

# CSE485 – Công nghệ Web

## Bài 03: PHP Căn bản 02

[dungkt@tlu.edu.vn](mailto:dungkt@tlu.edu.vn)

# Nội dung



- 1) Mảng
- 2) Các hàm dựng sẵn cho mảng
- 3) Các hàm dựng sẵn thao tác với String
- 4) Các hàm dựng sẵn thao tác với Số
- 5) Các hàm dựng sẵn thao tác với Time

# 1. Mạng

# Mảng – Định nghĩa



- Mảng là cấu trúc dữ liệu có thể lưu trữ một hoặc nhiều kiểu giá trị tại 1 thời điểm
- Các phần tử trong mảng có thể là bất cứ kiểu dữ liệu nào.
- Bài toán: viết chương trình PHP lưu thông tin nhân viên của công ty
- Sử dụng biến:

```
$staff1 = "Nhân viên A";
```

```
$staff2 = "Nhân viên B";
```

```
...
```

- Sử dụng mảng:  

```
$staffs = array('Nhân viên A', 'Nhân viên B', 'Nhân viên C');
```
- Xem cấu trúc mảng bằng các lệnh sau: `var_dump()`, `print_r()`

# Mảng - Khai báo



- **\$arr** = array(<Danh sách các phần tử>);
- **\$arr** = [<Danh sách các phần tử>]; // sử dụng từ php version >= 5.4
- Danh sách các phần tử được ngăn cách nhau bởi dấu ,
- Ví dụ  

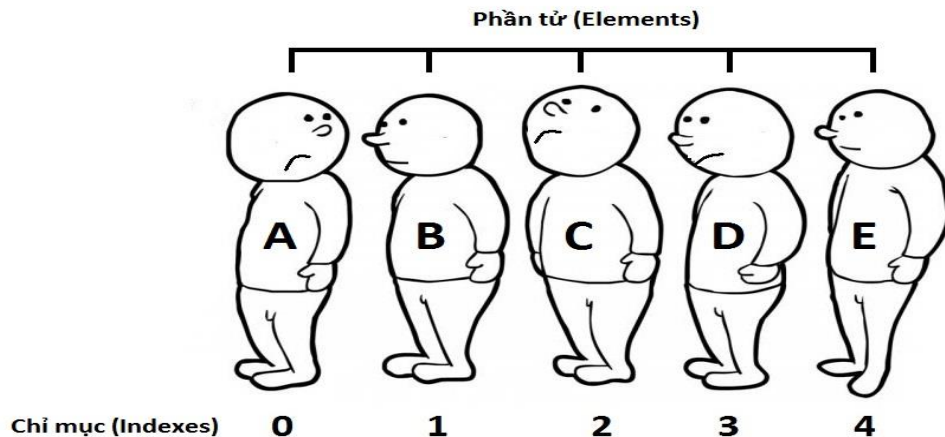
```
$staffs = array('Nhân viên A', 'Nhân viên B', 'Nhân viên C', 'Nhân viên D', 'Nhân viên E');  
$staffs = ['Nhân viên A', 'Nhân viên B', 'Nhân viên C', 'Nhân viên D', 'Nhân viên E'];  
$mangs = [0, 3, 6, 9];
```
- Cú pháp viết tắt khi viết lồng với HTML  

```
<?php foreach(<biểu-thức-lặp>): ?>  
<?php endforeach;?>
```

# Mảng - Key của mảng



- Là giá trị dùng để xác định vị trí của phần tử trong mảng
- Các giá trị của từng phần tử mảng được truy xuất thông qua key này
- Phần tử đầu tiên trong mảng luôn có key = 0



# Mảng – Vòng lặp For ... Each



- Vòng lặp chuyên sử dụng để lặp các phần tử trong mảng
- Ngoài ra cũng có thể sử dụng các vòng lặp for, while, do...while để lặp mảng
- Cú pháp

```
foreach ($array as $key => $value){  
    //code-logic;  
}
```

```
foreach ($array as $value){  
    //code-logic;  
}
```

