Họ và tên: Đinh Cống Phong\_Php Basics

Bài 4 cái đầu.

1. **array\_unique**

**Cú pháp:**

array **array\_unique** ( array $array [, int $sort\_flags = SORT\_STRING ] )

**Mô tả:**

Lấy một mảng đầu vào và trả về một mảng mới mà không có giá trị bản sao.

Chú ý rằng các từ khóa thì được tồn tại. Array\_unique () sẽ sắp xếp các giá trị điều trị như chuỗi lúc đầu, sau đó sẽ giữ từ khóa đầu tiên gặp phải cho mỗi giá trị, và bỏ qua tất cả các từ khóa sau đây. Nó không có nghĩa rằng từ khóa của các giá trị liên quan đầu tiên của mảng được phân loại sẽ được lưu giữ.

**Tham số:**

* Các mảng đầu vào.
* Sort\_flags(Cờ loại).

Các sort\_flags tham số tùy chọn thứ hai có thể được sử dụng để sửa đổi hành vi phân loại sử dụng các giá trị.

Phân loại kiểu cờ:

* SORT\_REGULAR - so sánh các mặt hàng bình thường (không thay đổi các loại)
* SORT\_NUMERIC - so sánh số lượng các mặt hang
* SORT\_STRING - so sánh các mặt hàng như các chuỗi
* SORT\_LOCALE\_STRING - so sánh các mặt hàng như các chuỗi, dựa trên ngôn ngữ hiện tại.

**Giá trị trả về:**

Trả về array filtered(mảng lọc).

**Chú ý**: Array\_unique () không có mục đích làm việc trên mảng đa chiều.

**Ví dụ:**

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<?php

$input = array(4, "4", "3", 4, 3, "3");

$result = array\_unique($input);

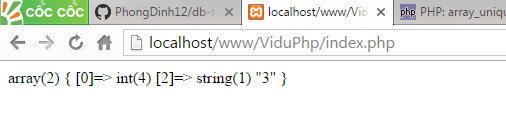
var\_dump($result);

?>

</body>

</html>

**KQ:**



1. **array\_unshift**

**Cú pháp:**

int **array\_unshift** ( array &$array , [mixed](http://php.net/manual/en/language.pseudo-types.php#language.types.mixed) $value1 [, [mixed](http://php.net/manual/en/language.pseudo-types.php#language.types.mixed) $... ] )

**Mô tả:**

Array\_unshift () thêm vào trước thông qua các yếu tố vào phía trước của mảng. Lưu ý rằng danh sách các yếu tố được thêm vào phía trước như một tổng thể, để các yếu tố được thêm vào trước ở trong cùng một trật tự. Tất cả các khóa mảng số sẽ được sửa đổi để bắt đầu đếm từ con số không trong khi các khóa bình thường sẽ không được thay đổi.

**Tham số:**

Các mảng đầu vào và giá trị đầu tiên để thêm vào trước.

**Giá trị trả về:**

Trả về số mới của các yếu tố trong mảng.

**Ví Dụ:**

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<?php

$queue = array("orange", "banana");

array\_unshift($queue, "apple", "raspberry");

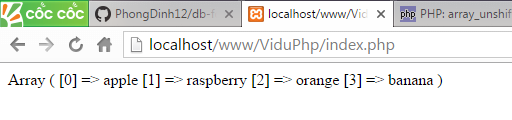
print\_r($queue);

?>

</body>

</html>

**KQ**



1. **array\_values**

**Cú pháp:**

array **array\_values** ( array $array )

**Mô tả:**

Array\_values () trả về tất cả các giá trị từ mảng và chỉ số mảng số.

**Tham số:**

Các mảng.

**Giá trị trả về:**

Trả về một mảng đánh chỉ mục của các giá trị.

**Ví Dụ:**

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<?php

$array = array("size" => "XL", "color" => "gold");

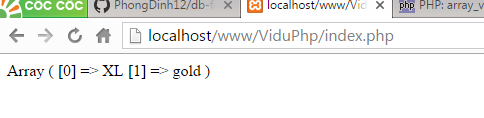
print\_r(array\_values($array));

?>

</body>

</html>

**KQ:**



1. **array\_** **walk\_recursive**

**Cú pháp:**

bool **array\_walk\_recursive** ( array &$array , [callable](http://php.net/manual/en/language.types.callable.php) $callback [, [mixed](http://php.net/manual/en/language.pseudo-types.php#language.types.mixed) $userdata = NULL ] )

**Mô tả:**

Áp dụng các hàm callback(gọi lại) người dùng định nghĩa để mỗi phần tử của mảng. Chức năng này sẽ được đệ quy vào mảng sâu hơn.

**Tham số:**

**Array:**

Các mảng đầu vào.

**Callback:**

Thông thường, gọi lại có trên hai tham số. Giá trị các tham số của mảng là vị trí đầu tiên, và các khóa / mục thứ hai.

Chú thích: Nếu gọi lại cần được làm việc với các giá trị thực tế của mảng, chỉ định các tham số đầu tiên của cuộc gọi lại như một tài liệu tham khảo. Sau đó, bất kỳ thay đổi nào trong những yếu tố sẽ được thực hiện trong các mảng ban đầu của chính nó.

**Usedata:**

Nếu tham số userdata tùy chọn được cung cấp, nó sẽ được thông qua như là tham số thứ ba để gọi lại.

**Giá trị trả về:**

Trả về TRUE nếu thành công hoặc FALSE nếu thất bại.

**Ví Dụ:**

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<?php

$sweet = array('a' => 'apple', 'b' => 'banana');

$fruits = array('sweet' => $sweet, 'sour' => 'lemon');

function test\_print($item, $key)

{

echo "$key holds $item\n<br>";

}

array\_walk\_recursive($fruits, 'test\_print');

?>

</body>

</html>

**KQ:**

