

CURSO
Técnico em Informática para Internet

DISCIPLINA
Desenvolvimento WEB II

PROFESSOR
Rômulo Silveira Ramos

Vetores / Arrays

Desenvolvimento WEB II

Vetores / Arrays

- Arrays em PHP podem ser observados como mapeamentos ou como vetores indexados.
- As variáveis vistas anteriormente são variáveis escalares, que armazenam um único valor, enquanto o array pode ter muitos elementos.
- Um array é uma variável que armazena um conjunto ou sequência de valores.
- Existem três tipos de arrays, são eles:
 - Arrays **Lineares**
 - Arrays **Associativos**
 - Arrays **Multidimensionais**

Array Linear

- Faz uso de índices inteiros como chave para acesso aos valores.

Lista de compras

- 
1. Feijão
 2. Arroz
 3. Batata
 4. Ketchup

Array Linear

Com variáveis

```
$variavel_1 = 'Feijão';  
$variavel_2 = 'Arroz';  
$variavel_3 = 'Batata';  
$variavel_4 = 'Ketchup';
```

Lista de compras

- 
1. Feijão
 2. Arroz
 3. Batata
 4. Ketchup

Com um único array

```
4 $lista_compras[1] = 'Feijão';  
5 $lista_compras[2] = 'Arroz';  
6 $lista_compras[3] = 'Batata';  
7 $lista_compras[4] = 'Ketchup';
```

Ou

```
11 $lista_compras = array(1 => 'Feijão', 2 => 'Arroz', 3 => 'Batata', 4 => 'Ketchup');  
12
```

Array Linear

É importante notar as seguintes características dos arrays em PHP:

- O primeiro elemento inicia no índice ou posição 0(zero), caso não seja especificado.
- O valor do índice ou posição do elemento acessado deve ser especificado entre colchetes, após o nome da variável do tipo array.

```
1  <?php
2  $lista_compras[1] = "Feijão";
3  $lista_compras[2] = "Arroz";
4  $lista_compras[3] = "Batata";
5  $lista_compras[4] = "Ketchup";
6  print $lista_compras[3];
7  ?>
```

Neste exemplo irá
imprimir na tela apenas
a palavra **Batata**

Array Linear

Para acessar todos os campos de um array podemos fazer de duas maneiras:

1ª → utilizando as funções **var_dump** ou **print_r** já existentes no php

```
var_dump($lista_compras);
```

```
array(4) { [1]=> string(7) "Feijão" [2]=> string(5) "Arroz" [3]=> string(6) "Batata" [4]=> string(7) "Ketchup" }
```

```
print_r($lista_compras);
```

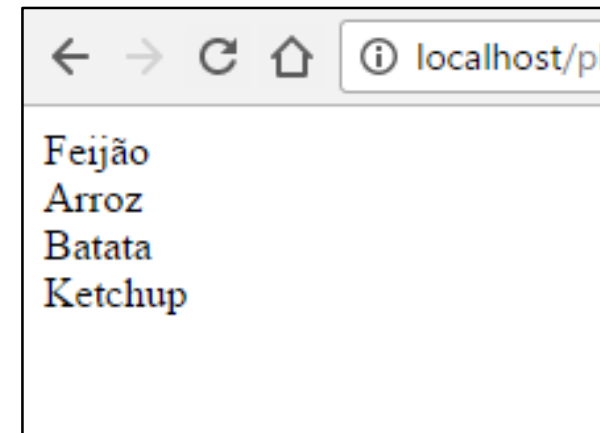
```
Array ( [1] => Feijão [2] => Arroz [3] => Batata [4] => Ketchup )
```

Array Linear

2ª → utilizando um laço de repetição para mostrar todos os valores.

Observação: Forma mais utilizada!

```
1  <?php
2  $lista_compras[1] = "Feijão";
3  $lista_compras[2] = "Arroz";
4  $lista_compras[3] = "Batata";
5  $lista_compras[4] = "Ketchup";
6
7  for($a=1; $a<=4; $a++){
8      print($lista_compras[$a]);
9      print"<br/>";
10 }
11 ?>
```



Atividades Array Linear

1. Faça um vetor com 30 casas. Guarde seu nome nas 30 casas do vetor. Mostre na tela seu nome e a posição do vetor que seu nome está ocupando.(Fazer utilizando repetição).
2. Faça um vetor receber todos os números pares de 1 a 100.
3. Crie um vetor A contendo 7 elementos, em cada posição guarde os números de 1 a 7. Crie um vetor B contendo 7 elementos, em cada posição guarde os dias da semana começando por Domingo. Crie um vetor C para receber o vetor A e o vetor B. Imprimir o vetor C.
Obs.: A saída da primeira casa do vetor C deverá ser: “1 - Domingo”.
4. Faça um programa que crie um vetor de inteiros de 10 posições. leia os valores e imprima somente os valores do vetor que forem impares.
5. Faça um programa que crie um vetor de inteiros de 10 posições leia os valores e imprima os valores onde o índice for impar.

