

## CHƯƠNG 3\_4

# NGÔN NGỮ LẬP TRÌNH PHP

---

# **Nội dung giảng dạy**

---

- Giới thiệu
- PHP cơ bản
- Các phép toán
- Cấu trúc điều khiển
- Hàm
- Xử lý Form
- Mảng
- Chuỗi
- Include và Require
- Cookie và Session



# Giới thiệu

- PHP (“Hypertext Preprocessor”) là một ngôn ngữ lập trình kịch bản chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở
- Tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học
- Là ngôn ngữ lập trình web được sử dụng rất phổ biến
  - PHP đủ mạnh để viết các hệ thống web blog lớn (WordPress, ...)
  - PHP đủ mạnh chạy các hệ thống web mạng xã hội lớn (Facebook, ...)
  - PHP dễ dàng cho người mới bắt đầu học làm trình web phía Server





# Giới thiệu

- Một số đặc điểm:
  - Tập tin bao gồm text, HTML, CSS, JavaScript, và mã PHP, ...)
  - Mã PHP được thực thi phía Server, sau đó trả kết quả về cho trình duyệt (browser) dạng HTML
  - Phần mở rộng tập tin PHP dạng .php
  - PHP chạy độc lập trên nhiều hệ điều hành: Windows, Linux, MacOS
  - Hỗ trợ nhiều cơ sở dữ liệu (thường sử dụng với cơ sở dữ liệu MySQL, MariaDB)
  - Có nhiều framework theo mô hình MVC để tạo website và dễ bảo trì (Lavarel, Zend, Codeigniter, ...)



