

# Phát Triển Phần Mềm Mã Nguồn Mở

## MẢNG, CHUỖI & HÀM TRONG PHP

LÊ THỊ BÍCH HẰNG  
NGUYỄN HẢI TRIỀU

Bộ môn Kỹ thuật phần mềm  
Khoa Công nghệ thông tin  
ĐH Nha Trang

# Mục lục

- 1 Ôn tập mảng trong PHP
- 2 Mảng nhiều chiều
- 3 Chuỗi (String)
- 4 Sessions/Cookies

# 1 Ôn tập mảng trong PHP

2 Mảng nhiều chiều

3 Chuỗi (String)

4 Sessions/Cookies

# Ôn tập mảng trong PHP

- Là một tập các phần tử có cùng tính chất
- Mỗi phần tử có thể có: Cùng kiểu dữ liệu hoặc Khác kiểu dữ liệu
- Một mảng gồm 2 phần thông tin
  - ▶ Thông tin chỉ mục (khóa)
  - ▶ Thông tin giá trị tương ứng với chỉ mục
- Mảng đánh chỉ mục từ 0
- Tạo mảng từ một mảng có sẵn  
`$Tên_Mảng_Mới=$Tên_Mảng_Cũ`

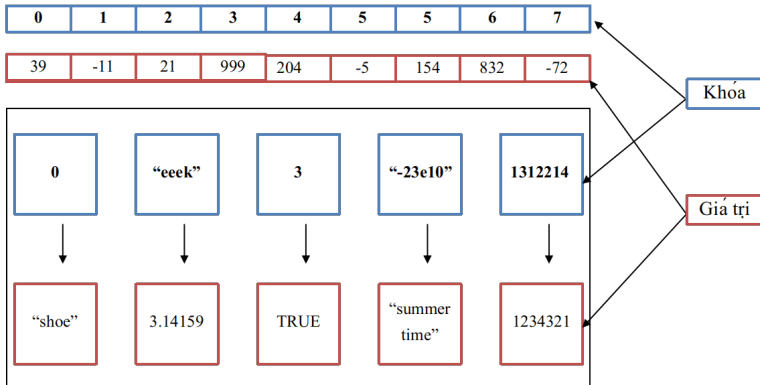
# Ôn tập mảng trong PHP

Ví dụ

Kiểu mảng

C/C++

Java/C#



Kiểu mảng  
PHP

# Ôn tập mảng trong PHP

## Khai báo mảng

- Không cần khai báo số lượng phần tử
- Không cần phải khai báo mảng trước khi sử dụng

## Khai báo tường minh thông qua `array()`. Ví dụ:

```
$colors = array("red", "green", "blue", "yellow");
```

```
$number = array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
```

```
$odddnumber=array(1, 3, 5, 7, 9);
```

## Khai báo không tường minh. Ví dụ:

```
$colors[] = "red";
```

```
$colors[] = "green";
```

```
$colors[] = "blue";
```

```
$odddnumber = range(1, 10, 2);
```

# Ôn tập mảng trong PHP

**Thêm một phần tử vào mảng:** `$tên_mảng[khóa] = giá trị`

- Phần tử luôn được **thêm vào cuối mảng**
- Nếu khóa đã tồn tại thì không có phần tử nào được thêm

**Xóa một phần tử khỏi mảng:** `unset($tên_mảng[khóa])`

**Đếm số phần tử của mảng:** `count($tên_mảng)`

**Rút trích nội dung:**

- Sử dụng hàm `list(items)`. Ví dụ:  

```
$prod = array("redz", "1", 77, "blue");
list($name, $id, $quantity, $color) = $prod;
// $name="redz", $id="1", $quantity="77", $color="blue"
```
- Sử dụng hàm `extract(array)`  

```
$prod = array("name"=>"redz", "id"=>"1",
"quantity"=>77, "color"=>"blue");
extract($prod);
echo "n=$name; id=$id; quan=$quantity; color=$color";
// n=redz; id=1; quan=77; color=red
```

