Trường ĐH CNTT & TT Việt Hàn
Khoa KHMT

LẬP TRÌNH WEB

Đà Nẵng tháng .../20...

Giảng viên: Ths. Võ Ngọc Đạt
Email: vongocdatit@gmail.com

KOICA

BÀI 3
MẢNG VÀ CHUỖI KÝ TỰ

NỘI DUNG BÀI 3

- GIỚI THIỆU VỀ MẢNG
- MẢNG MỘT CHIỀU
- MẢNG HAI CHIỀU
- CÁC HÀM XỬ LÝ TRÊN MẢNG
- CHUỖI KÝ TỰ
- CÁC HÀM XỬ LÝ TRÊN CHUỖI

KHÁI NIỆM MẢNG

- Mảng là một biến có khả năng lưu trữ nhiều phần tử, mỗi phần tử có thể mang bất cứ một kiểu giá trị nào mà PHP hỗ trợ như interger, float, string, array,…
- Riêng với PHP thì các phần tử của mảng có thể không cùng kiểu dữ liệu, và các phần tử của mảng được truy xuất thông qua các chỉ mục (vị trí) của nó nằm trong mảng.

PHÂN LOẠI MẢNG

- **Căn cứ vào chỉ số mảng**
  - Mảng có chỉ số kiểu number (mảng tuần tự)
  - Mảng có chỉ số kiểu kết hợp (associative) (mảng không tuần tự)

Ví dụ:

$tpho = array(“HoChiMinh”, “HaNoi”, “HaiPhong”, “DaNang”);

$tpho = array(“HCM” => “HoChiMinh”, “HN” => “HaNoi”, “HP” => “HaiPhong”, “DN” => “DaNang”);

- **Căn cứ vào số chiều của mảng**
  - Mảng một chiều
  - Mảng đa chiều

Ví dụ:

$lop= array(array(“LT01A”, 34), array(“LT01B”, 35));

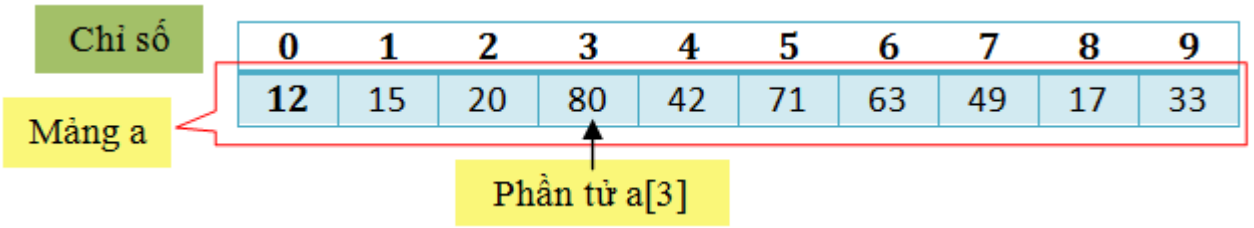
KHAI BÁO MẢNG 1 CHIỀU

- **Khai báo mảng 1 chiều có chỉ số kiểu number (mảng tuần tự):**

$a = array(); // khởi tạo một mảng gán vào biến $a

Ví dụ:

$a = array(12,15,20,80,42,71,63,49,17,33);



- **Khai báo mảng 1 chiều có chỉ số kiểu kết hợp (associative) (mảng không tuần tự)**

$a = array(key => value); // khởi tạo một mảng gán vào biến $a

Ví dụ:

$tpho = array("HCM" => "HoChiMinh", "HN" => "HaNoi", "HP" => "HaiPhong", "DN" => "DaNang");

KHAI BÁO MẢNG 1 CHIỀU

- Khi các giá trị gán cho mảng là các giá trị nguyên hoặc ký tự có tính chất sắp xếp (tăng hoặc giảm) dần, ta sử dụng hàm **range()** để khai báo mảng.

  $biến_mảng=range(giá trị đầu, giá trị cuối);

Ví dụ:

```
$number = range(0, 20);
$charac = range("A", "Z");
```

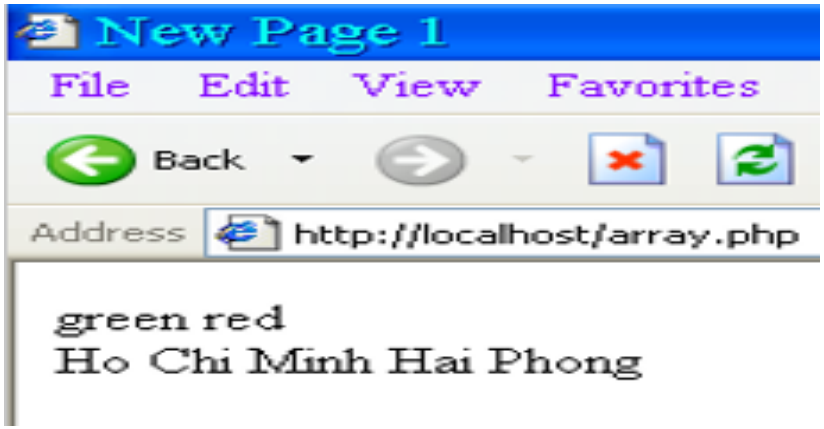
LÀM VIỆC VỚI MẢNG 1 CHIỀU

- Có thể truy cập giá trị của các phần tử mảng thông qua tên **biến_mảng** kèm với **chỉ số** của nó.