- Trong đó

\$array: mảng cần lặp

\$key: key của phần tử tại thời điểm lặp

\$value: giá trị tương ứng của phần tử với key là \$key

# Mảng – Vòng lặp For ... Each



- Ví dụ: in ra danh sách các nhân viên

```
$staffs = array('Nhân viên A', 'Nhân viên B', 'Nhân viên C', 'Nhân viên D', 'Nhân viên E');  
echo 'Danh sách các nhân viên và vị trí tương ứng';  
echo '<br />';  
foreach($staffs as $key => $value){  
    echo "Vị trí $key : $value";  
    echo '<br />';  
}
```

- Ngoài ra có thể lấy giá trị của phần tử trực tiếp thông qua key tương ứng:

VD: \$staff[0]; \$staff[1]; \$staff[2];



# Mảng số nguyên



- Key bắt buộc phải ở dạng số
- Phần tử đầu tiên luôn mặc định có key = 0
- Ví dụ

```
$arraysIndexes = ['phần tử thứ nhất', 2, '3'];
```

- Lấy giá trị của phần tử bằng vòng lặp foreach hoặc truy xuất trực tiếp thông qua key
- Ví dụ:

```
$element0 = $arraysIndexes[0];
```

```
$element1 = $arraysIndexes[1];
```

# Mảng kết hợp



- Khác biệt duy nhất so với mảng số nguyên là key của mảng có thêm dạng chuỗi
- Ví dụ

```
$salaries = [  
    'A' => 1000,  
    'B' => 2000,  
    'C' => 3000,  
    'D' => 4000,  
];
```

- Lấy giá trị của phần tử bằng vòng lặp foreach hoặc truy xuất trực tiếp thông qua key
- Ví dụ:

```
$elementA = $salaries['A'];  
$elementB = $salaries['B'];
```

# Mảng nhiều chiều



- Trong một mảng đa chiều, mỗi phần tử cũng có thể là một mảng. Và mỗi phần tử trong một mảng phụ cũng có thể là một mảng, và cứ tiếp tục như vậy.
- Trong mảng đa chiều, key có thể ở dạng số hoặc string
- Không nên tạo ra mảng lớn hơn 3 chiều vì rất khó quản lý

Ví dụ mảng 2 chiều:

```
$staffs = [  
    'group' => [  
        'staff1' => "Nhân viên A",  
        'staff2' => "Nhân viên B",  
    ],  
    'salary' => [  
        'staff1' => 1000,  
        'staff2' => 2000,  
    ]  
];
```

## 2. Các hàm dựng sẵn cho mảng

# Các hàm có sẵn cho thao tác với mảng



Tên hàm	Mô tả
array_sum(\$array)	Trả về tổng các giá trị của mảng \$array
array_key_exists(\$key, \$value)	Kiểm tra xem chỉ mục \$key có tồn tại trong mảng \$array hay không, trả về TRUE nếu tồn tại, ngược lại là FALSE
array_merge(\$array1, \$array2)	Gộp mảng array2 vào array1, các phần tử trong array2 sẽ thêm vào cuối mảng array1, trả về 1 mảng mới
array_search(\$keyword, \$array)	Tìm kiếm theo giá trị của mảng, trả về chỉ mục(key) của phần tử đó nếu có.
array_slice(\$array, \$begin, \$length)	Lấy các phần tử từ vị trí \$begin của mảng \$array, lấy \$length phần tử, trả về mảng mới
array_unique(\$array)	Loại bỏ các phần tử có giá trị trùng lặp, và trả về 1 mảng mới