# Giới thiệu

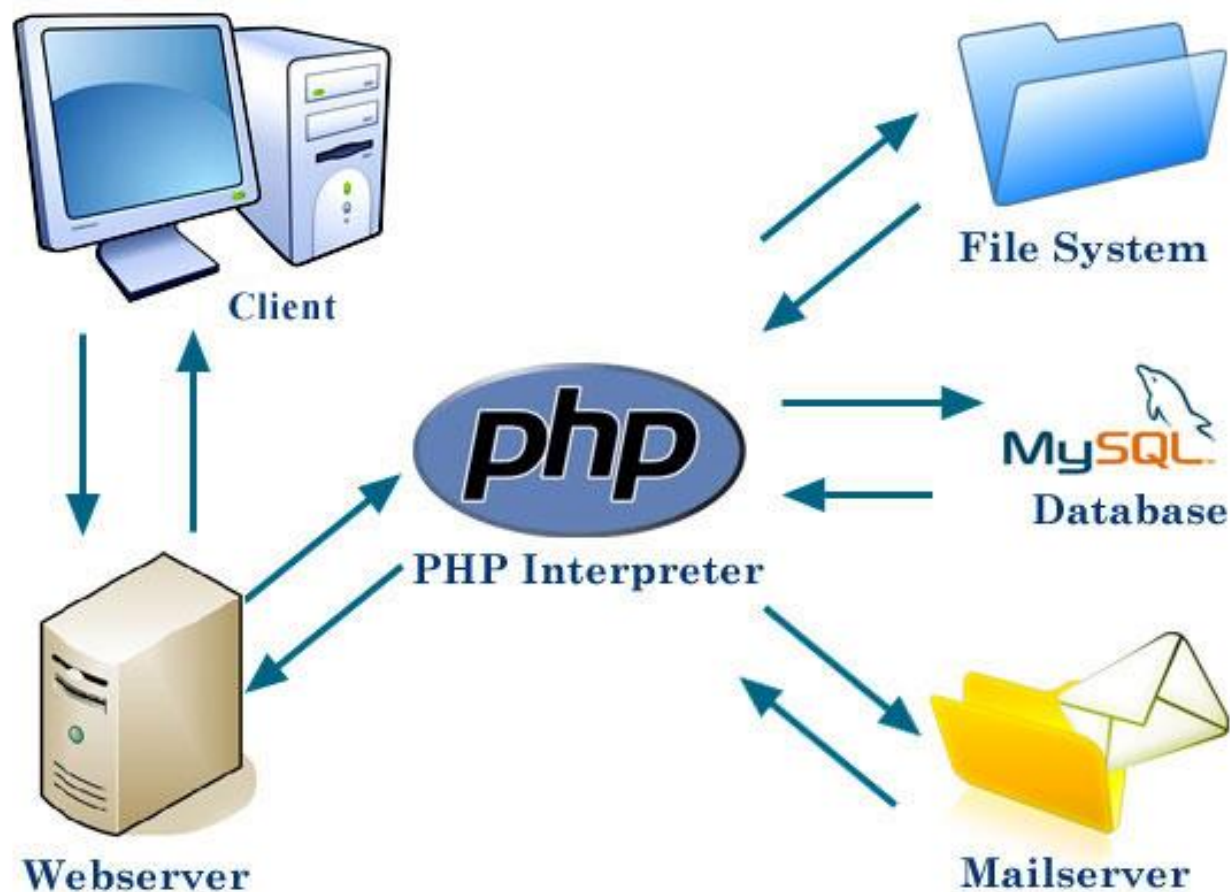
- So sánh ASP.NET và PHP

	ASP.NET	PHP
Dạng	Web Application Framework được phát triển bởi Microsoft	Ngôn ngữ Kịch bản phía máy chủ, mã nguồn mở
Quy mô	Phù hợp với các website trung bình và lớn	Phù hợp với các website trung bình và nhỏ
Chi phí	Mất phí bản quyền	Miễn phí và có sẵn
Giải pháp	Tập trung vào bảo mật và chức năng	Tập trung vào giao diện tương tác
Cộng đồng	Cộng đồng nhỏ hơn PHP	Cộng đồng khổng lồ, lớn hơn ASP.NET
Bảo mật	Bảo mật cao	Ít bảo mật hơn ASP.NET
Tốc độ	Nhanh (tùy từng phiên bản và mục đích)	Nhanh (tùy từng phiên bản và mục đích)
Tính tùy biến	Tính tùy biến thấp	Cho phép tùy biến cao



# Giới thiệu

- Mô hình hoạt động





# Giới thiệu

---

- Công cụ soạn thảo PHP
  - PhpStorm
  - Sublime Text
  - NetBeans
  - Eclipse
  - Notepad++
  - PHP Designer
  - Visual Studio Code
  - ...

# **Cú pháp cơ bản**

---

- Cú pháp:
  - Cách 01: Cú pháp chính `<?php Mã lệnh PHP ?>` (thường dùng)
  - Cách 02: Cú pháp ngắn gọn `<? Mã lệnh PHP ?>`
  - Cách 03: Cú pháp giống ASP `<% Mã lệnh PHP %>`
  - Cách 04: Cú pháp bắt đầu bằng Script
    - `<script language=php>`
    - ....
    - `</script>`
- Chú thích:
  - Dùng `//` nếu trên một dòng
  - Dùng `/* ... */` nếu trên nhiều dòng





# Khai báo biến

- Tuân theo quy tắc đặt tên biến
- Bắt đầu bằng dấu \$
- Cú pháp:
  - \$ten\_bien = gia\_tri\_khoi\_tao;
- Hiển thị biến chung với chuỗi:
  - Thêm dấu . hoặc ,
  - Thêm \$ten\_bien trong chuỗi
- Lấy giá trị biến làm tên biến:

```
<?php
    $fullName = "myString";
    $$fullName = "Dai hoc Vinh";
    echo $myString;

?>
```

```
<?php
    $fullName = "Dai hoc Vinh";
    echo $fullName;

?>
```

```
<?php
    $fullName = "Dai hoc Vinh";
    echo "Truong " . $fullName;

?>
```

```
<?php
    $fullName = "Dai hoc Vinh";
    echo "Truong {$fullName}";

?>
```

