Mảng trong PHP

Hùng Nguyễn

Mảng trong PHP

1. Khái niệm về mảng
2. Khai báo và sử dụng mảng
3. Sắp xếp mảng
4. Các hàm xử lý mảng

Khái niệm về mảng

- Mảng
  - Là một loại biến đặc biệt
  - Bao gồm một dãy các ô nhớ có nhiều ô nhớ con cho phép biểu diễn thông tin dạng danh sách trong thực tế
- Các phần tử trong mảng có thể có kiểu dữ liệu khác nhau
- Ví dụ:
  - 8 số nguyên => mảng có 8 phần tử
  - Danh sách học sinh => mảng thông tin các học sinh

Khai báo và sử dụng mảng

- Mảng một chiều
  - Khai báo và khởi tạo
  - Truy xuất phần tử trong mảng
  - Thao tác trên mảng một chiều
- Mảng hai chiều
  - Khai báo và khởi tạo
  - Truy xuất phần tử trong mảng
  - Thao tác trên mảng hai chiều

Mảng một chiều

- Khai báo và khởi tạo
  - Dùng array() để khai báo và khởi tạo
  - Khai báo 1 mảng rỗng
    - Cú pháp: $tên_mảng = array();
    - Ví dụ:

```php
<?php
    $mang = array();
?>
```

Mảng một chiều

- Khai báo và khởi tạo
  - Cú pháp:

    $tên_mảng = array([khóa=>] giá trị, ...,);

    Các phần tử trong mảng cách nhau bằng dấu ,

    Trong đó:
    - khóa: số nguyên dương / chuỗi
    - Nếu khóa là chuỗi: dùng cặp nháy đôi "giá trị của khóa" hoặc cặp nháy đơn 'giá trị của khóa'
    - Mặc dịnh, khóa tự động phát sinh, với phần tử đầu tiên của mảng khóa có giá trị là 0, phần tử thứ hai của mảng khóa có giá trị là 1, …

Mảng một chiều

- Khai báo và khởi tạo
  - Chú ý:
    - Không tạo hai khóa có giá trị trùng nhau
    - Có thể dùng tất cả các kiểu dữ liệu cho giá trị của khóa

  Ví dụ:

```php
<?php
    $mang_1 = array(1,2,3,4,5);
    $mang_2 = array(1=> "Một", 2=> "Hai", 3=> "Ba", 4=> "Bốn", 5=> "Năm");
    $mang_3 = array("mot"=>1, "hai"=>2, "ba"=>3, "bon"=>4, "nam"=>5);
?>
```

Mảng một chiều

- Truy xuất phần tử trong mảng
  - Cú pháp: $tên_mảng[<khóa>]
  - Ví dụ:

```php
<?php
    $mang1 = array(1, 5, 7);
    $gia_tri_1 = $mang1[0]; //→ 1

    $mang2 = array(1=>"Một", 2=>"Hai")
    $gia_tri_3 = $mang[1]; //→ "Một
    $mang2[3] = "Ba"; //Gán giá trị
?>
```

Mảng một chiều

- Thao tác trên mảng một chiều
  - Đếm số phần tử trong mảng
  - Duyệt mảng
  - Tạo mảng từ chuỗi
  - Xuất mảng

Mảng một chiều

- Thao tác trên mảng một chiều
  - Đếm số phần tử trong mảng: count()
    - Kết quả trả về của hàm là số phần tử có trong mảng
    - Cú pháp: $số_phần_tử = count($tên_mảng);
    - Ví dụ:

```php
<?php
    $mang_1 = array(1,2,3,4,5,6);
    $so_phan_tu = count($mang_1); → 6
?>
```

Mảng một chiều

- Thao tác trên mảng một chiều
  - Duyệt mảng
    - Duyệt mảng có khóa tự động
    - Duyệt mảng có khóa do người dùng tạo

Mảng một chiều

- Thao tác trên mảng một chiều
  - Duyệt mảng có khóa tự động
    - Dùng vòng lặp for để duyệt mảng
    - Cú pháp:

```php
<?php
    $mang_1 = array(1,2,3,4,5,6);
    $n=count($mang_1);
    for($i=0;$i<$n;$i++)
        echo "\t" .$mang_1[$i];
    //→  1  2  3  4  5  6
?>
```

