SW-I
SISTEMAS WEB I

Prof. Anderson Vanin

AULA 08 – Arrays em PHP

Conteúdo

- **Conceito de *array* em PHP**
- *Arrays* indexados: definição e sintaxe
- Acessando elementos por índice
- Inserindo/modificando valores
- Exercícios

Conceito de *array* em PHP

Um **array** é uma estrutura de dados fundamental na programação que armazena **uma coleção ordenada de elementos**. Esses elementos podem ser acessados por meio de **índices ou chaves**.

Principais Características:

- **Coleção de elementos:** Armazena múltiplos valores em uma única variável.
- **Organização indexada:** Cada elemento possui uma posição identificável (índice ou chave).
- **Tipos de dados:** Pode conter elementos de qualquer tipo (números, strings, objetos, outros arrays).
- **Tamanho flexível:** Na maioria das linguagens, o tamanho pode crescer ou diminuir dinamicamente.

Conceito de *array* em PHP

Analogia Prática:

Pense em um array como:

- Uma estante com compartimentos numerados (índices).
- Uma lista de contatos no seu telefone (cada nome tem uma posição).
- Uma fileira de caixas postais, cada uma com seu próprio número.

Conceito de *array* em PHP

Tipos Comuns de Arrays:

- **Indexados/numéricos**: Acessados por índices numéricos (0, 1, 2...)
- **Associativos**: Acessados por chaves nomeadas (como dicionários)
- **Multidimensionais**: *Arrays* que contêm outros *arrays*

***Arrays* são essenciais para manipulação eficiente de conjuntos de dados em programação, permitindo operações complexas com coleções de valores.**

Conteúdo

- Conceito de *array* em PHP
- **Arrays indexados: definição e sintaxe**
- Acessando elementos por índice
- Inserindo/modificando valores
- Exercícios

Arrays indexados

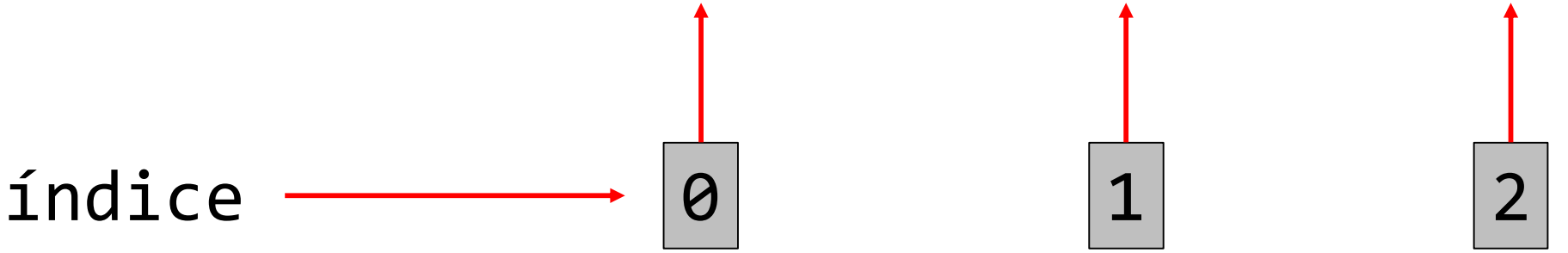
Em PHP, arrays indexados (também chamados de arrays numéricos ou listas) são um tipo de array onde os elementos são armazenados com índices numéricos sequenciais, começando do o (zero) por padrão.

Características Principais:

- Os índices são números inteiros.
- A indexação começa em o (zero) por padrão.
- Os elementos são acessados pela sua posição numérica.
- Mantêm uma ordem sequencial.

Arrays indexados

```
$cores = ["vermelho", "verde", "azul"];
```



Arrays indexados

Sintaxe básica:

```
// Sintaxe tradicional

$frutas = array("Maçã", "Banana", "Laranja");



// Sintaxe de colchetes (a partir do PHP 5.4)

$frutas = ["Maçã", "Banana", "Laranja"];
```

Arrays indexados

Funções Úteis para Arrays Indexados:

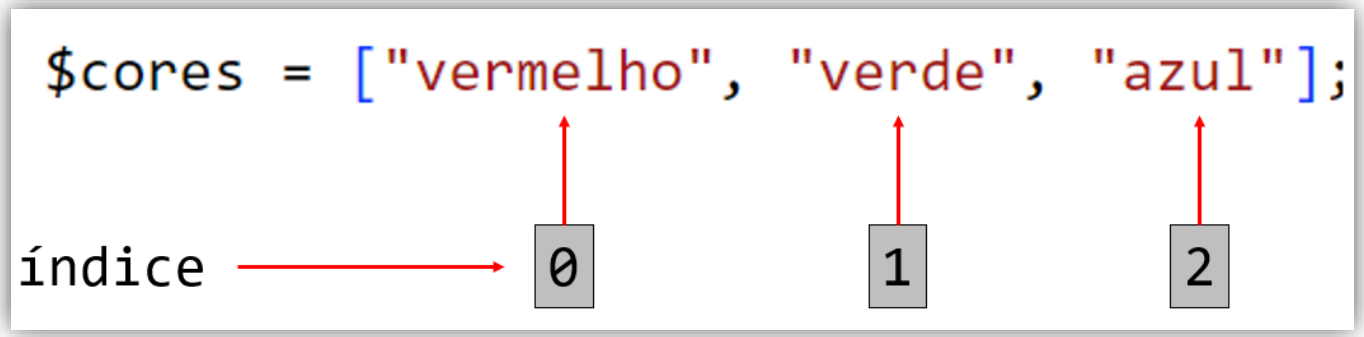
- **count()** - Conta o número de elementos.
- **array_push()** - Adiciona um ou mais elementos no final.
- **array_pop()** - Remove o último elemento.
- **array_shift()** - Remove o primeiro elemento.
- **array_unshift()** - Adiciona um ou mais elementos no início.
- **sort()** - Ordena o array.
- **array_slice()** - Extrai uma parte do array.

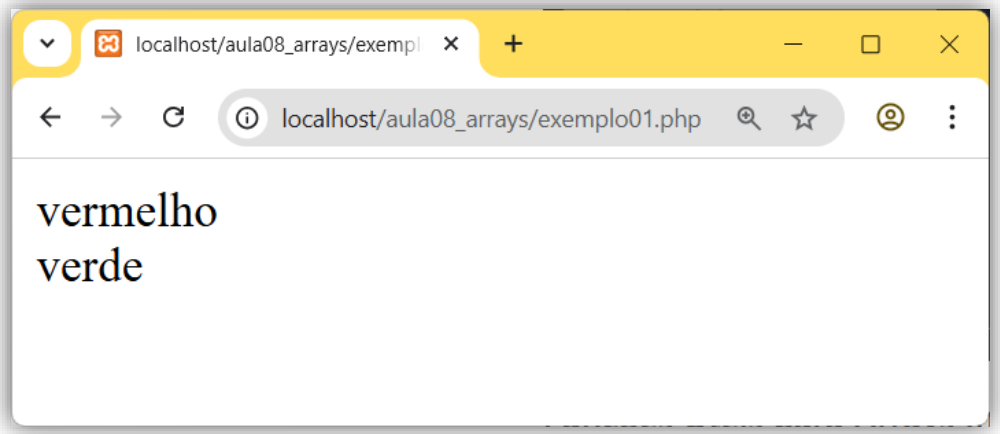
Conteúdo

- Conceito de *array* em PHP
- *Arrays* indexados: definição e sintaxe
- **Acessando elementos por índice**
- Inserindo/modificando valores
- Exercícios

Acessando elementos por índice

```php
// Criando um array indexado
$cores = ["vermelho", "verde", "azul"];


// Acessando elementos
echo $cores[0]; // Output: vermelho
echo "<br>";    // Output: Faz uma quebra de linha
echo $cores[1]; // Output: verde
```





Acessando elementos por índice - FOREACH

O ***foreach*** é uma estrutura de repetição da linguagem de programação PHP. Ele é usado para facilitar a iteração de estruturas como arrays, objetos e outros tipos que são iteráveis.