# Hằng số

- Hằng không có dấu \$ ở trước tên
- Hằng có thể truy cập bất cứ vị trí nào trong mã lệnh
- Hằng chỉ được phép gán giá trị duy nhất 1 lần
- Hằng thường viết bằng chữ in để phân biệt với biến
- Khai báo:
  - `define('ten_hang', 'gia_tri')`
  - `const ten_hang`

```
<?php
    define('SDT', '0909090909');
    echo SDT;
?>
```

```
<?php
    const MESSAGE="Hello const PHP";
    echo MESSAGE;
?>
```



# Lệnh echo và print

- Chức năng: in chuỗi ra màn hình
  - Echo: không có giá trị trả về
  - Print: trả về giá trị 1
  - In ký tự đặc biệt: dùng dấu \

```
<?php
    if(print "test"){
        print "kiem tra";
    }
?>
```

```
<?php
    $myArray = array("a", "b", "c");
    Echo "<pre>";
    print_r($myArray);
    Echo "</pre>";
?>
```

```
<?php
echo "Hello PHP Echo!";
?>
```

```
<?php
    echo "Hello PHP Echo!
        đây là nhiều dòng
        văn bản được in bởi
        câu lệnh echo PHP";
?>
```

```
<?php
    echo "Xin chao \"ban\"";
?>
```



# Kiểu dữ liệu

- Kiểu INT: số nguyên
  - (int)\$ten\_bien: ép kiểu
  - Is\_int(\$ten\_bien): kiểm tra có phải là int hay không
- Kiểu boolean
- Kiểu số thực (float, double)
- Kiểu chuỗi
  - Nháy ' (đơn): in ra một chuỗi dữ liệu bình thường
  - Nháy " (kép): in ra một chuỗi dữ liệu có đánh giá biến và các ký tự đặc biệt
- Kiểu NULL
- Kiểu đối tượng (object)

# Kiểu dữ liệu

- Kiểu mảng (array): Mảng một chiều
  - Khai báo: `$ten_mang = array()` hoặc `array('cac_phan_tu')`

```
<?php
$sinhvien = array('A', 'B');
print_r($sinhvien);
?>
```

```
<?php
$sinhvien = array(
0 => 'A',
1 => 'B'
);
print_r($sinhvien);
?>
```

```
<?php
$sinhvien = array();
$sinhvien[0] = 'A';
$sinhvien[1] = 'B';
print_r($sinhvien);
?>
```

- Truy xuất: `$ten_mang[index]`

```
<?php
$sinhvien = array('A', 'B');
echo $sinhvien[0];
echo $sinhvien[1];
?>
```

```
<?php
$sinhvien = array();
$sinhvien['sv_a'] = 'A';
$sinhvien['sv_b'] = 'B';
echo $sinhvien['sv_a'];
echo $sinhvien['sv_b'];
?>
```

# Kiểu dữ liệu

- Kiểu mảng (array): Mảng hai chiều
  - Khai báo:

```
<?php
$a = array();
$a[0][1] = 123;
$a[0][2] = 321;
?>
```

```
<?php
$a = array();
$a[0][1] = 123;
$a[0][2] = 321;
echo $a[0][1]; // in ra giá trị 123
echo $a[0][2]; // in ra giá trị 321
?>
```

- Kiểm tra một biến kiểu mảng: `is_array($ten_bien)`



# Phạm vi của biến

---

- PHP có ba phạm vi biến khác nhau:
  - Địa phương (local)
  - Toàn cục (global)
  - Tĩnh (static)
  - Các từ khóa:
    - Global: truy cập vào một biến toàn cầu từ bên trong một hàm
    - Static: biến không bị mất giá trị khi kết thúc hàm