Mảng một chiều

- Thao tác trên mảng một chiều
  - Duyệt mảng có khóa do người dùng tạo
    - Dùng vòng lặp foreach để duyệt mảng – Duyệt để lấy giá trị của các phần tử trong mảng
    - Cú pháp:

```php
<?php
    $mang_2 = array(1=> "Một", 2=> "Hai", 3=> "Ba", 4=> "Bốn", 5=> "Năm");
    foreach ($mang_2 as $gia_tri)
        {
            echo "\t $gia_tri";
        }
    //→ Một Hai Ba Bốn Năm
?>
```

Mảng một chiều

- Thao tác trên mảng một chiều
  - Tạo mảng từ chuỗi: dùng hàm explode

```php
<?php
    $chuoi = '1,6,3,12,8,2';
    //tạo mảng
    $mang = explode(',' , $chuoi);
?>
```

Mảng một chiều

- Thao tác trên mảng một chiều
  - Xuất mảng: dùng vòng lặp hoặc dùng hàm implode
    - Dùng vòng lặp:

```php
<?php
    $mang = array(1, 6, 3, 12, 8, 2);

    //tạo chuỗi kết quả
    foreach($mang as $pt)
        $kq .= "$pt,";

    //bỏ dấu , đầu tiên
    if($kq)   $kq = substr($kq,1);
    echo $kq;

    $mang = array(1, 6, 3, 12, 8, 2);
    //tạo chuỗi kết quả
    $kq = implode(',' , $mang);
    echo $kq;
?>
```

Mảng hai chiều

- Khai báo và khởi tạo
  - Mảng 2 chiều là mảng mà mỗi phần tử là mảng 1 chiều
  - Khai báo mảng chưa biết số phần tử và giá trị
    - Cú pháp: $tên_mảng = array(array(),array());
    - Ví dụ:

```php
<?php
    $mang_1 = array(array(1,2), array(3,4), array(5,6));
    $mang_2 = array(array(1=> "Một", 2=> "Hai"), array( 3=> "Ba", 4=> "Bốn"), array(5=> "Năm", 6=>"Sáu"));
    $mang_3 = [[1,2,3],['a','b','c'],[1,2,'3']];
    echo '<pre>',print_r($mang_3),'</pre>';
?>
```

Mảng hai chiều

- Thao tác trên mảng hai chiều
  - Duyệt từng phần tử trên mỗi dòng
    - Ví dụ: xuất giá trị của các phần tử trong mảng 1

```php
<?php
    $so_dong=count($mang_1);
    for($i=0;$i<$so_dong;$i++)
    {
        echo "<br>Dòng $i: ";
        foreach($mang_1[$i] as $gia_tri)
            echo $gia_tri;
    }
    //  Dòng 0: 1 2
    //  Dòng 1: 3 4
    //  Dòng 2: 5 6 7
?>
```

