

Referência da Linguagem

Olá,

Bem-vindos à Aula 5 da matéria Desenvolvimento Voltado à Web I. Na aula anterior, a programação orientada a objetos foi apresentada, desde suas classes, com os atributos e métodos aos objetos instanciados, bem como os relacionamentos possíveis entre classes, a associação, a agregação, a composição e a herança.

Nesta aula, será demonstrada a utilização de funções próprias do PHP, que facilitarão a manipulação de arrays, datas, json, e-mail e erros.

Leia atentamente a esta aula e, se tiver alguma dúvida, use os recursos que estão na sua área do aluno.

➡ Bons estudos!

Objetivos de aprendizagem

Ao término desta aula, vocês serão capazes de:

- desenvolver programas utilizando funções próprias do PHP.

Seções de estudo

- 1 – PHP Array
- 2 – PHP Date
- 3 – PHP JSON

1- PHP Array

Na aula 2, foi apresentado o array, que para ser instanciado utiliza a função `array()`, e sua informação pode ser inserida diretamente a uma determinada chave desse array. Nas subseções abaixo, veremos funções do PHP que facilitam a utilização do array no PHP.

array_combine

A partir de dois arrays, um contendo as chaves e outro os valores, um terceiro array combinando chaves e valores é retornado (COMBINE, 2019).

```
<?php
$a = array('verde', 'vermelho', 'amarelo');
$b = array('abacate', 'maçã', 'banana');
$c = array_combine($a, $b);

print_r($c);

/*resultado:
Array
(
    [green] => abacate
    [red]   => maçã
    [yellow] => banana
)
*/
?>
```

array_diff

Compara um array com um ou mais arrays, retornando somente os elementos que estiverem presentes no primeiro que não estiverem presentes nos demais. Caso no primeiro array contenha elementos repetidos, são tratados como um só (DIFF, 2019).

```
<?php
$array1 = array("a" => "verde", "vermelho", "azul", "vermelho");
$array2 = array("b" => "verde", "amarelo", "vermelho");
$result = array_diff($array1, $array2);

print_r($result);

/*resultado:
Array
(
    [1] => azul

```

```
)
*/
?>
```

array_key_exists

Essa função retorna TRUE, caso a chave pesquisada seja encontrada no array ou FALSE, caso não seja encontrada. As chaves de um array são únicas (KEYEXISTS, 2019).

```
<?php
$busca_array = array("primeiro" => 1, "segundo" => 4);
if (array_key_exists("primeiro", $busca_array)) {
    echo "O elemento 'primeiro' está no array!";
}

/*resultado:
O elemento 'primeiro' está no array!
*/
?>
```

array_map

Uma função é passada por parâmetro e ela será executada para cada um dos elementos do array. Caso a função tenha mais de um parâmetro, mais de um array deve ser passado por parâmetro para a função mapear (MAP, 2019).

```
<?php
function cube($n)
{
    return($n * $n * $n);
}

$a = array(1, 2, 3, 4, 5);
$b = array_map("cube", $a);

print_r($b);

/*resultado:
Array
(
    [0] => 1
    [1] => 8
    [2] => 27
    [3] => 64
    [4] => 125
)
*/
?>
```

array_merge

Essa função realiza a fusão de um ou mais arrays. Os elementos dos arrays serão incluídos na sequência que os arrays forem passados por parâmetro na função. Caso algum elemento contenha uma chave string que também esteja em outro array, o primeiro é sobrescrito, no fim a última ocorrência da chave é o valor que estará no array resultante. As chaves com valor numérico são incluídas (com novas chaves numéricas) mesmo que haja repetição da chave (MERGE, 2019).

```
<?php
$array1 = array("cor" => "vermelho",
2, 4);
$array2 = array("a", "b", "cor" =>
"verde", "forma" => "trapezoide", 4);

$result = array_merge($array1,
$array2);

print_r($result);

/*resultado:
Array
(
    [cor] => verde
    [0] => 2
    [1] => 4
    [2] => a
    [3] => b
    [forma] => trapezoide
    [4] => 4
)
*/
?>
```

array_pop e array_push

Funções que podem ser utilizadas quando o comportamento do array deve ser semelhante a uma pilha. Enquanto o array_pop remove e retorna o último elemento do array, o array_push adiciona um ou mais elementos no fim do array (POP, 2019; PUSH, 2019).

```
<?php
$cesta = array("laranja", "morango");
array_push($cesta, "melancia",
"batata");
$fruta = array_pop($cesta);

Echo $fruta;
print_r($cesta);

/*resultado:
batata
Array
(
    [0] => laranja
    [1] => morango
    [2] => melancia
)
*/
?>
```

