

## array sort – เรียงลำดับอาร์เรย์มีติเตียว

คำสั่งนี้ใช้เรียงลำดับค่าในอาร์เรย์มีติเตียว

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
$fruits = array("lemon", "orange", "ba-
nana", "apple");
sort($fruits);
foreach ($fruits as $key => $val) {
echo "fruits[" . $key . "] = " . $val . "<br>";
}
?>
```

ผลลัพธ์



## array shuffle – สลับที่ของสมาชิกในอาร์เรย์โดยการสุ่ม

คำสั่งนี้ใช้ สลับที่ของสมาชิกในอาร์เรย์ โดยการสุ่ม

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
$numbers = range(1, 20);
shuffle($numbers);
foreach ($numbers as $number) {
echo "$number ";
}
?>
```

ผลลัพธ์



## array range – สร้างอาร์เรย์ขึ้นตามเงื่อนไขที่กำหนด

คำสั่งนี้ใช้สำหรับ สร้างอาร์เรย์ขึ้นตามเงื่อนไขที่กำหนด โดยพารามิเตอร์ดังนี้

- 1.ค่าเริ่มต้น
- 2.ค่าสิ้นสุด
- 3.จำนวนขั้นในการเพิ่ม

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
foreach (range(0, 12) as $number) {
    echo $number;
}
echo "<br>";
// การเพิ่ม ทีละขั้น
foreach (range(0, 100, 10) as $number) {
    echo $number;
}
echo "<br>";
// การใช้ตัวอักษร
foreach (range('a', 'i') as $letter) {
    echo $letter;
}
?>
```

ผลลัพธ์



## array nat sort เรียงลำดับค่าอาร์เรย์โดยใช้ natural order algorithm

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
$array1 = $array2 = array("img12.png",
    "img10.png", "img2.png", "img1.
    png");
asort($array1);
echo " sort แบบธรรมดา";
var_dump($array1);
natsort($array2);
echo "Natural order sortingn";
var_dump($array2);
?>
```

บลัฟท์



## php array key – ใช้ดึงค่า key ออกจากอาเรย์

คำสั่งนี้ใช้ดึงค่า key ออกจากอาเรย์โดยใช้งานร่วมกับคำสั่งที่ใช้จัดการ pointer เช่น current ,next

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
$array = array(
    'fruit1' => 'apple',
    'fruit2' => 'orange',
    'fruit3' => 'grape',
    'fruit4' => 'apple',
    'fruit5' => 'apple');
while ($fruit_name = current($array)) {
    if ($fruit_name == 'apple') {
        echo key($array).<br />';
    }
    next($array);
}
?>
```

บลัฟท์



## array\_search – ค้นหาคีย์ที่ต้องการ

คือ ฟังก์ชันที่ใช้ค้นหาคีย์ที่เราต้องการ

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
$array = array(0 => 'blue', 1 => 'red', 2 =>
'green', 3 => 'red');
echo array_search('green', $array);
?>
```

ผลลัพธ์



## array\_reverse – พลิกลำดับของอาร์เรย์

ฟังก์ชันนี้ใช้เรียงลำดับของอาร์เรย์ใหม่

ตัวอย่าง

```
<?php
$input = array("php", "green", "red");
$reversed = array_reverse($input);
$preserved = array_reverse($input, true);
var_dump($reversed);
?>
```

ผลลัพธ์



## array\_replace – เปลี่ยนค่าในอาเรย์โดยระบุตำแหน่ง

ฟังก์ชันนี้ใช้สำหรับเปลี่ยนค่าในอาเรย์โดยระบุตำแหน่ง

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
$base = array("orange", "banana", "apple", "raspberry");
$replacements = array(0 => "pineapple", 4 => "cherry");
$basket = array_replace($base, $replacements);
var_dump($basket);
?>
```

ผลลัพธ์



## array\_diff\_ukey – เปรียบเทียบค่า key ของอาเรย์โดยใช้ฟังก์ชันเข้ามาจัดการ

ฟังก์ชันนี้ใช้เปรียบเทียบค่า key ของอาเรย์โดยใช้ฟังก์ชันเข้ามาจัดการ

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
function key_compare_func($key1, $key2)
{
    if ($key1 == $key2)
        return 0;
    else if ($key1 > $key2)
        return 1;
    else
        return -1;
}
$array1 = array('blue' => 1, 'red' => 2, 'green' => 3, 'purple' => 4);
$array2 = array('green' => 5, 'blue' => 6, 'yellow' => 7, 'cyan' => 8);
var_dump(array_diff_ukey($array1, $array2, 'key_compare_func'));
?>
```