$biến_mảng[“chỉ số”]

Ví dụ:

```
<?php
    $color=array(“gray”, “green”, “red”, “blue”, “yellow”);
    $tpho=array(“HCM”=>“Ho Chi Minh”, “HN”=>“Ha Noi”, “HP”=>“Hai
        Phong”, “DN”=>“Da Nang”);
    echo $color[1]. “ ”. $color[2]. “<br>”;
    echo $tpho[“HCM”]. “ ”. $tpho[“HP”];
?>
```

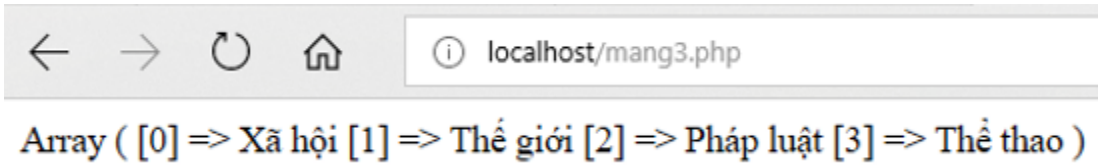


LÀM VIỆC VỚI MẢNG 1 CHIỀU

- **Hàm print_r():** cho phép xuất dữ liệu mảng phục vụ cho quá trình rà soát, kiểm tra dữ liệu. (duyệt nhanh giá trị các phần tử mảng)

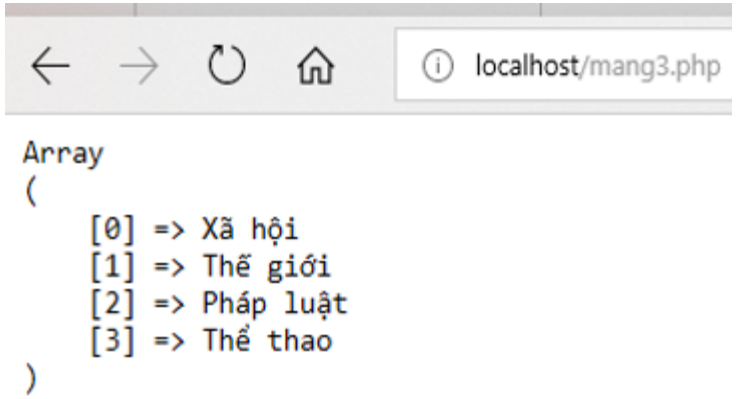
Ví dụ:

```
<?php
    $cat_news = array('Xã hội', 'Thế giới', 'Pháp luật', 'Thể thao');
    print_r($cat_news);
?>
```



Ví dụ: Hiển thị mảng trực quan

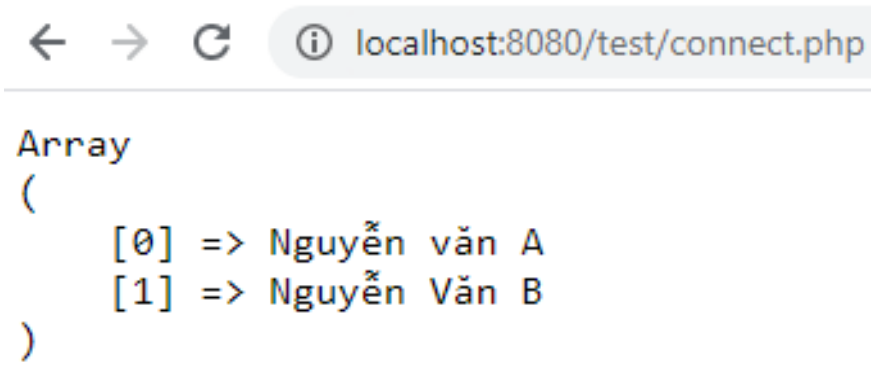
```
<?php
    echo "<pre>";
        print_r ($cat_news);
    echo "</pre>";
?>
```



LÀM VIỆC VỚI MẢNG 1 CHIỀU

❑ **Có thể gán giá trị trực tiếp cho các phần tử mảng**

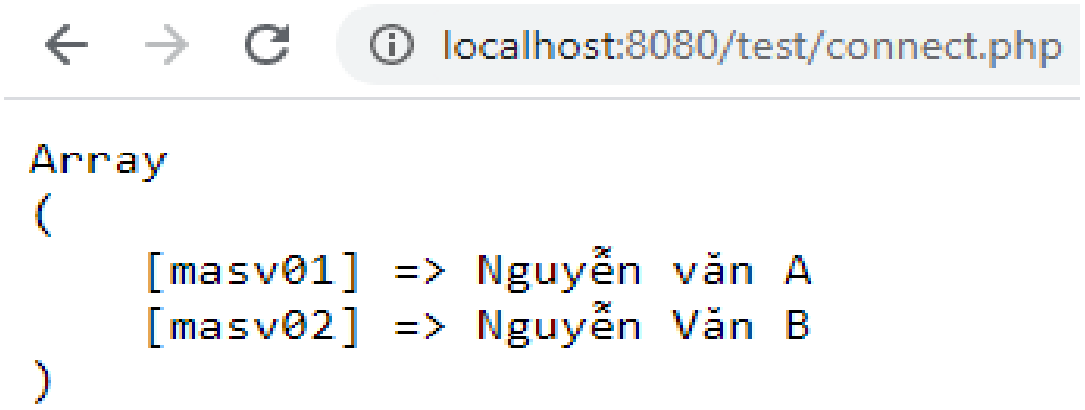
```
<?php
$sinhvien = array();
$sinhvien[] = 'Nguyễn văn A';
$sinhvien[] = 'Nguyễn Văn B';
echo "<pre>";
   print_r($sinhvien);
echo "</pre>";
?>
```



LÀM VIỆC VỚI MẢNG 1 CHIỀU

❑ **Có thể gán giá trị trực tiếp cho các phần tử mảng**

```
<?php
$sinhvien = array();
$sinhvien['masv01'] = 'Nguyễn văn A';
$sinhvien['masv02'] = 'Nguyễn Văn B';
echo "<pre>";
   print_r($sinhvien);
echo "</pre>";
?>
```



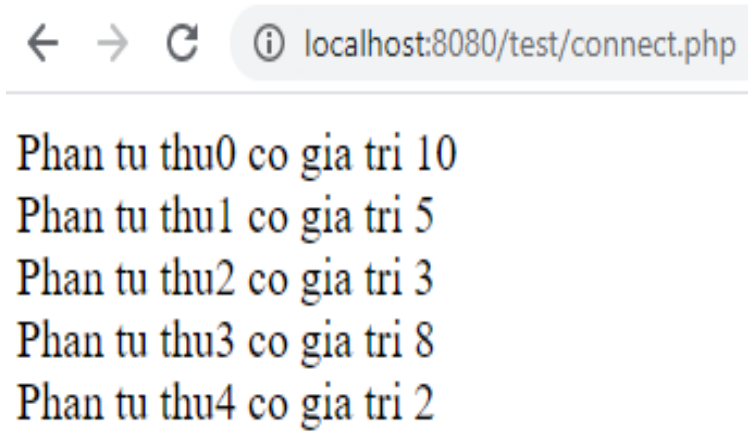
LÀM VIỆC VỚI MẢNG 1 CHIỀU

- **Sử dụng vòng lặp để duyệt các phần tử mảng**

Ví dụ:

```
<?php
  $mang = array(10,5,3,8,2);
  for ($i=0; $i<count($mang); $i++)
      echo "Phan tu thu" . $i ." co gia tri " . $mang[$i] . "<br>";
?>
```

=> Chú ý: Hàm count($biến_mảng) hoặc sizeof($biến_mảng) trả về số phần tử của mảng.



