INSTITUTO FEDERAL
São Paulo

# PHP: coleções (*arrays*) e JSON

Prof. DR. Eder Pansani
epansani@gmail.com / epansani@ifsp.edu.br

# Conteúdo da aula

INSTITUTO FEDERAL
São Paulo

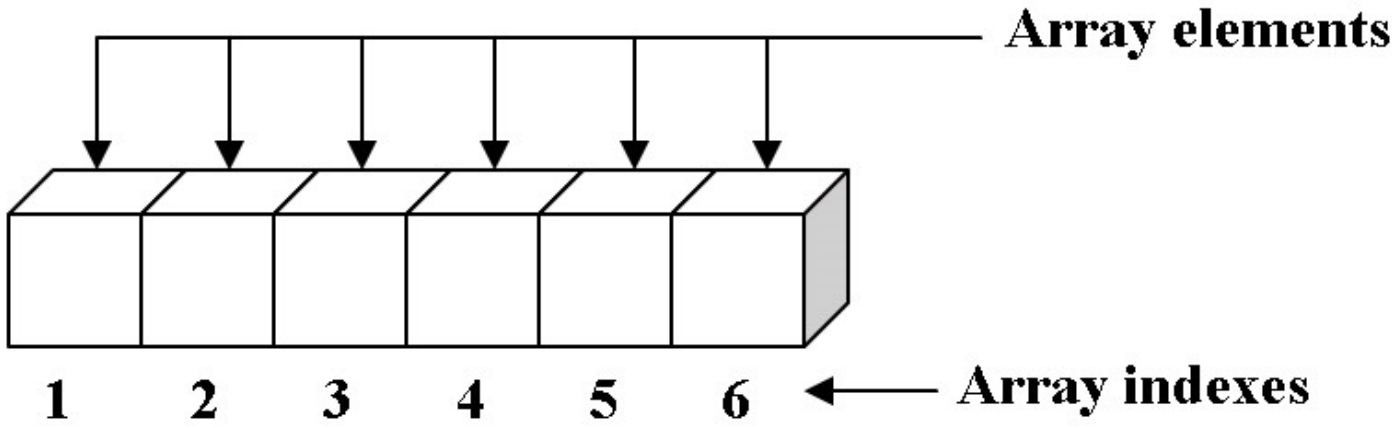
Nesta aula serão apresentados os conceitos de como o PHP lida com **coleções (*arrays*)** e **dados em formato JSON**.

INSTITUTO FEDERAL
São Paulo

# *Coleções (Arrays)*

# *Arrays*

Um *array* no PHP é um tipo de variável que relaciona valores a chaves.




**One-dimensional array with six elements**

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Um *array* é uma **variável especial**, que pode conter mais de um valor por vez. São especialmente úteis quando lidamos com uma lista grande de itens, pois permitem armazená-los em uma única variável e acessar os valores por meio dos índices.

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Também chamado de matriz, um *array* tem por função primordial armazenar vários valores em uma única variável.

https://www.w3schools.com/php/php_arrays.asp

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Exemplo:

```php
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>
```

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Considere a seguinte situação.

Se você tiver **uma lista de itens** (uma lista de nomes de carros, por exemplo), armazenar os carros em variáveis únicas pode ser uma opção viável.

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Considere a seguinte situação.

```
$cars1 = "Volvo";
$cars2 = "BMW";
$cars3 = "Toyota";
```

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Agora se você precisar **encontrar um carro específico**, você teria de percorrer todos os carros um a um, testar as três variáveis.

Sem problemas, certo?

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Porém e se ao invés de ter três carros **você tivesse 300**?

# *Arrays*

INSTITUTO FEDERAL
São Paulo

A solução para esse tipo de problema é o uso de *arrays*!

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Em PHP um array pode ser criado com o **construtor de linguagem** ***array()***. Ele leva qualquer quantidade de pares separados por vírgula chave => valor como argumentos.

```
array();
```

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Assim:

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

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Existe também uma **sintaxe curta** que substitui a declaração ***array()*** apenas por **[]**.

