

BÀI 3

MẢNG - ARRAY

Giáo viên: Vũ Đình Thắng

I. ĐỊNH NGHĨA

- Mảng là 1 tập hợp các giá trị có cùng một tên biến duy nhất
- Trong PHP. Các phần tử mảng có thể cùng 1 kiểu dữ liệu hoặc khác
- Các phần tử mảng được đặt liên tiếp trong các ô nhớ

Ví dụ:

```
$cars = array("Volvo", "BMW", "Toyota");
```

- Các kiểu mảng PHP
- Trong PHP, có ba loại mảng:
- Mảng có chỉ mục - Mảng có chỉ mục số
- Mảng kết hợp - Mảng có khóa được đặt tên
- Mảng đa chiều - Mảng chứa một hoặc nhiều mảng

II. CÁC MỤC TRONG MẢNG

- Các mục trong mảng có thể có bất kỳ kiểu dữ liệu nào.
- Phổ biến nhất là chuỗi và số (int, float), nhưng các mục mảng cũng có thể là đối tượng, hàm hoặc thậm chí là mảng.
- Bạn có thể có nhiều kiểu dữ liệu khác nhau trong cùng một mảng.

Ví dụ

- Mảng các mục có bốn kiểu dữ liệu khác nhau:
- `<?php`
- `function ham() {`
- `echo “Thang 2”;`
- `}`
- `// create array:`
- `$myArr = array(“tháng1”, 30, [“Tao”, “Chuoi”], ham);`
- `// calling the function from the array item:`
- `$myArr[3]();`
- `?>`

Kết quả
Thang 2

III. MẢNG CÓ CHỈ SỐ

1. Mảng được lập chỉ mục PHP

- Trong mảng có chỉ mục, mỗi mục có một số chỉ mục.
- Theo mặc định, mục đầu tiên có chỉ mục 0, mục thứ hai có chỉ mục 1, v.v.

Ví dụ:

```
<?php
$Sinhvien = array("Tran Van A", "Nguyen Thi B", "Hoang Thi C");
var_dump($Sinhvien);
?>
```

Kết quả

```
array(3) {
  [0]=>
  string(10) "Tran Van A"
  [1]=>
  string(12) "Nguyen Thi B"
  [2]=>
  string(11) "Hoang Thi C"
}
```

2. Truy cập mảng được lập chỉ mục

- Để truy cập vào một mục trong mảng, bạn có thể tham chiếu đến số chỉ mục.
- Các Chỉ mục được đánh số thứ tự từ 0 đến n-1

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo $cars[0];
?>
```

```
</body>
</html>
```

Phần tử đầu tiên của mảng

III. MẢNG CÓ CHỈ SỐ

3. Thay đổi giá trị

- Để thay đổi giá trị của một mục trong mảng, hãy sử dụng số chỉ mục:

Ví dụ: Thay đổi giá trị của mục thứ hai:

```
<?php

```

```
array(3) {
    [0]=>
    string(7) "Thang 1"
    [1]=>
    string(8) "Tháng 2"
    [2]=>
    string(7) "Thang 3"
}
```

4. Lặp qua một mảng có chỉ mục

Để lặp qua và in tất cả các giá trị của một mảng được lập chỉ mục, sử dụng foreach vòng lặp như thế này:

```
<?php

```

Thang 1, Tháng 2, Thang 3,

III. MẢNG KẾT HỢP

1. Mảng kết hợp: là mảng sử dụng các khóa được đặt tên mà bạn gán cho chúng.

Ví dụ:

```
<?php
$car = array("brand"=>"Ford", "model"=>"Mustang", "year"=>1964);
var_dump($car);
?>
```

Kết quả:

```
array(3) {
    ["brand"]=>
    string(4) "Ford"
    ["model"]=>
    string(7) "Mustang"
    ["year"]=>
    int(1964)
}
```

III. MẢNG KẾT HỢP

2. Truy cập mảng kết hợp

- Để truy cập vào một mục trong mảng, bạn có thể tham chiếu đến tên khóa.
- Ví dụ: Để hiển thị mẫu xe

```
<?php
$car = array("brand"=>"Ford", "model"=>"Mustang", "year"=>1964);
echo $car["model"];
?>
```

Mustang

3. Thay đổi giá trị

- Để thay đổi giá trị của một mục trong mảng, hãy sử dụng tên khóa:

Ví dụ

Thay đổi mục year:

```
<?php
$car = array("brand"=>"Ford", "model"=>"Mustang", "year"=>1964);
$car["year"] = 2024;
var_dump($car);
?>
```

III. MẢNG KẾT HỢP

4. Lặp qua một mảng kết hợp

Để lặp qua và in tất cả các giá trị của một mảng kết hợp, dùng vòng lặp foreach:

```
<?php
$car = array("brand"=>"Ford", "model"=>"Mustang", "year"=>1964);
foreach ($car as $x => $y) {
    echo "$x: $y <br>";
}
?>
```

brand: Ford

model: Mustang

year: 1964

IV. TẠO MẢNG

Sử dụng hàm **array()**;

1. Cú pháp 1:

`$Tên_biến_mảng=array(“Phần tử 1”, “Phần tử 2”,..., “Phần tử n”);`

Ví dụ:

```
$thang = array("Thang 1", "Thang 3", "Thang 3");
```

2. Cú pháp 2.

Sử dụng cú pháp ngắn hơn bằng cách sử dụng dấu ngoặc []:

```
$thang = ["Thang 1", "Thang 3", "Thang 3"];
```

Lưu ý: Có thể viết khai báo trên nhiều dòng

```
$thang = [  
    "Thang 1",  
    "Thang 3",  
    "Thang 3"  
];
```

IV. TẠO MẢNG

3. Khóa mảng-Array Keys

Khi tạo mảng có chỉ mục, các khóa sẽ được cung cấp tự động, bắt đầu từ 0 và tăng thêm 1 cho mỗi mục, do đó mảng ở trên cũng có thể được tạo bằng các khóa:

```
<?php
$cars = [
    0 => "Volvo",
    1 => "BMW",
    2 => "Toyota"
];

var_dump($cars);
?>
```



```
array(3) {
    [0]=>
        string(5) "Volvo"
    [1]=>
        string(3) "BMW"
    [2]=>
        string(6) "Toyota"
}
```

IV. TẠO MẢNG

4. Khai báo mảng rỗng

- Bạn có thể khai báo một mảng rỗng trước rồi thêm các mục vào sau:

Ví dụ:

```
$cars = [];  
$cars[0] = "Volvo";  
$cars[1] = "BMW";  
$cars[2] = "Toyota";
```

Tương tự với mảng kết hợp, trước tiên ta có thể khai báo mảng, sau đó thêm các mục vào đó:

```
$myCar = [];  
$myCar["brand"] = "Ford";  
$myCar["model"] = "Mustang";  
$myCar["year"] = 1964;
```

IV. TẠO MẢNG

5. Trộn các khóa

- Có thể có các mảng có cả khóa được lập chỉ mục và được đặt tên: Ví dụ:

```
$myArr = [];  
$myArr[0] = "apples";  
$myArr[1] = "bananas";  
$myArr["fruit"] = "cherries";
```

V. TRUY CẬP PHẦN TỬ MẢNG

1. Mảng có phần tử là hàm

```
function myFunction() {  
    echo "I come from a function!";  
}  
  
$myArr = array("car" => "Volvo", "age" => 15, "message" => myFunction);  
  
$myArr["message"]();
```

2. Lặp qua một mảng kết hợp

Để lặp qua và in tất cả các giá trị của một mảng kết hợp, sử dụng vòng lặp foreach

Ví dụ: Hiển thị tất cả các mục, khóa và giá trị của mảng:

```
$car = array("brand"=>"Ford", "model"=>"Mustang", "year"=>1964);
```

```
foreach ($car as $x => $y) {  
    echo "$x: $y <br>";  
}
```

V. TRUY CẬP PHẦN TỬ MẢNG

3. Lặp qua một mảng có chỉ mục

Để lặp qua và in tất cả các giá trị của một mảng được lập chỉ mục, sử dụng **foreach**

Ví dụ

Hiển thị tất cả các mục trong mảng:

```
$cars = array("Volvo", "BMW", "Toyota");  
foreach ($cars as $x) {  
    echo "$x <br>";  
}
```

V. TRUY CẬP PHẦN TỬ MẢNG

3. Lặp qua một mảng có chỉ mục

Để lặp qua và in tất cả các giá trị của một mảng được lập chỉ mục, sử dụng **foreach**

Ví dụ

Hiển thị tất cả các mục trong mảng:

```
$cars = array("Volvo", "BMW", "Toyota");  
foreach ($cars as $x) {  
    echo "$x <br>";  
}
```

VI. SẮP XẾP MẢNG

1. **Các hàm sắp xếp**
2. **sort()**- sắp xếp mảng theo thứ tự tăng dần
3. **rsort()**- sắp xếp các mảng theo thứ tự giảm dần
4. **asort()**- sắp xếp các mảng liên kết theo thứ tự tăng dần, theo giá trị
5. **ksort()**- sắp xếp các mảng liên kết theo thứ tự tăng dần, theo khóa
6. **arsort()**- sắp xếp các mảng liên kết theo thứ tự giảm dần, theo giá trị
7. **krsort()**- sắp xếp các mảng liên kết theo thứ tự giảm dần, theo khóa

VI. SẮP XẾP MẢNG

1. Các hàm sắp xếp

- a. **sort()**- sắp xếp mảng theo thứ tự tăng dần
- b. **rsort()**- sắp xếp các mảng theo thứ tự giảm dần
- c. **asort()**- sắp xếp các mảng liên kết theo thứ tự tăng dần, theo giá trị
- d. **krsort()**- sắp xếp các mảng liên kết theo thứ tự tăng dần, theo khóa
- e. **arsort()**- sắp xếp các mảng liên kết theo thứ tự giảm dần, theo giá trị
- f. **krsort()**- sắp xếp các mảng liên kết theo thứ tự giảm dần, theo khóa

VI. SẮP XẾP MẢNG

2. Sắp xếp mảng theo thứ tự tăng dần - sort()

Ví dụ sau đây sắp xếp các phần tử của \$carsmảng theo thứ tự bảng chữ cái tăng dần:

```
$cars = array("Volvo", "BMW", "Toyota");  
sort($cars);
```

Ví dụ sau đây sắp xếp các phần tử của \$numbersmảng theo thứ tự số tăng dần:

```
$numbers = array(4, 6, 2, 22, 11);  
sort($numbers);
```

3. Sắp xếp mảng theo thứ tự giảm dần - rsort()

Ví dụ

```
$cars = array("Volvo", "BMW", "Toyota");  
rsort($cars);
```

VI. SẮP XẾP MẢNG

4. Sắp xếp mảng (Thứ tự tăng dần), theo giá trị - asort()

- Ví dụ sau đây sắp xếp một mảng liên kết theo thứ tự tăng dần, theo giá trị:

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
asort($age);

foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

```
Key=Peter, Value=35
Key=Ben, Value=37
Key=Joe, Value=43
```

VI. SẮP XẾP MẢNG

5. Sắp xếp mảng (Thứ tự tăng dần), theo khóa - ksort()

- Ví dụ sau đây sắp xếp một mảng liên kết theo thứ tự tăng dần, theo khóa:

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
ksort($age);
```

```
foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

Key=Ben, Value=37

Key=Joe, Value=43

Key=Peter, Value=35

VI. SẮP XẾP MẢNG

6. Sắp xếp mảng (Thứ tự giảm dần), theo giá trị - arsort()

- Ví dụ sau đây sắp xếp một mảng liên kết theo thứ tự giảm dần, theo giá trị:

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
arsort($age);

foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

Key=Joe, Value=43

Key=Ben, Value=37

Key=Peter, Value=35

VI. SẮP XẾP MẢNG

7. Sắp xếp mảng (Thứ tự giảm dần), theo khóa - krsort()

- Ví dụ sau đây sắp xếp một mảng liên kết theo thứ tự giảm dần, theo khóa:

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
krsort($age);

foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

Key=Peter, Value=35

Key=Joe, Value=43

Key=Ben, Value=37

VII. THÊM CÁC PHẦN TỬ VÀO MẢNG

1. Thêm phần tử vào mảng

Để thêm các mục vào mảng hiện có, bạn có thể sử dụng [] cú pháp ngoặc vuông.

```
$fruits = array("Apple", "Banana", "Cherry");
```

```
$fruits[] = "Orange";
```

2. Thêm phần tử vào mảng kết hợp

Để thêm các mục vào mảng liên kết hoặc mảng khóa/giá trị, dùng dấu ngoặc [] cho khóa và gán giá trị bằng = toán tử.