LÀM VIỆC VỚI MẢNG 1 CHIỀU

❖ Duyệt mảng kết hợp

- Sử dụng vòng lặp **foreach()**

Cú pháp: **foreach**($biến_mảng as $key => $value)

{ //Các dòng lệnh;}

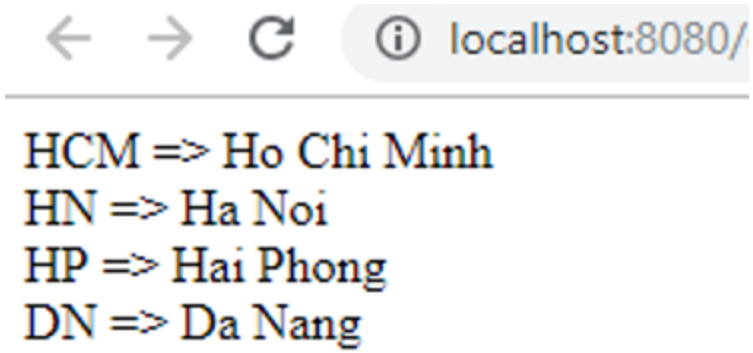
Hoặc: **foreach**($biến_mảng as $value)

{ //Các dòng lệnh;}

- **$biến_mảng** là mảng cần lặp, **$key** là chỉ mục, **$value** là giá trị của phần tử.

Ví dụ:

```
<?php
$tpho=array("HCM"=>"Ho Chi Minh", "HN"=>"Ha Noi", "HP"=>"Hai Phong",
            "DN"=>"Da Nang");
foreach($tpho as $key=>$value)
  echo "$key => $value <br/>";
  ?>
```



SẮP XẾP MẢNG 1 CHIỀU

- Đối với mảng một chiều, sau khi khai báo và gán giá trị cho các phần tử mảng, để sắp xếp mảng có một số hàm sau:
  - **Hàm sort()**
  - **Hàm rsort()**
  - **Hàm ksort()**
  - **Hàm krsort()**

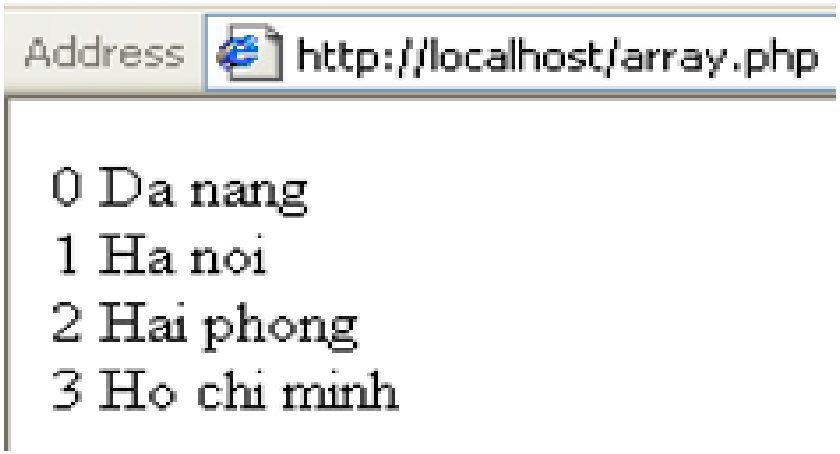
HÀM SORT()

- Sắp xếp mảng theo chiều **tăng dần** của giá trị các phần tử mảng nhưng **chỉ số tương ứng bị thay đổi**

Cú pháp: sort($biến_mảng)

Ví dụ:

```
<?php
    $tp=array("HCM"=>"Ho chi minh", "HN"=>"Ha noi", "DN"=>"Da
        nang", "HP"=>"Hai phong");
    sort($tp);
    while ($row=each($tp))
    {
            echo $row["key"]."\t";
            echo $row["value"]."<br>";
    }
?>
```



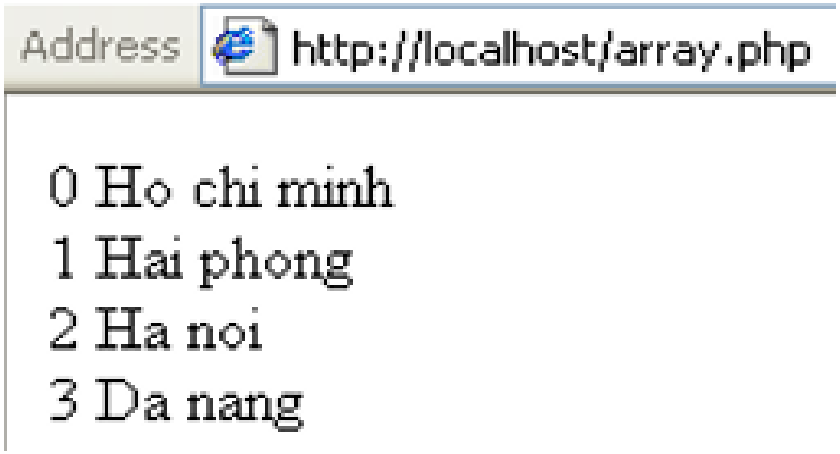
HÀM RSORT()

- Sắp xếp mảng theo chiều **giảm dần** của giá trị các phần tử mảng nhưng **chỉ số tương ứng bị thay đổi**

Cú pháp: rsort($biến_mảng)

Ví dụ:

```
<?php
    $tp=array("HCM"=>"Ho chi minh", "HN"=>"Ha noi", "DN"=>"Da
        nang", "HP"=>"Hai phong");
    rsort($tp);
    while ($row=each($tp))
    {
            echo $row["key"]."\t";
            echo $row["value"]."<br>";
    }
?>
```



