemento do array e retorna o valor deste elemento. `echo reset($vetor);`
- **End()** – Coloca o índice para o último elemento do array e retorna o valor deste elemento. `echo end($vetor);`
- **Prev()** – A partir da posição do índice, desloca-se para o elemento anterior e retorna o valor deste. Caso esteja no primeiro elemento, o resultado será vazio. `echo prev($vetor);`



Matrizes

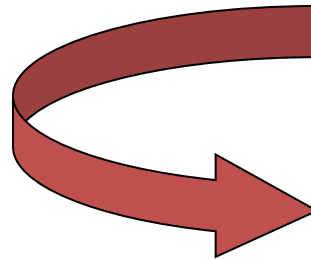
- **Next()** – A partir da posição do índice, avança para o próximo elemento e retorna o valor deste. Caso esteja no último elemento, o resultado será vazio. `echo next($vetor);`
- **Pos()** – Retorna o conteúdo do elemento atual do array. `echo pos($vetor);`
- **Key()** – Retorna o índice do elemento atual do array. `echo key($vetor);`
- **Sort()** – Coloca os elementos em ordem crescente. `sort($vetor); echo $vetor[0]. " | ". $vetor[1]. " | ". $vetor[2];`
- **Rsort()** – Coloca os elementos em ordem decrescente. `Rsort($vetor); echo $vetor[0]. " | ". $vetor[1]. " | ". $vetor[2];`



Matrizes

Usando matrizes para construir a página que exibe o dia da semana:

- Armazenar os dias da semana em uma matriz denominada `$diaSemana[]`.
- Uma vez carregada a matriz com os devidos valores, criar uma rotina para impressão do dia da semana de acordo com a data registrada no servidor.





Matrizes

```
<?php
$hoje = date("w");
$diaSemana[0] = "Domingo";
$diaSemana[1] = "Segunda-feira";
$diaSemana[2] = "Terça-feira";
$diaSemana[3] = "Quarta-feira";
$diaSemana[4] = "Quinta-feira";
$diaSemana[5] = "Sexta-feira";
$diaSemana[6] = "Sábado";
echo "A variável \$hoje contém o número: $hoje <br>";
echo "Hoje é: $diaSemana[$hoje]";
?>
```



Matrizes

Usando matrizes para construir a página “Prato do Dia”

- Construir uma página que mostre para cada dia da semana o prato do dia e seu referido preço, conforma a tabela:

Elemento	Dia Semana	Prato Dia	Preço
0	Domingo	Lasanha	R\$ 12,20
1	Segunda-feira	Frango	R\$ 10,00
2	Terça-feira	Arroz à grega	R\$ 9,40
3	Quarta-feira	Feijoada	R\$ 11,20
4	Quinta-feira	Nhoque	R\$ 8,50
5	Sexta-feira	Bacalhau	R\$ 15,20
6	Sábado	Feijão branco	R\$ 10,00



Vetores e Matrizes

- Como percorrer todos elementos de um vetor?
- Como ordenar os elementos?
- Como encontrar um elemento dentro do vetor?
- Como remover um elemento do vetor?



Vetores e Matrizes

➤ Como percorrer todos elementos de um vetor?

```
1 |  
2 <?  
3 // Exemplo de como percorrer por todos elementos de um vetor  
4 $alunos = array("Murilo", "Alan", "Ricardo", "Eder", "Rodrigo");  
5 for($x=0;$x < count($alunos);$x++)  
6 {  
7     echo $alunos[$x]. "<br>";  
8 }  
9 ?>  
10
```



Vetores e Matrizes

➤ Como ordenar os elementos?

```
1 |  
2 <?  
3 // Exemplo de como ordenar todos elementos de um vetor  
4 $alunos = array("Murilo", "Alan", "Ricardo", "Eder", "Rodrigo");  
5 sort($alunos);  
6 for($x=0;$x < count($alunos);$x++)  
7 {  
8     echo $alunos[$x]. "<br>";  
9 }  
10 ?>
```



