LAB 4: MẢNG TRONG PHP

1. Mục tiêu

- Nắm vững cấu trúc mảng trong php.
- Sử dụng thành thạo các thao tác trên mảng: tạo mảng, thêm - xóa phần tử, truy xuất dữ liệu trong mảng, duyệt mảng,…
- Sử dụng một số hàm thông dụng về mảng trong PHP

2. Tóm tắt lý thuyết

- Mảng là biến chứa nhiều phần tử. Trong php, các phần tử có thể có kiểu dữ liệu khác nhau. Mỗi phần tử của mảng có 2 thành phần: index và value. Index của mảng có thể là các số nguyên hay chuỗi. Value của phần tử mảng có thể là giá trị của bất cứ kiểu dữ liệu nào, thậm chí là một mảng (mảng nhiều chiều). Một phần tử value trong mảng có thể có kiểu dữ liệu khác với các phần tử khác.
- Tạo mảng: sử dụng hàm array().

```
$a = array();//mảng $a rỗng
$b = array(1, 3, 5); // Mảng $b có 3 phần tử
```

Mảng $b có 3 phần tử có index và value lần lượt là: 0,1, 2 và 1, 3, 5. Với mảng không xác định index cho các phần tử như thế này, php sẽ tạo mảng có index là các số nguyên và bắt đầu từ index là 0.

Tức là: : $b[0] = 1;
: $b[1] = 3;
: $b[2] = 5;

```
$c = array("x1"=>2, "x2 "=>4, "x3 "=>6);
```

Mảng $c có 3 phần tử có index và value lần lượt là: x1, x2, x3 và 2, 4, 6

Tức là : $c["x1"] = 2;
: $c["x2"] = 4;
: $c["x4"] = 6;

- Để debug xem cấu trúc và nội dung một mảng, ta sử dụng hàm print_r hay var_dump.

```
print_r($b);        var_dump($c);
```

- Mảng nhiều chiều: Là mảng mà giá trị của phần tử mảng là một mảng khác.

Ví dụ:

```
<?php
$a = array(1, 3, 5);          $b = array("x1"=>2, "x2"=>4);
$c = array($a, $b);           $d = array();  $d[] = $a; $d[] = $b;
```

```
print_r($c);
print_r($d);
$v1 = $d[1]["x2"];                  //$v1=4
$v2 = $c[0]; $v3 = $v2[2];          // $v2 = 5
?>
```

Kết quả:

```
Array
(
   [0] => Array
      (
         [0] => 1
         [1] => 3
         [2] => 5
      )

   [1] => Array
      (
         [x1] => 2
         [x2] => 4
      )

)
Array
(
   [0] => Array
      (
         [0] => 1
         [1] => 3
         [2] => 5
      )

   [1] => Array
      (
         [x1] => 2
         [x2] => 4
      )

)
```

- Các thao tác trên mảng:
    - Truy xuất tới một phần tử: sử dụng toán tử []. Ví dụ: $v = $b[1]
    - Để kiểm tra một index có trong mảng hay không, ta sử dụng hàm isset. Ví dụ:

```
<?php
if (isset($c["x1"]) $c["x1"] +=2;
?>
```

- Để thêm phần tử mới: sử dụng lệnh gán dữ liệu cho phần tử này trong mảng.
  - $b[3] = 5; //Tạo phần tử $b[3] (nếu chưa có) và gán giá trị bằng 5. Nếu $b[3] đã có, thay đổi nội dung của $b[3] bằng 5.
  - Hoặc $b[] = 6; Php tự động tạo phần tử $b[4] và gán giá trị là 6 nếu mảng $b mới chỉ có các phần tử như trên ($b[0], $b[1], $b[2], $b[3]) và chưa có phần tử $b[4].
- Để xóa phần tử trong mảng: sử dụng hàm unset(). Unset($b[2]);
- Duyệt qua mảng: có thể sử dụng vòng lặp for hay foreach.
  i. Vòng lặp for: sử dụng cho các mảng có các chỉ số nguyên liên tục (ví dụ mảng $b ở trên sau khi chưa xóa phần tử).
  ```
  for($i=0; $i< Count($b); $i++)
        echo $b[$i];
  ```
  ii. Vòng lặp foreach: Được sử dụng duyệt qua tất cả các loại mảng: mảng có index là số nguyên liên tục hay không liên tục, hoặc index là chuỗi. Mỗi lần lặp, vòng lặp sẽ duyệt qua một phần tử của mảng và gán dữ liệu (index hay value) cho các biến ta đưa vào.
  - ✓
  ```
  Foreach($c as $value)
  {
        //xử lý biến chứa giá trị của mảng là $value
  }
  ```
  - ✓
  ```
  Foreach($c as $key => $value)
  {
        /* xử lý biến chứa index và value của phần tử đang duyệt qua là $key và $value */
  }
  ```

- Các hàm hay sử dụng trên mảng:
  - Count: trả về số phần tử của mảng. $n = Count($b);
  - In_array(): kiểm tra một giá trị có trong mảng hay không.

```
<?php
$ext = "jpg"; $arr = array("jpg", "bmp", "gif");
If (in_array($ext, $arr)) echo “Đây là phần mở rộng của file hình”;
Else echo “Không phải”;
?>
```

  - Is_array(): Kiểm tra một biến có phải là một biến mảng hay không.
  - Array_rand(): Lấy ngẫu nhiên một số phần tử trong mảng.
    mixed array_rand (array $array [, int $num = 1]).
    $array: mảng được lấy giá trị. $num: số phần tử cần lấy.

