

TÀI LIỆU THỰC HÀNH PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ





01 Giới thiệu về php >

02 Cài đặt môi trường >

03 Cú pháp php cơ bản >

04 Hướng đối tượng >

05 Laragon >

06 Database >

PHP LÀ GÌ?

- PHP (Hypertext Preprocessor) là ngôn ngữ lập trình phía máy chủ (server)
- Ra đời năm 1994 bởi Rasmus Lerdorf
- Ban đầu là "Personal Home Page" tools, sau đổi thành PHP
- Dùng để tạo các trang web động, xử lý dữ liệu form, kết nối cơ sở dữ liệu
- PHP có thể nhúng vào HTML, dễ học và mạnh mẽ
- Mã nguồn mở và miễn phí



LỊCH SỬ HÌNH THÀNH VÀ PHÁT TRIỂN CỦA PHP

- 1994: PHP Tools v1.0 (Rasmus Lerdorf)
- 1995: PHP/FI (Form Interpreter)
- 1998: PHP 3.0 - Viết lại hoàn toàn bởi Andi Gutmans và Zeev Suraski
- 2000: PHP 4.0 với Zend Engine 1.0
- 2004: PHP 5.0 với Zend Engine 2.0, hỗ trợ OOP mạnh mẽ
- 2015: PHP 7.0 với hiệu suất cao hơn gấp đôi so với phiên bản 5.6
- 2020: PHP 8.0 với JIT compiler (Just-In-Time compilation), nâng cao hiệu suất
- 2025: PHP 8.4 với nhiều tính năng hiện đại

LÝ DO VÌ SAO CHỌN PHP?

- Mã nguồn mở hoàn toàn miễn phí, có cộng đồng hỗ trợ lớn
- Cú pháp đơn giản, gần gũi với ngôn ngữ C
- Tương thích cao có thể chạy trên nhiều hệ điều hành (Windows, Linux, MacOS)
- Dễ dàng kết nối với cơ sở dữ liệu MySQL
- Hỗ trợ Framework mạnh như: Laravel, CodeIgniter, Symfony
- Có hiệu suất cao với PHP 7+ có tốc độ nhanh hơn nhiều so với PHP 5

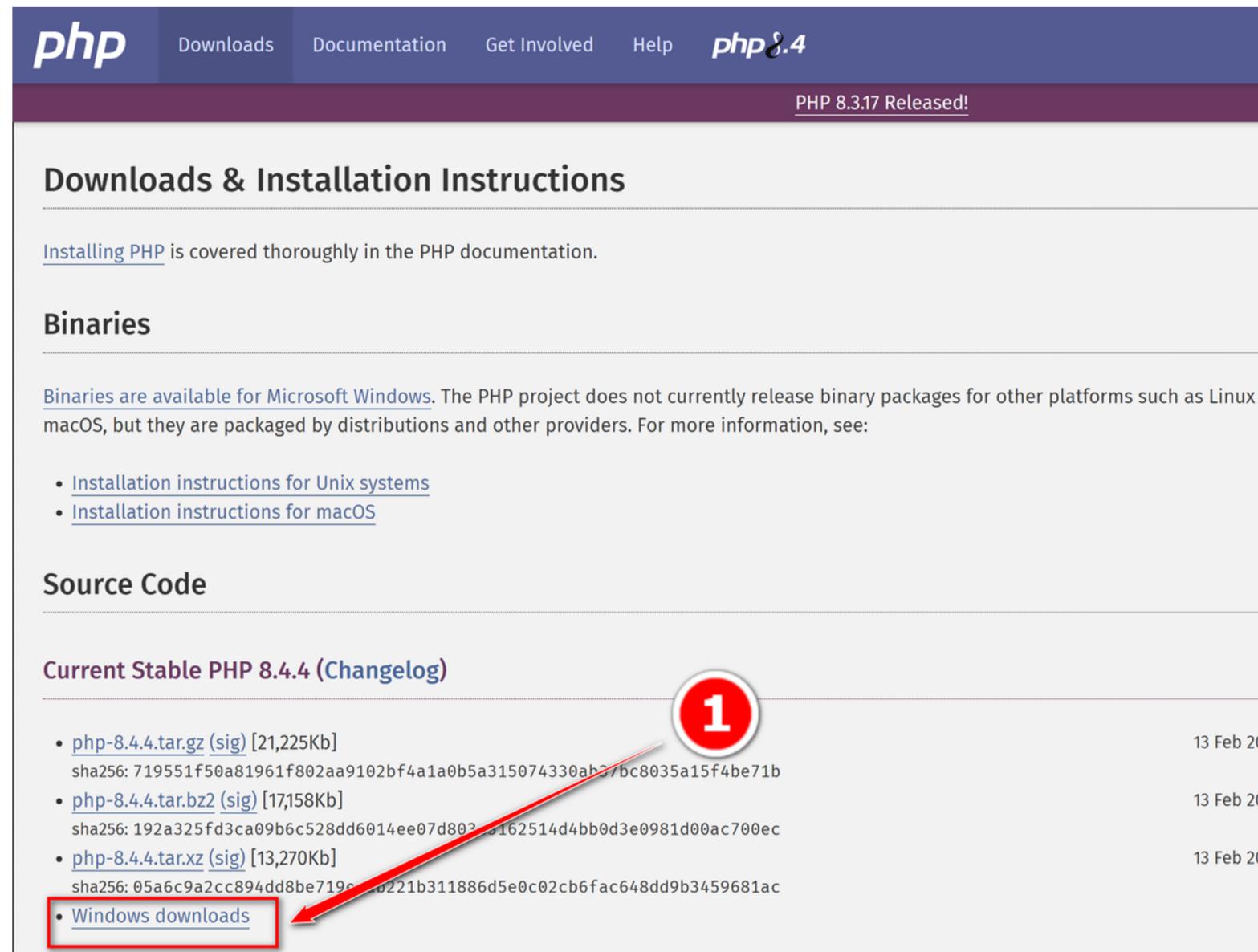


CÁC ÚNG DỤNG NỔI TIẾNG XÂY DỰNG BẰNG PHP

- Facebook: Sử dụng PHP với HHVM/Hack
- WordPress: CMS phổ biến nhất thế giới
- Wikipedia: Encyclopedia lớn nhất
- Shopify: Nền tảng thương mại điện tử
- Laravel.com: Framework PHP phổ biến nhất



CÀI ĐẶT PHP



PHP Downloads Documentation Get Involved Help **php8.4**
PHP 8.3.17 Released!

Downloads & Installation Instructions

Installing PHP is covered thoroughly in the PHP documentation.

Binaries

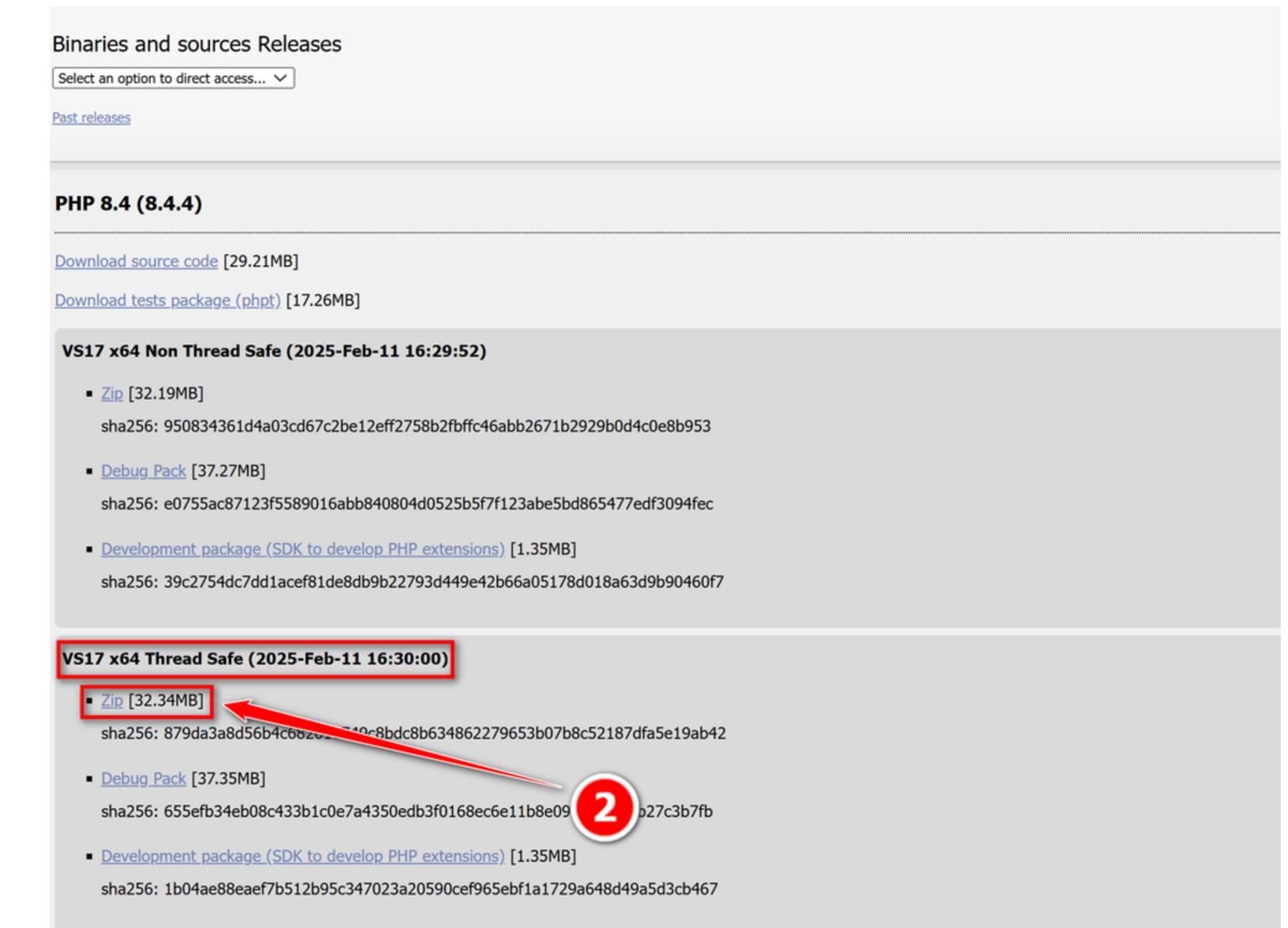
Binaries are available for Microsoft Windows. The PHP project does not currently release binary packages for other platforms such as Linux or macOS, but they are packaged by distributions and other providers. For more information, see:

