

# TRUNG TÂM ĐÀO TẠO LẬP TRÌNH WEB & ĐỒ HỌA

# VIETPRO EDUCATION



## BÀI GIẢNG PHP (BUỔI 2)

## NHẬP MÔN LẬP TRÌNH PHP (TIẾP THEO)

**Giảng viên: Hoàng Minh Tuấn**

# CÁC PHƯƠNG THỨC TRUYỀN DỮ LIỆU

## 1. Các Thuộc tính quan trọng của Form tương tác dữ liệu

- **name**: Tên của Form, dùng để dễ dàng nhận biết Form với các thành phần khác trong Form
- **method**: Phương thức truyền dữ liệu thông qua Form với 2 phương thức là POST và GET
- **action**: Điều hướng dữ liệu tới trang xử lý



# CÁC PHƯƠNG THỨC TRUYỀN DỮ LIỆU



## 2. Phương thức truyền dữ liệu POST

- *Post là phương thức truyền và nhận dữ liệu từ Form với phương thức method="post"*

**`$_POST["elementName"];`**

### **Chú ý:**

- *Có thể sử dụng phương thức GET để lấy giá trị trong Form*

## 3. Phương thức truyền dữ liệu GET

- *Get là phương thức truyền và nhận dữ liệu từ các tham số trên thanh địa chỉ của trình duyệt*

**`fileName.php?param=value;`**

**`$_GET["param"];`**

- **param:** Tham số truyền đi
- **value:** Giá trị mà tham số truyền đi

# CÁC PHƯƠNG THỨC TRUYỀN DỮ LIỆU

## 4. Truyền nhiều tham số với Phương thức GET

**fileName.php?param=value&param1=value1&...&paramN=valueN;**

- **param, param1,... paramN:** Các tham số được truyền đi
- **value, value1,...valueN:** Các giá trị tương ứng mà các tham số truyền đi

# MẢNG TRONG PHP



## 1. Định nghĩa Mảng trong PHP

- Mảng trong php là một danh sách các phần tử cùng hoặc không cùng kiểu dữ liệu. Mảng có thể là mảng một chiều hay nhiều chiều. Mảng có 2 thành phần là chỉ mục (key) và giá trị (value). Chỉ mục (key) có thể là số nguyên hoặc là chuỗi

## 2. Một số cách sử dụng để khai báo Mảng trong PHP

- Chúng ta có thể khai báo Mảng bao gồm các phần tử mảng hoặc chúng ta có thể khai báo một mảng rỗng, sau đó sẽ thêm các phần tử mảng vào sau.

**C1:**

```
$mang = array(Element1, Element2,... ElementN);
```

**C2:**

```
$mang = array();
```

```
$mang[] = Element1;
```

```
$mang[] = Element2;
```

```
...
```

```
$mang[] = ElementN;
```

- Phần tử đầu tiên trong mảng sẽ có Key là 0 và Value là Element1

# MẢNG TRONG PHP



## 3. Mảng kết hợp

- Mảng kết hợp là mảng chứa các phần tử có các chỉ mục (key) là các chuỗi

***\$mang=array('string1'=>value1,...'stringN'=>valueN);***

## 4. Thêm một phần tử mới vào Mảng

***\$mang[Key] = Value;***

### ***Chú ý:***

- Nếu chỉ mục (Key) để trống thì mảng sẽ tự động đánh số tiếp theo cho chỉ mục

## 5. Lấy giá trị của phần tử Mảng

***\$mang[Key] ;***

# MẢNG TRONG PHP



## 6. Duyệt Mảng với vòng lặp Foreach()

- Vòng lặp *Foreach* cho phép chúng ta duyệt qua tất cả các Phần tử Mảng cả Key và Value

**foreach(\$array as \$value){**

**Thực thi hành động;**

**}**

- \$array: Mảng cần duyệt
- \$value: Giá trị của phần tử trong mảng sau mỗi vòng duyệt

**foreach(\$array as \$key => \$value){**

**Thực thi hành động;**

**}**

- \$array: Mảng cần duyệt
- \$value: Giá trị của phần tử trong mảng sau mỗi vòng duyệt
- \$key: Key của phần tử sau mỗi vòng duyệt

# MỘT SỐ HÀM HỖ TRỢ MẢNG TRONG PHP



## 1. Hàm Print\_r()

- Cho phép duyệt đệ quy và hiển thị thông tin của một giá trị.