ผลลัพธ์

```
array (size=2)
    'red' => int 2
    'purple' => int 4
```

## array\_reduce – ลดจำนวนสมาชิกในอาเรย์ให้เหลือตัวเดียว โดยใช้ฟังก์ชันเข้ามาจัดการ

ฟังก์ชันนี้ลดจำนวนสมาชิกในอาเรย์ให้เหลือตัวเดียว โดยใช้ฟังก์ชันเข้ามาจัดการ  
ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
function sum($carry, $item)
{
    $carry += $item;
    return $carry;
}
$a = array(1, 2, 3, 4, 5);
var_dump(array_reduce($a, "sum")); // int(15)
?>
```

ผลลัพธ์



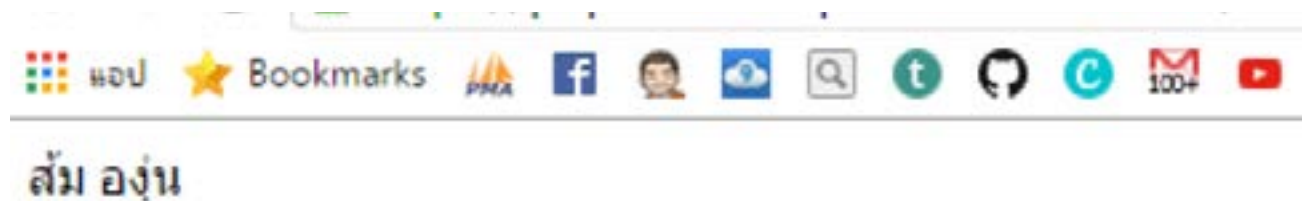
## array\_rand – สุ่มค่าจากอาเรย์

ฟังก์ชันนี้ใช้สุ่มค่าจากอาเรย์  
ตัวอย่างที่ 1 การใช้งานเบื้องต้น

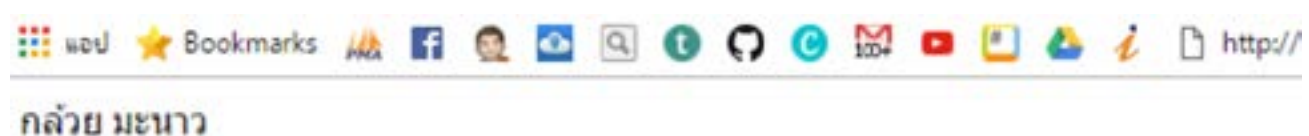
```
<?php
$input = array("กล้วย", "ส้ม", "มะนาว", "องุ่น", "แตงโม");
$rand_keys = array_rand($input, 2);
echo $input[$rand_keys[0]] . "n";
echo $input[$rand_keys[1]] . "n";
?>
```

ผลลัพธ์

การสุ่มครั้งที่ 1



การสุ่มครั้งที่ 2



## php array\_push – เพิ่มค่าลงในอาเรย์โดยเพิ่มต่อจากท้าย

ฟังก์ชันนี้ใช้เพิ่มค่าลงในอาเรย์โดย เพิ่ม ต่อ จากท้าย

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
$stack = array("orange", "banana");
array_push($stack, "apple", "raspberry");
var_dump($stack);
?>
```

ผลลัพธ์



## array\_product – ใช้หาผลรวมของค่าทั้งหมดในอาเรย์

ฟังก์ชันนี้ใช้หาผลรวมของค่าทั้งหมดในอาเรย์

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
$a = array(2, 4, 6);
echo "product(a) = " . array_product($a) . "\n";
?>
```

ผลลัพธ์



## array\_pop – ดึงค่าท้ายสุดของอาเรย์ออกมา

ฟังก์ชันนี้ใช้ดึงค่าท้ายสุดของอาเรย์ออกมาแล้วลบค่านั้นออกจากอาเรย์ที่ดึงมาด้วย

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
$stack = array("orange", "banana", "apple", "raspberry");
$fruit = array_pop($stack);
echo 'ค่าที่ดึงออกมาคือ ' . $fruit . '<br>';
echo 'ค่าที่เหลือคือ';
var_dump($stack);
?>
```

ผลลัพธ์



ค่าที่ตั้งออกมาคือ raspberry  
ค่าที่เหลือคือ

```
array (size=3)
  0 => string 'orange' (length=6)
  1 => string 'banana' (length=6)
  2 => string 'apple' (length=5)
```

