



**INSTITUTO
FEDERAL**

Sudeste de
Minas Gerais

Campus
Manhuaçu

Desenvolvimento Web III

Vetores e Funções

Prof. Leonardo C. R. Soares - leonardo.soares@ifsudestemg.edu.br

Instituto Federal do Sudeste de Minas Gerais

1 de novembro de 2024





Vetores

Definição

Vetor (*array*) é um tipo especial de variável que pode conter mais de um valor por vez.





Vetores

Definição

Vetor (*array*) é um tipo especial de variável que pode conter mais de um valor por vez. *Especificamente em PHP um vetor é um mapa ordenado, um tipo que associa valores a chaves.*





Vetores

Definição

Vetor (*array*) é um tipo especial de variável que pode conter mais de um valor por vez. *Especificamente em PHP um vetor é um mapa ordenado, um tipo que associa valores a chaves.*

Tipos de vetores em PHP

Existem três tipos de *vetores* em PHP:

- ▶ Vetores indexados;
- ▶ Vetores associativos;
- ▶ Vetores multidimensionais.





Vetores indexados

Podemos criar vetores indexados de duas formas:

1. Utilizando a função array. Os índices são gerados automaticamente iniciando-se em zero:

```
$professores = array("Leo", "Loham", "Antonio");
```





Vetores indexados

Podemos criar vetores indexados de duas formas:

1. Utilizando a função array. Os índices são gerados automaticamente iniciando-se em zero:

```
$professores = array("Leo", "Loham", "Antonio");
```

2. Associando manualmente os índices:

```
$professores[0] = "Leo";  
$professores[1] = "Loham";  
$professores[] = "Antonio";
```





Vetores associativos

Podemos criar vetores associativos de duas formas:

1. Utilizando a função array:

```
$age=array("Leo"=>41, "Loham"=>26, "Antonio"=>29);
```





Vetores associativos

Podemos criar vetores associativos de duas formas:

1. Utilizando a função array:

```
$age=array("Leo"=>41, "Loham"=>26, "Antonio"=>29);
```

2. Associando manualmente os índices:

```
$age['Leo'] = 41;  
$age['Loham'] = 26;  
$age['Antonio'] = 29;
```





Exemplo

Carregamos um vetor com 10 números inteiros entre -10 e 10. Depois, o programa calcula e imprime a quantidade de números positivos que o vetor possui.





Exemplo

Carregamos um vetor com 10 números inteiros entre -10 e 10. Depois, o programa calcula e imprime a quantidade de números positivos que o vetor possui.

```
<?php
for ($i=0;$i<10;++$i)
    $num[] = rand(-10,10);

$cont = 0;
for ($i=0;$i<10;++$i)
    if ($num[$i]>0)
        $cont++;
echo "O vetor possui $cont números positivos";
?>
```





Exercício rápido

Carregue um vetor de 100 posições com números aleatórios entre 0 e 2500. Imprima o maior valor carregado no vetor e sua posição.





Exercício rápido

Carregue um vetor de 100 posições com números aleatórios entre 0 e 2500. Imprima o maior valor carregado no vetor e sua posição.

```
<?php
for ($i=0;$i<100;++$i)
    $num[] = rand(0,2500);

$idx = 0;
for ($i=0;$i<count($num);++$i)
    if ($num[$i]>$num[$idx])
        $idx = $i;
echo "O maior valor é ".$num[$idx] . " e ele está na
    posição $idx.";

?>
```





Exercício rápido

Carregue um vetor de 100 posições com números aleatórios entre 0 e 2500. Imprima o maior valor carregado no vetor e sua posição.

```
<?php
for ($i=0;$i<100;++$i)
    $num[] = rand(0,2500);

$idx = 0;
for ($i=0;$i<count($num);++$i)
    if ($num[$i]>$num[$idx])
        $idx = $i;
echo "O maior valor é ".$num[$idx] . " e ele está na
    posição $idx.";
?>
```

Mas eu queria com foreach...





Exercício rápido

Carregue um vetor de 100 posições com números aleatórios entre 0 e 2500. Imprima o maior valor carregado no vetor e sua posição.

```
<?php
for ($i=0;$i<100;++$i)
    $num[] = rand(0,2500);
$idx = 0;
foreach ($num as $chave => $valor)
    if ($valor>$num[$idx])
        $idx = $chave;
echo "O maior valor é ".$num[$idx] .
" e ele está na posição $idx.";
?>
```





I need more...

Para saber mais

- ▶ Livro: PHP: programe de forma rápida e prática (livro disponível em minha biblioteca) - Capítulo 9;
- ▶ Documentação oficial (em português);
- ▶ Artigo Devmedia;
- ▶ PHP Arrays W3Schools.