- [Installation instructions for Unix systems](#)
- [Installation instructions for macOS](#)

Source Code

Current Stable PHP 8.4.4 (Changelog)

- [php-8.4.4.tar.gz \(sig\) \[21,225Kb\]](#)
sha256: 719551f50a81961f802aa9102bf4a1a0b5a315074330ab7/bc8035a15f4be71b
13 Feb 2025
- [php-8.4.4.tar.bz2 \(sig\) \[17,158Kb\]](#)
sha256: 192a325fd3ca09b6c528dd6014ee07d803...3162514d4bb0d3e0981d00ac700ec
13 Feb 2025
- [php-8.4.4.tar.xz \(sig\) \[13,270Kb\]](#)
sha256: 05a6c9a2cc894dd8be719...0221b311886d5e0c02cb6fac648dd9b3459681ac
13 Feb 2025
- [Windows downloads](#)



Binaries and sources Releases
Select an option to direct access... ▾
Past releases

PHP 8.4 (8.4.4)

[Download source code](#) [29.21MB]
[Download tests package \(.phpt\)](#) [17.26MB]

VS17 x64 Non Thread Safe (2025-Feb-11 16:29:52)

- [Zip](#) [32.19MB]
sha256: 950834361d4a03cd67c2be12eff2758b2fbff46abb2671b2929b0d4c0e8b953
- [Debug Pack](#) [37.27MB]
sha256: e0755ac87123f5589016abb840804d0525b5f7f123abe5bd865477edf309fec
- [Development package \(SDK to develop PHP extensions\)](#) [1.35MB]
sha256: 39c2754dc7dd1acef81de8db9b22793d449e42b66a05178d018a63d9b90460f7

VS17 x64 Thread Safe (2025-Feb-11 16:30:00)

- [Zip](#) [32.34MB]
sha256: 879da3a8d56b4c6820...710-8bcd8b634862279653b07b8c52187dfa5e19ab42
2
- [Debug Pack](#) [37.35MB]
sha256: 655efb34eb08c433b1c0e7a4350edb3f0168ec6e11b8e09...b27c3b7fb
- [Development package \(SDK to develop PHP extensions\)](#) [1.35MB]
sha256: 1b04ae88eaef7b512b95c347023a20590cef965ebf1a1729a648d49a5d3cb467

Link download: <https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170>

CÁCH FIX LỖI CÀI ĐẶT

Latest Microsoft Visual C++ Redistributable Version

The latest version is 14.42.34433.0

Use the following links to download this version for each supported architecture:

 Expand table

Architecture	Link	Notes
ARM64	https://aka.ms/vs/17/release/vc_redist.arm64.exe	Permalink for latest supported ARM64 version
X86	https://aka.ms/vs/17/release/vc_redist.x86.exe	Permalink for latest supported x86 version
X64	https://aka.ms/vs/17/release/vc_redist.x64.exe	Permalink for latest supported x64 version. The X64 Redistributable package contains both ARM64 and X64 binaries. This package makes it easy to install required Visual C++ ARM64 binaries when the X64 Redistributable is installed on an ARM64 device.

Download và chạy lại nhé



Trong quá trình cài đặt PHP gặp vấn đề trên thì thực hiện fix dựa theo cách sau:

```
C:\Users\Win>php --version
PHP Warning:  'C:\Windows\SYSTEM32\VCRUNTIME140.dll' 14.32 is not compatible with
this PHP build linked with 14.42 in Unknown on line 0
```

