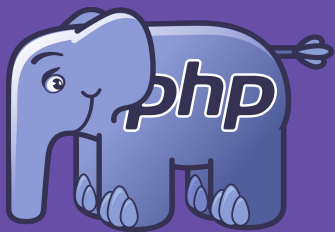


Curso PHP e MySQL

_estruturas de dados em PHP

Arrays e Objetos.

Adriel Sales



_arrays

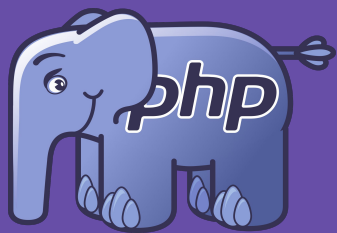
Array é um tipo de dado, assim como integer, float, string ou boolean. Contudo, um array **pode armazenar mais de um valor**, relacionando-o a uma chave. No PHP, um mesmo array pode conter diferentes tipos de dados, incluindo novos arrays.

A sintaxe de um php array funciona da seguinte forma:

```
array (  
    primeira_chave => primeiro_valor,  
    segunda_chave => segundo_valor,  
    terceira_chave => terceiro_valor,  
    ...  
);
```



Ou seja,
Um array é uma variável que nos permite armazenar mais de um valor nela.



_arrays

Tipos de arrays:

Arrays numéricos/indexados:

São arrays que têm um índice numérico como chave.

```
$cars = array("Volvo", "BMW", "Toyota");
```

or the index can be assigned manually:

```
$cars[0] = "Volvo";  
$cars[1] = "BMW";  
$cars[2] = "Toyota";
```

Arrays associativos:

São arrays que têm chaves nomeadas.

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

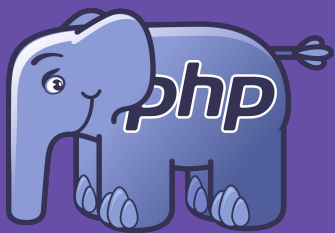
or:

```
$age['Peter'] = "35";  
$age['Ben'] = "37";  
$age['Joe'] = "43";
```

Arrays multidimensionais:

são arrays que têm um ou mais arrays dentro dele. Arrays multidimensionais também são chamados matrizes.

```
$cars = array (  
    array("Volvo",22,18),  
    array("BMW",15,13),  
    array("Saab",5,2),  
    array("Land Rover",17,15)  
);
```

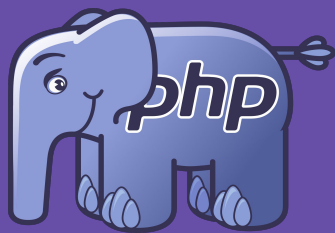


_arrays

Array é um tipo de dado, assim como integer, float, string ou boolean. Contudo, um array **pode armazenar mais de um valor**, relacionando-o a uma chave. No PHP, um mesmo array pode conter diferentes tipos de dados, incluindo novos arrays.

```
array(  
    chave => valor,  
    chave2 => valor2,  
    chave3 => valor3,  
    ...  
)
```

```
<?php  
$array = array(  
    "foo" => "bar",  
    "bar" => "foo",  
);  
  
// Utilizando a sintaxe curta  
$array = [  
    "foo" => "bar",  
    "bar" => "foo",  
];  
?>
```



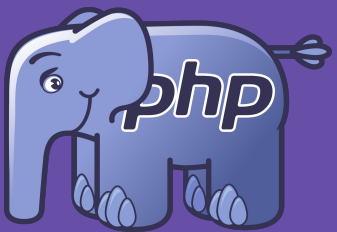
_arrays

A **chave** de um array **pode** ser um **int** ou uma **string**.
O **valor** pode ser de **qualquer tipo**.

```
<?php
$array = array(
    "foo" => "bar",
    "bar" => "foo",
    100   => -100,
    -100  => 100,
);
var_dump($array);
?>
```



```
array(4) {
    ["foo"]=>
    string(3) "bar"
    ["bar"]=>
    string(3) "foo"
    [100]=>
    int(-100)
    [-100]=>
    int(100)
}
```