A sintaxe básica do foreach no PHP é a seguinte:

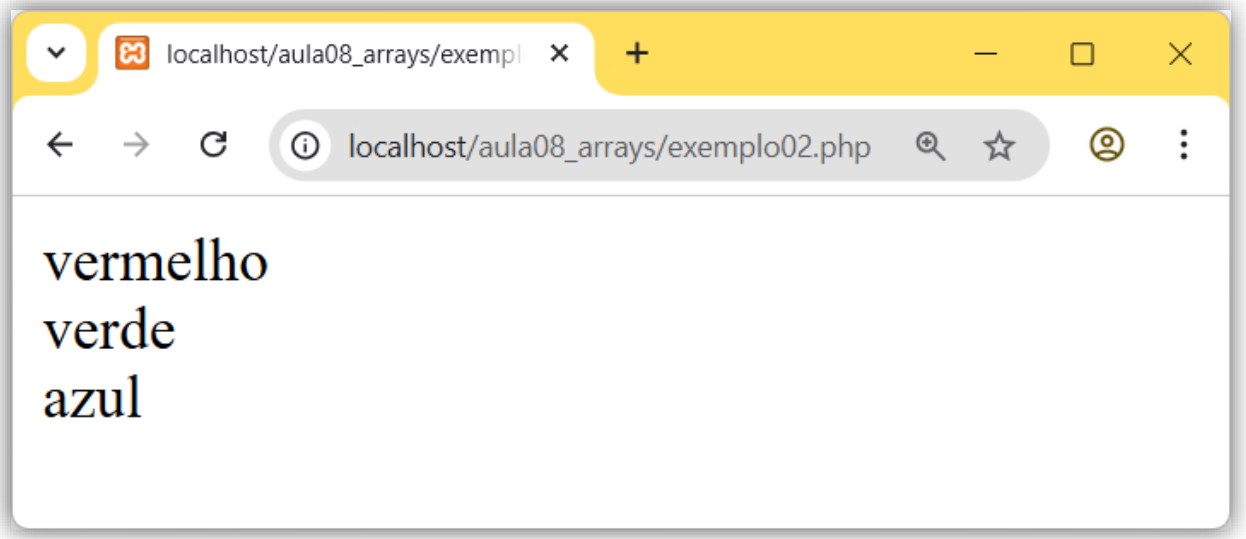
```
foreach ($coleção as $chave => $valor) {
      //disponível variáveis $chave e $valor
}
```

Podemos ler o código acima do seguinte modo:

Para cada item (na variável $coleção, coloque o índice na variável $chave e o valor na variável $valor)

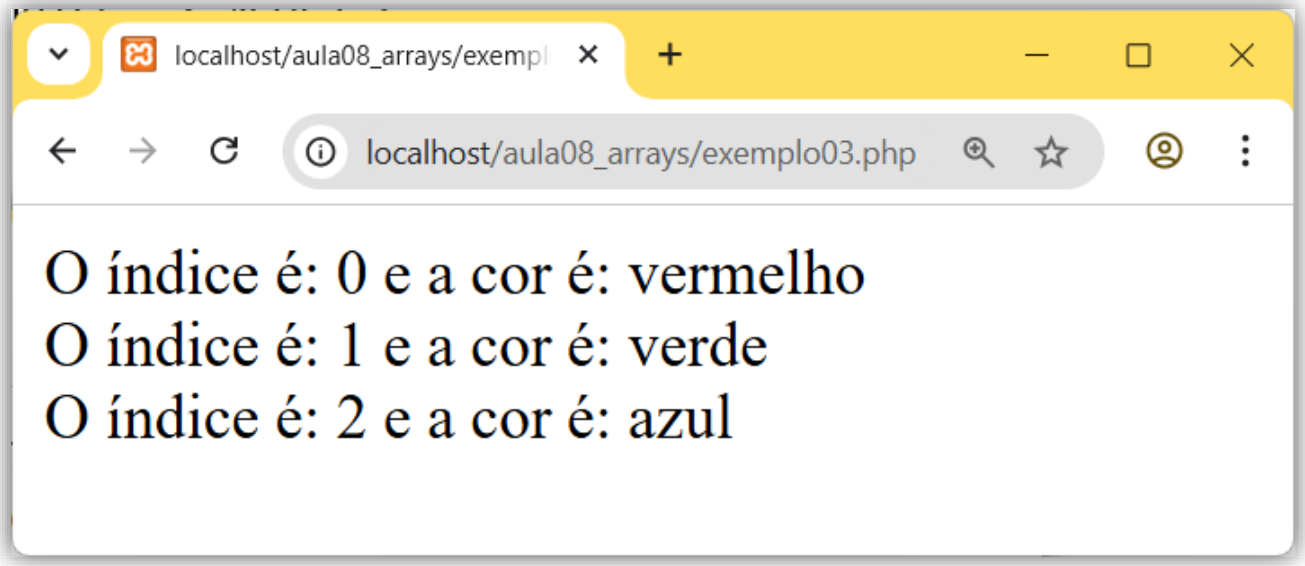
Acessando elementos por índice - FOREACH

```php
// Criando um array indexado
$cores = ["vermelho", "verde", "azul"];

// Acessando elementos - Iterando com foreach
foreach ($cores as $cor) {
    echo $cor . "<br>";
}
```