**print\_r(\$mang);**

## 2. Hàm Sort()

- Hàm sort() dùng để sắp xếp các phần tử trong mảng với giá trị tăng dần.

**sort(\$mang);**

## 3. Hàm Rsort()

- Hàm rsort() dùng để sắp xếp các phần tử trong mảng với giá trị giảm dần.

**rsort(\$mang);**

## 4. Hàm Count()

- Hàm count() dùng để đếm số phần tử của một mảng. Nó sẽ nhận vào một mảng và trả về một số nguyên bằng với số phần tử tồn tại trong mảng đó.

**count(\$mang);**



# MỘT SỐ HÀM HỖ TRỢ MẢNG TRONG PHP

## 5. Hàm Array\_merge()

- Hàm `array_merge()` dùng để gộp mảng. Nó sẽ nhận vào 2 mảng và trả về một mảng bao gồm tất cả các phần tử của mỗi mảng, các phần tử này có chỉ mục (key) được giữ nguyên.
- Nếu 2 mảng có chỉ mục (key) bị trùng nhau và là dạng chuỗi thì giá trị tương ứng của key đó ở mảng 1 sẽ bị ghi đè bởi giá trị ở mảng 2
- Nếu 2 mảng có chỉ mục (key) bị trùng nhau và là dạng số thì giá trị tương ứng của key đó ở mảng 2 sẽ được cung cấp 1 key mới mà không bị ghi đè

### Ví dụ:

```
$mang1 = array("Name"=>"John", "Age"=>18, 5=>9);  
$mang2 = array("Tên"=>"Minh", 5=>99, "Age"=>25);  
print_r($mang1);  
echo '<br />';  
print_r($mang2);  
echo '<br />';  
$mang = array_merge($mang1, $mang2);  
print_r($mang);
```

# MỘT SỐ HÀM HỖ TRỢ MẢNG TRONG PHP



## 6. Hàm Array\_combine()

- Hàm `array_combine()` nhận vào 2 mảng và trả về một mảng có chỉ mục (key) là giá trị của mảng 1 còn giá trị (value) là giá trị của mảng 2
- Nếu 2 mảng không có cùng số phần tử thì lập tức báo lỗi

### Ví dụ:

```
$mang1 = array("Language", "Database", "Server");  
$mang2 = array("PHP", "MySQL", "Linux");  
print_r($mang1);  
echo '<br />';  
print_r($mang2);  
echo '<br />';  
$mang = array_combine($mang1, $mang2);  
print_r($mang);
```

# MỘT SỐ HÀM HỖ TRỢ MẢNG TRONG PHP

## 7. Hàm Array\_intersect()

- Hàm `array_intersect()` nhận vào 2 mảng và trả về một mảng có chứa các phần tử có cùng giá trị ở 2 mảng.
- Nếu các phần tử có cùng giá trị (value) nhưng khác chỉ mục (key) thì phần tử đó sẽ nhận chỉ mục (key) của phần tử thuộc mảng 1

### Ví dụ:

```
$mang1 = array("Language"=>"PHP",  
"Database"=>"MySQL", "Server"=>"Linux");  
$mang2 = array("CMS"=>"Joomla",  
"Database"=>"MySQL", "Máy chủ"=>"Linux");  
print_r($mang1);  
echo '<br />';  
print_r($mang2);  
echo '<br />';  
$mang = array_intersect($mang1, $mang2);  
print_r($mang);
```

# MỘT SỐ HÀM HỖ TRỢ MẢNG TRONG PHP

## 8. Hàm In\_array()

- Hàm `in_array()` sẽ tìm sự tồn tại của một phần tử trong mảng, và trả về kết quả là `TRUE` hoặc `FALSE`