Veja no exemplo,

# Arrays

INSTITUTO FEDERAL
São Paulo

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

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

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Em PHP existem três tipos de *arrays*:

- ***Arrays indexados*** - *Arrays* com um índice numérico
- ***Arrays associativos*** - *Arrays* com chaves nomeadas
- ***Arrays multidimensionais*** - *Arrays* contendo um ou mais *arrays*

# *Arrays*

INSTITUTO FEDERAL
São Paulo

***Arrays indexados***:

Índices começam em 0 (automaticamente)

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

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

# *Arrays*

INSTITUTO FEDERAL
São Paulo

***Arrays associativos***:

Usam chaves nomeadas

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

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

# *Arrays*

INSTITUTO FEDERAL
São Paulo

***Arrays multidimensionais***:

São *arrays* que contém um ou mais *arrays*.

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

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Para acessar este tipo de *array* é preciso informar os dois índices (linha e coluna)

```php
<?php
echo $cars[0][0].": In stock: ".$cars[0][1].", sold: ".$cars[0][2].".<br>";
echo $cars[1][0].": In stock: ".$cars[1][1].", sold: ".$cars[1][2].".<br>";
echo $cars[2][0].": In stock: ".$cars[2][1].", sold: ".$cars[2][2].".<br>";
echo $cars[3][0].": In stock: ".$cars[3][1].", sold: ".$cars[3][2].".<br>";
?>
```

# *Arrays*

INSTITUTO FEDERAL
São Paulo

É possível realizar uma série de **operações e manipulações sobre os *arrays*** de forma muito simples usando as funções internas.

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Uma das mais básicas é a contagem do comprimento do *array* (a quantidade de elementos que ele contém).

A função *count()* faz isso:

```php
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo count($cars);
?>
```

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Outra necessidade básica em coleções é a ordenação. Alfabética ou numérica, crescente ou decrescente, existem funções para todas as situações.

https://www.w3schools.com/php/php_arrays_sort.asp

# *Arrays*

INSTITUTO FEDERAL
São Paulo

- `sort()` - ordenar arrays em ordem crescente
- `rsort()` - ordenar arrays em ordem decrescente
- `asort()` - classificar arrays associativos em ordem crescente, de acordo com o valor
- `ksort()` - classificar arrays associativos em ordem crescente, de acordo com a chave
- `arsort()` - classificar arrays associativos em ordem decrescente, de acordo com o valor
- `krsort()` - classificar arrays associativos em ordem decrescente, de acordo com a chave

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Além disso há outras funções interessantes, em minha opinião algumas delas são:

# *Arrays*

INSTITUTO FEDERAL
São Paulo

***array_unshift()*** - Adiciona um ou mais elementos no início de um array

https://www.php.net/manual/pt_BR/function.array-unshift.php

https://www.w3schools.com/php/func_array_unshift.asp

# *Arrays*

INSTITUTO FEDERAL
São Paulo

***array_push()*** - Adiciona um ou mais elementos no final de um array

https://www.php.net/manual/pt_BR/function.array-push.php

https://www.w3schools.com/php/func_array_push.asp

# *Arrays*

INSTITUTO FEDERAL
São Paulo

***array_shift()*** - Retira o primeiro elemento de um array

https://www.php.net/manual/en/function.array-shift.php

https://www.w3schools.com/php/func_array_shift.asp

# *Arrays*

INSTITUTO FEDERAL
São Paulo

***in_array()*** - Checa se um valor existe em um array

https://www.php.net/manual/pt_BR/function.in-array.php

https://www.w3schools.com/php/func_array_in_array.asp

# *Arrays*

INSTITUTO FEDERAL
São Paulo

***array_search()*** - Procura por um valor em um array e retorna sua chave correspondente caso seja encontrado

https://www.php.net/manual/pt_BR/function.array-search.php

https://www.w3schools.com/php/func_array_search.asp

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Existem ainda outras dezenas de funções para arrays que podem ser muitíssimo úteis em diversas situações, sempre procure antes de “fazer com suas próprias mãos”.

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Lista de funções:

https://www.php.net/manual/pt_BR/ref.array.php

https://www.w3schools.com/php/php_ref_array.asp

# *Arrays*

INSTITUTO FEDERAL
São Paulo

Referências e leitura complementar:

https://www.php.net/manual/pt_BR/language.types.array.php

https://www.w3schools.com/php/php_arrays.asp

INSTITUTO FEDERAL
São Paulo

# *JSON - JavaScript Object Notation*

# *JSON - JavaScript Object Notation*

INSTITUTO FEDERAL
São Paulo

JSON significa *JavaScript Object Notation* e é uma sintaxe para armazenar e trocar dados. Como é baseado em texto, ele pode ser facilmente enviado de e para um servidor e usado como formato de dados por qualquer linguagem de programação. https://www.w3schools.com/php/php_json.asp

# *JSON - JavaScript Object Notation*

INSTITUTO FEDERAL
São Paulo

JSON é uma formatação leve de troca de dados. Para seres humanos, é **fácil de ler e escrever**. Para máquinas, é **fácil de interpretar e gerar**. Está baseado em um subconjunto da linguagem de programação JavaScript, Standard ECMA-262 3a Edição -Dezembro - 1999.