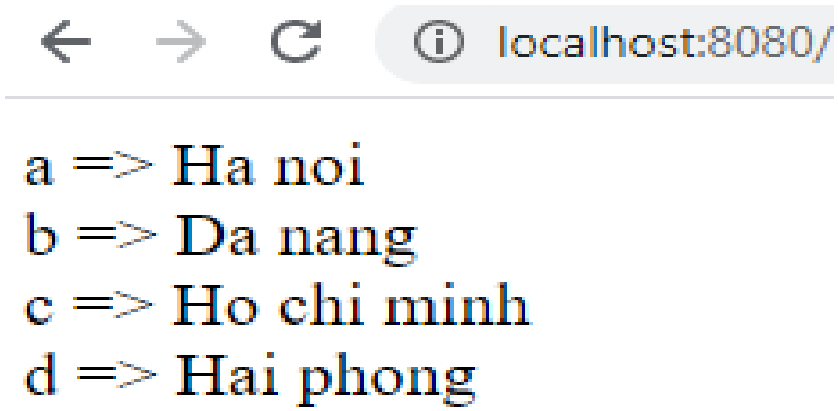
HÀM KSORT()

- Sắp xếp mảng theo **chiều tăng dần của chỉ số mảng**

Cú pháp: **ksort($biến_mảng)**

Ví dụ:

```
<?php
  $tp=array("c"=>"Ho chi minh", "a"=>"Ha noi", "b"=>"Da nang",
          "d"=>"Hai phong");
  ksort($tp);
  while ($row=each($tp))
  {
     echo $row["key"]." => ";
     echo $row["value"]."<br>";
  }
?>
```



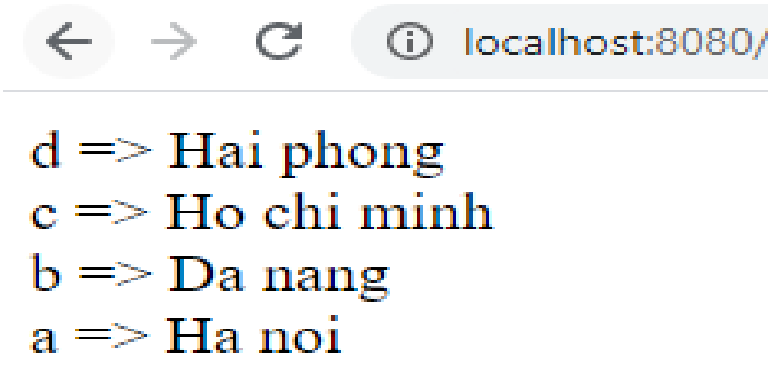
HÀM KRSORT()

- Sắp xếp mảng theo **chiều giảm dần của chỉ số mảng**

Cú pháp: **krsort($biến_mảng)**

Ví dụ:

```
<?php
  $tp=array("c"=>"Ho chi minh", "a"=>"Ha noi", "b"=>"Da nang",
            "d"=>"Hai phong");
  krsort($tp);
  while ($row=each($tp))
  {
     echo $row["key"]." => ";
     echo $row["value"]."<br>";
  }
?>
```



MẢNG 2 CHIỀU

- Khi mỗi phần tử của mảng được biểu diễn là mảng một chiều thì mảng đó được gọi là mảng hai chiều.
- Các thao tác trên mảng hai chiều tương tự với mảng một chiều

Khai báo mảng có chỉ số kiểu number

```
$biến_mảng=array(
        array(các giá trị mảng 1),
        array(các giá trị mảng 2),
        ….
);
```

Tên xe	Chổ	Đã bán
BMW	15	20
Toyota	24	30
Camry	4	10
Mazda	7	5

Khai báo mảng có chỉ số kiểu associative

```
$biến_mảng=array(
"chỉ số h1"=>array("chỉ số c1"=>giá trị, "chỉ số c2"=>giá trị 2, …),
"chỉ số h2"=>array("chỉ số c1"=>giá trị, "chỉ số c2"=>giá trị, …), …
);
```

MẢNG 2 CHIỀU

Ví dụ: Khai báo mảng hai chiều chỉ số kiểu **number** của bảng trên.

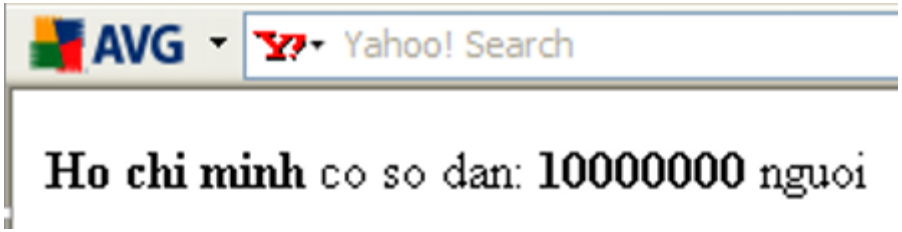
```
<?php
    $xehoi = array
     (
       //Cột 0 – Cột 1 – Cột 2
      array("BMW", "15", "20"), //hàng 0
      array("Toyota", "24", "30"), //hàng 1
      array("Camry", "4", "10"), hàng 2
      array("Mazda", "7", "5") //hàng 3
     );

    echo $xehoi[0][0].", loại ".$xehoi[0][1]." chổ đã bán ".$xehoi[0][2].". chiếc<br>";
    echo $xehoi[1][0].", loại ".$xehoi[1][1]." chổ đã bán ".$xehoi[1][2].". chiếc<br>";
    echo $xehoi[2][0].", loại ".$xehoi[2][1]." chổ đã bán ".$xehoi[2][2].". chiếc<br>";
    echo $xehoi[3][0].", loại ".$xehoi[3][1]." chổ đã bán ".$xehoi[3][2].". chiếc<br>";
?>
```

MẢNG 2 CHIỀU

Ví dụ: Khai báo mảng hai chiều chỉ số kiểu **associative**.

```
<?php
$tpho=array(
    "HCM"=>array("Ten"=>"Ho chi minh", "dan so"=>10000000),
    "HN"=>array("Ten"=>"Ha noi","dan so"=>6000000),
    "DN"=>array("Ten"=>"Da nang"),
    "HP"=>array("Ten"=>"Hai phong"));
echo "<b>".$tpho["HCM"]["Ten"]." </b>co so dan:
    <b>".$tpho["HCM"]["dan so"]."</b> nguoi <br>";
?>
```