# Phạm vi của biến

- Ví dụ:

```
<?php
$x = 5;
$y = 10;
function myTest() {
    global $x, $y;
    $y = $x + $y;
}
myTest();
echo $y;
?>
```

```
<?php
function myTest() {
    static $x = 0;
    echo $x . "<br>";
    $x++;
}
myTest();
myTest();
myTest();
?>
```



# **Biến siêu toàn cục**

- Là biến được định nghĩa trước
  - \$GLOBALS: truy cập các biến toàn cục từ bất kỳ đâu
    - Sử dụng: `$GLOBALS['ten_bien']`
  - \$\_SERVER: thông tin về tiêu đề, đường dẫn, IP, ...
  - \$\_REQUEST: thu thập dữ liệu sau khi gửi html form (thu thập giá trị của trường nhập, cả phương thức post và get)
  - \$\_POST: thu thập dữ liệu html form có method = "post" (dùng rộng rãi để truyền các biến)
  - \$\_GET: thu thập dữ liệu HTML Form có method = "get" (thu thập dữ liệu được gửi trong URL)
  - \$\_FILES: lưu thông tin file upload lên server
  - \$\_ENV: thông tin môi trường
  - \$\_COOKIE
  - \$\_SESSION



# Biến siêu toàn cục

- Một số ví dụ:

```
$x = 75;  
$y = 25;  
function addition()  
{  
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];  
}  
addition();  
echo $z;
```

```
echo $_SERVER['PHP_SELF'];  
echo "<br>";  
echo $_SERVER['SERVER_NAME'];  
echo "<br>";  
echo $_SERVER['HTTP_HOST'];  
echo "<br>";  
echo $_SERVER['HTTP_USER_AGENT'];  
echo "<br>";  
echo $_SERVER['SCRIPT_NAME'];
```



# Biến siêu toàn cục

- Một số ví dụ:

```
<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
    Name: <input type="text" name="fname">
    <input type="submit">
</form>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_REQUEST['fname'];
    if (empty($name)) {
        echo "Name is empty";
    } else {
        echo "Name is: " . $name;
    }
}
?>
```

```
<?php
echo "Hoc lap trinh " . $_GET["subject"] . " tai " . $_GET["location"];
?>
```

← → ↻ ⓘ localhost:8080/code/hello.php?subject=web&location=day



# Các phép toán

- Toán tử số học:
  - +, -, \*, /, %, ++, --
- Toán tử so sánh:
  - ==, ===, !=, <>, !==, >, <, >=, <=
- Toán tử gán:
  - =, +=, -=, \*=, /=, %=, &=, |=, ^=
- Toán tử logic:
  - &&, ||, !
- Toán tử điều kiện:
  - `dieu_kien ? bieu_thuc_1 : bieu_thuc_2`



# Cấu trúc điều khiển

- Câu lệnh if else
  - Câu lệnh if
  - Câu lệnh if else
  - Câu lệnh if ... elseif ... else
  - Câu lệnh điều kiện lồng nhau

```
<?php
$age = 20;
if ($age > 18) {
    echo "Tuổi lớn hơn 18";
}
?>
```

```
<?php
$number = 13;
if ($number % 2 == 0) {
    echo "Số $number là số c
hẵn.";
} else {
    echo "Số $number là số l
ẻ.";
}
?>
```

```
<?php
$marks = 75;
if ($marks < 50) {
    echo "Tach!";
} elseif ($marks >= 80 && $marks < 90) {
    echo "Xep loai B";
} elseif ($marks >= 90 && $marks < 100) {
    echo "xep loai A";
} else {
    echo "Xep loai TB";
}
?>
```



# Cấu trúc điều khiển

- Câu lệnh switch

```
<?php
switch (bieu_thuc) {
    case gia_tri_1:
        // Khối lệnh 1
        break; //tùy chọn
    case gia_tri_2:
        // Khối lệnh 2
        break; //tùy chọn
    .....
    case gia_tri_n:
        // Khối lệnh n
        break; //tùy chọn
    default:
        // Khối lệnh này được
        thực thi
        // nếu tất cả các điều
        kiện trên không thỏa mãn
}
?>
```

```
<?php
$num=20;
switch($num){
    case 10:
        echo("number is equals to 10");
        break;
    case 20:
        echo("number is equal to 20");
        break;
    case 30:
        echo("number is equal to 30");
        break;
    default:
        echo("Not in 10, 20 or 30");
}
?>
```