# Các hàm có sẵn cho thao tác với mảng



Tên hàm	Mô tả
<code>array_values(\$array)</code>	Tạo ra mảng chỉ mục mới với giá trị là giá trị của mảng ban đầu, đồng thời các chỉ mục (\$key) cũng sẽ được sắp xếp lại theo thứ tự 0,1,2.v.v
<code>array_keys(\$array)</code>	Tạo ra mảng chỉ mục mới với giá trị là chỉ mục của mảng ban đầu
<code>array_pop(\$array)</code>	Bỏ đi phần tử cuối cùng của mảng ban đầu và trả về giá trị của phần tử cuối cùng đó
<code>array_push(\$array, \$var, \$var..)</code>	Thêm 1 hoặc nhiều phần tử vào cuối mảng ban đầu, trả về số lượng phần tử của mảng ban đầu sau khi thêm
<code>array_shift(\$array)</code>	Xóa phần tử đầu tiên của mảng ban đầu, trả về giá trị của phần tử vừa bị xóa
<code>array_unshift(\$array, \$var, \$var ...)</code>	Ngược lại với <code>array_shift</code> , hàm này sẽ thêm các giá trị \$var vào mảng ban đầu, và trả về số lượng phần tử của mảng ban đầu sau khi thêm

# Các hàm có sẵn cho thao tác với mảng



Tên hàm	Mô tả
<code>array_flip(\$array)</code>	Trả về 1 mảng mới với từ việc đổi chỉ mục với giá trị của mảng ban đầu, lưu ý hàm này chỉ xử lý được key/value có kiểu dữ liệu là integer hoặc string
<code>count(\$array)</code>	Đếm số phần tử của 1 mảng
<code>explode(\$character, \$string)</code>	Chuyển chuỗi \$string thành mảng dựa vào ký tự phân tách \$character
<code>end(\$array)</code>	Trả về giá trị cuối cùng của mảng \$array
<code>is_array(\$array)</code>	Kiểm tra xem mảng \$array có phải là 1 mảng hay không, trả về TRUE nếu là mảng, ngược lại là FALSE
<code>sort(\$array)</code>	Sắp xếp mảng ban đầu theo chiều tăng dần các giá trị, trả về TRUE nếu thành công => mảng bị thay đổi, FALSE nếu thất bại

# Các hàm có sẵn cho thao tác với mảng



Tên hàm	Mô tả
asort(\$array)	tương tự như sort nhưng sẽ giữ nguyên key của phần tử ở mảng ban đầu (với sort thì sẽ bị reset key), trả về TRUE nếu thành công, FALSE nếu thất bại
ksort(\$array)	sắp xếp theo chiều tăng dần theo key của mảng ban đầu, trả về TRUE nếu thành công => mảng bị thay đổi, FALSE nếu thất bại
rsort(\$array)	đảo ngược vị trí của các phần tử trong mảng ban đầu, trả về TRUE nếu thành công, FALSE nếu thất bại
implode(\$character, \$array)	convert các giá trị của mảng \$array thành string ngăn cách nhau bởi ký tự \$character
in_array(\$value, \$array)	kiểm tra xem giá trị \$value có tồn tại trong mảng \$array hay không, trả về TRUE nếu tồn tại, FALSE nếu không tồn tại
list(\$var1, \$var2, ...)	gán các biến tương ứng với các giá trị của 1 mảng



# Các hàm có sẵn cho thao tác với mảng



Tên hàm	Mô tả
<code>min(\$array)</code>	Trả về giá trị nhỏ nhất của mảng \$array
<code>max(\$array)</code>	Trả về giá trị lớn nhất của mảng \$array
<code>print_r(\$array)</code>	Xem cấu trúc mảng
<code>reset(\$array)</code>	Trả về giá trị đầu tiên của mảng \$array
<code>unset(\$var1, \$var2, ....)</code>	Xóa phần tử \$var1, \$var2, ... khỏi mảng