Mảng hai chiều

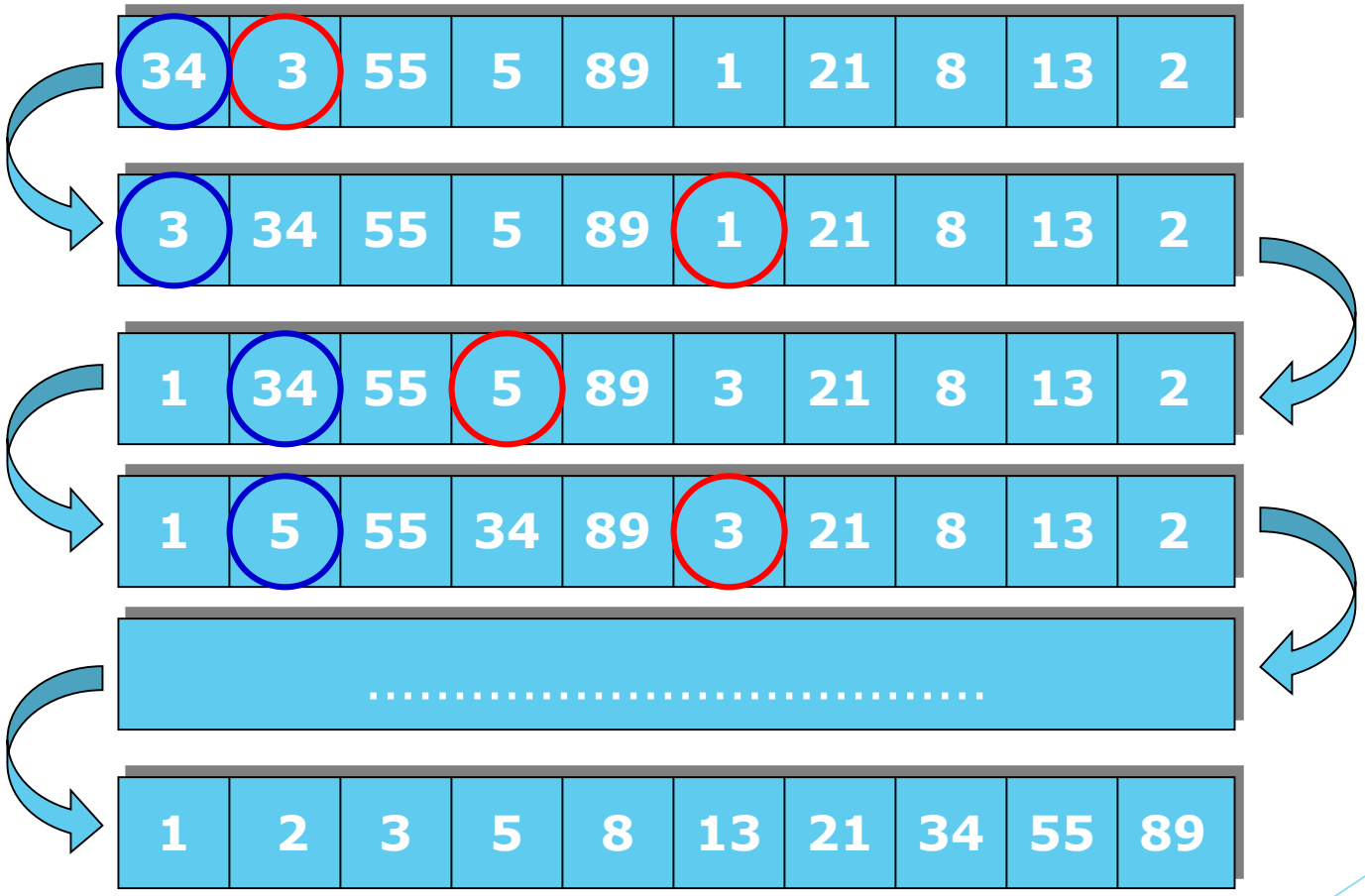
- Thao tác trên mảng hai chiều
  - Duyệt từng phần tử trên mỗi dòng
    - Ví dụ: xuất khóa và giá trị của các phần tử trong mảng 2

```php
<?php   $so_dong=count($mang_2);
    for($i=0;$i<$so_dong;$i++)
    {   echo "<br>Dòng $i: ";
        foreach ($mang_2[$i] as $key=>$gia_tri)
            echo " $key=>$gia_tri ";

    }
    //  Dòng 0: 1=>Một 2=>Hai
    //  Dòng 1: 3=>Ba 4=>Bốn
    //  Dòng 2: 5=>Năm 6=>Sáu 7=>Bảy
?>
```

Sắp xếp mảng một chiều

- Sắp tăng dần: đưa các phần tử nhỏ nhất về đầu



Sắp xếp mảng một chiều

- Sắp xếp mảng có khóa tự động
  - Để thực hiện việc sắp xếp mảng trên cần xây dựng hai hàm hoán vị hai giá trị và sắp xếp mảng
  - Hàm hoán vị

```php
<?php
    function hoan_vi(&$a, &$b)
    {
        $temp = $a;
        $a = $b;
        $b = $temp;
    }
?>
```

Sắp xếp mảng một chiều

- Sắp xếp mảng có khóa tự động
  - Hàm sắp xếp

```php
<?php
function sap_xep_mang_tang(&$tên_mảng)
{   $số_phần_tử = count($tên_mảng)
    for($i=0;$i<$số_phần_tử - 1; $i++)
    {   for($j = $i+1; $j<$số_phần_tử; $j++)
        {
        // sắp tăng dần
        if($tên_mảng[$i] > $tên_mảng[$j])
            hoan_vi($tên_mảng[$i],$tên_mảng[$j]);
        }
    }
}
?>
```

Sắp xếp mảng một chiều

- Sắp xếp mảng có khóa tự động
  - Ví dụ: sắp xếp mảng tăng dần và giảm dần

```php
<?php
    $mang = array(8,4,1,6,5,3,7,2);
    sap_xep_mang_tang(&$mang);
    echo implode(',', $mang_tang);
    //→ 1 2 3 4 5 6 7 8
    sap_xep_mang_giam($mang);
    echo implode(',', $mang_giam);
    //→ 8 7 6 5 4 3 2 1
?>
```