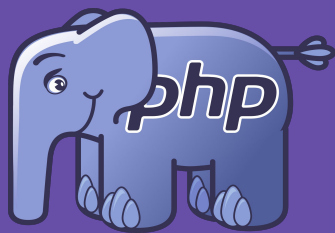


_arrays

```
1 <?php
2
3 //Declaração de arrays:
4
5 $array_exemplo1 = array(); //assim
6
7 $array_exemplo2 = []; //ou assim
8
9 $cars = array("Volvo", "BMW", "Toyota");
10 echo "I like $cars[0], $cars[1] and $cars[2]" . PHP_EOL;
11
12 $cars = ["Volvo", "BMW", "Toyota"];
13 echo "Yes, I like $cars[0], $cars[1] and $cars[2]" . PHP_EOL;
14
15 var_dump($array_exemplo1, $array_exemplo2);
16
17 ?>
```

Result for 8.1.10:

```
I like Volvo, BMW and Toyota
Yes, I like Volvo, BMW and Toyota
array(0) {
}
array(0) {
}
```



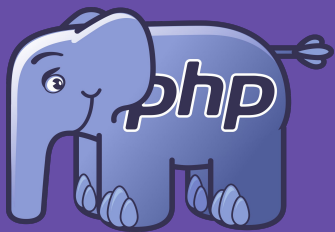
_arrays

A **chave** de um array **pode** ser um **int** ou uma **string**.
O **valor** pode ser de **qualquer tipo**.

```
<?php
$array = array(
    "foo" => "bar",
    "bar" => "foo",
    100   => -100,
    -100  => 100,
);
var_dump($array);
?>
```



```
array(4) {
    ["foo"]=>
    string(3) "bar"
    ["bar"]=>
    string(3) "foo"
    [100]=>
    int(-100)
    [-100]=>
    int(100)
}
```