```
$cars = array("brand" => "Ford", "model" => "Mustang");
```

```
$cars["color"] = "Red";
```

VII. THÊM CÁC PHẦN TỬ VÀO MẢNG

3. Thêm nhiều phần tử

Để thêm nhiều mục vào một mảng hiện có, hãy sử dụng hàm **array_push()**.

```
$fruits = array("Apple", "Banana", "Cherry");  
array_push($fruits, "Orange", "Kiwi", "Lemon");
```

4. Thêm nhiều mục vào mảng kết hợp

Để thêm nhiều mục vào một mảng hiện có, bạn có thể sử dụng toán tử +=.

Thêm hai mục vào `cars` mảng:

```
$cars = array("brand" => "Ford", "model" => "Mustang");  
$cars += ["color" => "red", "year" => 1964];
```


VIII. XÓA CÁC PHẦN TỬ VÀO MẢNG

1. Xóa phần tử

Để xóa một mục hiện có khỏi mảng, bạn có thể sử dụng `array_splice()` hàm.

Với hàm **`array_splice()`** này, bạn chỉ định chỉ mục (nơi bắt đầu) và số lượng mục bạn muốn xóa.

Ví dụ:

Xóa mục thứ hai:

```
$cars = array("Volvo", "BMW", "Toyota");  
array_splice($cars, 1, 1);
```

VIII. XÓA CÁC PHẦN TỬ VÀO MẢNG

2. Sử dụng hàm unset

Sử dụng hàm unset() để xóa các mục mảng hiện có.

Lưu ý: Hàm unset() không sắp xếp lại các chỉ mục, nghĩa là sau khi xóa, mảng sẽ không còn chứa các chỉ mục bị mất.

```
$cars = array("Volvo", "BMW", "Toyota");  
unset($cars[1]);
```

Hàm unset() có số lượng đối số không giới hạn và do đó có thể được sử dụng để xóa nhiều mục trong mảng:

Ví dụ. Xóa mục đầu tiên và mục thứ hai:

```
$cars = array("Volvo", "BMW", "Toyota");  
unset($cars[0], $cars[1]);
```

VIII. XÓA CÁC PHẦN TỬ VÀO MẢNG

3. Xóa nhiều phần tử mảng

Để xóa nhiều mục, `array_splice()` hàm này sẽ sử dụng tham số độ dài cho phép bạn chỉ định số lượng mục cần xóa.

- Ví dụ. Xóa 2 mục, bắt đầu từ mục thứ 2 (chỉ mục 1):

```
$cars = array("Volvo", "BMW", "Toyota");  
array_splice($cars, 1, 2);
```

VIII. XÓA CÁC PHẦN TỬ VÀO MẢNG

4. Xóa mục khỏi mảng kết hợp

Để xóa các mục khỏi mảng kết hợp, bạn có thể sử dụng hàm `unset()`.

Chỉ định khóa của mục bạn muốn xóa.

Ví dụ

Xóa "ô hình":

```
$cars = array('brand' => 'Ford', 'model' => 'Mustang', 'year' => 1964);  
unset($cars['model']);
```

VIII. XÓA CÁC PHẦN TỬ VÀO MẢNG

5. Sử dụng hàm `array_diff`

Bạn cũng có thể sử dụng `array_diff()` hàm này để xóa các mục khỏi mảng kết hợp. Hàm này trả về một mảng mới, không có các mục được chỉ định.

Ví dụ

Tạo một mảng mới không có "Mustang" và "1964":

```
$cars = array('brand' => 'Ford', 'model' => 'Mustang', 'year' => 1964);  
$newarray = array_diff($cars, ['Mustang', 1964]);
```

Lưu ý: Hàm `array_diff()` lấy giá trị làm tham số chứ không phải khóa .

VIII. XÓA CÁC PHẦN TỬ MẢNG

6. Xóa mục cuối cùng

Hàm **array_pop()** xóa phần tử cuối cùng của mảng.

Ví dụ. Xóa mục cuối cùng:

```
$cars = array("Volvo", "BMW", "Toyota");  
array_pop($cars);
```

7. Xóa mục đầu tiên

Hàm **array_shift()** xóa phần tử đầu tiên của mảng.

Ví dụ: Xóa mục đầu tiên:

```
$cars = array("Volvo", "BMW", "Toyota");  
array_shift($cars);
```