CÁC HÀM XỬ LÝ TRÊN MẢNG

- Hàm kiểm tra sự tồn tại của mảng
- Hàm di chuyển trên các phần tử
- Hàm trả về kích thước mảng
- Hàm tìm kiếm trên mảng
- Hàm thêm hoặc xóa phần tử mảng

HÀM KIỂM TRA SỰ TỒN TẠI CỦA MẢNG

- Dùng hàm **is_array()** để kiểm tra một biến mảng nào đó có tồn tại hay không.
- Cú pháp: **is_array($biến_mảng);**

Ví dụ:

```
$states = array("Florida");
$state = "Ohio";
printf("\$states is an array: %s <br />", (is_array($states) ? "TRUE" : "FALSE"));
printf("\$state is an array: %s <br />", (is_array($state) ? "TRUE" : "FALSE"));
```

Executing this example produces the following:

```
$states is an array: TRUE
$state is an array: FALSE
```

HÀM DI CHUYỂN TRÊN CÁC PHẦN TỬ

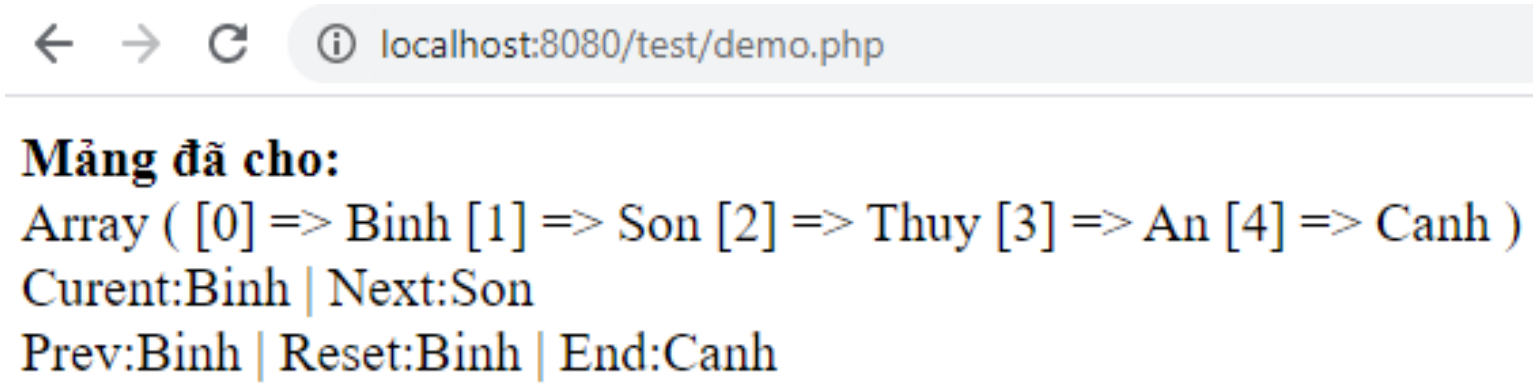
❖**Các hàm thường dùng để di chuyển con trỏ trên các phần tử mảng**

- **Hàm current():** trả về giá trị của phần tử hiện tại trong mảng.
- **Hàm next():** trả về giá trị của phần tử kế tiếp trong mảng.
- **Hàm prev():** trả về giá trị của phần tử trước phần tử hiện tại.
- **Hàm reset():** chuyển con trỏ mảng về đầu mảng và trả về giá trị của phần tử đầu tiên của mảng.
- **Hàm end():** chuyển con trỏ mảng về cuối mảng và trả về giá trị của phần tử cuối cùng của mảng.

HÀM DI CHUYỂN TRÊN CÁC PHẦN TỬ

Ví dụ: các hàm di chuyển con trỏ trên các phần tử trong mảng.

```php
<?php
    $list = ["Binh","Son","Thuy","An","Canh"];
    echo "<b>Mảng đã cho: </b><br>";
    print_r($list);
    echo "<br>";
    echo "Curent:".current($list)." | "."Next:".next($list)."<br>";
    echo "Prev:".prev($list)." | "."Reset:".reset($list)." | "."End:".end($list)."<br>";
?>
```



HÀM TRẢ VỀ KÍCH THƯỚC MẢNG

- Có thể sử dụng hàm **count()** hoặc hàm **sizeof()** để đếm tổng số phần tử có trong mảng.
- Cú pháp: **count($biến_mảng) hoặc sizeof($biến_mảng)**

Ví dụ:

```
<?php
    $people = array("Binh", "Thuy", "An",
    "Son", "Hung");
    $result1 = count($people);
    $result2 = sizeof($people);
    echo $result1; echo $result2;
?>
```

HÀM TÌM KIẾM TRÊN MẢNG

- Hàm in_array()
- Hàm array_key_exists()
- Hàm array_search()

HÀM IN_ARRAY()

- **Tác dụng:** Tìm xem một giá trị nào đó có trong mảng hay không? Nếu có trả về **true**, ngược lại trả về **false**.
- Cú pháp: **in_array(“giá trị cần tìm”, $biến_mảng, type)**

Trong đó:

type là tham số nhận 2 giá trị **true** hoặc **false**. Nếu thiết lập là **true** thì tìm kiếm có phân biệt xâu, số. Và mặc định là **false**.

Ví dụ: xét đoạn mã:

```
<?php
$list = array("Binh", "Son", "Thuy", "An", "Canh");
if (in_array("An",$list))
  echo "Match found";
else
  echo "Match not found";
?>
```

HÀM ARRAY_KEY_EXISTS()

- **Tác dụng:** Kiểm tra xem một khóa nào đó có trong mảng hay không? Trả về true nếu tìm thấy, ngược lại trả về false.
- Cú pháp: **array_key_exists(“giá trị khóa”, $biến_mảng)**

Ví dụ: Xét đoạn mã sau:

```
<?php
        $a=array("a"=>"Dog","b"=>"Cat");
        if (array_key_exists("a",$a))
          echo "Key exists!";
        else
          echo "Key does not exist!";
?>
```

HÀM ARRAY_SEARCH()

- **Tác dụng:** Kiểm tra xem một giá trị nào đó có trong mảng hay không? Trả về khóa tương ứng nếu tìm thấy.
- **Cú pháp: array_search(“giá trị cần tìm”, $biến_mảng, type)**

