

# THAM KHẢO NHANH PHP

Cú pháp, mảng, OOP, database, file I/O cơ bản

## Cơ Bản

### Hello World

```
<?php
echo "Hello, World!\n";
// PHP code must be inside <?php ... ?> tags
```

### Chạy PHP

```
php script.php # run a file
php -r 'echo "hi\n";' # run inline code
php -S localhost:8000 # built-in dev server
```

### Chú Thích

```
// single-line comment
# also single-line
/* multi-line
comment */
```

## Biến & Kiểu Dữ Liệu

### Biến

```
$name = "PHP"; // string
$version = 8.3; // float
$count = 42; // int
$active = true; // bool
$itens = null; // null
```

### Kiểm Tra Kiểu

**gettype(\$x)** Trả về kiểu dưới dạng chuỗi  
**is\_string(\$x)** Kiểm tra nếu là chuỗi  
**is\_int(\$x)** Kiểm tra nếu là số nguyên  
**is\_array(\$x)** Kiểm tra nếu là mảng  
**is\_null(\$x)** Kiểm tra nếu là null  
**isset(\$x)** Kiểm tra nếu đã đặt và không null  
**empty(\$x)** Kiểm tra nếu rỗng (falsy)

### Ép Kiểu

```
$n = (int) "42"; // 42
$s = (string) "3.14"; // "3.14"
$b = (bool) ""; // false
$a = (array) $obj; // object to array
```

### Hằng Số

```
define("MAX_SIZE", 100);
const API_VERSION = "v2";
echo MAX_SIZE; // 100
```

## Strings

### String Cơ Bản

```
$name = "World";
echo "Hello, $name!"; // variable interpolation
echo "Hello, $name!"; // literal (no interpolation)
echo "Value: ${arr['key']}"; // complex expression
```

### Hàm String

**strlen(\$s)** Độ dài chuỗi tính bằng byte  
**mb\_strlen(\$s)** Độ dài chuỗi tính theo ký tự (an toàn với multibyte)  
**strtolower(\$s)** Chuyển thành chữ thường  
**strtoupper(\$s)** Chuyển thành chữ hoa  
**trim(\$s)** Xóa khoảng trắng ở hai đầu  
**str\_replace(a, b, \$s)** Thay 'a' bằng 'b' trong '\$s'  
**substr(\$s, 0, 5)** Chuỗi con từ vị trí 0, độ dài 5  
**strpos(\$s, 'find')** Tìm vị trí chuỗi con (false nếu không tìm thấy)  
**explode(' ', \$s)** Tách chuỗi thành mảng  
**implode(' ', \$a)** Nối mảng thành chuỗi

### Heredoc & Nowdoc

```
$html = <<<HTML
<p>Hello, $name</p>
HTML;
$raw = <<<TEXT'
No $interpolation here
TEXT;
```

## Mảng

### Indexed & Associative

```
$nums = [1, 2, 3]; // indexed
$user = ["name" => "Alice", "age" => 30]; // associative
$nums[] = 4; // append
echo $user["name"]; // access
```

### Hàm Mảng

**count(\$a)** Số phần tử  
**array\_push(\$a, \$v)** Thêm vào cuối  
**array\_pop(\$a)** Xóa và trả về phần tử cuối  
**array\_merge(\$a, \$b)** Kết hợp hai mảng  
**in\_array(\$v, \$a)** Kiểm tra giá trị tồn tại  
**array\_key\_exists(\$k, \$a)** Kiểm tra key tồn tại  
**array\_map(\$fn, \$a)** Áp dụng hàm lên mỗi phần tử  
**array\_filter(\$a, \$fn)** Lọc phần tử theo callback  
**sort(\$a)** Sắp xếp in-place (danh chỉ số)  
**array\_keys(\$a)** Trả về tất cả keys

### Lặp

```
foreach ($users as $user) { echo $user; }
foreach ($map as $key => $value) {
    echo "$key: $value\n";
}
```

## Hàm

### Hàm Cơ Bản

```
function add(int $a, int $b): int {
    return $a + $b;
}
echo add(3, 5);
```

### Tham Số Mặc Định & Có Tên

```
function greet(string $name, string $greeting = "Hello"): string {
    return "$greeting, $name!";
}
greet("Alice");
greet($greeting: "Hi", name: "Bob"); // named args (PHP 8+)
```

### Array Functions

```
$double = fn(int $x): int => $x * 2;
$nums = array_map(fn($n) => $n * 10, [1, 2, 3]);
```

### Closures

```
$factor = 3;
$multiply = function(int $x) use ($factor): int {
    return $x * $factor;
};
echo $multiply(5); // 15
```

