

NHẬP MÔN LẬP TRÌNH WEB VỚI PHP

**Please purchase a personal
license.**

Bài 5: Mạng

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

1. Khái niệm về mạng

- Mạng
 - 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 mạng có thể có kiểu dữ liệu khác nhau
- Ví dụ:
 - 8 số nguyên => mạng có 8 phần tử
 - Danh sách học sinh => mạng thông tin các học sinh{

2. Khai báo và sử dụng mạng

- Mạng một chiều
 - Khai báo và khởi tạo
 - Truy xuất phần tử trong mạng
 - Thao tác trên mạng một chiều
- Mạng hai chiều
 - Khai báo và khởi tạo
 - Truy xuất phần tử trong mạng
 - Thao tác trên mạng 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  
    $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 định, 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
    $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
    $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
 - Tích Tổng / Tích
 - Tìm kiếm

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

Mạng một chiều

- Thao tác trên mạng một chiều
 - Duyệt mạng
 - Duyệt mạng có khóa tự động
 - Duyệt mạng 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
for($i=0; $i<$số_phần_tử; $i++)
{
    Xử lý các phần tử trong mảng (đọc, tính toán, thay
    đổi, ...)
    với mỗi phần tử: $tên_mảng[i]
}
?>
```

Mảng một chiều

- Thao tác trên mảng một chiều
 - Duyệt mảng có khóa tự động
 - Ví dụ:

```
<?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
foreach ($tên_mảng as $giá_trị)
{
    Xử lý các giá trị trong mảng (đọc, tính toán, thay đổi, ...)
    với mỗi phần tử: $giá_trị
}
?>
```

Mảng một chiều

- Thao tác trên mảng một chiều
 - Duyệt mảng có khóa do người dùng tạo
 - Ví dụ:

```
<?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
 - Duyệt mảng có khóa do người dùng tạo
 - Dùng vòng lặp foreach để duyệt mảng – Duyệt để lấy cả giá trị của khóa và giá trị của phần tử
 - Cú pháp:

```
<?php  
foreach($tên_mang as $khóa=>$giá_trị)  
{  
    Xử lý  
    với mỗi phần tử: $khóa=>$giá_trị  
}  
?>
```


Mảng một chiều

- Thao tác trên mảng một chiều
 - Duyệt mảng có khóa do người dùng tạo
 - Ví dụ:

```
<?php
    $mang_3 = array("mot"=>1, "hai"=>2, "ba"=>3,
    "bon"=>4, "nam"=>5);
    foreach ($mang_3 as $khoa=>$gia_tri)
    {
        echo "<br> [$khoa] => $gia_tri";
    }
→      [mot] => 1      [hai] => 2      [ba] => 3
        [bon] => 4     [nam] => 5
?>
```

Mảng một chiều

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

```
<?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
    $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;

?>
```

Mảng một chiều

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

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

Mảng một chiều

- Thao tác trên mảng một chiều
 - Tính tổng, tích:

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

    $tong = 0; $tich = 1;
    for($i=0; $i<count($mang); $i++)
    {
        $tong += $mang[$i];
        $tich *= $mang[$i];
    }
?>
```

Mảng một chiều

- Thao tác trên mảng một chiều
 - Tìm kiếm:

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

    $vi_tri = -1; //dùng để lưu vị trí tìm thấy x
    $x = 12; //giá trị cần tìm
    for($i=0; $i<count($mang); $i++)
    {
        if($mang[$i]==$x)
        {
            $vi_tri = $i;
            break;
        }
    }
    if($vi_tri!=-1) echo "Tìm thấy x tại vị trí $vi_tri";

?>
```

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();`
 - Ví dụ:

```
<?php  
    $mang_2_chieu = array();  
?>
```

Mảng hai chiều

- Khai báo và khởi tạo

- Mảng biết trước giá trị của các phần tử

```
$tên_mảng = array( array([$khóa]=$giá_trị, ...),  
                  array(...),  
                  ...  
                  );
```

- Ví dụ:

```
<?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"));  
?>
```


Mảng hai chiều

- Khai báo và khởi tạo

- Khởi tạo giá trị cho từng phần tử của mảng:

- Mảng có khóa tự động (người dùng không thiết lập giá trị cho khóa)

- Cú pháp:

`$tên_mảng[vị trí dòng][vị trí cột] = <giá trị>;`

- Ví dụ:

```
<?php  
    $mang_1[2][3] = 7;  
?>
```

Mảng hai chiều

- Khai báo và khởi tạo
 - Khởi tạo giá trị cho từng phần tử của mảng:
 - Mảng có khóa do người thiết lập giá trị
 - Cú pháp:
`$tên_mảng[vị trí dòng][giá trị khóa] = <giá trị>;`
 - Ví dụ:

```
<?php  
    $mang_2[2][7] = "Bảy";  
?>
```

Mảng hai chiều

- Truy xuất phần tử trong mảng
 - Cú pháp:
 $\text{\$giá_trị} = \text{\$tên_mảng}[\text{vị trí dòng}][\text{vị trí cột}];$
 - Ví dụ: truy xuất phần tử đầu tiên trong các mảng ở ví dụ trên

```
<?php
```

```
    $gia_tri_1 = $mang_1[0][0]; → 1
```

```
    $gia_tri_2 = $mang_2[0][1]; → "Một"
```

```
?>
```

Mạng hai chiều

- Thao tác trên mạng hai chiều
 - Đếm số dòng của mạng
 - Duyệt từng phần tử trên mỗi dòng

Mảng hai chiều

- Thao tác trên mảng hai chiều
 - Đếm số dòng của mảng **count()**
 - Cú pháp: **\$số_dòng = count(\$tên_mảng);**

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
    $so_dong=count($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 $so_dong=count($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

```
?>
```

3. Sắp xếp mạng

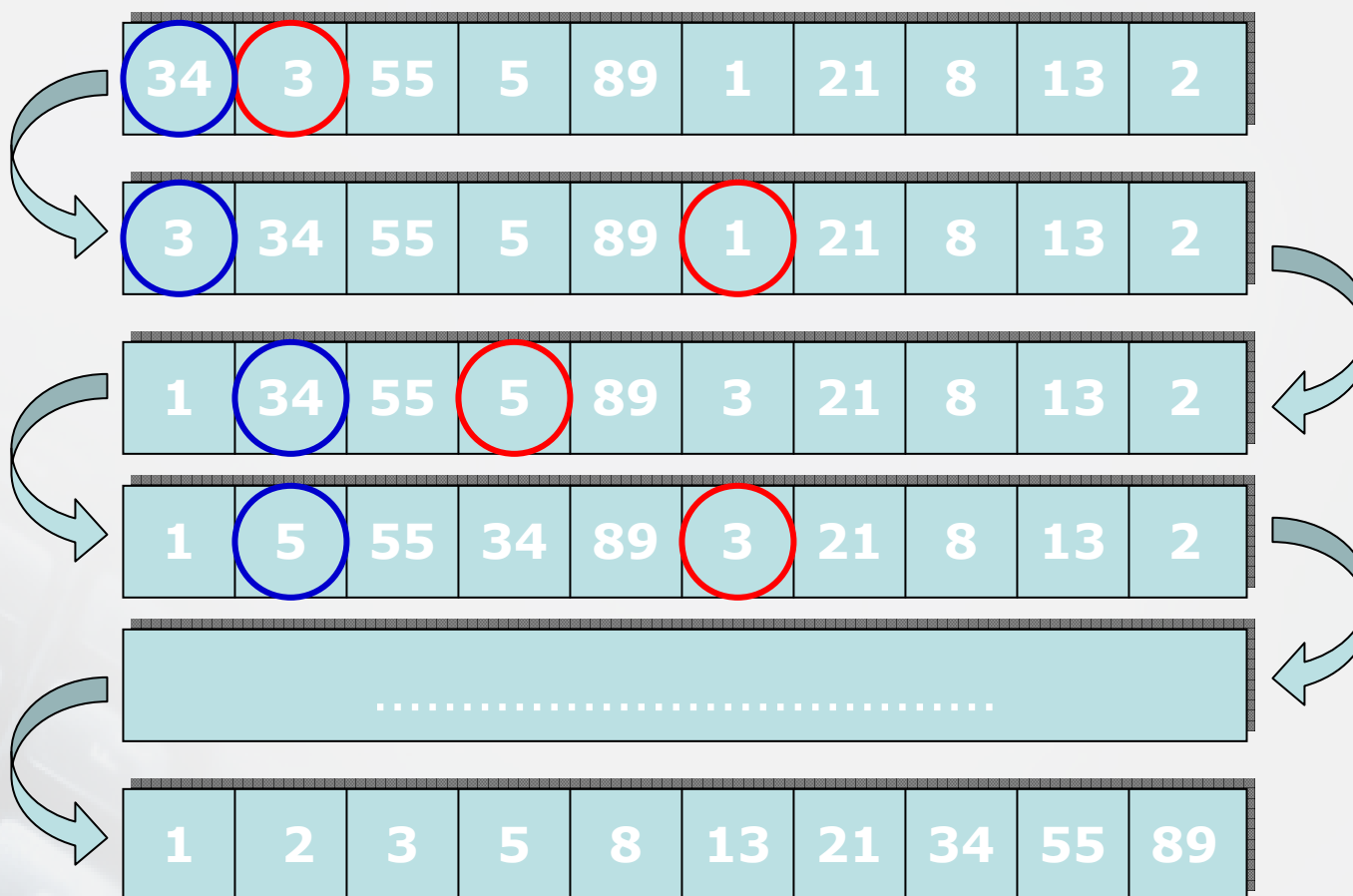
- Sắp xếp mạng một chiều
- Sắp xếp mạng hai chiều

