

# ClickBlocks\Utils\Arrays

## General information

Inheritance	no
Subclasses	no
Interfaces	no
Source	Framework/utils/arrays.php

**Arrays** is an utility class that contains a few static methods simplifying the frequently used operations with arrays.

## Public static methods

### swap()

```
public static void swap(array &$amp;array, integer $n1, integer $n2)
```

<b>\$array</b>	array	the array in which the two elements will be swapped.
<b>\$n1</b>	integer	the index number of the first element.
<b>\$n2</b>	integer	the index number of the second element.

Swaps the two elements of the array. The elements are determined by their index numbers.

### aswap()

```
public static void aswap(array &$amp;array, mixed $key1, mixed $key2)
```

<b>\$array</b>	array	the array in which the two elements will be swapped.
<b>\$key1</b>	mixed	the key of the first element.
<b>\$key2</b>	mixed	the key of the second element.

Swaps the two elements of the array. The elements are determined by their keys.

### insert()

```
public static void insert(array &$amp;array, mixed $value, integer $offset = 0)
```

<b>\$array</b>	array	the input array in which a value will be inserted.
<b>\$value</b>	mixed	the inserting value.
<b>\$offset</b>	integer	the position in the first array.

Inserts a value or an array to the input array at the specified position. Example:

```
$a = ['a' => 1, 'b' => 2, 'c' => 3]; // the input array.
Arrays::insert($a, ['d' => 4], 1); // $a is ['a' => 1, 'd' => 4, 'b' => 2, 'c' => 3]
Arrays::insert($a, [1, 2, 3], 4); // $a is ['a' => 1, 'd' => 4, 'b' => 2, 'c' => 3, 0 => 1, 1 => 2, 2 => 3]
```

# unsetByKeys()

```
public static void unsetByKeys(array &$array, array $keys)
```

<b>\$array</b>	array	the array from which an element will be removed.
<b>\$keys</b>	array	the element keys.

Completely removes the element of a multidimensional array, defined by its keys. Example:

```
// the input array.
$a = [ 'k1' => [ 'k11' => 'a', 'k12' => 'b', 'k13' => 'c'],
      'k2' => [ 'k21' => [ 'k211' => 1], 'k22' => 2],
      'k3' => 3];

Arrays::unsetByKeys($a, [ 'k1', 'k12' ]);
print_r($a);
Arrays::unsetByKeys($a, [ 'k2', 'k21' ]);
print_r($a);
Arrays::unsetByKeys($a, [ 'k1' ]);
print_r($a);
```

The above example will output:

```
Array
(
    [k1] => Array
        (
            [k11] => a
            [k13] => c
        )

    [k2] => Array
        (
            [k21] => Array
                (
                    [k211] => 1
                )

            [k22] => 2
        )

    [k3] => 3
)
Array
(
    [k1] => Array
        (
            [k11] => a
            [k13] => c
        )

    [k2] => Array
        (
            [k22] => 2
        )

    [k3] => 3
)
Array
(
    [k2] => Array
        (
```

[k22] => 2

)

[k3] => 3

)