Link download: <https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170>

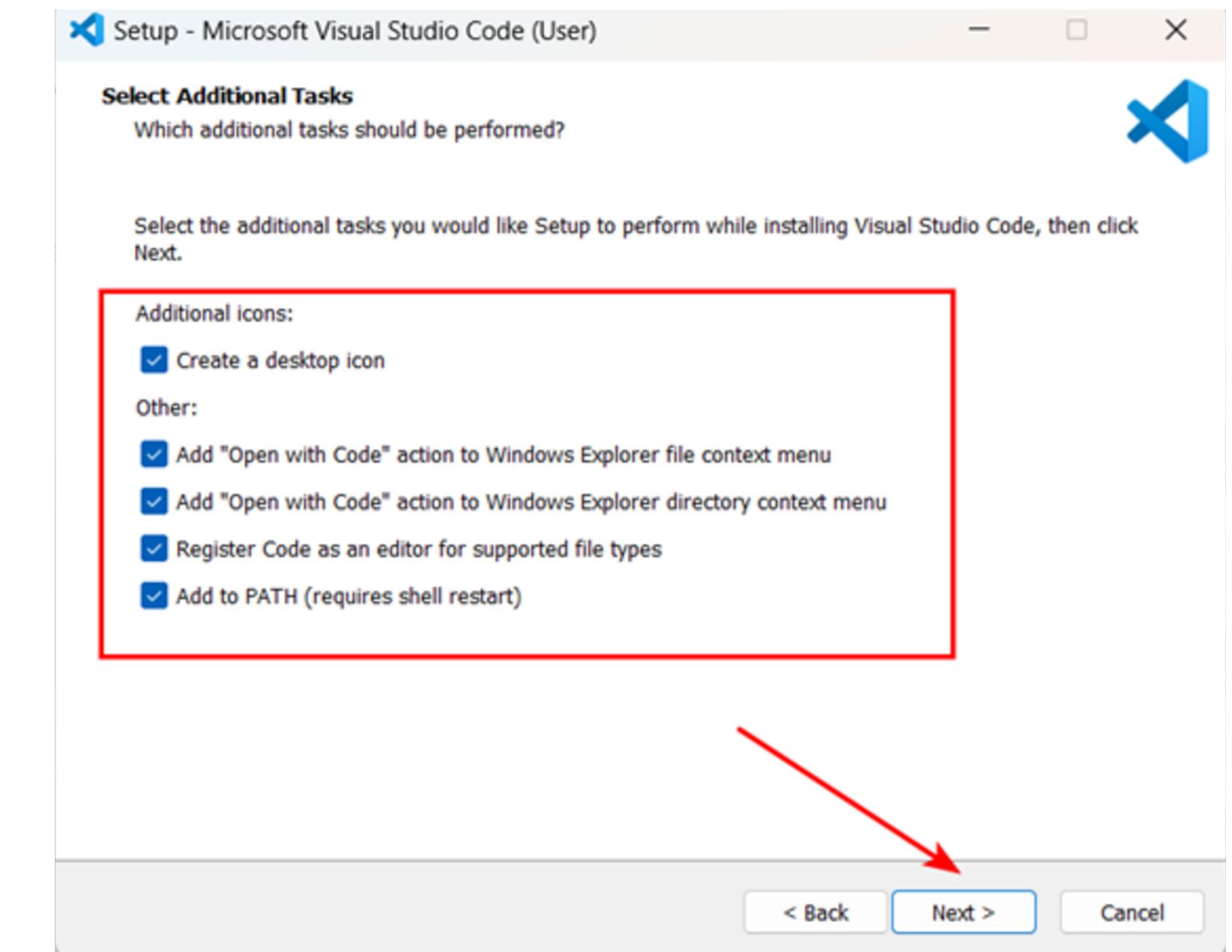
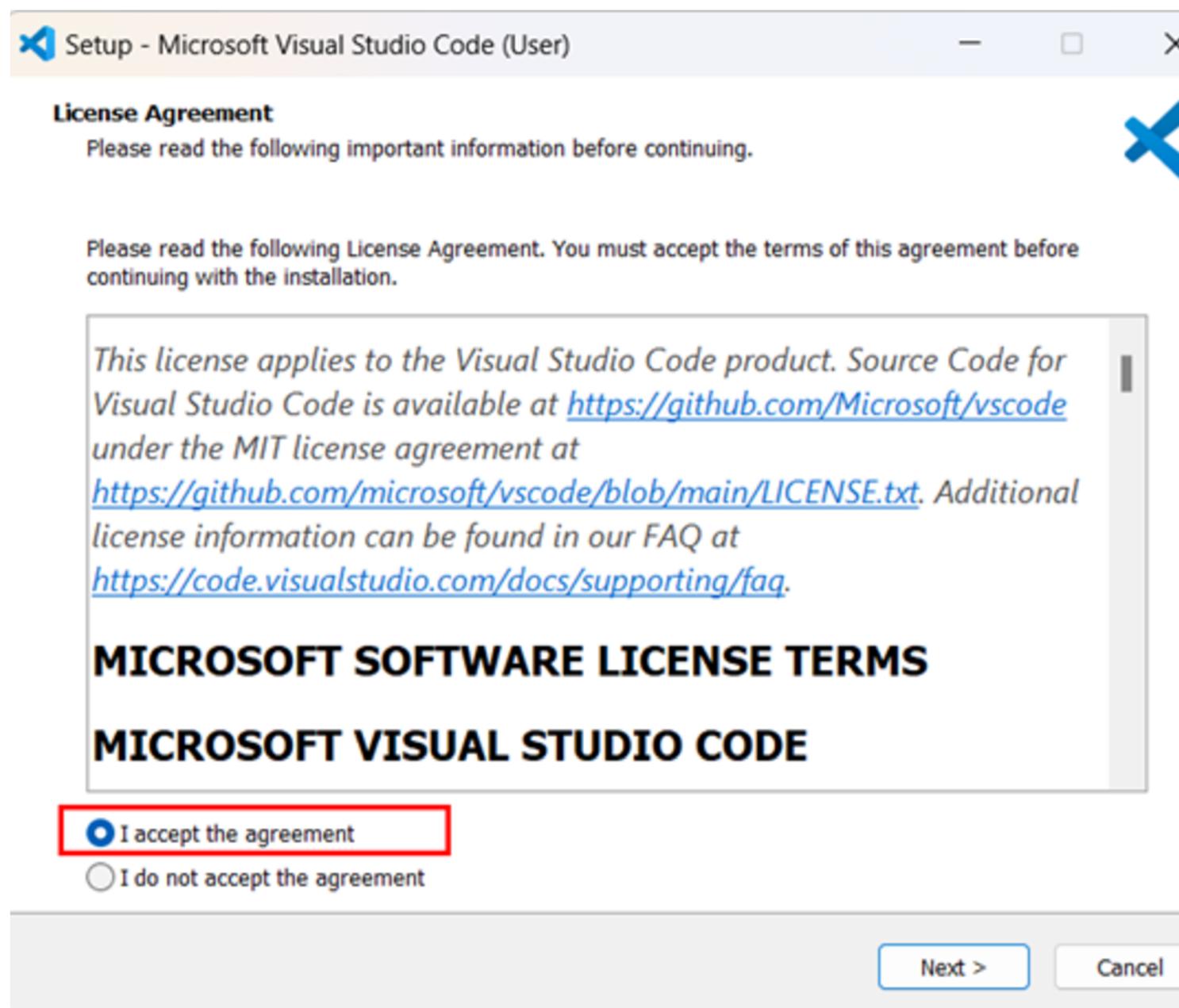
```
C:\Users\Win>php --version
PHP 8.4.4 (cli) (built: Feb 11 2025 16:25:02) (ZTS Visual C++
Copyright (c) The PHP Group
Zend Engine v4.4.4, Copyright (c) Zend Technologies
```

IDE VÀ TEXT EDITOR HỖ TRỢ

- IDE/Editor:
 -  Visual Studio Code
 -  PHPStorm
 -  Sublime Text
- Server cục bộ:
 -  Laragon
 -  XAMPP
 -  WAMP
- Cơ sở dữ liệu:
 -  MariaDB
 -  MySQL
- Version control:
 -  Github
 -  Git
 -  Gitlab
- API Testing:
 -  Postman



CÀI ĐẶT VISUAL STUDIO CODE



EXTENSION HỖ TRỢ CODE PHP



PHP Intelephense

Ben Mewburn  intelephense.com | ⚡ 14,277,786 |

PHP code intelligence for Visual Studio Code

[Install](#)



Auto Update 



PHP Extension Pack

Xdebug  xdebug.org | ⚡ 5,640,292 |

Everything you need for PHP development

[Install](#)



Auto Update 



PHP IntelliSense

Damjan Cvetko | ⚡ 4,732,844 | ★★★★☆(20)

Advanced Autocompletion and Refactoring support for PHP

[Install](#)



Auto Update 



PHP Server

brapifra | ⚡ 2,449,756 | ★★★★★(33)

Serve your Project with PHP

[Install](#)



Auto Update 



PHP Debug

Xdebug  xdebug.org | ⚡ 13,230,463 |

Debug support for PHP with Xdebug

[Install](#)



Auto Update 



Code Runner

Jun Han | ⚡ 32,150,747 | ★★★★★(272) |

Run C, C++, Java, JS, PHP, Python, Perl, Ruby, Go, Lua,

[Disable](#)



Uninstall 

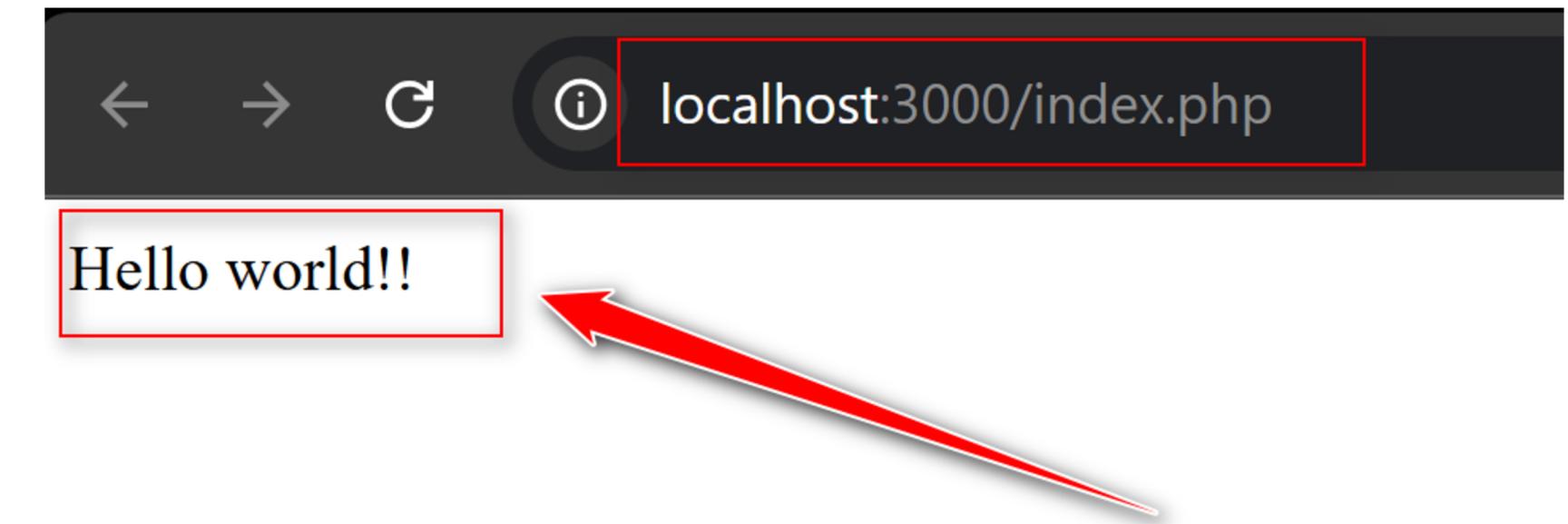
Auto Update 

CẤU TRÚC THƯ MỤC TRONG WEB SERVER

```
● ● ●  
1 /PROJECT  
2 |-- .vscode           # Cấu hình cho VS Code  
3 |-- /app              # Chứa toàn bộ mã nguồn của ứng dụng  
4 |   |-- /controllers    # Xử lý yêu cầu và điều hướng dữ liệu  
5 |   |-- /models          # Quản lý truy xuất dữ liệu và kết nối cơ sở dữ liệu  
6 |   |-- /views            # Hiển thị giao diện cho người dùng  
7 |-- /public             # Chứa tài nguyên tĩnh của hệ thống  
8 |   |-- /css               # Chứa file CSS  
9 |   |-- /images            # Chứa hình ảnh sử dụng trong ứng dụng  
10 |   |-- /js                # Chứa file JavaScript  
11 |-- .htaccess           # Cấu hình server Apache, giúp viết lại URL đẹp hơn  
12 |-- index.php            # File chính để xử lý request vào hệ thống  
13
```

VIẾT MÃ PHP CƠ BẢN

```
● ● ●  
1 <?php  
2 echo "Hello world!!"  
3 ?>
```



- Mã PHP được viết trong cặp thẻ `<?php ... ?>`
- Mỗi tập tin PHP có phần mở rộng .php
- Có thể kết hợp với JavaScript, CSS để tạo ứng dụng web động.