## Classes & Objects

### Định Nghĩa Class

```
class User {
    public function construct(
        private string $name,
        private int $age = 0,
    ) {}
    public function greet(): string { return "Hi, {$this->name}"; }
}
```

### Kế Thừa & Interfaces

```
interface Printable {
    public function toString(): string;
}
class Admin extends User implements Printable {
    public function toString(): string { return "Admin"; }
}
```

### Phạm Vi Truy Cập

**public** Truy cập từ bất kỳ đâu  
**protected** Truy cập từ class và các lớp con  
**private** Chỉ truy cập trong class  
**readonly** Chỉ gán được một lần (PHP 8.1+)  
**static** Thuộc class, không phải instance  
**abstract** Phải được implement bởi lớp con

### Traits

```
trait Timestamped {
    public function createdAt(): string {
        return date('Y-m-d H:i:s');
    }
}
class Post { use Timestamped; }
```

## Xử Lý Lỗi

### Try / Catch / Finally

```
try {
    $result = riskyOperation();
} catch (InvalidArgumentException $e) {
    echo "Bad input: " . $e->getMessage();
} catch (Exception $e) {
    echo "Error: " . $e->getMessage();
} finally { cleanup(); }
```

### Custom Exceptions

```
class ApiException extends RuntimeException {
    public function __construct(string $message, private int $statusCode = 500) {
        parent::__construct($message, $statusCode);
    }
}
```

### Null Safety (PHP 8+)

```
$len = $user?->address?->zip; // nullsafe operator
$name = $input ?? "default"; // null coalescing
$data ??= []; // null coalescing assignment
```

## File I/O

### Đọc & Ghi Files

```
$content = file_get_contents("data.txt");
file_put_contents("out.txt", $content);
$lines = file("data.txt", FILE_IGNORE_NEW_LINES);
```

### File Handle

```
$f = fopen("log.txt", "a");
fwrite($f, "entry\n");
fclose($f);
```

### Hàm File

**file\_exists(\$path)** Kiểm tra file tồn tại  
**is\_dir(\$path)** Kiểm tra path là thư mục  
**mkdir(\$path, 0755, true)** Tạo thư mục đệ quy  
**unlink(\$path)** Xóa file  
**glob('\*.\*.txt')** Tìm file khớp pattern  
**realpath(\$path)** Giải quyết đường dẫn tuyệt đối đầy đủ

## Database

### Kết Nối PDO

```
$pdo = new PDO(
    "mysql:host=localhost;dbname=app",
    "user", "password",
    [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]
);
```

### Prepared Statements

```
$stmt = $pdo->prepare("SELECT * FROM users WHERE id = :id");
$stmt->execute(["id" => 42]);
$user = $stmt->fetch(PDO::FETCH_ASSOC);
```

### Insert & Update

```
$stmt = $pdo->prepare("INSERT INTO users (name, email) VALUES
(?, ?)");
$stmt->execute(["Alice", "alice@example.com"]);
$id = $pdo->lastInsertId();
```

### Chế Độ Fetch PDO

**fetch()** Lấy một hàng

**fetchAll()** Lấy tất cả hàng  
**FETCH\_ASSOC** Trả về dạng mảng kết hợp  
**FETCH\_OBJ** Trả về dạng object ẩn danh  
**FETCH\_CLASS** Trả về dạng instance của class cụ thể

## Hàm Phổ Biến

### JSON

```
$json = json_encode(["name" => "Alice", "age" => 30]);
$data = json_decode($json, true); // true = assoc array
$data = json_decode($json); // object
```

### Ngày & Giờ

```
echo date("Y-m-d H:i:s"); // 2026-03-26 12:00:00
$time = strtotime("+1 week");
$date = new DateTime("2026-01-01");
echo $date->format("D, M j"); // Thu, Jan 1
```

### Toán & Ngẫu Nhiên

**abs(\$n)** Giá trị tuyệt đối  
**round(\$n, 2)** Làm tròn đến 2 chữ số thập phân  
**ceil(\$n) / floor(\$n)** Làm tròn lên / xuống  
**min(\$a, \$b) / max(\$a, \$b)** Nhỏ nhất / lớn nhất  
**random\_int(1, 100)** Số nguyên ngẫu nhiên bảo mật  
**number\_format(\$n, 2)** Định dạng với dấu phân cách hàng nghìn

### Biểu Thức Chính Quy

```
preg_match('/^[a-z]+$/', $str, $matches);
preg_match_all('/\d+/', $str, $all);
$result = preg_replace('/\s+/', ' ', $str);
```