Sắp xếp mảng một chiều

- Sắp xếp mảng có khóa tự động (người dùng không thiết lập giá trị cho khóa): dùng hai vòng lặp for lồng nhau để sắp xếp

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
    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
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
```

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

- Sắp giảm dần: **rsort(\$tên_mảng)**
 - Ví dụ:

```
<?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 hai chiều

- Hàm so sánh `compare()`
- Hàm sắp xếp tăng dần theo giá trị của cột `usort($tên_mảng, "compare")`:

Sắp xếp mảng hai chiều

- Hàm `compare()`
 - So sánh hai phần tử có khóa là cột muốn sắp xếp.
 - Nếu sắp xếp dựa theo cột đầu tiên - khóa có giá trị là 0:

```
<?php
function compare($a,$b)
{
    if ($a[0]==$b[0])
        return 0;
    if($a[0]<$b[0])
        return -1;
    else
        return 1;
}
?>
```


Sắp xếp mảng hai chiều

- Hàm `usort($mang, "compare")`
 - Ví dụ: sắp xếp mảng hai chiều tăng dần theo cột đầu tiên (cột 0) => truyền vào cho tham số thứ hai của hàm `usort()` giá trị là hàm `compare`

```
<?php
    $san_pham = array(
        array("XB", "Xà bông", 5500),
        array("KDR", "Kem đánh răng", 14500),
        array("DGD", "Dầu gội đầu", 28500));
    usort($san_pham, "compare");
    print_r($san_pham);
→   Array ([0] => Array ([0] => DGD [1] => Dầu gội đầu [2] => 28500)
      [1] => Array ([0] => KDR [1] => Kem đánh răng [2] => 14500)
      [2] => Array ([0] => XB [1] => Xà bông [2] => 5500) )
?>
```

4. Các hàm xử lý mảng

- Tìm kiếm trên mảng
- Ghép mảng
- Đếm số lần xuất hiện
- Tạo mảng duy nhất
- Tìm các giá trị khác nhau của một mảng so với mảng khác

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);
```

Tìm kiếm trên mảng: array_search()

- Ví dụ:

```
<?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.

Ghép mảng: array_merge()

■ Ví dụ:

```
<?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);`

Đếm số lần xuất hiện: `array_count_values()`

- Ví dụ:

```
<?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);`

Tạo mảng duy nhất: array_unique()

- Ví dụ:

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

Tìm các giá trị khác nhau của một mảng so với mảng khác: `array_diff()`

- So sánh giữa hai mảng và lọc ra những phần tử 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 chỉ xuất hiện duy nhất trong mảng một.
- Cú pháp:
`array_diff($tên_mảng_1, $tên_mảng_2);`

Tìm các giá trị khác nhau của một mảng so với mảng khác: `array_diff()`

- Ví dụ:

```
<?php
    $mang_1 = array("a" => "xanh", "đỏ", "tím", "vàng");
    $mang_2 = array("b" => "xanh", "vàng", "đỏ");
    $mang_con_1 = array_diff($mang_1, $mang_2);
    print_r($mang_con_1);
    → Array ( [1] => tím )
?>
```

Hết