

ԳՈՐԾՈՂՈՒԹՅՈՒՆՆԵՐ ԶԱՆԳՎԱԾՆԵՐԻ ՀԵՏ

Մաս 2

Array_merge ֆունկցիան

Array_merge () ֆունկցիայի կանչը մի քանի զանգվածների միավորելու ամենահեշտ ձևն է: Բոլոր տվյալները կմիացվեն միասին, կրկնվող տվյալները կվերագրվեն:

```
$array1 = ['a' => 'a', 'b' => 'b', 'c' => 'c'];  
$array2 = ['a' => 'A', 'b' => 'B', 'D' => 'D'];  
  
$merge = array_merge($array1, $array2);  
print_r($merge);  
  
// Array  
// (  
//   [a] => A  
//   [b] => B  
//   [c] => c  
//   [D] => D  
// )
```

array_diff և array_intersect ֆունկցիաները

Մեկ զանգվածի արժեքները մյուս զանգվածից հեռացնելու համար օգտագործեք array_diff () ֆունկցիան: Հետադարձ գործողությունը իրականացվում է array_intersect () ֆունկցիայի միջոցով:

```
$array1 = [1, 2, 3, 4];  
$array2 = [3, 4, 5, 6];  
  
$diff = array_diff($array1, $array2);
```

```

print_r($diff);

// [0 => 1, 1 => 2]

$intersect = array_intersect($array1, $array2);
print_r($intersect);

// [2 => 3, 3 => 4]

```

Array_sum և array_product ֆունկցիաները

Array_sum () ֆունկցիայի օգնությամբ կարելի է հաշվել զանգվածի տարրերի գումարը, array_product()-ի օգնությամբ՝ բազմապատկել բոլոր արժեքները:

```

$numbers = [1, 2, 3, 4, 5];

echo(array_sum($numbers));    // 15
echo(array_product($numbers)); // 120

```

Array_count_values ֆունկցիան

Array_count_values () ֆունկցիան կօգնի հաշվարկել զանգվածի բոլոր եզակի արժեքների քանակը:

```

$things = ['apple', 'apple', 'banana', 'tree', 'tree', 'tree'];
$values = array_count_values($things);

print_r($values);

// Array
// (
//   [apple] => 2
//   [banana] => 1
//   [tree] => 3
// )

```

array_fill ֆունկցիան

Array_fill ֆունկցիան օգտագործվում է նույն արժեքով որոշակի քանակությամբ տարրերից զանգված ձևավորելու համար:

```
$bind = array_fill(0, 5, '?');  
print_r($bind);  
  
// ['?', '?', '?', '?', '?']
```

Range ֆունկցիան

Տրված միջակայքին պատկանող արժեքներից զանգված ստանալու համար կարելի է օգտագործել Range ֆունկցիան:

```
$letters = range('a', 'z');  
print_r($letters);  
  
// ['a', 'b', ..., 'z']
```

```
$hours = range(0, 23);  
print_r($hours);  
  
// [0, 1, 2, ..., 23]
```

Եթե մեզ անհրաժեշտ է, որ թվերը պահպանվեն ոչ անընդմեջ, ապա range-ում պետք է ավելացնել երրորդ պարամետրը.

```
$odd = range(1, 52, 2);
```

Վերը նշված կոդը թույլ կտա ստեղծել զանգված, որը պարունակում է 1-ից 52 բոլոր կենտ թվերը:

Զույգ թվերի համար ամեն ինչ նույնն է, փոխվում է միայն առաջին պարամետրի արժեքը.

```
$even = range(2, 52, 2);
```

Որոնել տարր

Չանգվածում տարրերի առկայությունը ստուգելու համար օգտագործում են հետևյալ ֆունկցիաները.

- `in_array()` - եթե տարրը հայտնաբերվել է, վերադարձնում է `true`, հակառակ դեպքում՝ `false`:
- `array_search()` - եթե տարրը հայտնաբերվել է, վերադարձնում է իր բանալին, հակառակ դեպքում՝ `false`:

```
<html>
<head>
    <title> Որոնել տարրը զանգվածում </title>
</head>
<body>
    <?php
        $a = array("առաջին" => 6, "երկրորդ" => 2, "երրորդ" => 1);
        if (in_array (2, $a)) echo "2 գտնվեց <br>";
        echo "գտնված տարրի բանալին - ".array_search(2, $a);
    ?>
</body>
</html>
```

`array_shift`, `array_pop`, `array_pop`, `array_push` և `array_unshift` ֆունկցիաները

`array_shift ()` - առաջին տարրի հեռացում:

`array_pop ()` - վերջին տարրի հեռացում:

`array_push ()` - ավելացնել տարրեր զանգվածի վերջում:

`array_unshift ()` - ավելացնել տարրեր զանգվածի սկզբում:

```
<html>
<head>
    <title> Ավելացնել տարրեր զանգվածի սկզբում </title>
</head>
<body>
    <?php
```

```
$a = array(1, 2, 3);
echo "Չանգված \${a}:";
foreach ($a as $tmp) echo " $tmp";
$k = array_unshift($a, 4, 5);
echo "<br> Չանգված \${a} ավելացնելուց հետո:";
foreach ($a as $tmp) echo " $tmp";
echo "<br>Երկարությունը = \${k}";
?>
</body>
</html>
```

Արդյունքում կստանանք.

Չանգված \${a}: 1 2 3

Չանգված \${a} ավելացնելուց հետո: 4 5 1 2 3

Երկարությունը = 5