

Bài 6

Thao tác trên mạng

1. Khái niệm về mảng

- Mảng
 - Bao gồm một dãy các ô nhớ có nhiều ô nhớ con cho phép biểu diễn thông tin dạng danh sách trong thực tế
- PHP: 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

2. Khai báo và sử dụng mảng

Mảng một chiều

- Khai báo và khởi tạo
 - Dùng **array()** để khai báo và khởi tạo
 - Mảng chưa biết số phần tử và giá trị
 - Cú pháp: **\$tên_mảng = array();**
 - Ví dụ:

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

Mảng một chiều

- Mảng biết trước số phần tử
 - Cú pháp: `$tên_mảng = array(<số phần tử>);`
 - Ví dụ:

```
<?php
    $mang = array(10);
?>
```

Mảng một chiều

- Mảng biết trước giá trị của các phần tử trong mảng
 - Vừa khai báo vừa 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

- Mảng biết trước giá trị của các phần tử trong mảng
 - 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

- Khởi tạo giá trị cho từng phần tử của mảng
 - Phần tử dùng khóa mặc định (người dùng không tạo giá trị cho khóa)

Cú pháp: `$tên_mảng[] = <giá trị>;`

- Phần tử có giá trị khóa

Cú pháp: `$tên_mảng[“giá trị khóa”] = <giá trị>;`

```
<?php
    $mang_1[] = 6;
    $mang_2[6] = “Sáu”;
    $mang_3[“sau”] = 6;
?>
```


Mảng một chiều

- Truy xuất phần tử trong mảng
 - Cú pháp: $\$gia_tri = \$tên_mảng[<giá\ tri\ khóa>];$
 - Ví dụ:

```
<?php
    $gia_tri_1 = $mang_1[0]; → 1
    $gia_tri_2 = $mang_2[1]; → “Một”
    $gia_tri_3 = $mang_3[“mot”]; → 1
?>
```

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
?>
```

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

- 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

- 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

- 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

– 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 hai chiều

- Khai báo và khởi tạo
 - Lồng **array()** hai cấp
 - Mảng chưa biết số phần tử và giá trị
 - Cú pháp: **\$tên_mảng = array(array(...));**
 - Ví dụ:

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

Mảng hai chiều

▪ Khai báo và khởi tạo

– Mảng biết trước giá trị của các phần tử

- Vừa khai báo vừa khởi tạo giá trị cho các phần tử của mảng
- Cú pháp: `$tên_mảng = array(array([$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: `$giá_trị = $tên_mảng[vi trí dòng][vi 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”
```

```
?>
```

Mảng hai chiều

- Thao tác trên mảng hai chiều
 - Đếm số dòng của mảng `count()`
 - Cần phải biết số dòng trên mảng vì các thao tác duyệt mảng hai chiều sẽ duyệt lần lượt từng dòng trên mảng, và cứ mỗi dòng này lại duyệt tiếp từng phần tử có trên dòng
 - 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
?>
```


Quizes

- Cho biết sau khi thực hiện đoạn mã dưới đây thì biến mảng `$array` chứa những phần tử nào?

```
<?php
    $array = array ('1', '1');
    foreach ($array as $k => $v) {
        $v = 2;
    }
?>
```

Quizes

- Cho mảng `$multi_array` được khai báo như dưới đây. Cho biết câu lệnh truy xuất đến giá trị **cat** trong mảng?

```
<?php
$multi_array = array("red", "green", 42 => "blue",
                    "yellow" => array("apple",
                                      9 => "pear",
                                      "banana",
                                      "orange" =>
array("dog", "cat", "iguana")
                                      )
                    );
?>
```

Quizes

- Hãy cho biết kết quả của đoạn script PHP sau:

```
<?php
    $s = '12345';
    $s[$s[1]] = '2';
    echo $s;
?>
```

Quizes

- Hãy cho biết kết quả của đoạn script PHP sau:

```
<?php
    $array = array (0.1 => 'a', 0.2 => 'b');
    echo count ($array);
?>
```

Quizes

–Hãy cho biết kết quả của đoạn script PHP sau:

```
<?php
    $array = array (1, 2, 3, 5, 8, 13, 21, 34, 55);
    $sum = 0;
    for ($i = 0; $i < 5; $i++) {
        $sum += $array[$array[$i]];
    }
    echo $sum;
?>
```

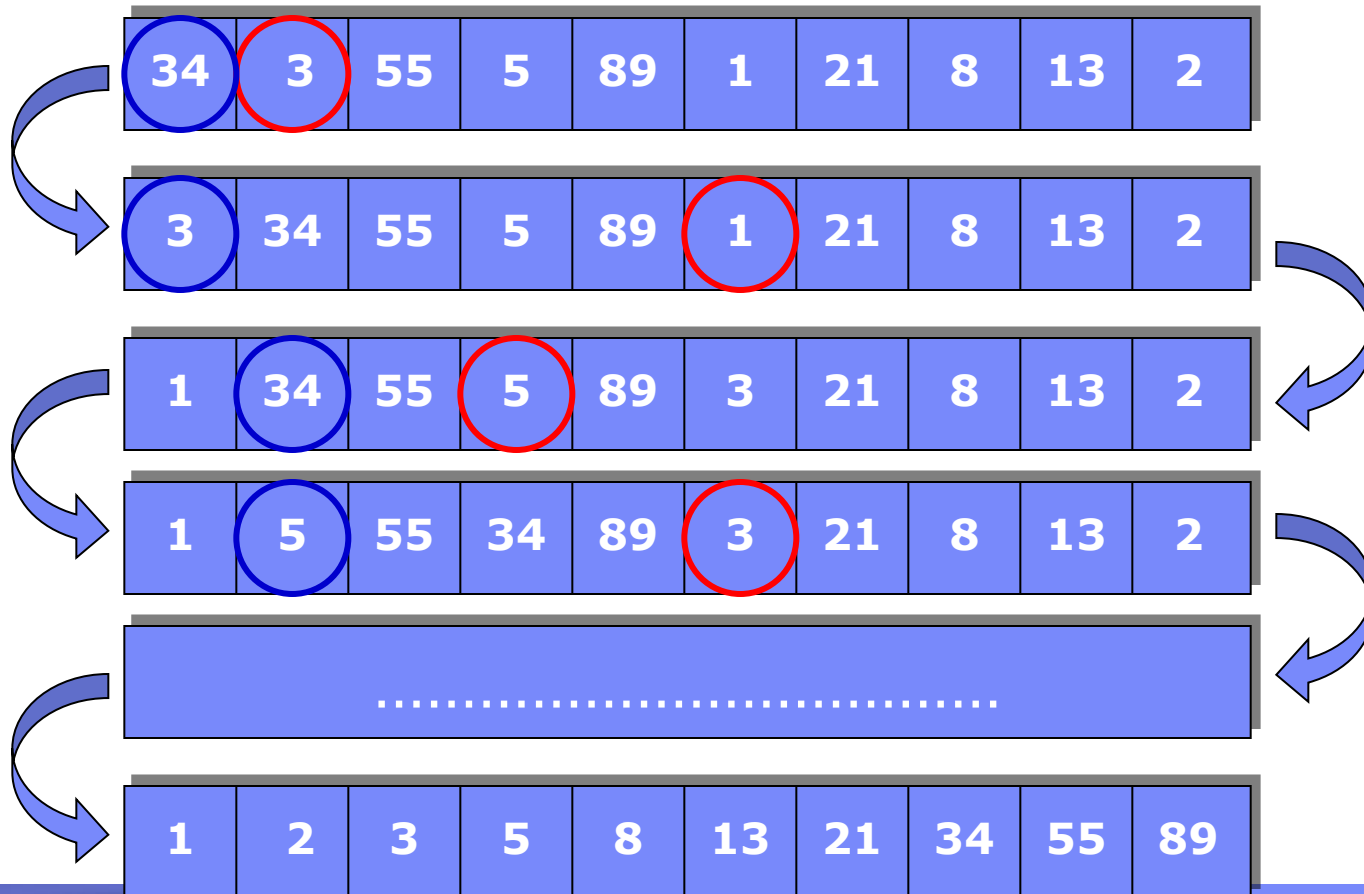
3. Sắp xếp mảng

Sắp xếp mảng một chiều

- Sắp xếp mảng có khóa mặc định (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



Sắp xếp mảng một chiều

- Sắp xếp mảng dùng vòng lặp
 - Thuật toán sắp xếp đổi chỗ hai phần tử

```
<?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]);
        }
    }
    return $tên_mảng;
}
?>
```

Sắp xếp mảng một chiều

- 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 dùng vòng lặp
 - Ví dụ: sắp xếp mảng tăng dần và giảm dần

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

Sắp xếp mảng một chiều

- Sử dụng hàm sắp xếp trong thư viện mảng
 - Sắp xếp mảng (có khóa mặc định) theo giá trị của các phần tử
 - Sắp tăng dần
- Cú pháp: **sort(\$tên_mảng)**
- Ví dụ:

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

Sắp xếp mảng một chiều

- Sử dụng hàm sắp xếp trong thư viện mảng
 - Sắp xếp mảng có khóa do người dùng nhập
 - Sắp xếp theo giá trị: Dùng hàm **sort()** để sắp tăng và **rsort()** để sắp giảm tương tự như sắp xếp mảng có khóa mặc định

Sắp xếp mảng một chiều

- Sử dụng hàm sắp xếp trong thư viện mảng
 - Sắp xếp mảng có khóa do người dùng nhập
 - Sắp xếp theo khóa - Sắp xếp tăng dần
- Cú pháp: **ksort(\$tên_mảng)**
- Ví dụ:

```
<?php
$trai_cay = array("d"=>"chanh", "a"=>"lê", "b"=>"chuối", "c"=>"táo");
ksort($trai_cay);
foreach ($trai_cay as $khoa => $gia_tri)
{
    echo "$khoa = $gia_tri \t";
}
→a = lê   b = chuối   c = táo   d = chanh
?>
```

Sắp xếp mảng một chiều

- Sử dụng hàm sắp xếp trong thư viện mảng
 - Sắp xếp mảng có khóa do người dùng nhập
 - Sắp xếp theo khóa - Sắp xếp giảm dần
- Cú pháp: **ksort(\$tên_mảng)**
- Ví dụ:

```
<?php
$trai_cay = array("d"=>"chanh", "a"=>"lê", "b"=>"chuối", "c"=>"táo");
ksort($trai_cay);
foreach ($trai_cay as $khoa => $gia_tri)
{
    echo "$khoa = $gia_tri \t";
}
→d = chanh   c = táo   b = chuối   a = lê
?>
```

Sắp xếp mảng một chiều

- Sắp xếp mảng có khóa do người dùng tạo
 - Xây dựng hàm so sánh dùng để so sánh hai giá trị với nhau.
 - Tùy theo yêu cầu sắp xếp => sử dụng các hàm sắp xếp khác nhau

Sắp xếp mảng một chiều

- Sắp xếp mảng có khóa do người dùng tạo
 - Hàm so sánh: dùng để so sánh hai giá trị với nhau

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

Sắp xếp mảng một chiều

- Sắp xếp mảng có khóa do người dùng tạo
 - Sắp xếp theo giá trị của các phần tử trong mảng
 - Sắp tăng dần
Cú pháp: **usort(\$tên_mảng, “compare”)**
 - Sắp giảm dần
Cú pháp: **ursort(\$tên_mảng, “compare”)**

Sắp xếp mảng một chiều

- Sắp xếp mảng có khóa do người dùng tạo
 - Sắp xếp theo khóa của các phần tử trong mảng
 - Sắp tăng dần
- Cú pháp: **uksort(\$tên_mảng, "compare")**
- Ví dụ:

```
<?php
$a = array(4 => "four", 3 => "three", 20 => "twenty", 10 => "ten");
uksort($a, "compare");
foreach ($a as $key => $value)
{
    echo "$key: $value\n";
}
→ 3: three 4: four 10: ten 20: twenty
?>
```

4. Các hàm xử lý khác trên 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 mới trong đó:
 - khóa của phần tử trong mảng mới là phần tử trên mảng cần đếm
 - giá trị của phần tử trong mảng mới sẽ là số lần xuất hiện trong mảng cần đếm.
- 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);
    $mang_uy_nhat= array_unique($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 )
?>
```

Quizes

- Hãy cho biết kết quả khi thực hiện đoạn script PHP sau đây?

```
<?php  
$array = array (0.1 => 'a', 0.2 => 'b');  
echo count ($array);  
?>
```

Quizes

- Trang “testscript.php” được xây dựng như sau, hãy cho biết khi người dùng truy nhập với địa chỉ truy nhập trên trình duyệt là “**http://localhost/testscript.php?c=25**” thì kết quả thu được là gì?

```
<?php
function process($c, $d = 25)
{
    global $e;
    $retval = $c + $d - $_GET['c'] - $e;
    return $retval;
}
$e = 10;
echo process(5);
?>
```


Bài tập

- Xây dựng đoạn mã PHP thực hiện:

TÌM KIẾM

Nhập mảng:	<input style="width: 90%;" type="text" value="1, 3, 5, 7, 9, 10, 11, 12, 14, 5, 6"/>
Nhập số cần tìm:	<input style="width: 50%;" type="text" value="9"/>
	<input style="width: 50%; background-color: #4682b4; color: white;" type="button" value="Tìm kiếm"/>
Mảng:	<input style="width: 90%;" type="text" value="1, 3, 5, 7, 9, 10, 11, 12, 14, 5, 6"/>
Kết quả tìm kiếm:	<input style="width: 90%; background-color: #e0ffff;" type="text" value="Tìm thấy 9 tại vị trí thứ 5 của mảng"/>

(Các phần tử trong mảng sẽ cách nhau bằng dấu ",")

Quizes

- Hãy cho biết kết quả khi thực hiện đoạn script PHP sau đây?

```
<?php  
$array = array (true => 'a', 1 => 'b');  
print_r ($array);  
?>
```

Quizes

- Hãy cho biết kết quả khi thực hiện đoạn script PHP sau đây?

```
<?php  
$dog = "Dogzilla";  
$dragon = &$dog;  
$dragon = "Dragonzilla";  
echo $dog . " " . $dragon;  
?>
```

Quizes

- Hãy cho biết kết quả khi thực hiện đoạn script PHP sau đây?

```
<?php  
define(myvalue, "10");  
$myarray[10] = "Dog";  
$myarray[] = "Human";  
$myarray['myvalue'] = "Cat";  
$myarray["Dog"] = "Cat";  
print "The value is: ";  
print $myarray[myvalue]."<br>";  
?>
```