Acessando elementos por índice - FOREACH

```
// Criando um array indexado
$cores = ["vermelho", "verde", "azul"];


// Acessando elementos - Iterando com foreach
foreach ($cores as $indice => $cor) {
    echo "O índice é: $indice e a cor é: $cor <br>";
}
```



Acessando elementos por índice - FOR

O **for** é a estrutura de repetição do PHP que utilizamos quando sabemos a quantidade de repetições que devem ser executadas.

Sintaxe do for:

```
for (expressão 1; expressão 2; expressão 3) {
    // bloco de código
}
```

- **Expressão 1:** Utilizamos para declarar e inicializar as variáveis que faremos uso para controlar o número de iterações do loop;
- **Expressão 2:** Expressão booleana, validada antes de cada iteração do loop. O loop somente será executado enquanto essa expressão retornar **true**;
- **Expressão 3:** Executada ao final de cada iteração, normalmente a utilizamos para declarar a forma de atualização do valor da variável avaliada na expressão 2.

Acessando elementos por índice - FOR

Para utilizar a estrutura **for** para acessar os elementos de um *array*, precisamos primeiramente saber o número exato de iterações necessárias, que nesse caso é a quantidade de elementos (índices) no *array*.

O PHP oferece uma função interna (*built-in*) que fornece a quantidade de elementos que um array possui, a função **count** que conta todos os elementos de um array ou de um objeto.

```
   // Criando um array indexado
$cores = ["vermelho", "verde", "azul"];

// Contando o número de elementos
$qtde = count($cores);

echo "Existem $qtde elementos no array";
//Output: Existem 3 elementos no array
```

Acessando elementos por índice - FOR

Agora que sabemos a quantidade exata de iterações, podemos usar a estrutura for para percorrer cada um dos índices do array.

```
   // Criando um array indexado
$cores = ["vermelho", "verde", "azul"];

// Contando o número de elementos
$qtde = count($cores);

echo "Existem $qtde elementos no array <br>";
//Output: Existem 3 elementos no array

echo "<hr>";

for ($i=0; $i < $qtde ; $i++) {
    echo "Cor: $cores[$i] <br>";
}
```

