

Algumas funções úteis de Arrays em PHP

Prof. Edkallenn

- Algumas funções úteis de Arrays em PHP
 - ksort
 - is_array
 - isset
 - in_array
 - array_search
 - range
 - array_keys
 - array_values
 - count
 - array_merge
 - array_pop
 - array_push()
 - array_unique
 - compact
 - filter
 - array_map()
- Como base será utilizado o seguinte array associativo:

```
$notas = [  
    'Edkallenn' => null,  
    'John' => 8,  
    'Paul' => 9,  
    'George' => '7',  
    'Ringo' => 10  
];
```

ksort

Ordena um array pelas chaves

Uso:

```
ksort($notas);  
var_dump($notas);
```

Saída:

```
array(5) {  
    ["Edkallenn"]=>  
    NULL  
    ["George"]=>  
    string(1) "7"  
    ["John"]=>  
    int(8)  
    ["Paul"]=>  
    int(9)  
    ["Ringo"]=>  
    int(10)  
}
```

is_array

Verifica se a variável é um array

Uso:

```
if (is_array($notas)) {  
    echo 'Sim, é um array' . '<br>';  
}
```

Saída:

```
Sim, é um array
```

isset

Informa se a variável foi iniciada

Uso:

```
echo 'Edkallenn fez a prova: ' . '<br>';  
var_dump(isset($notas['Edkallenn']));
```

Saída:

```
Edkallenn fez a prova:  
bool(false)
```

in_array

Checa se um valor existe em um array

Uso:

```
echo 'Alguém tirou 10? - ';  
var_dump(in_array(10, $notas));
```

Saída:

```
Alguém tirou 10? - bool(true)
```

Sintaxe alternativa:

```
echo 'Alguém tirou 7? - ';  
var_dump(in_array(7, $notas, true));
```

Saída:

```
Alguém tirou 7? - bool(false)
```

Explicação: `strict = true` usa comparação `firme` (não-frouxa), ou seja, verifica por tipo. Sem essa cláusula o resultado seria `true`.

array_search

Procura por um valor em um array e retorna sua chave correspondente caso seja encontrado

Uso:

```
echo 'Quem tirou 10? ';  
echo array_search(10, $notas);
```

Saída:

```
Quem tirou 10? Ringo
```

range

Cria um array contendo uma faixa de elementos Sintaxe:

```
range(mixed $start, mixed $end, number $step = 1): array
```

Uso:

```
$array3 = range(1, 20);  
print_r($array3);
```

Saída:

```
Array  
(  
    [0] => 1  
    [1] => 2  
    [2] => 3  
    [3] => 4  
    [4] => 5  
    [5] => 6  
    [6] => 7  
    [7] => 8  
    [8] => 9  
    [9] => 10  
    [10] => 11  
    [11] => 12  
    [12] => 13  
    [13] => 14  
    [14] => 15  
    [15] => 16  
    [16] => 17  
    [17] => 18  
    [18] => 19  
    [19] => 20  
)
```

array_keys

Retorna todas as chaves ou uma parte das chaves de um array

Uso:

```
print_r(array_keys($notas));
```

Saída:

```
(  
    [0] => Edkallenn  
    [1] => George  
    [2] => John  
    [3] => Paul  
    [4] => Ringo  
)
```

array_values

Retorna todos os valores de um array Uso:

```
print_r(array_values($notas));
```

Saída:

```
Array  
(  
    [0] =>  
    [1] => 7  
    [2] => 8  
    [3] => 9  
    [4] => 10  
)
```

count

Conta o número de elementos de uma variável, ou propriedades de um objeto Uso:

```
echo "Tamanho do array: " . count($notas) . "<br>;
```

Saída:

```
Tamanho do array: 5
```

array_merge

Juntar Um ou Mais Arrays (Combinar): Uso:

```
$notas = [  
    'Edkallenn' => null,
```

```
'John' => 8,
'Paul' => 9,
'George' => '7',
'Ringo' => 10
];

$treinamentos = [
    'Treinamentos' => [
        'PHP',
        'JS',
        'HTML5',
        'Laravel',
    ]
];
print_r(array_merge($notas, $treinamentos));
```