# Ôn tập mảng trong PHP

## Các hàm xử lý mảng

- Nối ghép hai mảng

- ▶ `array_merge($mảng1, $mảng2)`
- ▶ Ghép tương ứng **Key, Value**: `array_combine($mảng1, $mảng2)`

```
<?php
$arr1=array(54,"a"); $arr2=array("b",100);
$arr=array_merge($arr1,$arr2);
foreach ($arr as $value){
    echo $value." ";} //->54 a b 100
?>
```

- Tìm kiếm

- ▶ `array_search($giá_trị, $mảng)`

```
<?php
$arr=array(54,2,3,4,5,6);
echo array_search(54,$arr); //->0
?>
```



- 1 Ôn tập mảng trong PHP
- 2 Mảng nhiều chiều**
- 3 Chuỗi (String)
- 4 Sessions/Cookies

# Mảng nhiều chiều

- Mảng nhiều chiều là mảng chứa **một** hoặc **nhiều mảng** khác.
- Ví dụ:

```
<?php
$movies = array("film1"=> array(
    "title" => "Rear_Window",
    "director" => "Alfred_Hitchcock",
    "year" => 1954),
    "film2"=> array(
    "title" => "Full_Metal_Jacket",
    "director" => "Stanley_Kubrick",
    "year" => 1987),
    "film3"=> array(
    "title" => "Mean_Streets",
    "director" => "Martin_Scorsese",
    "year" => 1973));
?>
```

# Mảng 2 chiều

## Ví dụ duyệt mảng 2 chiều

```
<?php
$superheroes = array(
    array("Peter_Parker", "peterparker@mail.com"),
    array("Clark_Kent", "clarkkent@mail.com"),
    array("Harry_Potter", "harrypotter@mail.com"),
    array("Steve_Rogers", "steverogers@mail.com")
);
for ($i=0;$i<count($superheroes);$i++){
    echo "<ul>";
    for ($j=0;$j<count($superheroes[$i]);$j++){
        echo "<li>". $superheroes[$i][$j]. "</li>";
    }
    echo "</ul>";
}
```

# Mảng 2 chiều

**Bài tập:** cho mảng hai chiều \$superheroes

```
<?php
$superheroes = array(
    "spider-man" => array(
        "name" => "Peter_Parker",
        "email" => "peterparker@mail.com"),
    "super-man" => array(
        "name" => "Clark_Kent",
        "email" => "clarkkent@mail.com"),
    "iron-man" => array(
        "name" => "Tony_Stark",
        "email" => "tonystark@mail.com")
);
?>
```

Hãy in ra **Keys & Value** của mảng trên.

## Mảng 2 chiều

Kết quả của mảng trên được hiện như sau.

spider-man:

name: Peter Parker

email: peterparker@mail.com

super-man:

name: Clark Kent

email: clarkkent@mail.com

iron-man:

name: Tony Stark

email: tonystark@mail.com

- 1 Ôn tập mảng trong PHP
- 2 Mảng nhiều chiều
- 3 Chuỗi (String)**
- 4 Sessions/Cookies

- Một chuỗi là một **dãy các ký tự**
- Một ký tự giống như một byte
- PHP **không** giới hạn kích thước kiểu chuỗi, nó chỉ phụ thuộc vào bộ nhớ mà PHP đang chạy.
- Các cách để biểu diễn một chuỗi:
  - ▶ Dùng dấu **ngoặc đơn** (`'`). Đây là cách đơn giản nhất để thể hiện chuỗi, đặt chuỗi cần hiện trong dấu ngoặc đơn.
    - **Các biến** trong dấu ngoặc đơn và **các ký tự đặc biệt** cho xuống dòng sẽ **không được** PHP biên dịch trong dấu ngoặc đơn.
  - ▶ Dùng dấu **ngoặc kép** (`"`)
    - PHP sẽ **biên dịch các biến** và **các ký tự đặc biệt** ở trong dấu ngoặc kép.