array_search e in_array

Muitas vezes é necessário descobrir se um determinado valor se encontra em um array. Para isso, utiliza-se as funções array_search e in_array, a primeira ao encontrar o valor no array retorna a chave em que se encontra, e a segunda retorna TRUE, se o valor existir no array, e FALSE, caso contrário (SEARCH, 2019; IN_ARRAY, 2019).

```
<?php
$array = array(0 => 'blue', 1 =>
'red', 2 => 'green', 3 => 'red');

$key = array_search('green', $array);
echo $key;
$key = array_search('red', $array);
echo $key;
$res = in_array('black', $array);
echo $res;

/*resultado:
2
1
FALSE
*/
?>
```

asort e ksort

A ordenação dos elementos de um array, tanto pelo seu valor quanto por sua chave pode simplificar muito o desenvolvimento. Para isso, usamos as funções asort (ordena pelos valores) e o ksort (ordena pelas chaves) (ASORT, 2019; KSORT, 2019).

```
<?php
$frutas = array("d" => "limao", "a"
=> "laranja", "b" => "banana", "c" =>
"melancia");

asort($frutas);
foreach( $frutas as $chave => $valor
){
    echo "$chave = $valor\n";
}
echo "-----\n";
ksort($frutas);
foreach ($fruits as $key => $val) {
    echo "$chave = $valor\n";
}

/*resultado:
b = banana
a = laranja
d = limao
c = melancia
-----
a = laranja
b = banana
c = maçã
d = limao
*/
?>
```

count

Durante a programação, em alguns momentos, não é possível saber ao certo qual será a quantidade de elementos que um array pode conter, já que a inclusão de elementos pode ser dinâmica. A função count() retorna a quantidade de elementos que podem ser encontrados no array (COUNT, 2019).

```
<?php
$frutas = array("d" => "limao", "a"
=> "laranja", "b" => "banana", "c" =>
"melancia");

$result = count($frutas);
echo $result;

/*resultado:
4
*/
?>
```

explode e implode

Essas funções manipulam dois tipos que são muito utilizados no PHP, os arrays e as Strings. A função explode consegue dividir uma string a partir de uma substring delimitadora para criar um array. Toda vez que a função encontra essa substring o conteúdo antes dela na string é colocado em uma posição do array (EXPLODE, 2019).

O implode vai na linha contrária, por parâmetro é indicado o elemento que irá concatenar cada valor encontrado no array. É utilizado para gerar uma string contendo todos os valores separados por vírgula (IMPLode, 2019).

```
<?php
$pizza = "piece1 piece2 piece3
piece4 piece5 piece6";
$pieces = explode(" ", $pizza);

print_r($pieces);
echo "-----\n";

echo implode(", ", $pieces);

/*resultado:
Array (
[0] => piece1
[1] => piece2
[2] => piece3
[3] => piece4
[4] => piece5
[5] => piece6
)
-----
piece2,piece3,piece4,piece5,piece6
*/
?>
```

2- PHP Date

Outros tipos de dados muito trabalhados no PHP são as datas e horas. As funções do PHP simplificam a manipulação das datas, já que para cada região o padrão de data e hora pode ser modificado.

date

A função date retorna uma string no padrão desejado da data e hora local ou até de um valor predefinido. Muito utilizada quando um sistema deve atender vários idiomas

(DATE, 2019)

```
<?php
echo date("d/m/y h:i:s")."\n";
echo date("d/m/Y")."\n";
echo date("Y/m/d")."\n";

/*resultado:
25/10/19 10:43:10
25/10/2019
2019/10/25
*/
?>
```

date_default_timezone_set

Como já explicado, além de cada região ter um padrão de exibição de data, cada uma pode ter também um fuso horário diferente. O fuso horário que será retornado será o que estiver setado no servidor que o PHP está instalado, mas é possível setar o fuso horário em tempo de execução utilizando a função date_default_timezone_set que modifica o padrão de todas as funções de data e hora do script (TIMEZONE, 2019).

A lista com todos os fusos horários das Américas pode ser encontrada no link: https://www.php.net/manual/pt_BR/timezones.america.php.

```
<?php
date_default_timezone_set('America/
Campo_Grande');
?>
```

diff

Existem diversas outras funções que o PHP já possui para manipular datas, que podem ser encontradas na documentação do PHP. Como ponto de partida explicaremos o funcionamento da função diff, que como o próprio nome já sugere, retorna a diferença entre dois objetos do tipo DateTime (DIFF, 2019).

Os objetos do tipo DateTime podem ser criados a partir da função DateTime() que pode receber por parâmetro a data que o objeto deverá ter. Após criados os objetos a serem comparados, o método diff é chamado para comparar uma data a outra. Após isso, utiliza-se o format() para criar um padrão de retorno para o resultado da diferença entre as datas.

```
<?php
$datetime1 = new
DateTime('2019-10-25');
$datetime2 = new
DateTime('2019-10-27');
$interval = $datetime1-
>diff($datetime2);
echo $interval->format('%R%a dias');

/*resultado:
+2 dias
*/

?>
```

3- PHP JSON

O JSON é uma notação que simplifica a troca de dados entre sistemas. Por mais que seu nome seja um acrônimo para JavaScript Object Notation, ela não se limita a ser utilizada somente no JavaScript. O PHP possui algumas funções que facilitam a manipulação do JSON (GAMA, 2011).