# Cấu trúc điều khiển

- Các lệnh lặp:
  - For
  - Foreach

```
foreach ($array as $key => $value){  
    // Các dòng lệnh  
}
```

```
foreach ($array as $value){  
    // Các dòng lệnh  
}
```

```
<?php  
$season=array("summer","winter","spring","autumn");  
foreach( $season as $arr ){  
    echo "Season is: $arr<br>";  
}  
?>
```

```
<?php  
for ($n = 1; $n <= 10; $n++) {  
    echo "$n<br/>";  
}  
?>
```

```
<?php  
for ($i = 1; $i <= 3; $i++) {  
    for ($j = 1; $j <= 3; $j++) {  
        echo "$i    $j<br>";  
    }  
}  
?>
```

```
<?php  
$sinhvien = array(  
    'SV01' => "Nguyen A",  
    'SV01' => "Nguyen B"  
);  
foreach ($sinhvien as $masv => $tensv) {  
    echo $masv . ' => ' . $tensv . '<br>';  
}  
?>
```



# Cấu trúc điều khiển

- Các lệnh lặp
  - While
  - Do while

```
<?php
$i = 1;
while ($i < 10) {
    $j = $i;
    while ($j < 10) {
        echo $j;
        $j++;
    }
    echo '<br>';
    $i++;
}
?>
```

```
<?php
$i = 1;
while ($i <= 10) {
    echo "$i <br>";
    $i++;
}
?>
```

```
<?php
$a = 1;
$sum = 0;
do {
    $sum += $a;
    $a++;
} while ($a <= 5);
echo("Sum of 1 to 5 is " . $sum);
?>
```





# Cấu trúc điều khiển

- Lệnh break và continue
  - Break: thoát khỏi vòng lặp
  - Continue: tiếp tục vòng lặp

```
<?php
for ($i = 1; $i <= 10; $i++) {
    if ($i == 5) {
        break;
    }
    echo "$i <br>";
}
?>
```

```
<?php
for ($i = 1; $i <= 10; $i++) {
    if ($i == 5) {
        continue;
    }
    echo "$i <br>";
}
?>
```

# Hàm

- Cú pháp
- Hàm không tham số
- Hàm có tham số
- Hàm có tham số mặc định
- Hàm có giá trị trả về

```
<?php
function setHeight($minheight = 50) {
    echo "The height is : $minheight <br>";
}
setHeight(350);
setHeight();?>
```

```
<?php
function sum($x, $y) {
    $z = $x + $y;
    return $z;
}
echo "5 + 10 = " . sum(5, 10) . "<br>";
?>
```

```
function functionName($vars)
{
    // code được thực thi
}
```

```
<?php
function writeMsg() {
    echo "Hello world!";
}
writeMsg(); // gọi hàm
?>
```

```
<?php
function showName($name) {
    echo "$name <br>";
}
showName("dai");
showName("hoc");
?>
```

# Hàm

- Truyền tham số bằng giá trị: giá trị biến không thay đổi sau khi hàm chạy xong
- Truyền tham số bằng tham chiếu: tham số bên trong hàm thay đổi sau khi gọi hàm
  - Cú pháp: thêm & trước tham số muốn làm tham chiếu

```
<?php
$a = 1;
function tang_len_1($a)
{
    $a = $a + 1;
    return $a;
}
echo tang_len_1($a);
echo $a;
?>
```

```
<?php
$a = 1;
function tang_len_1_(&$a)
{
    $a = $a + 1;
    return $a;
}
echo tang_len_1_($a);
echo $a;
?>
```