PHP VỚI HTML

Nhúng PHP vào HTML

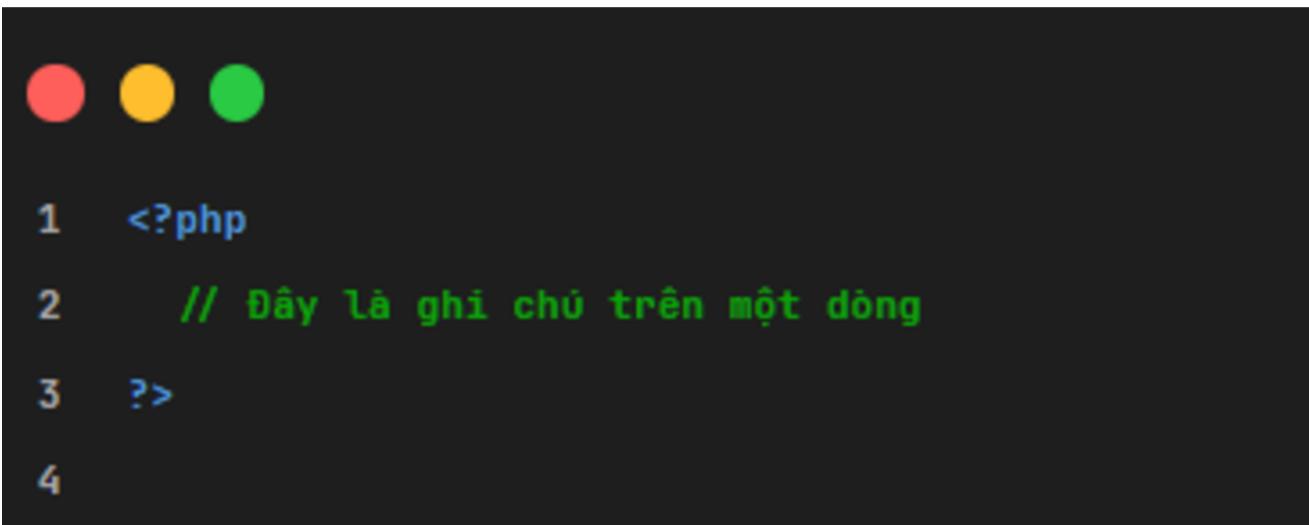
```
● ● ●  
1 <!DOCTYPE html>  
2 <html>  
3   <body>  
4     <h1><?php echo "Xin chào, HUTECH!"; ?></h1>  
5   </body>  
6 </html>
```

Nhúng HTML vào PHP

```
● ● ●  
1 <?php  
2   echo "<h1>Chào mừng đến với HUTECH!</h1>";  
3 ?>  
4
```

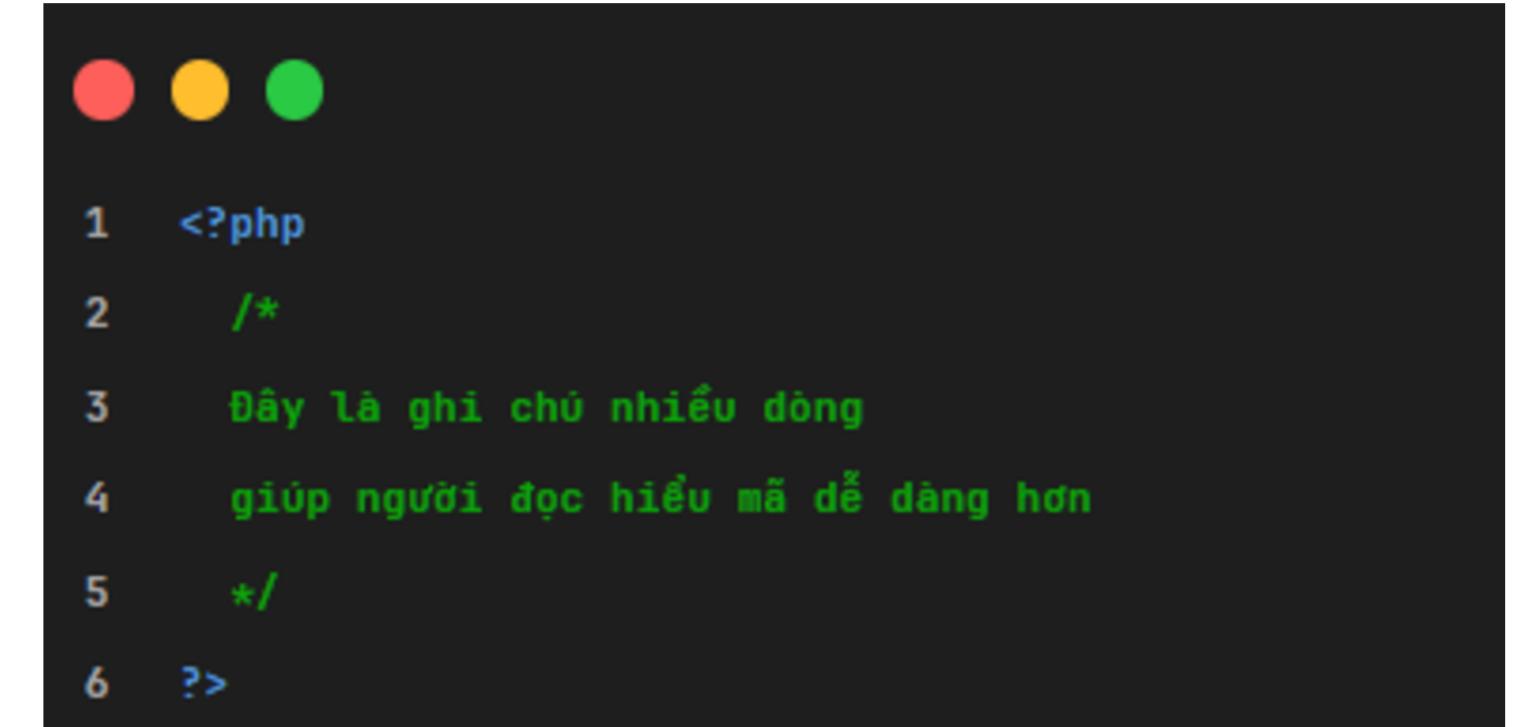
CÁCH GHI CHÚ (COMMENT) TRONG PHP

Ghi chú một dòng



```
● ● ●  
1 <?php  
2 // Đây là ghi chú trên một dòng  
3 ?>  
4
```

Ghi chú nhiều dòng



```
● ● ●  
1 <?php  
2 /*  
3 Đây là ghi chú nhiều dòng  
4 giúp người đọc hiểu mã dễ dàng hơn  
5 */  
6 ?>
```

BIẾN TRONG PHP



```
1 <?php
2     $name = "Dương Quốc Nam"; // String - Chuỗi
3     $age = 25;                // Integer - Số nguyên
4     $height = 1.80;            // Float - Số thực
5     $isTeacher = true;         // True/False - Boolean
6     $subjects = ["PHP", "ASP.Net", "Java"]; // Array
7
8     // In giá trị biến
9     echo "Tên: " . $name;
10    echo "Tuổi: $age"; // PHP cho phép biến trong chuỗi
11 ?>
```

Quy tắc đặt tên biến:

- Bắt đầu bằng ký tự \$
- Chữ cái hoặc dấu gạch dưới (_)
- Không chứa *khoảng trắng*
- Phân biệt chữ hoa và chữ thường