Gama (2011) cita ainda algumas vantagens do JSON sobre outras notações como o XML, são elas:

1. Leitura mais simples;
2. Analisador mais fácil;
3. JSON é tipado, suporta objetos;
4. Transporte e execução de dados com maior velocidade;
5. Arquivo com tamanho reduzido.

O exemplo de JSON abaixo representa vários carros que serão retornados pelo servidor.

```
{
  "carros": [
    {
      "id": 0,
      "modelo": "Celta",
      "placa": "AAA1234"
    },
    {
      "id": 1,
      "modelo": "Palio",
      "placa": "BBB1234"
    }
  ]
}
```

json_decode

Essa função decodifica uma string que foi enviada e a converte em um objeto que pode ser manipulado diretamente pelo PHP (DECODE, 2019).

A função `var_dump()` mostra informações sobre a variável que é passada por parâmetro. Uma representação estruturada com tipos e valores, indicada para visualizar o conteúdo de arrays e objetos.

```
<?php
$json
='{ "a":1,"b":2,"c":3,"d":4,"e":5}';
```

```
echo $json["a"]."/n";
var_dump(json_decode($json));
```

```
/*resultado:
1
object(stdClass)#1 (5) {
    ["a"] => int(1)
    ["b"] => int(2)
    ["c"] => int(3)
    ["d"] => int(4)
```

```
["e"] => int(5)
}
*/
?>
```

json_encode

Contrária ao `json_decode`, a função `json_encode` retorna uma string a partir de um elemento que é passado por parâmetro. Pode ser qualquer tipo do PHP, tal como o array (ENCODE, 2019).

```
<?php
$arr = array('a' => 1, 'b' => 2, 'c'
=> 3, 'd' => 4, 'e' => 5);

echo json_encode($arr);

/*resultado:
{"a":1,"b":2,"c":3,"d":4,"e":5}
*/
?>
```

Retomando a aula

Chegamos ao final da quinta aula. Vamos recordar?

1 – PHP Array

Na primeira seção, as funções predefinidas no PHP que são capazes de manipular o array foram apresentadas.

2 – PHP Date

Na segunda seção, foram detalhadas as funções que manipulam data e hora.

3 – PHP JSON

Em seguida, as funções que convertem string para JSON e o contrário foram exemplificadas.

Vale a pena

Vale a pena ler,

SOARES, W. *Programação Web com PHP 5*, Érica, 2014.
BEIGHLEY, L; MORRISON, M. *Use a Cabeça! PHP & MySQL*. Alta Books, 2010.

SILVA, J. M. C. *PHP na prática 200 Exercícios resolvidos*. Elsevier, 2014.

BENTO, E. J. *Desenvolvimento web com PHP e MySQL*. Editora Casa do Código, 2014.





Vale a pena **acessar,**

Combine. In: PHP Documentation, array_combine, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.array-combine.php. Acesso em: 25 out. 2019.

Diff. In: PHP Documentation, array_diff, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.array-diff.php. Acesso em: 25 out. 2019.

KeyExists. In: PHP Documentation, array_key_exists, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.array-key-exists.php. Acesso em: 25 out. 2019.

Map. In: PHP Documentation, array_map, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.array-map.php. Acesso em: 25 out. 2019.

Merge. In: PHP Documentation, array_merge, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.array-merge.php. Acesso em: 25 out. 2019.

Pop. In: PHP Documentation, array_pop, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.array-pop.php. Acesso em: 25 out. 2019.

Push. In: PHP Documentation, array_push, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.array-push.php. Acesso em: 25 out. 2019.

Search. In: PHP Documentation, array_search, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.array-search.php. Acesso em: 25 out. 2019.

In_array. In: PHP Documentation, in_array, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.in-array.php. Acesso em: 25 out. 2019.

Asort. In: PHP Documentation, *asort*, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.asort.php. Acesso em: 25 out. 2019.

Ksort. In: PHP Documentation, ksort, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.ksort.php. Acesso em: 25 out. 2019.

Date. In: PHP Documentation, date, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.date.php. Acesso em: 25 out. 2019.

Timezone. In: PHP Documentation, `date_default_timezone_set`, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.date-default-timezone-set.php. Acesso em: 25 out. 2019.

Diff. In: PHP Documentation, date_diff, 2019. Disponível em: https://www.php.net/manual/pt_BR/datetime.diff.php. Acesso em: 25 out. 2019.

GAMA, A. In: O que é JSON, 2011. Disponível em: <https://www.devmedia.com.br/o-que-e-json/23166>. Acesso em: 25 out. 2019.

Decode. In: PHP Documentation, json_decode, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.json-decode.php. Acesso em: 25 out. 2019.

Encode. In: PHP Documentation, json_encode, 2019. Disponível em: https://www.php.net/manual/pt_BR/function.json-encode.php. Acesso em: 25 out. 2019.



Minhas anotações