# Hàm

- Hàm có n tham số (n chưa biết trước):
  - Cú pháp: đặt dấu 3 chấm trước tham số
- Hàm đệ quy

```
<?php
function add(...$numbers) {
    $sum = 0;
    foreach ($numbers as $n) {
        $sum += $n;
    }
    return $sum;
}
echo add(1,2,3,4,5);
?>
```

```
<?php
function giaiithua($n) {
    if ($n == 1)
        return 1;
    else
        return ($n * giaiithua($n - 1));
}
echo "5! = " . giaiithua(5);
?>
```

- Đối với mảng (array):
  - Count: đếm số phần tử
  - Sort/rsort: sắp xếp tăng dần/giảm dần
  - array\_reverse: đảo ngược phần tử
  - Array\_search: tìm kiếm
  - Array\_intersect: tìm các phần tử giống nhau của 2 mảng
  - Json\_decode/json\_encode: mã hóa/giải mã mảng  $\Leftrightarrow$  json
- Đối với chuỗi (string):
  - Strlen: độ dài chuỗi
  - Str\_word\_count: đếm số từ
  - Strrev: đảo ngược chuỗi
  - Str\_replace: thay thế chuỗi ký tự
  - Md5/sha1: mã hóa thành chuỗi md5/sha1 (32/40 ký tự)

- Các hàm kiểm tra dữ liệu
  - Isset: kiểm tra biến có tồn tại không
  - Empty: kiểm tra biến có trống không
  - Is\_string: kiểm tra có phải chuỗi không
  - Is\_int hoặc is\_integer: kiểm tra có phải kiểu int không
  - Is\_float: kiểm tra có phải kiểu float không
  - Is\_null: kiểm tra có phải null không
  - Is\_array: kiểm tra có phải mảng không
- Hàm thời gian: date
  - Các định dạng: "Y/m/d", "d/m/Y", "d/m/Y H:i:s", "Y:m:d", "Y-m-d", "Y m d h:i:s"



# GET và POST

- GET: gửi thông tin qua URL (\$\_GET để lấy thông tin)
- POST: gửi thông tin ẩn (\$\_POST để lấy thông tin)

```
<form action="GET_method_1.php" method="get">  
Name: <input type="text" name="name"><br>  
Email: <input type="text" name="email"><br>  
<input type="submit">  
</form>
```

```
Wellcome: <?php echo $_GET['name'] ?><br>  
Your Email: <?php echo $_GET['email'] ?>
```

```
<form action="POST_method_1.php" method="post">  
Name: <input type="text" name="name"><br>  
Email: <input type="text" name="email"><br>  
<input type="submit">  
</form>
```

```
Wellcome: <?php echo $_POST['name'] ?><br>  
Your Email: <?php echo $_POST['email'] ?>
```



# Lệnh include và require

- Chức năng: chèn nội dung file php này vào file php khác
- Nếu không tìm thấy file:
  - Include: sinh ra cảnh báo và vẫn chạy chương trình
  - Require: sinh ra lỗi và dừng chạy chương trình
  - Include\_once/require\_once: chèn file một lần duy nhất
- Cú pháp: include 'filename' hoặc require 'filename'

```
<!-- include_menu.php -->
<?php
echo '<a href="">Home</a> -
<a href="/html">HTML</a> -
<a href="/css">CSS</a> -
<a href="/php">PHP</a>';
?>
```