# Các xử lý cơ bản

- `trim($chuỗi)`//Loại bỏ khoảng trắng (hoặc các ký tự khác) ở đầu và cuối chuỗi.
- `ltrim($chuỗi)`//Loại bỏ khoảng trắng (hoặc các ký tự khác) ở đầu chuỗi
- `rtrim($chuỗi)`//Loại bỏ khoảng trắng (hoặc các ký tự khác) ở cuối chuỗi
- `strlen($chuỗi)`://trả về độ dài của một chuỗi.
- `substr($chuỗi, $vị trí, $chiều_dài)`://Trả về một phần của chuỗi.
- `strtoupper($chuỗi)`
- `strtolower($chuỗi)`
- `ucfirst($chuỗi)`; //đổi ký tự đầu tiên thành chữ hoa
- `ucwords($chuỗi)`;//đổi ký tự đầu mỗi từ thành chữ hoa
- `str_word_count()`;//đếm số lượng từ trong một chuỗi
- `str_replace()` ;//thay thế một vài kí tự trong chuỗi bằng các kí tự khác



# Ví dụ

```
<?php
$str="abcxyz";
echo strlen($str); //->6
?>
```

```
<?php
$str="   abcxyz";
echo trim($str); //->abcxyz
?>
```

```
<?php echo substr($str,2,3);?> //->cxy
```

```
<?php echo str_word_count($str);?> //->1
```

```
<?php echo str_replace("world", "Dolly", "Hello_world!"); ?>
//->Hello Dolly!
```

# Các xử lý cơ bản

- Tìm kiếm

- ▶ `strpos($chuỗi, $chuỗi_con, [$vi_trí_bắt_đầu])`: tìm kiếm vị trí một text cụ thể trong một chuỗi, trả ra vị trí của kí tự đầu tiên tìm thấy.
- ▶ `strrpos($chuỗi, $chuỗi_con)`: Tìm vị trí của lần xuất hiện cuối cùng của một text cụ thể trong một chuỗi khác (phân biệt chữ hoa chữ thường).

- So sánh

- ▶ `strcmp($chuỗi_1, $chuỗi_2)`: So sánh hai chuỗi (phân biệt chữ hoa chữ thường)
- ▶ `strncmp($chuỗi_1, $chuỗi_2, $chiều_dài)`: So sánh chuỗi của  $n$  ký tự đầu tiên (phân biệt chữ hoa chữ thường)
- ▶ `strcasecmp($chuỗi_1, $chuỗi_2)`
- ▶ `strncasecmp($chuỗi_1, $chuỗi_2, $chiều_dài)`
- ▶ `strnatcmp($chuỗi_1, $chuỗi_2)`
- ▶ `strnatcasecmp($chuỗi_1, $chuỗi_2)`

# Các xử lý cơ bản

- Tách hay kết hợp chuỗi

- ▶ `explode(separator,string,limit)`: tách một chuỗi thành một mảng

```
<?php
$str="a,b,c";
var_dump(explode(",",$str)); // array("a","b","c")
?>
```

- ▶ `implode()`: Trả về một chuỗi bằng cách nối các phần tử của một mảng với một chuỗi được chỉ định.

```
<?php
$arr = array('Hello','World!');
echo implode("-", $arr); //->Hello-World!
?>
```

# Lưu ý

- Không nên dùng toán tử Equal (==) mà nên dùng toán tử Identical (===). Bởi vì == không so sánh kiểu dữ liệu và có thể dẫn đến sai lệch trong việc so sánh.
- Dùng toán tử so sánh === để so sánh 2 chuỗi

```
<?php echo ('1e3' == '1000') ? 'true' : 'false'; ?>  
//return true  
<?php echo ('1e3' === '1000') ? 'true' : 'false'; ?>  
//return false
```

# Bài tập

Thiết kế Form nhập và tính tích các số trên dãy số, các số ngăn cách nhau bởi dấu phẩy (,). Yêu cầu:

- Sử dụng điều khiển textfield, button
- Textfield *Tích dãy số* **không** cho phép nhập liệu và chỉnh sửa
- Khi chọn button *Tích dãy số*, thực hiện:
  - ▶ Tách chuỗi và gán vào mảng
  - ▶ Tính tích các phần tử của mảng và xuất tích các số ra textfield *Tích dãy số*