# 3. Các hàm dựng sẵn thao tác với String

# Các hàm có sẵn thao tác với String



Tên hàm	Mô tả
Nối chuỗi sử dụng ký tự .	<pre>\$string = "string1" . " , string2"; echo \$string; //string1, string2</pre>
strlen(\$string)	Trả về độ dài của chuỗi \$string
str_word_count(\$string, \$format, \$charlist)	<p>Đếm tổng số từ có trong chuỗi \$string</p> <p>\$format: chỉ định kiểu giá trị trả về của hàm str_word_count(). Các giá trị này có thể là:</p> <ul style="list-style-type: none"><li>0 - Mặc định - trả về số lượng từ đếm được</li><li>1 - Trả về một mảng chứa các từ trong chuỗi</li><li>2 - Trả về một mảng với key là vị trí của từ trong chuỗi và value là từ trong chuỗi</li></ul> <p>\$charlist: chỉ định các ký tự đặc biệt sẽ được xem như một từ trong chuỗi</p>

# Các hàm có sẵn thao tác với String



Tên hàm	Mô tả
strtoupper(\$string)	Chuyển tất cả các ký tự trong chuỗi \$string sang chữ hoa, trả về 1 chuỗi mới
strtolower(\$string)	Chuyển tất cả các ký tự trong chuỗi \$string sang chữ thường, trả về 1 chuỗi mới
ucfirst(\$string)	Đổi ký tự đầu tiên của chuỗi \$string sang chữ in hoa, trả về 1 chuỗi mới
ucwords(\$string)	Đổi các ký tự đầu tiên của các từ trong chuỗi \$string thành chữ in hoa

# Các hàm có sẵn thao tác với String



Tên hàm	Mô tả
<code>trim(\$string, \$character)</code>	Xoá các ký tự <code>\$character</code> ở đầu và cuối của chuỗi <code>\$string</code> , nếu không nhập <code>\$character</code> thì mặc định là khoảng trắng
<code>ltrim(\$string, \$character)</code>	Tương tự <code>trim</code> nhưng chỉ xóa bên trái
<code>rtrim(\$string, \$character)</code>	Chuyển tất cả các ký tự trong chuỗi <code>\$string</code> sang chữ hoa, trả về 1 chuỗi mới
<code>str_replace(\$search, \$replace, \$string)</code>	Tìm kiếm chuỗi <code>\$search</code> trong chuỗi <code>\$string</code> , thay thế bằng chuỗi <code>\$replace</code> , trả về 1 chuỗi mới
<code>preg_replace(\$pattern, \$replace, \$string)</code>	Tìm kiếm theo khuôn mẫu <code>\$pattern</code> trong chuỗi <code>\$string</code> , thay thế bằng chuỗi <code>\$replace</code> , trả về 1 chuỗi mới

# Các hàm có sẵn thao tác với String



Tên hàm	Mô tả
<code>substring(\$string, \$start, \$length)</code>	<p>Cắt chuỗi \$string, vị trí cắt từ \$start, và cắt \$length ký tự, trả về chuỗi mới</p> <p>\$start: nếu \$start là một số dương, chuỗi được trả về sẽ bắt đầu từ vị trí \$start(ký tự đầu tiên của chuỗi được tính là 0). Nếu \$start là một số âm chuỗi con trả về sẽ được cắt từ vị trí \$start tính từ cuối chuỗi</p> <p>\$length: độ dài của chuỗi trả về. Mặc định length được tính đến cuối chuỗi</p> <ul style="list-style-type: none"><li>• \$length&gt;0: độ dài của chuỗi trả về được tính từ đầu chuỗi</li><li>• \$length&lt;0: độ dài của chuỗi trả về được tính từ cuối chuỗi</li></ul>
<code>strstr(\$string, \$needle)</code>	<p>Tách chuỗi \$string bắt đầu từ \$needle đến hết chuỗi, trả về 1 chuỗi mới</p>