Vetores e Matrizes

➤ Como encontrar um elemento dentro do vetor?

```
1
2 <?
3 // Exemplo de encontrar um elemento dentro do vetor
4 $alunos = array("Murilo", "Alan", "Ricardo", "Eder", "Rodrigo");
5
6 $indice = array_search("Eder", $alunos);
7
8 if ($indice === false)
9 {
10     echo "Elemento não encontrado";
11 }
12 else
13 {
14     echo "Elemento encontrado na posicao $indice do vetor";
15 }
16
17 ?>
```




Vetores e Matrizes

Aviso



Esta função pode retornar o **booleano** FALSE, mas também pode retornar um valor não-booleano que pode ser avaliado como FALSE, como 0 ou "". Leia a seção em **Booleanos** para maiores informações. Utilize o **operador ===** para testar o valor retornado por esta função.

Ao converter para booleano, os seguintes valores são considerados **FALSE**:

- ✓ o próprio **booleano** FALSE
- ✓ o **inteiro** 0 (zero)
- ✓ o **ponto flutuante** 0.0 (zero)
- ✓ uma **string** vazia e a **string** "0"
- ✓ um **array** sem elementos
- ✓ um **objeto** sem elementos membros (somente PHP 4)
- ✓ o tipo especial **NULL** (incluindo variáveis não definidas)
- ✓ o objeto **SimpleXML** criado de tags vazias
- ✓ Qualquer outro valor é considerado **TRUE** (incluindo qualquer **recurso**).



Vetores e Matrizes

Operadores de comparação

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



Vetores e Matrizes

➤ Como remover um elemento do vetor?

```
1 <?
2 // Exemplo de como remover um elemento de um vetor
3 $alunos = array("Murilo", "Alan", "Ricardo", "Eder", "Rodrigo");
4 print_r($alunos);
5
6 // unset remove conteúdo da posição 2
7 unset($alunos[2]);
8 echo "<BR>";
9 print_r($alunos);
10
11 // array_values reindexa o vetor
12 $alunos = array_values($alunos);
13 echo "<BR>";
14 print_r($alunos);
15
16 ?>
```



Exemplo

Suponha que tenha que construir uma aplicação em PHP onde apenas os usuários autorizados possam acessar a aplicação, porém, sem utilizar um banco de dados para gravar os usuários autorizados, neste caso poderíamos criar um vetor com os usuários autorizados e a senha de acesso.



Login HTML

```
1 <HTML>
2 <HEAD>
3   <TITLE>Login</TITLE>
4 </HEAD>
5 <BODY>
6 <h2 align="center">Acesso ao Sistema Acadêmico</h2>
7 <hr align="center">
8 <form method = "POST" action = "Login.php">
9 <table align="center">
10 <tr>
11   <td>Login:</td>
12   <td><input type="text" name="login"size="20"></td>
13 </tr>
14 <tr>
15   <td>Senha:</td>
16   <td><input type="text" name="Senha"size="20"></td>
17 </tr>
18 <tr>
19   <td></td>
20   <td><input type=submit align="center" value="Entrar" ></td>
21 </tr>
22 </table>
23
24 </form>
25 </BODY>
26 </HTML>
```



Login PHP

```
1 <?
2 $LoginDigitado = $_POST['Login'];
3 $SenhaDigitada = $_POST['Senha'];
4
5 $Usuarios = array("Murilo" => "123",
6                  "Ciclano" => "321",
7                  "Beltrano" => "abcd",
8                  "Fulano" => "54321");
9 // Verifica se existe o usuário no vetor
10 if (array_key_exists($LoginDigitado,$Usuarios))
11 {
12     if($Usuarios[$LoginDigitado] == $SenhaDigitada)
13     {
14         echo "Acesso permitido!";
15     }
16     else
17     {
18         echo "Senha incorreta!";
19     }
20 }
21 else
22 {
23     echo "Usuário não cadastrado!";
24 }
25 ?>
```