Ví dụ: xét đoạn mã sau:

```
<?php
  $a=array("a"=>"Dog","b"=>"Cat","c"=>"Horse");
  echo array_search("Dog",$a);

  $b=array("a"=>"5","b"=>5,"c"=>“15");
  echo array_search(5,$b,true);
?>
```

HÀM THÊM HOẶC XÓA PHẦN TỬ MẢNG

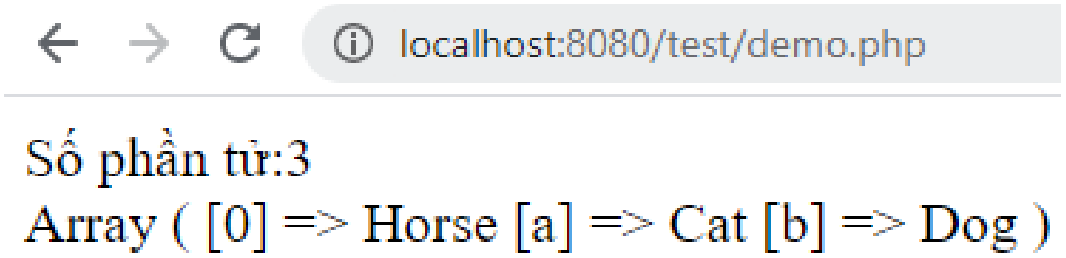
- Hàm array_unshift()
- Hàm array_push()
- Hàm array_shift()
- Hàm array_pop()

HÀM ARRAY_UNSHIFT()

- **Tác dụng:** thêm các phần tử vào đầu mảng. Giá trị trả về của hàm là số phần tử của mảng sau khi thêm.
- **Cú pháp: array_unshift($biến_mảng, giá trị 1, …)**

Ví dụ: xét đoạn mã sau:

```
<?php
        $a=array("a"=>"Cat","b"=>"Dog");
        echo "Số phần tử:".array_unshift($a,"Horse")."<br>";
        print_r($a);
?>
```

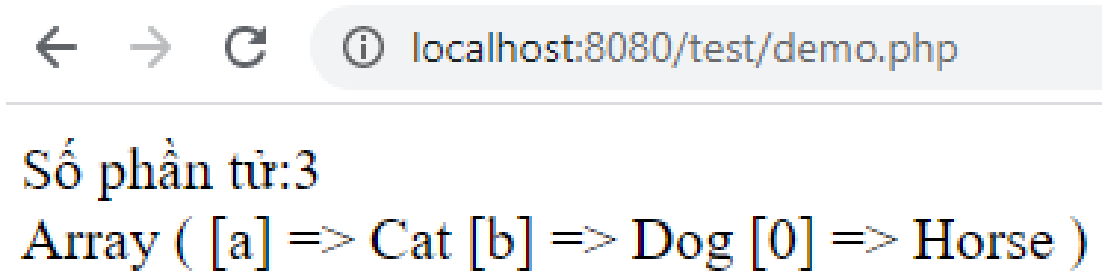


HÀM ARRAY_PUSH()

- **Tác dụng:** giống như hàm **array_unshift()** nhưng lại thêm vào cuối mảng.
- **Cú pháp: array_push($biến_mảng, giá trị 1, giá trị 2, …)**

Ví dụ: xét đoạn mã sau:

```
<?php
        $a=array("a"=>"Cat","b"=>"Dog");
        echo "Số phần tử:" . array_push($a,"Horse") . "<br>";
        print_r($a);
?>
```

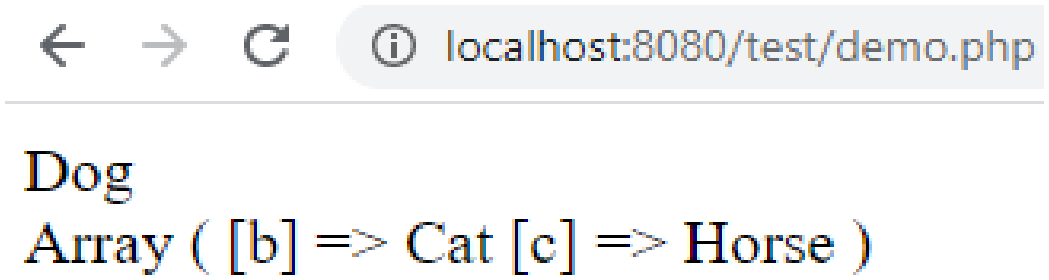


HÀM ARRAY_SHIFT()

- **Tác dụng:** loại bỏ phần tử đầu tiên của mảng. Kết quả trả về của hàm là giá trị phần tử vừa bị loại bỏ.
- **Cú pháp: array_shift($biến_mảng)**

Ví dụ: xét đoạn mã sau:

```
<?php
        $a=array("a"=>"Dog","b"=>"Cat","c"=>"Horse");
        echo array_shift($a) . "<br>";
        print_r($a);
?>
```

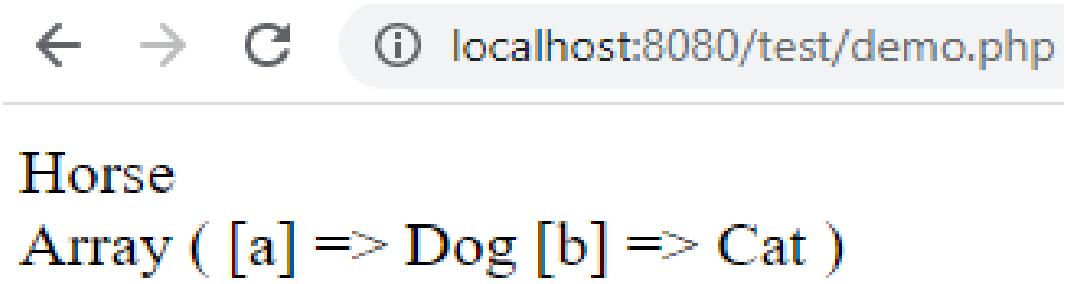


HÀM ARRAY_POP()

- **Tác dụng:** giống như **array_shift** nhưng loại bỏ phần tử cuối cùng.
- **Cú pháp:** **array_pop($biến_mảng)**

Ví dụ: xét đoạn mã sau:

```
<?php
   $a=array("a"=>"Dog","b"=>"Cat","c"=>"Horse");
   echo array_pop($a) . "<br>";
   print_r($a);
?>
```



CHUỖI KÝ TỰ

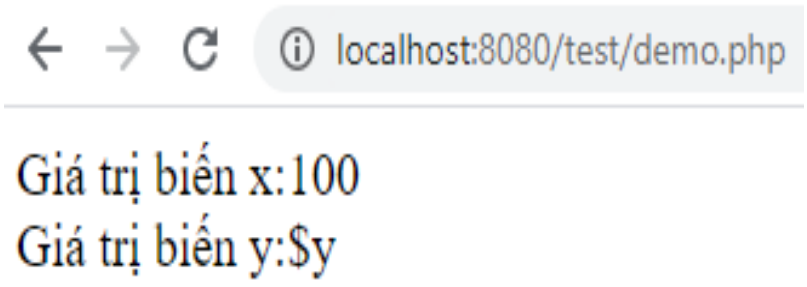
- Chuỗi trong PHP là một chuỗi các ký tự, mỗi ký tự chiếm 1 byte.