## array\_pad – ขยายขนาดอาร์เรย์และเพิ่มค่าเข้าไปใหม่

ฟังก์ชันนี้ใช้ขยายขนาดของอาร์เรย์และเพิ่มค่าตามที่กำหนด

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
$input = array(12, 10, 9);
$result = array_pad($input, 5, 0);
var_dump($result);
?>
```

ผลลัพธ์

array (size=5)  
0 => int 12  
1 => int 10  
2 => int 9  
3 => int 0  
4 => int 0

## array\_multisort – เรียงลำดับค่าในอาร์เรย์ด้วย เงื่อนไขหลายๆ ตัว

คำสั่งนี้ช่วยเรียงลำดับข้อมูลในอาร์เรย์โดยใช้หลายๆ เงื่อนไขและหลายแบบ

พารามิเตอร์ที่ใช้กับคำสั่ง

1.array ที่ต้องการเรียงลำดับ

2.รูปแบบการเรียงลำดับ

SORT\_ASC – น้อยไปมาก

SORT\_DESC – มากไปน้อย

3.ชนิดของการเรียง

SORT\_NUMERIC – เรียงลำดับตามตัวเลข

SORT\_STRING – เรียงลำดับตามตัวอักษร

SORT\_LOCALE\_STRING – เปรียบเทียบเป็นตัวอักษรตามภาษาที่ใช้อยู่โดยการตั้งค่าภาษาจะใช้ฟังก์ชัน

setlocale()

SORT\_NATURAL – เปรียบเทียบเป็นตัวอักษรโดยใช้ฟังก์ชัน natsort

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
$a=array("จ","บ","ค","ก","ร");
array_multisort($a);
print_r($a);
?>
```



ผลลัพธ์



Array ( [0] => ข [1] => ค [2] => จ [3] => ก [4] => ร )

ตัวอย่างที่ 2 การใช้ SORT\_ASC

```
<?php
$a=array("จ","ข","ค","ก","ร");
array_multisort($a,SORT_ASC);
print_r($a);
?>
```

ผลลัพธ์



Array ( [0] => ข [1] => ค [2] => จ [3] => ก [4] => ร )

ตัวอย่างที่ 3 การใช้ SORT\_DESC

```
<?php
$a=array("จ","ข","ค","ก","ร");
array_multisort($a,SORT_DESC);
print_r($a);
?>
```

ผลลัพธ์



Array ( [0] => ร [1] => ก [2] => จ [3] => ค [4] => ข )

## php array\_merge\_recursive – รวมอาเรย์ หลายตัว โดยรวม key ที่เหมือนกันด้วย

ต่างจาก array\_merge คือจะรวมค่าจาก key ที่มีชื่อเหมือนกันด้วย โดย array\_merge นั้น เมื่อใช้กับตัวอย่างเดียวกัน ตัวอย่างที่ 1 เปรียบเทียบกับ array\_merge

```
<?php
$ar1 = array("color" => array("favorite" => "red"), 5);
$ar2 = array(10, "color" => array("favorite" => "green"));
$result = array_merge($ar1, $ar2);
var_dump($result);
?>
```

ผลลัพธ์



```
array (size=3)
  'color' =>
    array (size=1)
      'favorite' => string 'green' (length=5)
  0 => int 5
  1 => int 10
```

เมื่อใช้ array\_merge\_recursive จะเห็นว่าการรวมค่าจากอาเรย์ตัวที่สอง ที่มี key เหมือนกันเข้าไปด้วย



```
array (size=3)
  'color' =>
    array (size=1)
      'favorite' =>
        array (size=2)
          0 => string 'red' (length=3)
          1 => string 'green' (length=5)
  0 => int 5
  1 => int 10
```

