

```

<?php
/*
    array_push:
    adiciona elementos ao final de uma array. tem o mesmo efeito de utilizar
    a sintaxe $array[]=$valor

    int array_push (array nome_array, mixed valor [, mixed...])
*/
$a = array("verde", "azul", "vermelho");
array_push($a, "amarelo");
var_dump($a);

/*
    array_pop:
    remove um valor do final de um array.

    mixed array_pop(array nome_array)
*/
$a = array("verde", "azul", "vermelho", "amarelo");
array_pop($a);
var_dump($a);

/*
    array_shift:
    remove um elemento do início de um array.

    mixed array_shift(array nome_array)
*/
$a = array("verde", "azul", "vermelho", "amarelo");
array_shift($a);
var_dump($a);

/*
    array_unshift:
    Adiciona um elemento no início de um array.

    int array_unshift(array nome_array, mixed valor [,mixed])
*/
$a = array("verde", "azul", "vermelho");
array_unshift($a, "amarelo");
var_dump($a);

/*
    array_pad:
    preenche um array com um dado valor, determinada quantidade de
    posições.

    array array_pad(array nome_array int tamanho, mixed valor)
*/
$a = array("verde", "azul", "vermelho");
$a = array_pad($a, 6, "branco");
var_dump($a);

/*
    array_reverse:
    recebe um array e retorna-o na ordem inversa.

```

```

    array array_reverse(array nome_array, boolean preservar_chaves)
*/

$a[0] = 'verde';
$a[1] = 'amarelo';
$a[2] = 'vermelho';
$a[3] = 'azul';
$b = array_reverse($a, true);
var_dump($b);

/*
    array_merge:
    mescla dois ou mais arrays. Um array é adicionado ao final do outro.
    O resultado é um novo array. se ambos arrays tiverem conteúdo
    indexado pela mesma chave, o segundo irá se sobrepor ao primeiro.

    array array_merge(array nome_array, array nome_array2 [,array...])
*/

$a = array("verde", "azul");
$b = array("vermelho", "amarelo");
$c = array_merge($a, $b);

var_dump($c);

/*
    array_keys:
    Retorna as chaves de um array. Se o segundo parâmetro for indicado,
    a função retornará apenas índices que apontam para um conteúdo
    igual ao parâmetro.

    array array_keys (array nome-array [, mixed valor_procurado])
*/

$exemplo = array('cor' => 'vermelho', 'volume' => '5', 'animal' =>
'cachorro');
$indices = array_keys($exemplo);
print_r($indices);

/*
    array_values:
    Retorna um array contendo os valores de outro array

    array array_values(array nome_array)
*/

$exemplo = array('cor' => 'vermelho', 'volume' => '5', 'animal' =>
'cachorro');
$valores = array_values($exemplo);
print_r($valores);

/*
    array_slice:
    Extrai uma porção de um array.

    array array_slice(array nome_array, int inicio, int tamanho)
*/

```

```

$a[0] = 'verde';
$a[1] = 'amarelo';
$a[2] = 'vermelho';
$a[3] = 'azul';
$a[4] = 'cinza';
$a[5] = 'branco';
$b = array_slice($a, 2, 3);
print_r($b);

/*
count:
retorna a quantidade de elementos de um array.

int count(array nome_array)
*/

$a = array('refrigerante', 'cerveja', 'vodka', 'suco natural');
echo 'O array $a contém '. count($a).' posições.';

/*
in_array:
verifica se um array contém um determinado valor.

boolean in_array(mixed valor, array nome_array)
*/

$a = array('refrigerante', 'cerveja', 'vodka', 'suco natural');
if (in_array('suco natural', $a))
{
    echo 'suco natural encontrado';
}

/*
sort:
ordena um array pelo seu valor, não mantendo a associação de índices.

boolean sort(array nome_array)
*/

$a = array('refrigerante', 'cerveja', 'vodka', 'suco natural');
sort($a);
print_r($a);

/*
rsort:
Ordena um array em ordem reversa pelo seu valor, não mantendo a
associação de índices.

boolean rsort(array nome_array)
*/

$a = array('refrigerante', 'cerveja', 'vodka', 'suco natural');
rsort($a);
print_r($a);

/*

```

asort:
Ordena um array pelo seu valor, mantendo a associação de índices. Para ordenar de forma reversa, use `arsort()`.

```
void asort(array nome_array)
*/
```

```
$a[0] = 'verde';
$a[1] = 'amarelo';
$a[2] = 'vermelho';
$a[3] = 'azul';
$a[4] = 'cinza';
$a[5] = 'branco';
$a = asort($a);
print_r($a);
```

```
/*
ksort:
Ordena um array pelos seus índices. Para ordem reversa, utilize
krsort().
*/
```

```
$carro['potencia'] = '1.0';
$carro['cor'] = 'branco';
$carro['modelo'] = 'celta';
$carro['opcionais'] = 'ar condicionado';
ksort($carro);
print_r($carro);
```

```
/*
explode:
converte uma string em um array, separando os elementos por meio de
um separador.
```

```
array explode(string separado, string padrão)
*/
```

```
$string = "31/12/2004";
var_dump(explode("/", $string));
```

```
/*
implode:
Converte um array em uma string, separando os elementos do array por
meio de um separador.
```

```
array implode(string separador, array padrão)
*/
```

```
$padrao = array('Maria', 'Paulo', 'José');
$resultado = implode(' + ', $padrao);
var_dump($resultado);
```

?>