

# Funções para *arrays*

## ○ array push

Adiciona um ou mais elementos ao final de um *array*

Tem o mesmo efeito de utilizar a sintaxe: `$array[] = $valor`

### ► Exemplo

```
<?php
$a = array('verde', 'azul',
'vermelho');

array_push($a, 'amarelo');
```

```
var_dump($a);
?>
```

#### Resultado:

```
array(4) {
  [0]=> string(5) "verde"
  [1]=> string(4) "azul"
  [2]=> string(8) "vermelho"
  [3]=> string(7) "amarelo"
}
```

## ○ array pop

Retira um elemento do final de um *array*

### ► Exemplo

```
<?php
$a = array('verde', 'azul',
'vermelho');

array_pop($a);

var_dump($a);
?>
```

#### Resultado:

```
array(2) {
  [0]=> string(5) "verde"
  [1]=> string(4) "azul"
}
```



# Funções para *arrays* (continuação)

## ○ array shift

Retira um elemento do início de um *array*

### ► Exemplo

```
<?php
$a = array('verde', 'azul',
'vermelho', 'amarelo');

array_shift($a);

var_dump($a);
?>
```



**Resultado:**

```
array(3) {
  [0]=> string(4) "azul"
  [1]=> string(8) "vermelho"
  [2]=> string(7) "amarelo"
}
```



INSTITUTO FEDERAL  
GOIANO

## ○ array unshift

Adiciona um ou mais elementos do início de um *array*

### ► Exemplo

```
<?php
$a = array('verde', 'azul',
'vermelho');

array_unshift($a, 'amarelo');

var_dump($a);
?>
```



**Resultado:**

```
array(4) {
  [0]=> string(7) "amarelo"
  [1]=> string(5) "verde"
  [2]=> string(4) "azul"
  [3]=> string(8) "vermelho"
}
```

# Funções para *arrays* (continuação)

## ○ array reverse

Retorna um *array* na ordem inversa

### ► Exemplo

```
<?php
$a[0] = 'green';
$a[1] = 'yellow';
$a[2] = 'red';
$a[3] = 'blue';

$b = array_reverse($a);

var_dump($b);
?>
```



**Resultado:**

```
array(4) {
    [3]=> string(4) "blue"
    [2]=> string(3) "red"
    [1]=> string(6) "yellow"
    [0]=> string(5) "green"
}
```



GOIANO

## ○ array merge

Mescla dois ou mais *arrays*

Obs: se dois *arrays* tiverem conteúdo indexado pela mesma chave (*string*), o segundo sobrepõe o primeiro

### ► Exemplo

```
<?php
$a = array('verde', 'azul');
$b = array('vermelho',
'amarelo');
$c = array_merge($a, $b);

var_dump($c);
?>
```



**Resultado:**

```
array(4) {
    [0]=> string(5) "verde"
    [1]=> string(4) "azul"
    [2]=> string(8) "vermelho"
    [3]=> string(7) "amarelo"
}
```

# Funções para *arrays* (continuação)

- in\_array

Verifica se um valor existe em um *array*

- ▶ Exemplo

```
<?php
$a[] = 'refrigerante';
$a[] = 'cerveja';
$a[] = 'vodka';
$a[] = 'suco natural';

if(in_array('suco natural', $a)){
    echo 'suco natural encontrado';
}
?>
```



**Resultado:**

suco natural encontrado

- sort

Ordena um *array* pelos valores

- ▶ Exemplo

```
<?php
$a[] = 'refrigerante';
$a[] = 'cerveja';
$a[] = 'vodka';
$a[] = 'suco natural';

sort($a);

print_r($a);
?>
```



**Resultado:**

Array

```
(
    [0] => cerveja
    [1] => refrigerante
    [2] => suco natural
    [3] => vodka
)
```

Veja também as funções de ordenação:

- ▶ rsort

- ▶ asort

- ▶ arsort

- ▶ ksort

- ▶ krsort



# Funções para *arrays* (continuação)

## ○ count

Retorna a quantidade de elementos de um *array*

### ► Exemplo

```
<?php
$a[] = 'refrigerante';
$a[] = 'cerveja';
$a[] = 'vodka';
$a[] = 'suco natural';

echo 'o array $a contém ' . count($a) . ' posições';
?>
```



#### Resultado:

o array \$a contém 4 posições

A função `count` também pode ser usada para iterar em *arrays*

```
<?php
//...considerando que o array anterior já está no código

for($i = 0; $i < count($a); $i++){
    echo $a[$i] . '<br>';
}
?>
```



#### Resultado:

refrigerante  
cerveja  
vodka  
suco natural

# Funções para *arrays* (continuação)

## ○ explode

Converte uma *string* em um *array*, separando os elementos por meio de um separador

### ► Exemplo

```
<?php
$string = '31/12/2004';
$array = explode('/', $string);

var_dump($array);
?>
```



#### Resultado:

```
array(3) {
  [0]=>
  string(2) "31"
  [1]=>
  string(2) "12"
  [2]=>
  string(4) "2004"
}
```



## ○ implode

Converte um *array* em uma *string*, separando os elementos por meio de um separador

### ► Exemplo

```
<?php
$padrao[] = 'Maria';
$padrao[] = 'Paulo';
$padrao[] = 'José';

$resultado = implode(' + ', $padrao);

var_dump($resultado);
?>
```



#### Resultado:

```
string(20) "Maria + Paulo + José"
```

# Exercícios 28, 29 e 30

- 28) Crie uma matriz 5x5 e insira valores inteiros aleatórios (use um laço de repetição, como o FOR, e a função `rand`). Depois, faça uma função que receba uma matriz de qualquer dimensão como parâmetro e retorne o maior valor encontrado na matriz.

Salve seu arquivo como `ex28.php`

- 29) Crie dois *arrays*, cada um com 50 posições preenchidas com números inteiros aleatórios (use um laço de repetição, como o FOR, e a função `rand`). Em seguida, mescle estes dois *arrays* em um terceiro *array* e, por fim, ordene o terceiro *array* e imprima-o utilizando um laço de repetição (como o FOREACH).

Salve seu arquivo como `ex29.php`

- 30) Crie uma função que verifique se um determinado valor existe em um *array* e retorne sua chave caso exista ou uma mensagem informando que o valor não existe no *array*.

Dica: pesquise o funcionamento da função [array\\_search](#)

Salve seu arquivo como `ex30.php`