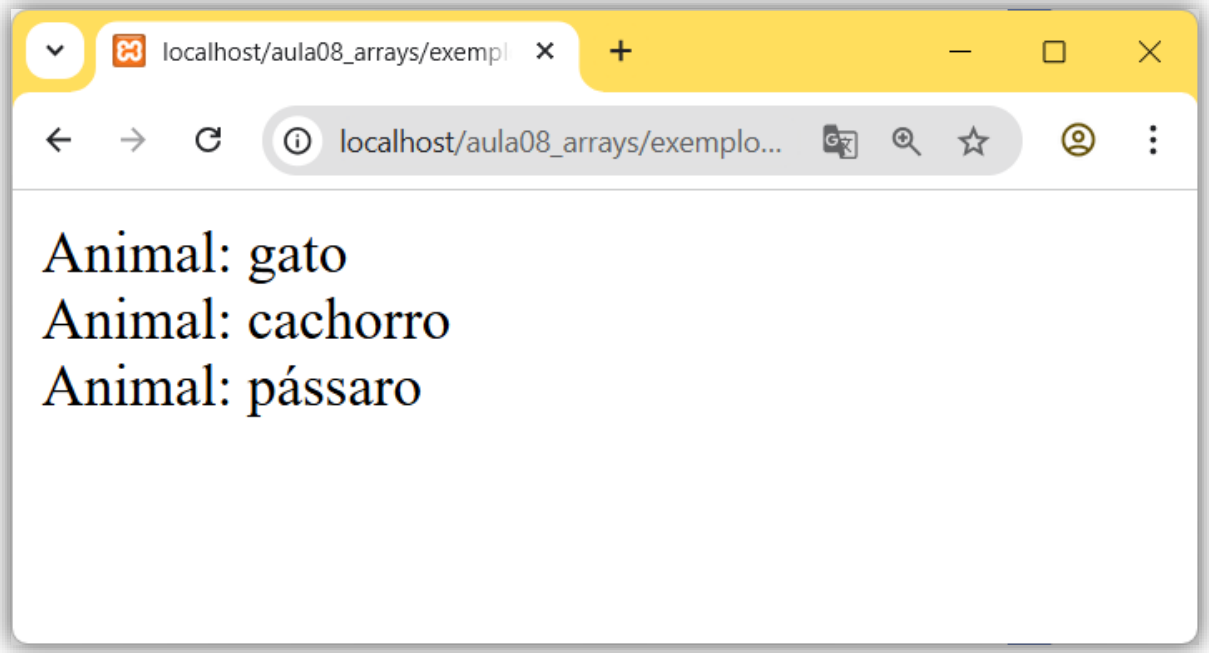
Conteúdo

- Conceito de *array* em PHP
- *Arrays* indexados: definição e sintaxe
- Acessando elementos por índice
- **Inserindo/modificando valores**
- Exercícios

Inserindo/modificando valores

Para inserir um novo valor ao final de um array podemos usar:

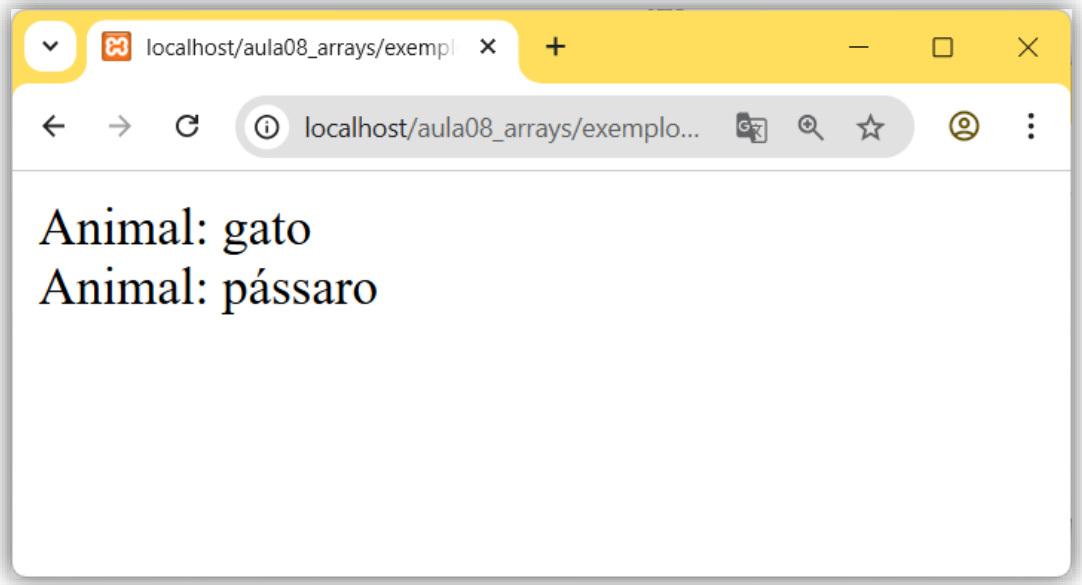
```
$animais = ["gato", "cachorro"];
$animais[] = "pássaro"; // Inserido no final
// Acessando elementos
foreach ($animais as $animal) {
     echo "Animal: $animal <br>";
}
```



Inserindo/modificando valores

Podemos, também, inserir um novo valor em um índice específico de um array (**desde que este índice exista**):

```
$animais = ["gato", "cachorro"];
$animais[1] = "pássaro";
// Inserido no índice 1 – substitui o que tiver

// Acessando elementos
foreach ($animais as $animal) {
   echo "Animal: $animal <br>";
}
```



Inserindo/modificando valores

Funções úteis:

- **array_push():** insere no final
- **array_unshift():** insere no início

Conteúdo

- Conceito de *array* em PHP
- *Arrays* indexados: definição e sintaxe
- Acessando elementos por índice
- Inserindo/modificando valores
- **Exercícios**

Exercícios

1. Crie um array com 5 nomes e exiba todos.
2. Adicione um novo nome ao array e exiba.
3. Modifique o segundo nome da lista.
4. Crie um array de 10 números e calcule a média.
5. Crie um array com os dias da semana e exiba os dias úteis.