# *JSON - JavaScript Object Notation*

INSTITUTO FEDERAL
São Paulo

JSON é completamente independente de linguagem, pois usa convenções que são familiares às linguagens C, incluindo C++, C#, Java, JavaScript, Perl, Python e muitas outras. Estas propriedades fazem com que JSON seja um formato ideal de troca de dados.

# *JSON - JavaScript Object Notation*

INSTITUTO FEDERAL
São Paulo

PHP é capaz de lidar com JSON de forma nativa usando as funções internas ***json_encode()*** e ***json_decode()***.

https://www.w3schools.com/php/php_json.asp

# JSON - JavaScript Object Notation

INSTITUTO FEDERAL
São Paulo

A função ***json_encode()*** é usada para codificar um valor para o formato JSON.

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

echo json_encode($age);
?>
```

# JSON - JavaScript Object Notation

INSTITUTO FEDERAL
São Paulo

## *json_encode()*

```
<!DOCTYPE html>
<html>
<body>

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

echo json_encode($age);
?>

</body>
</html>
```

```
{"Peter":35,"Ben":37,"Joe":43}
```

# *JSON - JavaScript Object Notation*

INSTITUTO FEDERAL
São Paulo

A função ***json_encode()*** é usada para decodificar um objeto JSON em um objeto PHP ou um array associativo.

```php
<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

var_dump(json_decode($jsonobj));
?>
```

# JSON - JavaScript Object Notation

INSTITUTO FEDERAL
São Paulo

## *json_decode()*

```
<!DOCTYPE html>
<html>
<body>

<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

var_dump(json_decode($jsonobj));
?>

</body>
</html>
```

```
object(stdClass)#1 (3) { ["Peter"]=> int(35) ["Ben"]=> int(37)
["Joe"]=> int(43) }
```

# *JSON - JavaScript Object Notation*

INSTITUTO FEDERAL
São Paulo

Referências e leitura complementar:

https://www.php.net/manual/en/function.json-decode.php

https://www.php.net/manual/en/function.json-encode.php

https://www.w3schools.com/php/php_json.asp

# *JSON - JavaScript Object Notation*

INSTITUTO FEDERAL
São Paulo

**Importante**, há duas formas para se acessar os valores decodificados de um objeto JSON. A primeira é usando a sintaxe de acesso à objetos e a segunda é usando índices de um array associativo.

# JSON - JavaScript Object Notation

INSTITUTO FEDERAL
São Paulo

Como acessar os valores de um **objeto PHP**:

```php
<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

$obj = json_decode($jsonobj);

echo $obj->Peter;
echo $obj->Ben;
echo $obj->Joe;
?>
```

# JSON - JavaScript Object Notation

INSTITUTO FEDERAL
São Paulo

Como acessar os valores de um ***array* associativo PHP**:

```php
<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

$arr = json_decode($jsonobj, true);

echo $arr["Peter"];
echo $arr["Ben"];
echo $arr["Joe"];
?>
```

# *JSON - JavaScript Object Notation*

INSTITUTO FEDERAL
São Paulo

Você pode ainda percorrer os valores com um *loop* ***foreach()*** usando valores de um objeto PHP:

```php
<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

$obj = json_decode($jsonobj);

foreach($obj as $key => $value) {
  echo $key . " => " . $value . "<br>";
}
?>
```

# *JSON - JavaScript Object Notation*

INSTITUTO FEDERAL
São Paulo

Você pode ainda percorrer os valores com um *loop* ***foreach()*** usando valores de um *array* associativo PHP:

```php
<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

$arr = json_decode($jsonobj, true);

foreach($arr as $key => $value) {
  echo $key . " => " . $value . "<br>";
}
?>
```

INSTITUTO FEDERAL
São Paulo

Dúvidas?

Perguntas?

Sugestões?