Arrays Associativos

- Até o momento, exemplificamos array que faz uso de índices inteiros como chave para acesso aos valores. Estes são mais conhecidos como arrays lineares. Entretanto, você pode especificar uma chave do tipo string também.
- Arrays com essa característica são conhecidos como arrays associativos. Veja exemplo de como defini-los:

```
<?php
$dadosUsuario = array(
    "nome" => "Fulano de Tal",
    "email" => "fulano@email.com",
    "senha" => "a1b2c3");
print $dadosUsuario["nome"] . "<br/>";
print $dadosUsuario["senha"] . "<br/>";
print $dadosUsuario["email"] . "<br/>";
?>
```

Arrays Associativos

- E se você desejar utilizar um laço de repetição para este tipo de array?
- Como os índices nesse array não são números, não podemos utilizar um contador simples em um loop **for** para trabalhar com o array. A maneira de listar o conteúdo desse vetor portanto seria utilizando o loop **foreach**.
- Sintaxe básica do **foreach**

```
foreach ($array as $apelido) {  
    # code...  
}
```

Estrutura foreach

Exemplo: Utilizando **foreach** para arrays com índices numéricos

```
1  <?php
2  $lista_compras[1] = "Feijão";
3  $lista_compras[2] = "Arroz";
4  $lista_compras[3] = "Batata";
5  $lista_compras[4] = "Ketchup";
6
7  foreach ($lista_compras as $compras) {
8      print"$compras <br/>";
9  }
10 ?>
```

Estrutura foreach

Exemplo: Utilizando **foreach** para arrays com diferentes índices

```
<?php
$dadosUsuario = array(
    "nome" => "Fulano de Tal",
    "email" => "fulano@email.com",
    "senha" => "a1b2c3");

foreach ($dadosUsuario as $atributo => $valor) {
    print "$atributo - $valor <br/>";
}
?>
```

Arrays Associativos

- A vantagem dos arrays associativos é que, não só permite o acesso a elementos individualizados, mas também não nos obriga a conhecer qual a posição ou ordem dos elementos dentro do array.
- O array **\$_POST**, já estudado, é um exemplo de array associativo já disponível ao programador PHP. Que armazena todos os valores de um formulário, usando os atributos **name** como chave.

Atividades Arrays Associativos

1. Faça um formulário que peça para o usuário digitar: Nome, Senha, e-mail, idade e altura(em centímetros). Guardar estes valores em apenas um vetor chamado `$dados_usuario[]`, criar um índice neste vetor para cada campo recebido. Exemplo.

`$dados_usuario["nome"]` = Nome enviado do formulário.

`$dados_usuario["senha"]` = Senha enviada do formulário.

Mostre os valores do array na tela.

2. O mesmo exemplo anterior mostre os valores do array utilizando `foreach()`, depois mostre utilizando `var_dump()`, em seguida, `print_r()`. Observe a diferença entre cada um deles.

Arrays Multidimensionais (Matriz)

- O que é?

Tabuleiro de xadrez



Arrays Multidimensionais (Matriz)

- É importante conhecermos como criar arrays com mais de uma dimensão. Basta adicionar mais um par de colchetes para cada dimensão adicional.
- Exemplo:

```
1  <?php
2  // Descrição: array linear bidimen
3  $tabela[0][0] = 10;
4  $tabela[0][1] = 20;
5  $tabela[0][2] = 30;
6  $tabela[1][0] = 40;
7  $tabela[1][1] = 50;
8  $tabela[1][2] = 60;
9
10 for($i = 0; $i < 3; $i++){
11     for($j = 0; $j < 3; $j++){
12         print $tabela[$i][$j]. " ";
13     }
14     print "<br/>";
15 }
16 ?>
```

Dúvidas



Atividades Arrays Multidimensionais (Matriz)

1. Crie uma matriz 5x5, guarde os valores de 1 a 25 dentro dessa matriz. Imprimir a matriz utilizando a estrutura de repetição (for).
2. Utilizar o exemplo anterior e somar todos os valores desta matriz.
3. Utilizar o exemplo da questão 1 e mostrar os valores na tela dentro de uma tabela.
4. Criar uma matriz 3x3, guardar os valores de -1 a -9 dentro dessa matriz. Mostrar na tela apenas os valores da matriz que o índice da linha e coluna são iguais. Para verificar se os índices são iguais utilize a estrutura condicional if.