**Ví dụ:**

```
$mang = array("Học", "Lập", "Trình", "Web", "PHP");  
$result = in_array("PHP", $mang);  
echo $result.'<br />';  
$result = in_array("HTML", $mang);  
echo $result.'<br />';
```



# HÀM TRONG PHP



## 1. Định nghĩa về Hàm trong PHP

- Để giảm thời gian lặp lại 1 thao tác code nhiều lần, PHP hỗ trợ người lập trình việc tự định nghĩa cho mình những hàm có khả năng lặp lại nhiều lần trong website. Việc này cũng giúp cho người lập trình kiểm soát mã nguồn một cách mạch lạc. Đồng thời có thể tùy biến ở mọi trang. Mà không cần phải khởi tạo hay viết lại mã lệnh như HTML thuần túy.

```
function functionName(){  
    Thực thi lệnh;  
}
```

### Chú ý:

- Tên hàm có thể là một tổ hợp bất kỳ những chữ cái, con số và dấu gạch dưới, nhưng phải bắt đầu từ chữ cái hoặc dấu gạch dưới.

## 2. Thực thi Hàm

```
functionName();
```

# HÀM TRONG PHP



## 3. Hàm tự định nghĩa có Tham số

```
function functionName(Param1, Param2,... ParamN){  
    Thực thi lệnh;  
}
```

## 4. Hàm tự định nghĩa có giá trị trả về

```
function functionName(){  
    Thực thi lệnh;  
    return $result;  
}
```

## 5. Hàm gọi lại Hàm trong PHP

- PHP cho phép chúng ta sử dụng một số hàm sau đây để triệu gọi File như: **`require_once()`**, **`include_once()`**  
**`required_once(filename.php);`**
- **`include_once(filename.php);`**

# MỘT SỐ HÀM XÂY DỰNG SẴN TRONG PHP



## 1. Hàm Print\_r()

- Cho phép duyệt đệ quy và hiển thị thông tin của một giá trị.

**print\_r(\$bien);**

## 2. Hàm Explode()

- Tách chuỗi ban đầu theo một ký hay một chuỗi ký tự mẫu và trả về một mảng các ký tự trong chuỗi ban đầu.

**explode(Character, String);**

## 3. Hàm Implode()

- Kết hợp các phần tử trong một mảng lại thành một chuỗi theo một ký tự hay một chuỗi ký tự mẫu.

**implode(Character, String);**

## 4. Hàm str\_replace()

- Tìm ký tự hoặc chuỗi ký tự trong chuỗi cha để thay thế bởi ký tự hoặc chuỗi ký tự mới. Kết quả trả về một chuỗi mới giữ nguyên chuỗi cũ.

**str\_replace(oldString, newString, String);**

# MỘT SỐ HÀM XÂY DỰNG SẴN TRONG PHP



## 5. Hàm Strlen()

- *Đếm số ký tự trong một xâu (chuỗi).*

**strlen(String);**

## 6. Hàm Substr()

- *Hàm này dùng để cắt ra một phần của xâu.*

**substr(String, Index, Count);**

### Chú ý:

- *Vị trí bắt đầu của chuỗi được tính là 0*
- *Vị trí cuối cùng của chuỗi được tính là 1 (nếu đếm ngược lại từ cuối chuỗi lên)*
- *Chuỗi con luôn luôn lấy cả ký tự ở vị trí bắt đầu và kết thúc của việc cắt*
- *Trả về 1 chuỗi mới và giữ nguyên chuỗi ban đầu*
- *Nếu hàm không có tham số thứ 3 thì sẽ cắt từ vị trí cắt đến hết chuỗi*

## 7. Hàm Strrchr()

- *Tìm kiếm và bắt đầu lấy từ ký tự xuất hiện cuối cùng của ký tự đó trong xâu đến hết xâu.*

**strrchr(String, Endchar);**



# THỰC HÀNH



## I – BÀI TẬP THỰC HÀNH

### 1. Bài 1

### 2. Bài 2

## II – BÀI TẬP VỀ NHÀ

### 1. Bài 1

### 2. Bài 2