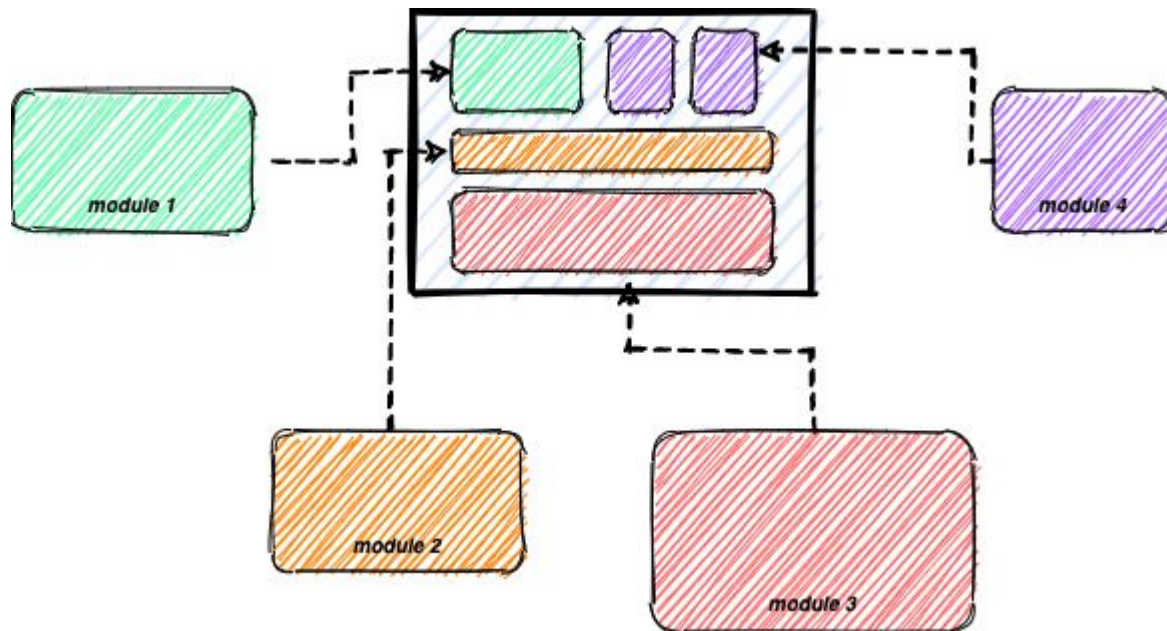
_arrays multidimensionais

são arrays que têm um ou mais arrays dentro dele.
Arrays multidimensionais também são chamados matrizes.

```
1  <?php
2
3  $turma = [
4      [
5          'nome' => 'Joao da sIlva',
6          'idade' => 18,
7          'rg' => '5556666',
8          'cpf' => '010.200.256-89',
9      ]
10 ];
11
12 var_dump($turma[0]['nome'], $turma[0]['idade'], $turma[0]['rg'], $turma[0]['cpf']);
13
14 /**
15  * Resultado:
16  string(13) "Joao da sIlva" //nome
17  int(18) // idade
18  string(7) "5556666" // rg
19  string(14) "010.200.256-89" //cpf
20  */
```




Incluindo conteúdo

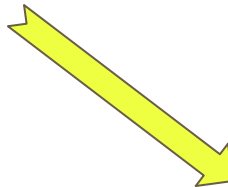




Incluindo conteúdo

Há quatro funções básicas:

- include “meu-arquivo.php”;
- include_once “meu-arquivo.php”;
- require “meu-arquivo.php”;
- require_once “meu-arquivo.php”;





Incluindo conteúdo

Include

A declaração include inclui e avalia o arquivo informado. Se o arquivo não for encontrado no `include_path`, a declaração include checará no diretório do script que o executa e no diretório de trabalho corrente, antes de falhar.

```
include 'file.txt'; // Works.  
include 'file.php'; // Works.
```



Incluindo conteúdo

include_once

Avalia o arquivo informado durante a execução do script. Este é um comportamento similar a declaração include, com a única diferença que, se o código do arquivo já foi incluído, não o fará novamente, e o include_once retornará TRUE.

```
<?php
include_once "a.php"; // this will include a.php
include_once "A.php"; // this will include a.php again! (PHP 4 only)
?>
```



Incluindo conteúdo

require

É idêntica à `include`, exceto que em caso de falha também produzirá um erro fatal de nível `E_COMPILE_ERROR`. Em outras palavras, **ele parará** o script enquanto que o `include` apenas emitirá um alerta (`E_WARNING`) permitindo que o script continue.

```
<?php
require('somefile.php');
?>
```



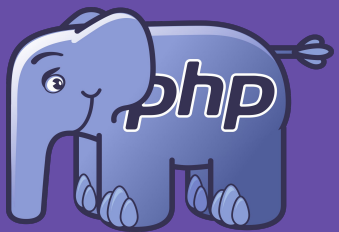
Incluindo conteúdo

require_once

É idêntica à *require*, exceto que o PHP verificará se o arquivo já foi incluído, e em caso afirmativo, não o incluirá (exigirá) novamente.

```
<?php
define('__ROOT__', dirname(dirname(__FILE__)));
require_once(__ROOT__.'/config.php');
?>
```

Dica: utilize a função `fgets(STDIN)` para capturar o valor digitado no terminal.



_atividades de fixação

1. Utilizando o seu conhecimento em laços de repetição, crie dois arrays: **listaCrescente:** com uma sequência de inteiros de 0 a 100; **listaDecrescente:** com uma sequência de inteiros de 100 para 0; Ao final, faça a impressão de cada array no terminal;
2. Faça um programa PHP que devolve um array de números lidos do teclado.
3. Percorra o array a seguir e exiba a mensagem indicando o nome e a posição deste quando for localizado no array: [“João”, “José”, “Maria”, “Severino”, “Antônio”, “Marcos”, “Manoel”, “Filipe”, “André”, “Paulo”]; Use a linha de comando para digitar o nome;
4. Crie um array chamado “alunosDeMinhaTurma” e, com o recurso da função `fgets(STDIN)`, adicione os nomes de todos os seus colegas de sala. Ao final, faça a impressão dos nomes. (Será necessário usar um laço de repetição).
5. Crie um programa PHP que recebe um array de inteiros predefinido e retorna as quantidades de elementos do array que são números negativos e números positivos.



Professor:
Adriel Sales

Fonte? Vá na fonte!



<https://www.php.net>

<https://github.com/adrielacademico>