Funções

Definição

Funções são blocos de códigos desenvolvidos para resolver um determinado problema.





Funções

Definição

Funções são blocos de códigos desenvolvidos para resolver um determinado problema.

Sintaxe

```
function functionName(parameters): tipo de retorno {  
    code to be executed;  
}
```





Funções

Tipos de retorno

Os tipos de retorno possíveis para uma função em PHP são:

- ▶ void: Não haverá retorno;
- ▶ bool: Valor lógico (true ou false);
- ▶ int: Valor inteiro;
- ▶ float: Valor com ponto flutuante;
- ▶ string: Sequência de caracteres;
- ▶ array: Um mapa ordenado de chave/valor (vetor);
- ▶ object: Uma instância de uma classe pré-definida.





Funções

Tipos de retorno

Os tipos de retorno possíveis para uma função em PHP são:

- ▶ void: Não haverá retorno;
- ▶ bool: Valor lógico (true ou false);
- ▶ int: Valor inteiro;
- ▶ float: Valor com ponto flutuante;
- ▶ string: Sequência de caracteres;
- ▶ array: Um mapa ordenado de chave/valor (vetor);
- ▶ object: Uma instância de uma classe pré-definida.

A partir do PHP 8.0 podemos usar **Union Types**. Pode-se especificar mais de um tipo separado por pipe: int|float.





Exemplo

```
<?php
function fatorial(int $n): float{
    $f = 1;
    while($n>1){
        $f*=$n;
        $n--;
    }
    return $f;
}
```

```
$x = rand(1,10);
$fat = fatorial($x);
echo "O fatorial de $x é $fat";
?>
```





Exemplo

```
<?php
function geraVetor(int $tamanho):array{
    for($i=0;$i<=$tamanho;$i++)
        $vet[]=$vet[rand(0,10)];
    return $vet;
}
function somaVet(array $vet):int{
    $soma = 0;
    for ($i=0; $i<count($vet);++$i)
        $soma+=$vet[$i];
    return $soma;
}
$x = geraVetor(10);
print_r($x);
echo "A soma dos elementos do vetor é: ". somaVet($x);
?>
```





Funções

Algumas considerações

- ▶ Pode-se declarar as funções em um arquivo separado para dar maior usabilidade às mesmas;





Funções

Algumas considerações

- ▶ Pode-se declarar as funções em um arquivo separado para dar maior usabilidade às mesmas;
- ▶ As variáveis declaradas dentro das funções são locais;





Funções

Algumas considerações

- ▶ Pode-se declarar as funções em um arquivo separado para dar maior usabilidade às mesmas;
- ▶ As variáveis declaradas dentro das funções são locais;
- ▶ Caso uma função precise utilizar uma variável declarada fora do seu escopo, deve utilizar a palavra-chave **global**.





Exemplo

```
<?php
function ajuste(float $sal):float{
    global $fator;
    return $sal*$fator;
}

$fator = 1.20;
$salario = 1000;
$novo = ajuste($salario);
echo "Seu novo salário é $novo<br>";
?>
```





Funções

Algumas considerações

- ▶ As variáveis enviadas como parâmetros das funções são passadas **por valor**, ou seja, uma nova variável é criada e inicializada com o valor da variável original. Alterar o valor dos parâmetros não interfere nas variáveis originais.





Exemplo

```
<?php
function troca(int $x, int $y):void{
    $aux = $x;
    $x = $y;
    $y = $aux;
}

$x=10;
$y=5;
echo "\$x = $x e \$y = $y<br>";
troca($x,$y);
echo "\$x = $x e \$y = $y<br>";
?>
```





Funções

Algumas considerações

- ▶ Caso seja necessário passar parâmetros **por referência**, ou seja, os parâmetros irão apontar para o mesmo endereço de memória das variáveis originais, utiliza-se o caractere & antes do nome do parâmetro.





Exemplo

```
<?php
function troca(int &$x, int &$y):void{
    $aux = $x;
    $x = $y;
    $y = $aux;
}

$x=10;
$y=5;
echo "\$x = $x e \$y = $y<br>";
troca($x,$y);
echo "\$x = $x e \$y = $y<br>";

?>
```





Dúvidas?





Exercícios

1. Faça uma função que receba como parâmetro um vetor e retorne a quantidade de números pares que o vetor possui.
2. Faça uma função que receba como parâmetro um vetor e retorne a média dos valores contidos no mesmo.
3. Faça uma função que receba como parâmetro dois vetores e retorne um vetor contendo a soma deles, elemento a elemento.
4. Faça uma função que receba como parâmetro um vetor e retorne um vetor de conteúdo igual ao primeiro mas sem valores repetidos (não utilize funções prontas do PHP).