Ví dụ: \$Name khác \$name

CÁC KIỂU DỮ LIỆU TRONG PHP

PHP có các kiểu dữ liệu chính:

- String (chuỗi)
- Integer (số nguyên)
- Float (số thực)
- Boolean (true / false)
- Array (mảng)
- Object (đối tượng)
- NULL

```
● ● ●  
1  <?php  
2      $name = "Dương Quốc Nam"; // String - Chuỗi  
3      $age = 25;                // Integer - Số nguyên  
4      $height = 1.80;           // Float - Số thực  
5      $isTeacher = true;       // True/False - Boolean  
6      $subjects = ["PHP", "ASP.Net", "Java"]; // Array  
7      $value = null;          // NULL  
8  
9      // Object  
10     class SinhVien {  
11         public $name;  
12         public $age;  
13     }  
14     $sv = new SinhVien();  
15 ?>
```

KIỂU DỮ LIỆU UNION

Định nghĩa:

- Union Type cho phép một biến có thể có nhiều kiểu dữ liệu
- Hữu ích khi một hàm có thể nhận nhiều loại dữ liệu đầu vào.

```
● ● ●  
1 <?php  
2     function setValue(int|string $value) {  
3         echo "Giá trị: " . $value;  
4     }  
5  
6     setValue(10); // Hợp lệ  
7     setValue("Hello"); // Hợp lệ  
8 ?>
```

ÉP KIỂU TRONG PHP

Ép kiểu giúp chuyển đổi dữ liệu sang kiểu khác.

Kiểu	Cách dùng
Integer	(int) \$var hoặc (integer) \$var
Float	(float) \$var hoặc (double) \$var
String	(string) \$var
Boolean	(bool) \$var hoặc (boolean) \$var
Array	(array) \$var
Object	(object) \$var



```
1 <?php
2 $var = "100";
3 $num = (int) $var; // Chuyển sang số nguyên
4 echo $num; // 100
5 ?>
```

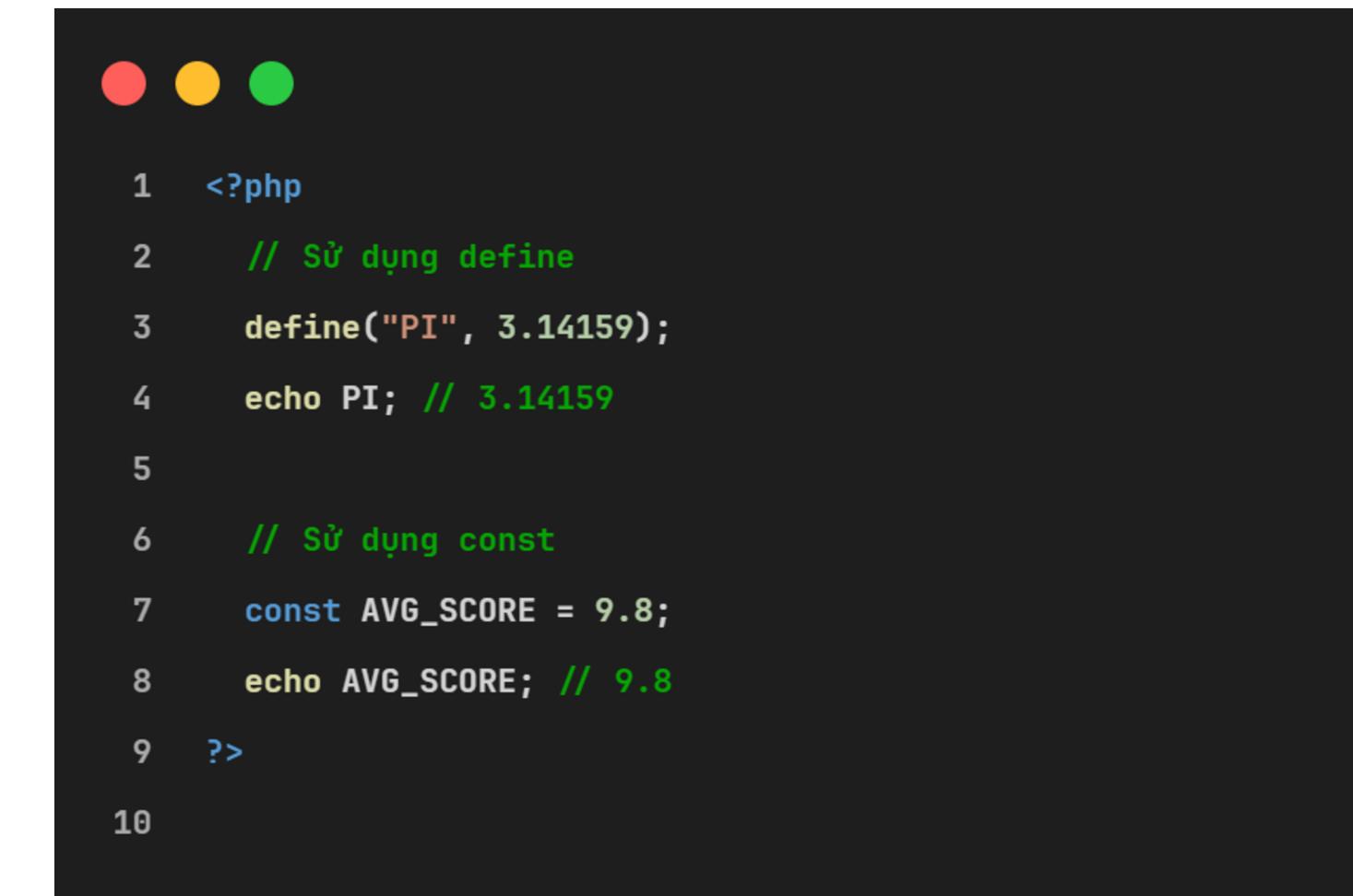
CÁCH ĐỊNH NGHĨA HẰNG SỐ

Định nghĩa:

- Hằng số là giá trị cố định trong suốt chương trình
- Có thể khai báo bằng define() hoặc const

So sánh:

Cách khai báo	Hỗ trợ trong phạm vi hàm?	Hỗ trợ hằng số mảng?
define()	<input checked="" type="checkbox"/> Có	<input checked="" type="checkbox"/> Có
const	<input checked="" type="checkbox"/> Không	<input checked="" type="checkbox"/> Có



```
1 <?php
2 // Sử dụng define
3 define("PI", 3.14159);
4 echo PI; // 3.14159
5
6 // Sử dụng const
7 const AVG_SCORE = 9.8;
8 echo AVG_SCORE; // 9.8
9 ?>
10
```

ESCAPE CHARACTERS TRONG PHP

Escape Characters dùng để hiển thị các ký tự đặc biệt

Các ký tự phổ biến:

Ký tự	Ý nghĩa
\n	Xuống dòng mới
\t	Tab
\\"	Hiển thị dấu \
\\"	Hiển thị dấu " "
\\$	Hiển thị ký hiệu \$



```
1 <?php
2 echo "Dòng 1\nDòng 2"; // Xuống dòng
3 echo "Giá trị: \$100"; // Giá trị: $100
4 ?>
5
```

PHÉP TOÁN TRONG PHP

Toán tử số học

```
1 <?php
2 $a = 10;
3 $b = 3;
4
5 echo $a + $b; // Cộng: 13
6 echo $a - $b; // Trừ: 7
7 echo $a * $b; // Nhân: 30
8 echo $a / $b; // Chia: 3.333...
9 echo $a % $b; // Chia lấy dư: 1
10 echo $a ** $b; // Lũy thừa: 1000
11 ?>
```

Toán tử gán

```
1 <?php
2 $x = 5;           // Gán giá trị
3 $x += 3;          // $x = $x + 3
4 $x -= 2;          // $x = $x - 2
5 $x *= 4;          // $x = $x * 4
6 $x /= 2;          // $x = $x / 2
7 $x %= 3;          // $x = $x % 3
8 ?>
```

TOÁN TỬ SO SÁNH TRONG PHP

Toán tử so sánh dùng để so sánh hai giá trị, kết quả trả về true hoặc false

Toán tử	Ý nghĩa	Ví dụ	Kết quả
<code>==</code>	Bằng	<code>5 == 5</code>	true
<code>===</code>	Bằng (và cùng kiểu)	<code>"5" === 5</code>	false
<code>!=</code> hoặc <code><></code>	Khác	<code>5 != 3</code>	true
<code>!==</code>	Khác (hoặc khác kiểu)	<code>"5" !== 5</code>	true
<code>></code>	Lớn hơn	<code>10 > 5</code>	true
<code><</code>	Nhỏ hơn	<code>3 < 8</code>	true
<code>>=</code>	Lớn hơn hoặc bằng	<code>10 >= 10</code>	true
<code><=</code>	Nhỏ hơn hoặc bằng	<code>5 <= 8</code>	true