Nếu $num=1: kết quả trả về chỉ số của phần từ cần lấy.
Nếu $num>1: Trả về mảng các chỉ số cần lấy.

```
<?php
$input = array("Neo", "Morpheus", "Trinity", "Cypher", "Tank");
$rand_keys = array_rand($input, 2);
echo $input[$rand_keys[0]] . "\n";
echo $input[$rand_keys[1]] . "\n";
?>
```

- sort, asort: sắp xếp mảng.
  - bool sort (array &$array [, int $sort_flags = SORT_REGULAR]): sắp xếp lại một mảng theo giá trị tang dần.
    - $array: mảng cần sắp xếp
    - $sort_flags: Cách so sánh khi sắp xếp: có các giá trị hay sử dụng sau:
      SORT_REGULAR – So sánh không thay đổi kiểu dữ liệu
      SORT_NUMERIC – So sánh theo dạng số
      SORT_STRING – So sánh các phần tử theo dạng chuỗi
  - bool asort (array &$array [, int $sort_flags = SORT_REGULAR]): Sắp xếp các giá trị của mảng và giữ nguyên index. Hàm trả về true/false nếu thành công hay không.

```
<?php
$fruits = array("d" => "lemon", "a" => "orange", "b" => "banana", "c" => "apple");
asort($fruits);
foreach ($fruits as $key => $val) {
   echo "$key = $val\n";
}
?>
```

Kết quả:

```
c = apple
b = banana
d = lemon
a = orange
```

- Array_sum(): tính tổng các giá trị số của các phần tử trong mảng
  number array_sum (array $array);

```
<?php
$a = array(2, 4, 6, 8);
echo "sum(a) = " . array_sum($a) . "\n";

$b = array("a" => 1.2, "b" => 2.3, "c" => 3.4);
echo "sum(b) = " . array_sum($b) . "\n";
?>
```

Kết quả:

```
sum(a) = 20
sum(b) = 6.9
```

3. Ví dụ

3.1 Ví dụ lab4_1.php: Tạo mảng, xem nội dung của mảng bằng các hàm print_r và var_dump, thêm một phần tử, xóa và kiểm tra một phần tử có tồn tại hay không.

3.2 Ví dụ lab4_2.php . Duyệt qua mảng một chiều sử dụng vòng lặp foreach.

3.3 Ví dụ lab4_3.php Sử dụng một số hàm trên mảng một chiều:

a. array_rand : lấy một số phần tử ngẫu nhiên trong mảng
b. sort, asort: sắp xếp mảng
c. array_sum: tính tổng các giá trị trong mảng

3.4 Ví dụ lab4_4.php: Làm việc trên mảng nhiều chiều.

a. Tạo mảng
b. Truy xuất phần tử trong mảng
c. Duyệt mảng

4. Vận dụng

4.1 Sửa lại lab4_1.php, kiểm tra xem một phần tử nào đó có trong mảng không, nếu có, hãy xóa hoặc thay đổi dữ liệu của phần tử này và sử dụng hàm print_r() in mảng sau khi thay đổi ra màn hình.

4.2 Từ lab4_2.php, hãy sửa code lại để:

a. Đếm số phần tử có giá trị dương của mảng $a.
b. Tạo mảng mới, lưu các phần tử dương trong mảng $b. Ví dụ, mảng $c được tạo thành từ mảng $b ban đầu có giá trị như sau:
$c = array("a"=>2, "b"=>4)

4.3 Trong ví dụ lab4_3.php: hãy tìm và sử dụng các hàm để sắp xếp mảng theo chiều giảm dần.

5. Nâng cao

5.1 Xây dựng một hàm php in ra mảng một chiều và hiển thị trên trình duyệt web theo dạng bảng HTML. Tạo mảng và gọi hàm vừa tạo.

```
function showArray($arr)
{
//code in bảng html từ mảng một chiều $arr
}
```

Index	Value
...	....
...	...

5.2 Viết hàm duyệt và in ra danh sách mảng 2 chiều $arr sau ra dạng bảng HTML.

```
$arr= array();
$r = array("id"=> "sp1", "name "=> "Sản phẩm 1 ");
$arr[] = $r;
$r = array("id"=> "sp2", "name "=> "Sản phẩm 2 ");
$arr[] = $r;
$r = array("id"=> "sp3", "name "=> "Sản phẩm 3 ");
$arr[] = $r;
```

Stt	Mã Sản Phẩm	Tên Sản Phẩm
1	Sp1	Sản phẩm 1
2	Sp2	Sản phẩm 2
3	Sp3	Sản phẩm 3

5.3 Xây dựng một mảng chứa danh sách $n câu hỏi trắc nghiệm. Hãy viết trang để in ra một đề thi lấy ngẫu nhiên $m câu hỏi ($m<$n) trong danh sách câu hỏi từ mảng câu hỏi đã cho ban đầu.