Saída:

```
Array
(
    [Edkallenn] =>
    [George] => 7
    [John] => 8
    [Paul] => 9
    [Ringo] => 10
    [Treinamentos] => Array
        (
            [0] => PHP
            [1] => JS
            [2] => HTML5
            [3] => Laravel
        )
)
```

array_pop

Remover o Último Elemento do Array Uso:

```
$frutas = ['Uva', 'Abacaxi', 'Abacate', 'Laranja'];
array_pop($frutas);
print_r($frutas);
```

Saída:

```
Array
(
```

```
[0] => Uva
[1] => Abacaxi
[2] => Abacate
)
```

Para remover o primeiro elemento use `array_shift()`

array_push()

Adicionar um Novo Elemento no Final do Array Uso:

```
$frutas = ['Uva', 'Abacaxi', 'Abacate'];
$novaFruta = 'Tangerina';
array_push($frutas, $novaFruta);
print_r($frutas);
```

Saída:

```
Array
(
    [0] => Uva
    [1] => Abacaxi
    [2] => Abacate
    [3] => Tangerina
)
```

Pode-se adicionar um elemento ao final somente fazendo `$frutas[] = 'Tangerina';` É possível adicionar um novo elemento no final de um array é possível adicionar no início com a função `array_unshift()`

array_unique

Remover Elementos Repetidos de um Array Uso:

```
$cesta = ['Arroz', 'Feijão', 'Macarrão', 'Batata', 'Arroz', 'Batata'];
$cesta = array_unique($cesta);
print_r($cesta);
```

Saída:

```
Array
(
    [0] => Arroz
    [1] => Feijão
)
```

```
[2] => Macarrão  
[3] => Batata  
)
```

compact

Criar array contendo variáveis e seus valores É possível utilizar a função compact() para criar um array a partir do nome de variáveis Uso:

```
$nome = 'Edkallenn Lima';  
$email = 'ed1rac@unipe.edu.br';  
$empresa = 'Unipê';  
$cursos = [  
    'PHP',  
    'JS',  
    'HTML5',  
    'Laravel',  
];  
$info = compact('nome', 'email', 'empresa', 'cursos');  
print_r($info);
```

Saída:

```
Array  
(  
    [nome] => Edkallenn Lima  
    [email] => ed1rac@unipe.edu.br  
    [empresa] => Unipê  
    [cursos] => Array  
        (  
            [0] => PHP  
            [1] => JS  
            [2] => HTML5  
            [3] => Laravel  
        )  
)
```

filter

Filtrar elementos do array Uso:

```
$numeros = [1, 2, 726, 5, 6, 788, 900, 212, 2, 3, 1998];  
function checaPar($value)  
{  
    return $value % 2 == 0;  
}
```



```
$numeros = array_filter($numeros, 'checaPar');  
print_r($numeros);
```

Saída:

```
Array  
(  
    [1] => 2  
    [2] => 726  
    [4] => 6  
    [5] => 788  
    [6] => 900  
    [7] => 212  
    [8] => 2  
    [10] => 1998  
)
```

array_map()

Altera os elementos dos arrays passando uma função de callback Uso:

```
$carrinho = [  
    'Uva',  
    'Arroz',  
    'Maionese',  
    'Batata',  
];  
function mudaUppercase($value)  
{  
    return strtoupper($value);  
}  
$carrinho = array_map('mudaUppercase', $carrinho);  
print_r($carrinho);
```

Saída:

```
Array  
(  
    [0] => UVA  
    [1] => ARROZ  
    [2] => MAIONESE  
    [3] => BATATA  
)
```

- Essas são algumas das principais funções de manipulação de arrays em PHP. Para maiores informações (e dezenas de outras funções úteis), siga o link: https://www.php.net/manual/pt_BR/

Prof. Edkallenn