TOÁN TỬ LOGIC TRONG PHP

Toán tử logic dùng để kết hợp các biểu thức điều kiện

Toán tử	Ý nghĩa	Ví dụ
&& hoặc and	Và	$($x > 0) \&\& ($y > 0)$
'		hoặc or `
!	Phủ định	$!($x > 5)$
xor	Hoặc duy nhất (chỉ một trong hai đúng)	true xor false // true



```
1 <?php
2 $age = 25;
3 $hasID = true;
4
5 if ($age > 18 && $hasID) {
6     echo "Bạn được phép vào.";
7 }
8 ?>
```

XỬ LÝ CHUỖI TRONG PHP

Định nghĩa:

- Chuỗi là một dãy ký tự nằm trong dấu nháy đơn (' ') hoặc nháy kép (" ")
- Có thể chứa chữ cái, số, ký tự đặc biệt

Sự khác biệt:

- Nháy kép (" "): biến trong chuỗi sẽ được nội suy (thay thế giá trị của biến)
- Nháy đơn (' '): biến trong chuỗi không được nội suy (hiển thị nguyên dạng)

```
1  <?php
2      $lastName = "Duong";
3      $firstName = "Quoc Nam";
4
5      // Cách 1: Nối chuỗi sử dụng dấu chấm
6      $fullName = $lastName . " " . $firstName; // "Duong Quoc Nam"
7
8      // Cách 2: Nối chuỗi sử dụng nháy kép
9      $introduce = "Xin chào $fullName!";
10
11     // Một số hàm xử lý chuỗi
12     echo strlen($fullName);           // Độ dài chuỗi
13     echo strtoupper($fullName);       // Chuyển thành chữ hoa
14     echo strtolower($fullName);        // Chuyển thành chữ thường
15     echo trim($fullName);            // Xóa khoảng trắng đầu cuối
16 ?>
```

TOÁN TỬ NỐI CHUỖI

Định nghĩa:

- Dùng dấu chấm (.) để nối chuỗi.
- Sử dụng toán tử gán nối (.=) để nối chuỗi vào một biến

So sánh các dùng chuỗi:

Cách dùng	Biến trong chuỗi	Ví dụ	Kết quả
Nháy đơn ('')	Không thay thế biến	'Hello \$name'	Hello \$name
Nháy kép ("")	Thay thế giá trị biến	Hello \$name	Hello John
Nối chuỗi (.)	Kết hợp các chuỗi	\$a . "World"	Hello World
Gán nối (.=)	Thêm chuỗi vào biến	\$a .= "World"	Hello World



```
1 <?php
2 $a = "Hello";
3 $b = $a . " World!";
4
5 echo $b; // Output: Hello World!
6
7 $a = "Hello";
8 $a .= " World!"; // Tương đương $a = $a . " World!";
9
10 echo $a; // Output: Hello World!
11 ?>
```

XỬ LÝ MẢNG TRONG PHP

PHP có các loại mảng:

- Mảng một chiều (*Indexed Array*)
- Mảng kết hợp (*Associative Array*)
- Mảng đa chiều

```
● ● ●  
1  <?php  
2      // Mảng một chiều  
3  $fruits = ["Apple", "Orange", "Banana"];  
4  echo $fruits[0]; // In ra: Apple  
5  
6      // Mảng kết hợp  
7  $students = [  
8      "fullName" => "Duong Quoc Nam",  
9      "age" => 20,  
10     "score" => 8.5  
11 ];  
12 echo $students["fullName"]; // In ra: Duong Quoc Nam  
13  
14      // Mảng đa chiều  
15  $class = [  
16      ["fullName" => "Nam", "score" => 8],  
17      ["fullName" => "Tuyet", "score" => 10]  
18  ];  
19 echo $class[1]["fullName"]; // In ra: Tuyet  
20 ?>
```

CẤU TRÚC ĐIỀU KIỆN IF - ELSE

Cấu trúc chung

```
1 <?php  
2     if ($dieu_kien) {  
3         // Lệnh nếu điều kiện đúng  
4     } else {  
5         // Lệnh nếu điều kiện sai  
6     }  
7 ?>
```

Ví dụ minh họa

```
1 <?php  
2     $score = 75;  
3  
4     if ($score ≥ 90) {  
5         echo "Xuất sắc";  
6     } elseif ($score ≥ 80) {  
7         echo "Giỏi";  
8     } elseif ($score ≥ 70) {  
9         echo "Khá";  
10    } else {  
11        echo "Trung bình";  
12    }  
13 ?>
```

CẤU TRÚC SWITCH - CASE

Cấu trúc chung



```
1 <?php
2     switch ($variable) {
3         case 'value_1':
4             // Lệnh thực thi nếu biến có giá trị 1
5             break;
6         case 'value_2':
7             // Lệnh thực thi nếu biến có giá trị 2
8             break;
9         default:
10            // Lệnh thực thi nếu không có giá trị nào khớp
11    }
12 ?>
```

Ví dụ minh họa



```
1 <?php
2     $color = "red";
3
4     switch ($color) {
5         case "red":
6             echo "You choose red!";
7             break;
8         case "blue":
9             echo "You choose blue!";
10            break;
11        default:
12            echo "Invalid color. Please select again!";
13    }
14 ?>
```

PHƯƠNG THỨC MATCH (PHP 8)

match giống switch nhưng ngắn gọn hơn, dễ đọc hơn so với switch-case, không cần break

```
● ● ●  
1 <?php  
2 // Kiểu cũ  
3 $status = 2;  
4 switch ($status) {  
5     case 1:  
6         echo "Chờ xử lý";  
7         break;  
8     case 2:  
9         echo "Đã xác nhận";  
10        break;  
11    default:  
12        echo "Không hợp lệ";  
13    }  
14 ?>
```

```
● ● ●  
1 <?php  
2 // Kiểu mới sử dụng match  
3 $status = 2;  
4  
5 $message = match($status) {  
6     1 => "Chờ xử lý",  
7     2 => "Đã xác nhận",  
8     3 => "Hoàn thành",  
9     default => "Không hợp lệ"  
10 };  
11  
12 echo $message; // Đã xác nhận  
13 ?>
```

VÒNG LẶP TRONG PHP

Cấu trúc chung

```
● ● ●  
1  <?php  
2      // Vòng lặp for  
3      for ($i = 1; $i ≤ 10; $i++) {  
4          // Lệnh thực thi  
5      }  
6  
7      // Vòng lặp while  
8      while ($dieu_kien) {  
9          // Lệnh thực thi  
10     }  
11  
12     // Vòng lặp foreach với mảng  
13     foreach ($mang as $phan_tu) {  
14         // Lệnh thực thi  
15     }  
16 ?>
```

Ví dụ minh họa

```
● ● ●  
1  <?php  
2      // Vòng lặp for  
3      for ($i = 1; $i ≤ 5; $i++) {  
4          echo "Số: $i<br>";  
5      }  
6  
7      // Vòng lặp while  
8      $count = 0;  
9      while ($count < 5) {  
10         echo "Count: $count<br>";  
11         $count++;  
12     }  
13  
14     // Vòng lặp foreach với mảng  
15     $subjects = ["PHP", "Java", "ASP.Net"];  
16     foreach ($subjects as $item) {  
17         echo "Môn học: $item<br>";  
18     }  
19 ?>
```

HÀM (FUNCTION)

Hàm có tham số

```
● ● ●  
1 <?php  
2     function tong($a, $b) {  
3         return $a + $b;  
4     }  
5  
6 // In giá trị của hàm ra màn hình  
7 echo tong(5, 3); // 8  
8 ?>
```

Hàm không có tham số

```
● ● ●  
1 <?php  
2     function sayHello(){  
3         echo "Hello World";  
4     }  
5  
6 // Gọi hàm thực thi  
7 sayHello() // Hello World  
8 ?>
```

HÀM NẶC DANH (ANONYMOUS FUNCTION)

Hàm không có tên, thường dùng để truyền vào hàm khác.



```
1 <?php
2     $sayHello = function($name) {
3         return "Xin chào, $name!";
4     };
5
6     echo $sayHello("Quốc Nam"); // Xin chào, Quốc Nam!
7 ?>
```

CALLBACK FUNCTION

Hàm callback là hàm được truyền như một đối số

```
● ● ●

1 <?php
2     function sayHello($name, $callback) {
3         $msg = "Xin chào, $name!";
4         $callback($msg);
5     }
6
7     sayHello("Quốc Nam", function($msg) {
8         echo strtoupper($msg); // XIN CHÀO, QUỐC NAM!
9     });
10 ?>
```