Khai báo chuỗi:

- Trong PHP, chuỗi ký tự được khai báo theo 2 cách
  - Dấu nháy đơn (single quote) . **Ví dụ:** $st1 = 'Hello Php String ';
  - Dấu nháy kép (double quote). **Ví dụ:** $st2 = “Hello Php String “;

=> **Chú ý:** chuỗi sử dụng dấu nháy kép "..." tự động chèn giá trị của biến chứa bên trong khai báo chuỗi.

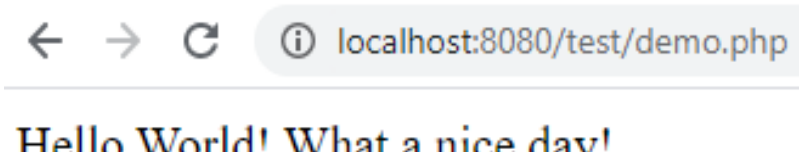
```php
<?php
  $x = 100;
  $y = 200;
  echo "Giá trị biến x:$x <br>";
  echo 'Giá trị biến y:$y <br>';
?>
```



LÀM VIỆC VỚI CHUỖI

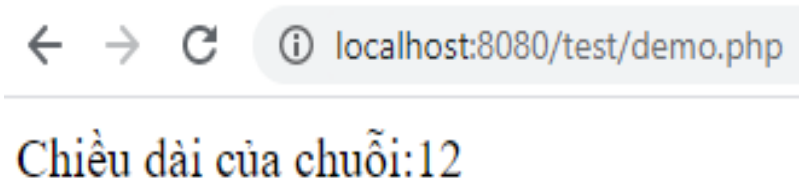
- **Toán tử trên chuỗi:** toán tử **“.”** sử dụng cộng chuỗi.

```php
<?php
  $txt1="Hello World!";
  $txt2="What a nice day!";
  echo $txt1 . " " . $txt2;
?>
```



- **Lấy chiều dài của chuỗi:** sử dụng hàm **strlen()**.

```php
<?php
  $txt1="Hello World!";
  echo "Chiều dài của chuỗi:" . strlen($txt1);
?>
```

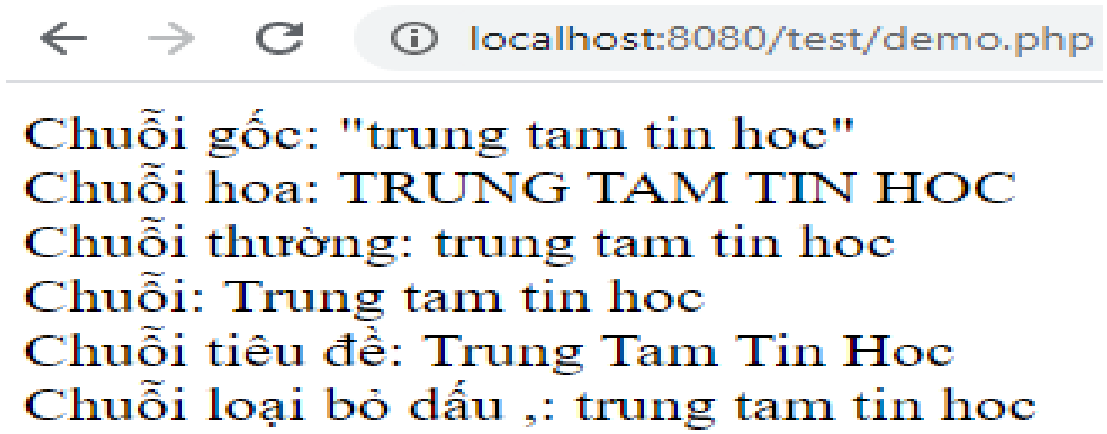


HÀM ĐỊNH DẠNG CHUỖI

- **strtoupper($string)**:chuyển tất cả các ký tự trong chuỗi thành chữ hoa
- **strtolower(($string)**:chuyển tất cả các ký tự trong chuỗi thành chữ thường
- **ucfirst(($string)**:chuyển ký tự đầu tiên trong chuỗi thành chữ hoa.
- **ucwords(($string)**:chuyển các ký tự đầu của mỗi từ trong chuỗi thành chữ hoa
- **trim(($string,$kytulietke):**xóa các ký tự liệt kê nằm ở đầu và cuối chuỗi, nếu không có thành phần **$kytulietke** thì mặc định hiểu xóa khoảng trắng.

HÀM ĐỊNH DẠNG CHUỖI

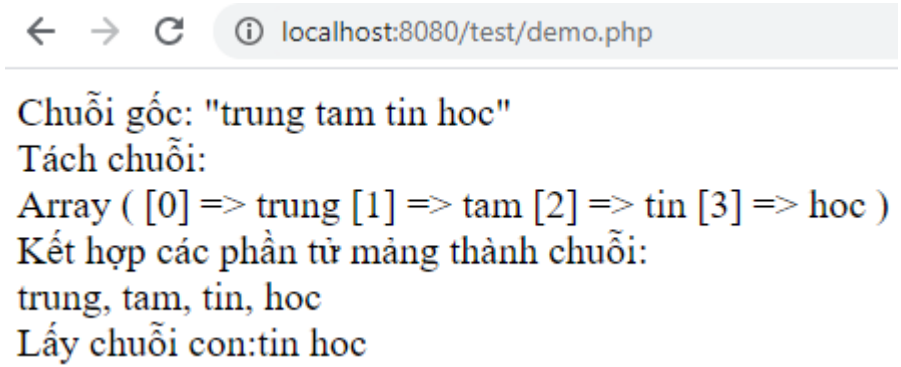
```php
<?php
  $st = "trung tam tin hoc";
  echo "Chuỗi gốc: \"$st\"<br>";
  echo "Chuỗi hoa: ".strtoupper($st)."<br>";
  echo "Chuỗi thường: ".strtolower($st)."<br>";
  echo "Chuỗi: ".ucfirst($st)."<br>";
  echo "Chuỗi tiêu đề: ".ucwords($st)."<br>";
  $temp = ",trung tam tin hoc,";
  echo "Chuỗi loại bỏ dấu ,: ".trim($temp,",");
?>
```