- 1 Ôn tập mảng trong PHP
- 2 Mảng nhiều chiều
- 3 Chuỗi (String)
- 4 Sessions/Cookies**

**Sessions/Cookies** dùng để lưu trữ thông tin tạm thời như users, thông tin hệ thống ...

- Sessions: `session_start()`, `session_destroy()`, `session_unset()`
- Cookies: `setcookie()`

# Cookies

- cookie là một tập tin nhỏ mà máy chủ nhúng vào máy tính của end-user thông qua trình duyệt
- thường có chức năng xác định thông tin người dùng thông qua thông tin lưu trữ trong file
- khi sử dụng cùng một máy tính, trình duyệt web đã lưu cookie gửi request đến một server thì nó sẽ gửi kèm theo cookie



# Cookies

## Hàm tạo cookie

```
setcookie(name, value, expire, path, domain, secure, httponly);
```

Trong đó, tham số **name** bắt buộc phải có. Ví dụ:

```
<?php
$cookie_name = "user";
$cookie_value = "John_Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/",
); // 86400 = 1 day
?>

<html>
<body>
<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie_named_" . $cookie_name . "'_is_not_set!";
} else {
    echo "Cookie_" . $cookie_name . "'_is_set!<br>";
    echo "Value_is:" . $_COOKIE[$cookie_name];
}??>
</body>
</html>
```

# Bài tập Cookies

Hãy tạo ra một form đăng nhập/login như hình:

Login Form	
User	<input type="text"/>
pass	<input type="password"/>
<input type="button" value="Login"/>	

Thực hiện các yêu cầu về cookie như sau:

- Khi nhấn **Login**, thì lưu thông tin **user** vào file cookie. Trang web sẽ thông báo: **Hi user . Welcome to my site!**, với user là thông tin mà người dùng nhập từ form.
- Tính năng đăng nhập tự động: sau khi đã lưu trữ thông tin đăng nhập vào cookie và tắt web browser, tiến hành mở và truy cập lại trang web cũ thì phải xuất hiện thông báo: **Hi user . Welcome to my site!**
- Kiểm tra điều kiện đã nhập user, password chưa. Nếu chưa thì xuất thông báo nhập lại.

# Sessions

Cũng giống như cookies, sessions cũng được dùng để lưu trữ thông tin. Session sẽ **giữ thông tin người dùng cuối gần nhất khi đóng trình duyệt** và có giá trị cho tất cả các trang trong một trình duyệt web. Tuy nhiên session khác với cookie ở chỗ, session **không lưu thông tin ở máy người dùng**.

- Các biến Sessions **`$_SESSION`** là các biến php toàn cục
- Bắt đầu Session với hàm **`session_start()`**
- Ví dụ:
  - ▶ **`$_SESSION['username']=$_POST['user'];`**

# Ví dụ Sessions

## session\_start()

```
//demo_session1.php
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>
</body>
</html>
```

# Ví dụ Sessions

Trang `demo_session2.php` sẽ truy cập thông tin từ trang `demo_session1.php` ở trên

```
//demo_session2.php
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
//print_r($_SESSION); //show all the session variable values for
    a user session
// Echo session variables that were set on previous page
echo "Favorite_color_is_" . $_SESSION["favcolor"] . "<br>";
echo "Favorite_animal_is_" . $_SESSION["favanimal"] . ".";
?>
</body>
</html>
```

# Xóa/Hủy Sessions

Để xóa toàn bộ các biến toàn cục session, sử dụng `session_unset()`.  
Để hủy một session, sử dụng `session_destroy()`

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
// remove all session variables
session_unset();
// destroy the session
session_destroy();
?>
</body>
</html>
```

## Reference

- [1] Lê Thị Bích Hằng, Bài giảng *Phát Triển Phần Mềm Mã Nguồn Mở*.
- [2] <https://www.w3schools.com/php/>