HÀM CÓ THAM SỐ TÙY CHỌN

Tham số có giá trị mặc định, không bắt buộc truyền

```
1 <?php
2     function sayHi($name = "Nam") {
3         echo "Xin chào, $name!";
4     }
5
6     sayHi(); // Xin chào, Nam!
7     sayHi("Dev"); // Xin chào, Dev!
8 ?>
```

ĐỐI SỐ ĐƯỢC ĐẶT TÊN

PHP 8 hỗ trợ đối số có tên, giúp truyền tham số không theo thứ tự

```
1 <?php
2     function introduce($name, $age, $country) {
3         echo "$name, $age tuổi, đến từ $country.";
4     }
5
6     // Hiển thị: Tuyết, 22 tuổi, đến từ Việt Nam
7     introduce(age: 22, name: "Tuyết", country: "Việt Nam");
8 ?>
```

ARROW FUNCTION

Phiên bản rút gọn của hàm nặc danh. Viết gọn hơn so với function()

```
1 <?php
2 // This only works in PHP 7.4 and above
3 $bindingNumber = fn($x) => $x * 2;
4
5 $printFullName = fn($fullName) => $fullName;
6
7 echo $bindingNumber(4); // 8
8
9 echo $printFullName("Dương Quốc Nam") // Dương Quốc Nam
10 ?>
```

CLASS VÀ OBJECT TRONG PHP

Giới thiệu về class:

- Class là một khuôn mẫu dùng để tạo ra các đối tượng
- Một class chứa:
 - Thuộc tính (properties): biến lưu trữ dữ liệu
 - Phương thức (methods): hàm xử lý dữ liệu

Giới thiệu về object:

- Object là một thực thể cụ thể của class, được tạo bằng từ khóa **new**

```
● ● ●  
1  <?php  
2  class Car {  
3      public $brand; // Thuộc tính  
4      public $color;  
5  
6      public function showInfo() { // Phương thức  
7          return "Xe hãng: $this->brand, màu: $this->color";  
8      }  
9  }  
11 // Tạo đối tượng  
12 $car1 = new Car();  
13 $car1->brand = "Toyota";  
14 $car1->color = "Đỏ";  
15  
16 echo $car1->showInfo(); // Output: Xe hãng: Toyota, màu: Đỏ  
17 ?>
```

CONSTRUCTOR VÀ DESTRUCTOR

Constructor:

- Hàm khởi tạo tự động chạy khi đối tượng được tạo.
- Được khai báo với từ khóa `__construct()`.

Destructor:

- Hàm hủy đối tượng khi chương trình kết thúc.
- Được khai báo với từ khóa `__destruct()`.

```
1  <?php
2      class Car {
3          public $brand;
4          public $color;
5
6          public function __construct($brand, $color) {
7              $this->brand = $brand;
8              $this->color = $color;
9          }
10
11         public function showInfo() {
12             return "Xe hãng: $this->brand, màu: $this->color";
13         }
14
15         public function __destruct() {
16             echo "Đối tượng đã bị hủy.";
17         }
18     }
19
20     $car1 = new Car("Honda", "Xanh");
21     echo $car1->showInfo();
22 ?>
```

TÍNH ĐÓNG GÓI (ENCAPSULATION)

Định nghĩa:

- Đóng gói giúp bảo vệ dữ liệu và chỉ cho phép truy cập có kiểm soát.
- Dùng các modifier: public, private, protected.

Modifier	Truy cập trong class	Truy cập trong class kế thừa	Truy cập từ bên ngoài
public	✓ Có	✓ Có	✓ Có
protected	✓ Có	✓ Có	✗ Không
private	✓ Có	✗ Không	✗ Không

```

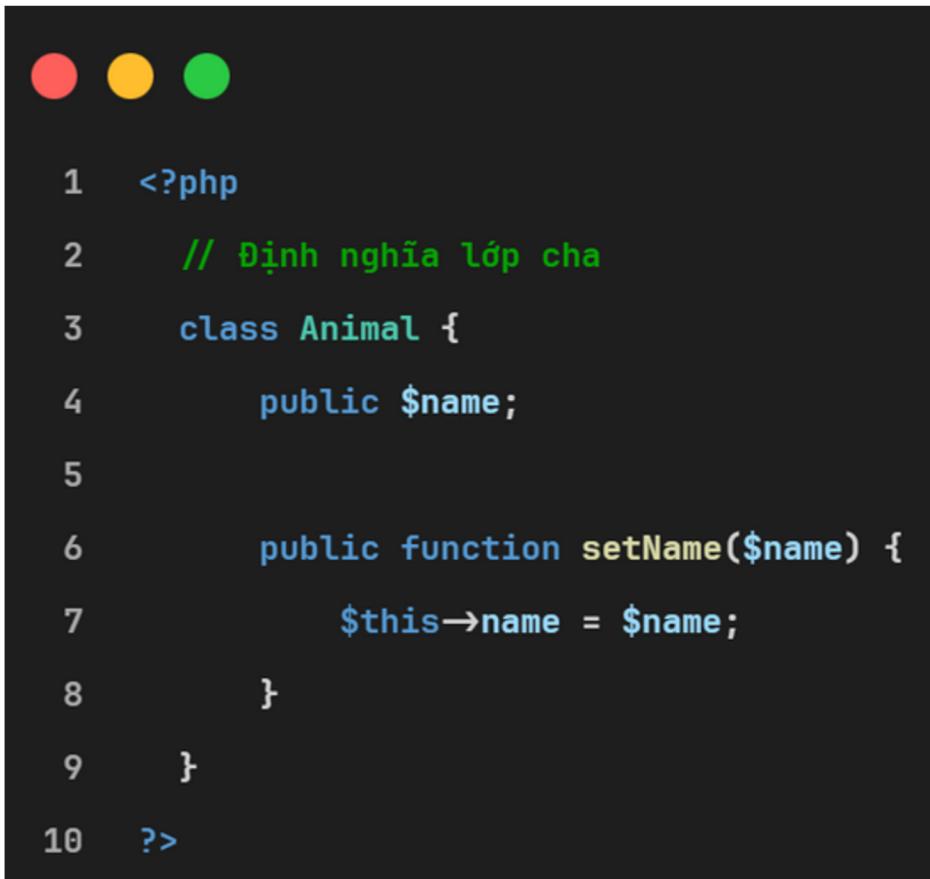
1  <?php
2  class Person {
3      private $name; // Chỉ truy cập được trong class
4      protected $salary; // Truy cập trong class và class kế thừa
5      public $class; // Truy cập ở mọi nơi trong class
6
7      public function setName($name) {
8          $this->name = $name;
9      }
10
11     public function getName() {
12         return $this->name;
13     }
14 }
15
16 $p = new Person();
17 $p->setName("Dương Quốc Nam");
18 echo $p->getName(); // Output: Dương Quốc Nam
19 ?>

```

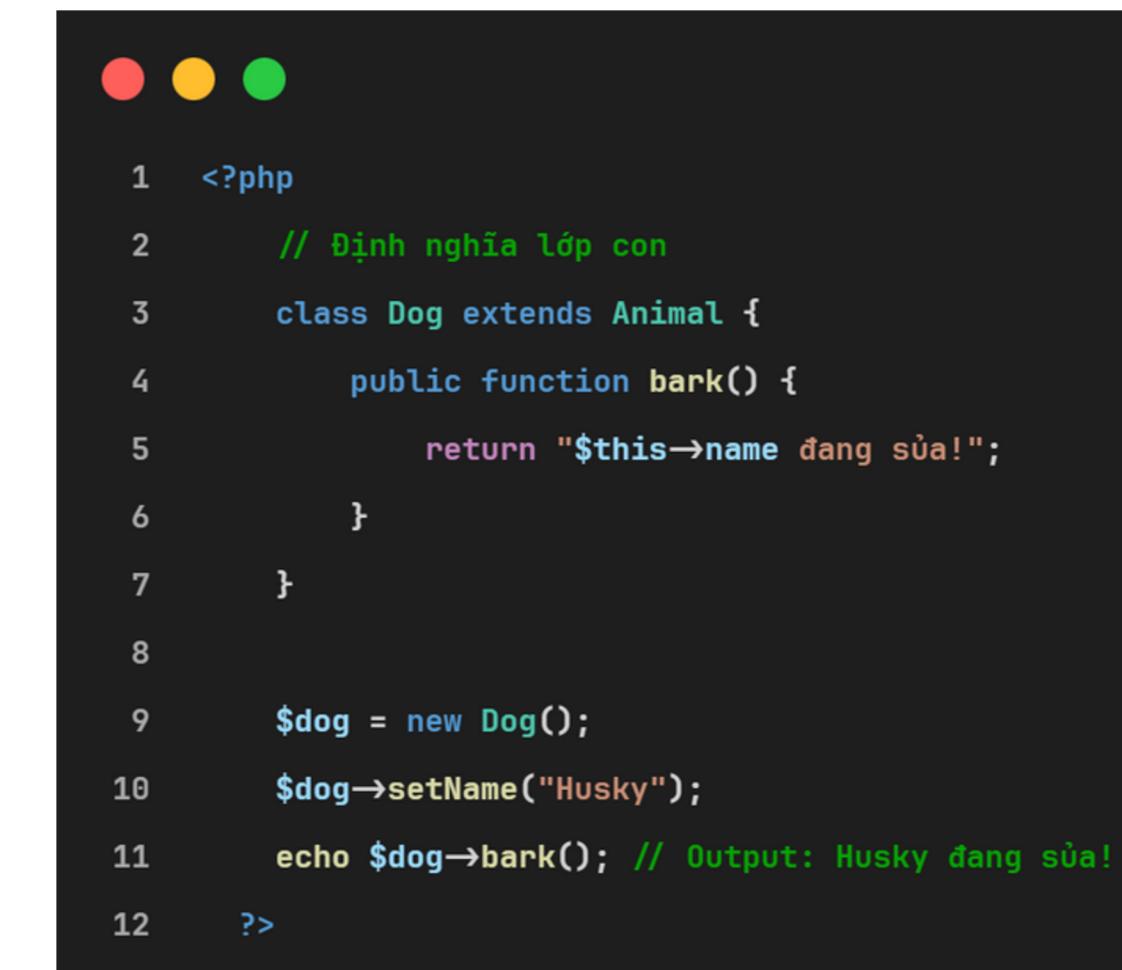
TÍNH KẾ THỪA (INHERITANCE)