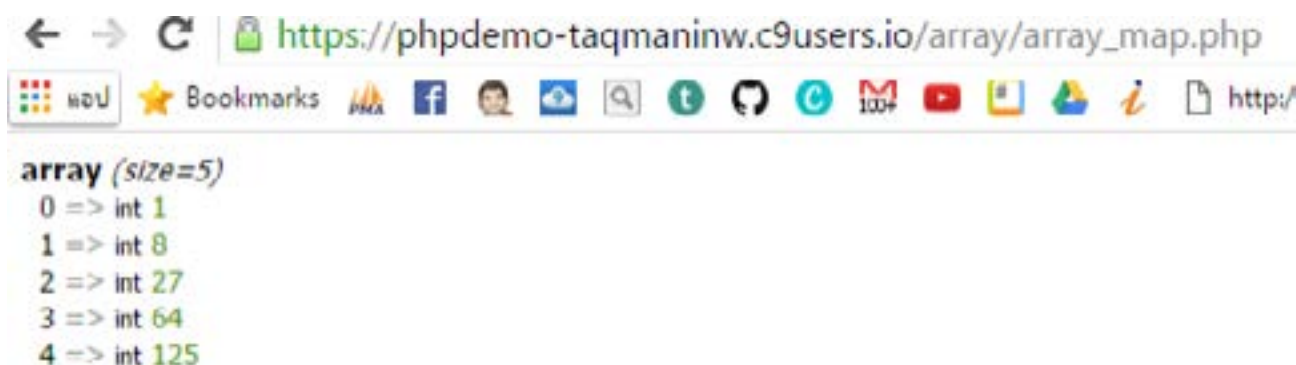
## array\_map – จัดการค่าทั้งหมดในอาเรย์โดยใช้ฟังก์ชันเพิ่มเติม

คำสั่งนี้ใช้แปลงค่าในอาเรย์ทั้งหมดโดยใช้ฟังก์ชัน callback เข้ามาจัดการ

ตัวอย่างที่ 1 การใช้งานเบื้องต้น

```
<?php
function cube($n)
{
    return($n $n $n);
}
$a = array(1, 2, 3, 4, 5);
$b = array_map("cube", $a);
print_r($b);
?>
```

ผลลัพธ์ที่ได้จะเห็นว่าค่าในอาเรย์ใหม่ถูกแปลงด้วยฟังก์ชัน a



```
array (size=5)
  0 => int 1
  1 => int 8
  2 => int 27
  3 => int 64
  4 => int 125
```

## php array merge – รวมอาเรย์หลายตัว

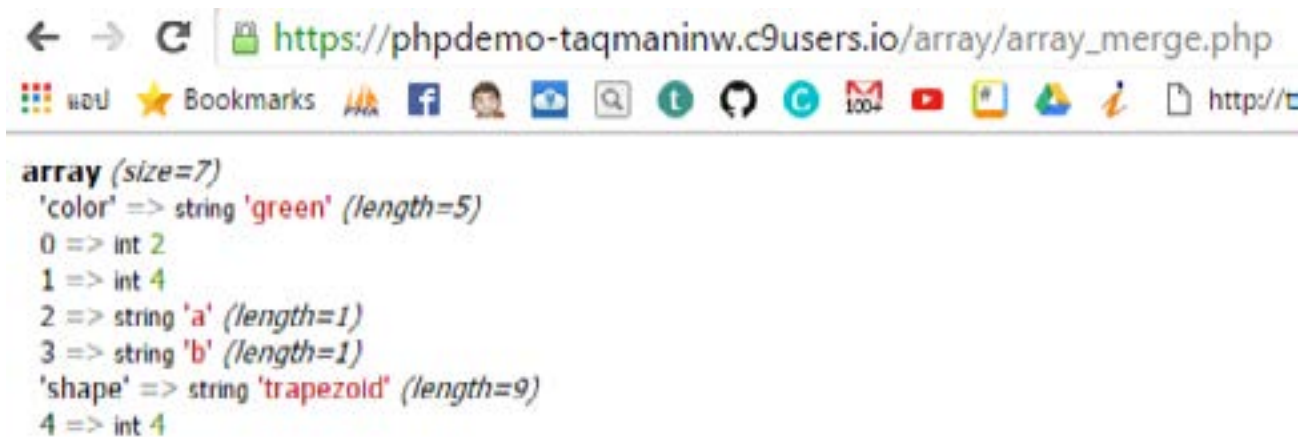
array merge คือฟังก์ชันใช้รวมอาเรย์หลายๆ ตัวเข้าด้วยกัน

ตัวอย่าง

```
<?php
$array2 = array("a", "b", "color" => "green", "shape" => "trapezoid",
4);
$result = array_merge($array1, $array2);
print_r($result);
?>
```

ผลลัพธ์

จะเห็นว่าการสร้าง key ให้อัตโนมัติ



← → ↺ [https://phpdemo-taqmaninw.c9users.io/array/array\\_merge.php](https://phpdemo-taqmaninw.c9users.io/array/array_merge.php)

แอป Bookmarks 🔥 f 👤 ☁ 🔍 t 🔄 100% 📺 📄 http://t

```
array (size=7)
  'color' => string 'green' (length=5)
  0 => int 2
  1 => int 4
  2 => string 'a' (length=1)
  3 => string 'b' (length=1)
  'shape' => string 'trapezoid' (length=9)
  4 => int 4
```