# Các hàm có sẵn thao tác với String



Tên hàm	Mô tả
<code>strpos(\$string, \$needle)</code>	Tìm vị trí xuất hiện lần đầu tiên của <code>\$needle</code> trong chuỗi <code>\$string</code> , trả về vị trí xuất hiện nếu có, và ngược lại trả về <code>FALSE</code> nếu không thấy
<code>strrev(\$string)</code>	Đảo ngược chuỗi <code>\$string</code> , trả về 1 chuỗi mới
<code>strcmp(\$string1, \$string2)</code>	So sánh hai chuỗi phân biệt chữ hoa chữ thường, hàm trả về số âm nếu chuỗi thứ nhất nhỏ hơn chuỗi thứ 2, trả về 0 nếu hai chuỗi bằng nhau và trả về số dương nếu chuỗi thứ nhất lớn hơn chuỗi thứ 2.

# 4. Các hàm dựng sẵn thao tác với Số



# Các hàm dựng sẵn thao tác với Số



Tên hàm	Mô tả
is_numeric(\$var)	Kiểm tra xem biến \$var có phải là kiểu Number hay không, trả về TRUE nếu phải, ngược lại là FALSE
is_int(\$var)	Kiểm tra xem biến \$var có phải kiểu integer hay không, trả về TRUE nếu phải, ngược lại là FALSE
is_float(\$var)	Kiểm tra xem biến \$var có phải kiểu float hay không, trả về TRUE nếu phải, ngược lại là FALSE
round(\$var)	Trả về số nguyên gần biến \$var nhất
ceil(\$var)	Trả về số nguyên được làm tròn lên số nguyên lớn nhất gần với \$var nếu là số dương, ngược lại trả về phần nguyên nếu là số âm
floor(\$var)	Trả về số nguyên được làm tròn xuống số nguyên nhỏ nhất gần với \$var nếu là số âm, ngược lại trả về phần nguyên nếu là số dương

# Các hàm dựng sẵn thao tác với Số



Tên hàm	Mô tả
<code>max(\$var1, \$var2, ...)</code>	Trả về giá trị lớn nhất trong các biến <code>\$var1</code> , <code>\$var2</code> , ... truyền vào
<code>min(\$var1, \$var2, ...)</code>	Trả về giá trị nhỏ nhất trong các biến <code>\$var1</code> , <code>\$var2</code> , ... truyền vào
<code>rand(\$number1, \$number2)</code>	Trả về số ngẫu nhiên nằm trong khoảng <code>\$number1</code> và <code>\$number2</code>
<code>number_format(\$number)</code>	Định dạng số <code>\$number</code> theo hàng nghìn, trả về 1 string
<code>abs(\$number)</code>	Trả về giá trị tuyệt đối của số <code>\$number</code>
<code>pow(\$number, \$exponent)</code>	Trả về <code>\$number</code> mũ <code>\$exponent</code>
<code>sqrt(\$number)</code>	Trả về căn bậc 2 của <code>\$number</code>

# 5. Các hàm dựng sẵn cho Time

# Các hàm dựng sẵn cho Time



Tên hàm	Mô tả
<code>getdate()</code>	trả về mảng chứa các thông tin về thời gian hiện tại của máy chủ
<code>date_default_timezone_get()</code>	trả về string của timezone(múi giờ) mặc định mà hệ thống máy chủ đang sử dụng
<code>date_default_timezone_set(\$timezone_identifier)</code>	thiết lập timezone mặc định cho hệ thống bằng string <code>\$timezone_identifier</code> , trả về TRUE nếu thành công, FALSE nếu thất bại.
<code>time()</code>	lấy giá trị unix timestamp tại thời điểm hiện tại (unix timestamp là số giây tại thời điểm hiện tại cho với 01/01/1970), trả về số int
<code>date(\$format, \$timestamp)</code>	định dạng các hiển thị thời gian theo format <code>\$format</code> và giá trị unix timestamp <code>\$timestamp</code>
<code>strtotime(\$format)</code>	chuyển <code>\$format</code> thành giá trị Unix timestamp