Sắp xếp mảng một chiều

- Sắp tăng dần: sort($tên_mảng)

```php
<?php
    $mang = array(8,4,1,6,5,3,7,2);
    sort($mang);
    echo implode(',', $mang);
    //→ 1 2 3 4 5 6 7 8
?>
```

- Sắp giảm dần: rsort($tên_mảng)

```php
<?php
    $mang = array(8,4,1,6,5,3,7,2);
    rsort($mang);
    echo implode(',', $mang);
    //→ 8 7 6 5 4 3 2 1
?>
```

Sắp xếp mảng một chiều

- Sắp tăng dần theo key: ksort($tên_mảng)
  - Ví dụ:

```php
<?php
$mang = array('1' => 'Luân Nguyễn', 'a' => 'Hùng Nguyễn', 'aa' => 'Linh Nguyễn', '2' => 'Minh Trần', '3' => 'Hạnh Trần');
ksort($mang);
echo '<pre>',print_r($mang),'</pre>';
?>
```

  - Kết quả:

```
Array
(
    [a] => Hùng Nguyễn
    [aa] => Linh Nguyễn
    [1] => Luân Nguyễn
    [2] => Minh Trần
    [3] => Hạnh Trần
)
1
```

Sắp xếp mảng một chiều

- Sắp giảm dần theo key: krsort($tên_mảng)
  - Ví dụ:

```php
<?php
$mang = array('1' => 'Luân Nguyễn', 'a' => 'Hùng Nguyễn', 'aa' => 'Linh Nguyễn', '2' => 'Minh Trần', '3' => 'Hạnh Trần');
krsort($mang);
echo '<pre>',print_r($mang),'</pre>';
?>
```

  - Kết quả:

```
Array
(
    [3] => Hạnh Trần
    [2] => Minh Trần
    [1] => Luân Nguyễn
    [aa] => Linh Nguyễn
    [a] => Hùng Nguyễn
)
1
```

Các hàm xử lý mảng

- Tìm kiếm trên mảng
- Ghép mảng
- Đếm số lần xuất hiện
- Tạo mảng duy nhất
- Tạo mảng các giá trị của một cột, thuộc tính của một mảng truyền vào
- Tìm các giá trị khác nhau của một mảng so với mảng khác
- Tìm các từ khóa(key) khác nhau của một mảng so với mảng khác
- Thêm một hay nhiều phần tử vào mảng
- Xóa phần tử cuối cùng của mảng
- Tính tổng tất cả phần tử của mảng

Tìm kiếm trên mảng: array_search()

- Tìm kiếm một giá trị trên mảng
  - Nếu tìm thấy sẽ trả về khóa của phần tử chứa giá trị đó, nếu không tìm thấy sẽ trả về giá trị NULL.
- Cú pháp:

  $khóa = array_search(giá_trị_cần_tìm, $tên_mảng);
- Ví dụ:

```php
<?php
$mang = array(0 => 'xanh', 1 => 'đỏ', 2 => 'tím', 3 => 'vàng');
$khoa_do = array_search('đỏ', $mang); → 1
$khoa_vang = array_search('vàng', $mang);  → 3
?>
```

Ghép mảng: array_merge()

- Ghép hai hay nhiều mảng với nhau
- Kết quả trả về là một mảng mới được tạo ra từ các mảng
- Cú pháp:

$mảng_ghép = array_merge($mảng_1, $mảng_2, ...);

- Chú ý: khi các mảng dùng để ghép có khóa trùng nhau thì mảng ghép sẽ chỉ lấy phần tử có khóa trùng của mảng cuối cùng.

```php
<?php
    $mang1 = array("màu" => "đỏ", 2, 4);
    $mang2 = array("a", "b", "màu" => "xanh", "hình" => "tròn", 4);
    $mang_chung = array_merge($mang1, $mang2);
    print_r($mang_chung);
    // →    Array (
    //  [màu] => xanh   [0] => 2    [1] => 4    [2] => a
    //  [3] => b    [hình] => tròn  [4] => 4 )
?>
```

Đếm số lần xuất hiện: array_count_values()

- Đếm số lần xuất hiện của các phần tử trong mảng.
- Kết quả trả về là một mảng trong đó khóa chính là giá trị trên mảng cần đếm và giá trị sẽ là số lần xuất hiện của nó trong mảng.
- Cú pháp:

  $mảng_slxh = array_count_values($tên_mảng);

```php
<?php
    $mang = array(1, "hello", 1, "world", "hello", 2, "Xin chào", 1);
    $mang_slxh = array_count_values($mang);
    print_r($mang_slxh);
    // → Array (
    //   [1] => 3
    //   [hello] => 2
    //   [world] => 1
    //   [2] => 1
    //   [Xin chào] => 1 )
?>
```

Tạo mảng duy nhất: array_unique()

- Hàm này sẽ bỏ đi những giá trị lặp lại trong mảng.
- Kết quả trả về là một mảng mới mà trong đó mỗi phần tử trong mảng chỉ xuất hiện một lần.
- Cú pháp:

  $mảng_duy_nhất = array_unique($tên_mảng);

```php
<?php
    $mang = array(1,3,1,2,5,1,3,4);
    $mang_duy_nhat= array_unique($mang);
    // → 1, 3, 2, 5, 4
?>
```

Tạo mảng các giá trị của một cột, thuộc tính của một mảng truyền vào: array_column()

- Hàm sẽ lấy toàn bộ giá trị từ một từ khóa (cột, thuộc tính) từ các phần tử một mảng truyền vào và trả về một mảng là các giá trị đó.
- Tất nhiên hàm này cũng có thể áp dụng cho một mảng đối tượng.
- Cú pháp:

  $mang_cot = array_column($ten_mang,"tên_cột");

```
$ds_sinh_vien = array(array("id"=>"SV001", "ho_ten"=>"Hùng Nguyễn"),
                      array("id"=>"SV002", "ho_ten"=>"Linh Nguyễn"),
                      array("id"=>"SV003", "ho_ten"=>"Hoàng Nguyễn"),
                      array("id"=>"SV004", "ho_ten"=>"Vân Phan"));

$mang_ho_ten = array_column($ds_sinh_vien, "ho_ten");
echo "<pre>",print_r($mang_ho_ten),"</pre>";
```

Tìm các từ khóa(key) khác nhau của một mảng so với mảng khác: array_diff_key()

- So sánh giữa hai mảng và lọc ra những phần tử có từ khóa(key) chỉ có trong mảng thứ nhất mà không có trong mảng thứ hai.
- Kết quả trả về là một mảng mới với những phần tử có từ khóa(key) chỉ xuất hiện duy nhất trong mảng một.
- Cú pháp:

  array_diff_key($tên_mảng_1, $tên_mảng_2);

```
$ds_sinh_vien = array("SV001"=>"Hùng Nguyễn","SV002"=>"Linh Nguyễn",
                      "SV003"=>"Hoàng Nguyễn","SV004"=>"Vân Phan");

$ds_sinh_vien_2 = array("SV005"=>"Hoàng Nguyễn","SV006"=>"Chiêu Hoa",
                        "SV003"=>"Trang Lê","SV004"=>"Quỳnh Như");

$mang_diff_key = array_diff_key($ds_sinh_vien, $ds_sinh_vien_2);
echo "<pre>",print_r($mang_diff_key),"</pre>";
```

Thêm một hay nhiều phần tử vào mảng: array_push()

- Để thêm một hay nhiều phần tử mới vào mảng ta dùng hàm array_push()
- Nhưng thường hàm array_push chỉ được sử dụng cho trường hợp thêm nhiều, còn thêm một trong PHP có thể sử dụng cách sau:

  $mang[] = giá trị cần thêm vào;
- Cú pháp hàm array_push:

  array_push($tên_mảng, phần_tử_1, phần_tử_2,…);

Xóa phần tử cuối cùng của mảng: array_pop()

- Để xóa phần tử cuối cùng của mảng trong PHP ta dùng hàm array_pop()
- Sau khi xóa phần tử cuối của mảng hàm cũng xóa luôn index của phần tử đó, nên khi thêm vào phần tử mới thì phần tử mới sẽ có index là index của phần tử vừa xóa.
- Giá trị của hàm trả về sẽ là mảng nếu giá trị truyền vào là mảng hợp lệ, ngược lại thi trả về null
- Cú pháp hàm array_pop:

  array_pop($tên_mảng);

Tính tổng tất cả phần tử của mảng: array_sum()

- PHP cung cấp hàm array_sum giúp tính tổng toàn bộ phần tử trong mảng.
- Với các phần tử có giá trị không phải là số thì PHP sẽ ngầm chuyển nó về giá trị số phù hợp
- Cú pháp của hàm array_sum:

  array_sum($tên_mảng);

Thảo luận