HÀM TÁCH HAY KẾT HỢP CHUỖI

- **explode(str, biến_chuỗi)**: Tách chuỗi thành mảng các chuỗi con, sử dụng str làm chuỗi xác định cách tách.
- **implode(str, $mang)**: Kết hợp các phần tử của mảng thành chuỗi, lấy str làm chuỗi liên kết.
- **substr(biến_chuỗi, k [, n])**: Trả về chuỗi con từ biến chuỗi, tại vị trí k, lấy n ký tự.

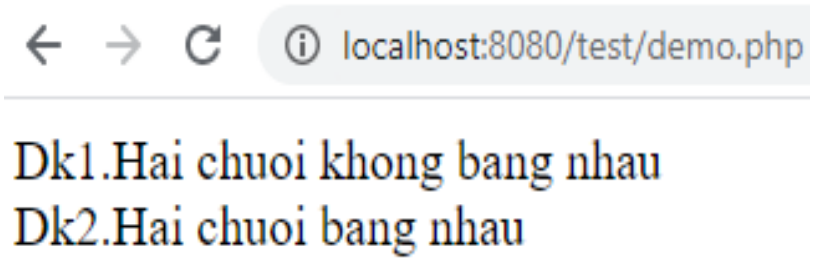
```php
<?php
  $st = "trung tam tin hoc";
  echo "Chuỗi gốc: \"$st\"<br>";
  $arr = explode(" ",$st);
  echo "Tách chuỗi:<br>";
  print_r($arr);
  echo "<br>";
  echo "Kết hợp các phần tử mảng thành chuỗi:<br>";
  $temp = implode(", ",$arr);
  echo "$temp<br>";
  echo "Lấy chuỗi con:" . substr($st,10,7);
?>
```



HÀM SO SÁNH CHUỖI

- **(int) strcmp (str1, str2)**: 0 nếu str1 == str2, n nếu str1>str2, -n nếu str1<str2 (phân biệt chữ hoa, chữ thường, n là số ngẫu nhiên)
- **(int) strcasecmp (str1, str2)**: 0 nếu str1 == str2, n nếu str1>str2, -n nếu str1<str2 (không phân biệt chữ hoa, chữ thường, n là một giá trị ngẫu nhiên)

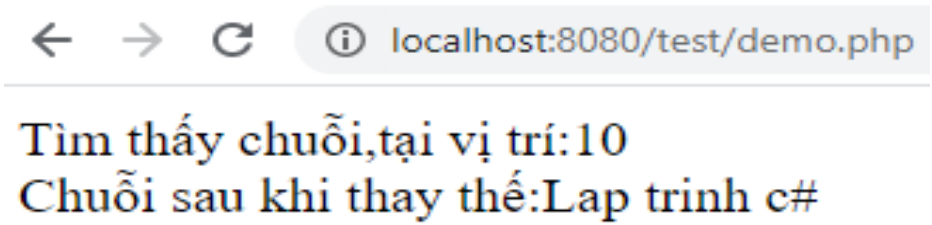
```php
<?php
  $st1 = "Lap trinh php";
  $st2 = "Lap trinh PHP";
  if(strcmp($st1,$st2) == 0)
    echo "Dk1.Hai chuoi bang nhau<br>";
  else
    echo "Dk1.Hai chuoi khong bang nhau<br>";
  if(strcasecmp($st1,$st2) == 0)
    echo "Dk2.Hai chuoi bang nhau<br>";
  else
    echo "Dk2.Hai chuoi khong bang nhau<br>";
?>
```



HÀM TÌM KIẾM VÀ THAY THẾ CHUỖI

- **strpos(biến_chuỗi,str)**: trả về vị trí chuỗi con str xuất hiện đầu tiên trong biến_chuỗi, nếu không tìm thấy trả về false.
- **str_replace(str1, str2, biến_chuỗi)**: thay chuỗi con str1 bằng str2 trong biến_chuỗi.

```php
<?php
  $st = "Lap trinh php";
  $find = "php";
  if(strpos($st,$find) > 0)
    echo "Tìm thấy chuỗi,tại vị trí:".strpos($st,$find)."<br>";
  else
    echo "Không tim thấy chuỗi<br>";
  $replace = "c#";
  echo "Chuỗi sau khi thay thế:".str_replace($find,$replace,$st);
?>
```



BÀI THỰC HÀNH

1. Mảng

- Viết chương trình sử dụng hàm **rand()** (đưa ra số interger ngẫu nhiên) để nhập dữ liệu cho mảng có độ dài n, với n được gán sẵn, rồi in các số đó ra màn hình.
- Sắp xếp các số đó theo thứ tự tăng dần rồi lại in ra màn hình.
- Sắp xếp các số đó theo thứ tự giảm dần rồi lại in ra màn hình.

BÀI THỰC HÀNH

2. Mảng

- Tạo mảng 1 chiều có các giá trị 1, 5, 7, 2, 6, 10, 13, 15, 28, 30, 1
- Đếm và in danh sách các số chẵn
- Đếm và in danh sách các số lẻ
- Tính tổng các số chia hết cho 3
- Tính tích các số không chia hết cho 3
- In số lớn nhất và số nhỏ nhất
- Sắp xếp mảng theo thứ tự tăng dần và in kết quả
- In ra vị trí các phần tử trong mảng có giá trị chia hết cho 2.

BÀI THỰC HÀNH

3. Mảng hai chiều

- Tạo mảng hai chiều quản lý danh sách sinh viên trong trường ĐH bao gồm các thông tin: Tên, Ngày sinh, Giới tính.
- In danh sách.

Q & A

THANK YOU!