Định nghĩa:

- Kế thừa giúp một class tái sử dụng các thuộc tính/phương thức của class khác
- Dùng từ khóa **extends** để kế thừa



```
1 <?php
2 // Định nghĩa lớp cha
3 class Animal {
4     public $name;
5
6     public function setName($name) {
7         $this->name = $name;
8     }
9 }
10 ?>
```

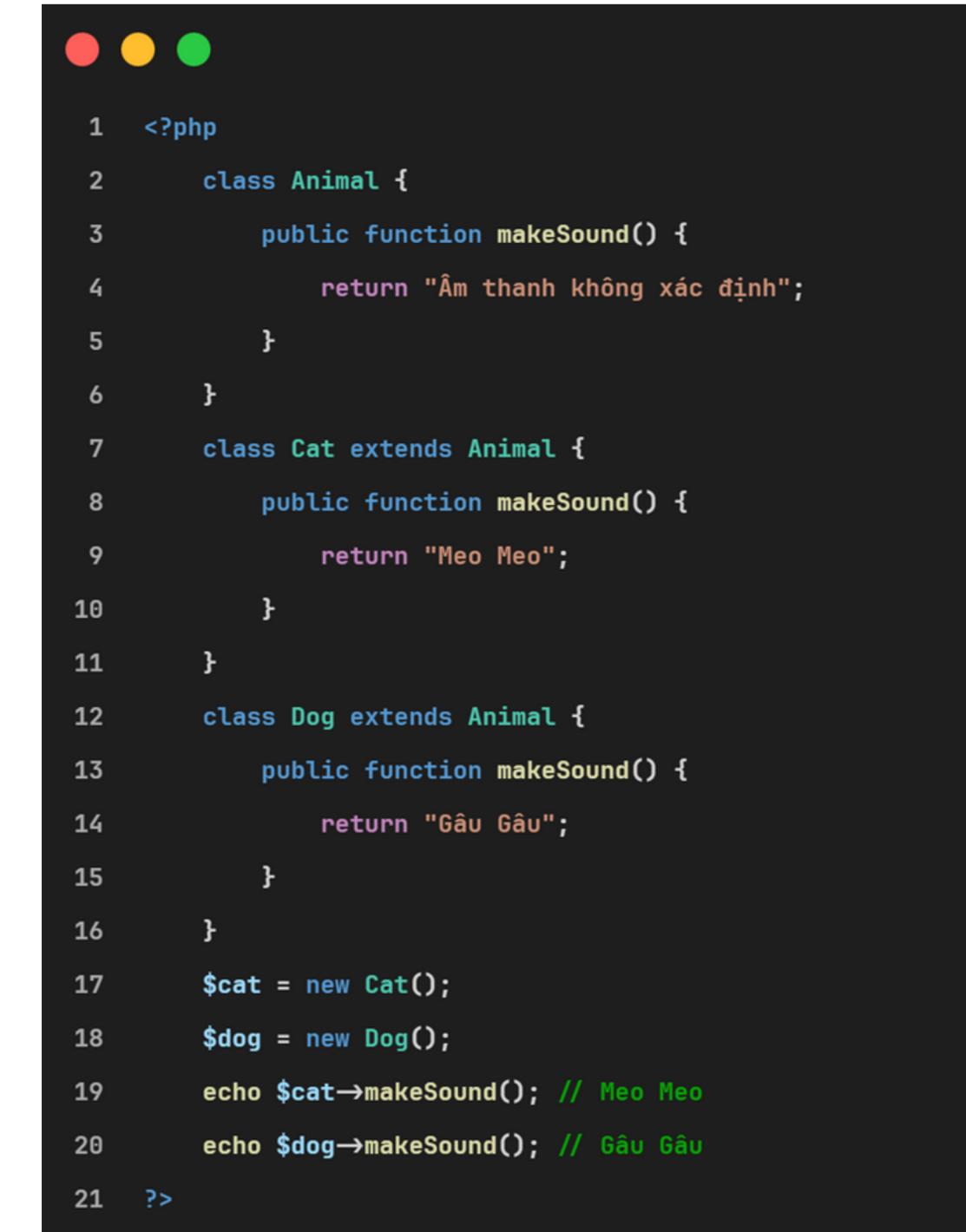


```
1 <?php
2 // Định nghĩa lớp con
3 class Dog extends Animal {
4     public function bark() {
5         return "$this->name đang sủa!";
6     }
7 }
8
9 $dog = new Dog();
10 $dog->setName("Husky");
11 echo $dog->bark(); // Output: Husky đang sủa!
12 ?>
```

TÍNH ĐA HÌNH (POLYMORPHISM)

Định nghĩa:

- Đa hình cho phép một phương thức có nhiều cách triển khai khác nhau



The screenshot shows a code editor with a dark theme. At the top left are three colored circular icons: red, yellow, and green. The code itself is written in PHP and illustrates polymorphism through inheritance. It defines an `Animal` class with a `makeSound()` method that returns a generic sound. Two subclasses, `Cat` and `Dog`, inherit from `Animal` and override the `makeSound()` method to return specific cat and dog sounds respectively. Finally, the code creates instances of `Cat` and `Dog`, and prints their sounds using the `echo` statement.

```
1 <?php
2     class Animal {
3         public function makeSound() {
4             return "Âm thanh không xác định";
5         }
6     }
7     class Cat extends Animal {
8         public function makeSound() {
9             return "Meo Meo";
10        }
11    }
12    class Dog extends Animal {
13        public function makeSound() {
14            return "Gâu Gâu";
15        }
16    }
17    $cat = new Cat();
18    $dog = new Dog();
19    echo $cat->makeSound(); // Meo Meo
20    echo $dog->makeSound(); // Gâu Gâu
21 ?>
```

ABSTRACT CLASS

Định nghĩa:

- Abstract Class là class không thể khởi tạo trực tiếp, dùng để làm cha cho các class con
- Chứa các phương thức abstract mà các class con phải triển khai

```
● ● ●  
1  <?php  
2  abstract class Animal {  
3      abstract public function makeSound();  
4  }  
5  
6  class Dog extends Animal {  
7      public function makeSound() {  
8          return "Gâu Gâu";  
9      }  
10 }  
11  
12 $dog = new Dog();  
13 echo $dog->makeSound(); // Output: Gâu Gâu  
14 ?>
```

INTERFACE TRONG PHP

Constructor:

- Interface là một bản mẫu chứa các phương thức trừu tượng.
- Một class có thể implement nhiều interface.

```
● ● ●  
1  <?php  
2      interface Animal {  
3          public function makeSound();  
4      }  
5  
6      interface CanRun {  
7          public function run();  
8      }  
9  
10     class Dog implements Animal, CanRun {  
11         public function makeSound() {  
12             return "Gâu Gâu";  
13         }  
14  
15         public function run() {  
16             return "Chạy rất nhanh!";  
17         }  
18     }  
19  
20     $dog = new Dog();  
21     echo $dog->makeSound(); // Gâu Gâu  
22     echo $dog->run(); // Chạy rất nhanh!  
23 ?>
```