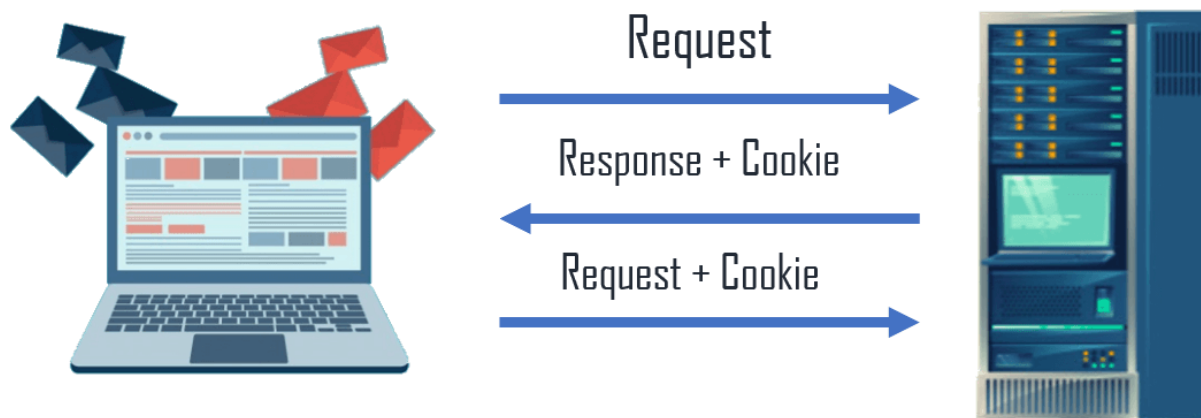
```
<!-- include_footer.php -->
<?php
echo "<p>Copyright -
" . date("Y") . "</p>";
?>
```

```
<?php include 'include_menu.php' ?>
<p>Xin chao</p>
<?php include 'include_footer.php' ?>
```



# Cookie

- Đặc điểm:
  - Là một file nhỏ được Server tạo ra
  - Lưu trữ trên Client
  - Được gửi kèm trong các request từ Client lên Server
  - Thường dùng để xác thực người dùng
  - Không bị mất khi đóng trang web



- Thiết lập cookie:
  - `setcookie(name, value, expire, path, domain)`
    - `$name`: là tên của Cookie (bắt buộc)
    - `$value`: giá trị của Cookie (bắt buộc)
    - `$expire`: thời gian sống của Cookie
    - `$path`: đường dẫn lưu trữ Cookie
    - `$domain`: tên của domain
- Đọc cookie:
  - `$_COOKIE['cookie_name']` (hoặc `$HTTP_COOKIE_VARS['cookie_name']`)
- Xóa cookie: (thiết lập ngày hết hạn)
  - `setcookie("cookie_name", "", time() - 3600)`



# Cookie

- Ví dụ:

```
<?php  
setcookie('name', 'test', time() + 3600);  
?>
```

```
<?php  
if (isset($_COOKIE['name'])) {  
    echo 'name: ' . $_COOKIE['name'];  
}  
?>
```

```
<?php setcookie('name', '', time() - 3600); ?>
```



# Session

---

- Đặc điểm:
  - Được lưu trữ hoàn toàn trên Server (một file trong thư mục trên Server)
  - Mỗi Client sẽ có một ID session khác nhau
  - Tính bảo mật cao
  - Session kết thúc khi tắt trình duyệt hoặc rời khỏi trang, hoặc hết thời gian (thường 30 phút)
  - Thường dùng để lưu thông tin của người dùng



# Session

- Thiết lập session
  - session\_start() (bắt đầu một session)
    - Lưu ý: Nếu thiết lập session.auto\_start=1 trong php.ini thì không cần gọi session\_start()
  - \$\_SESSION['session\_name'] = \$session\_value
- Đọc session
  - \$ten\_bien = \$\_SESSION['session\_name']
- Xóa session
  - unset(\$\_SESSION['session\_name']): xóa một biến session
  - session\_destroy(): xóa tất cả session
- In session
  - print\_r(\$\_SESSION)



# Session

- Ví dụ:

```
<?php
$_SESSION["user_id"] = "100";
$_SESSION["user_name"] = "test";
?>
```

```
<?php
echo "user_id là: " . $_SESSION["user_id"] . "<br>";
echo "username là: " . $_SESSION["user_name"];
?>
```

```
<?php session_destroy(); ?>
```



# Session

- Ví dụ:

```
<?php session_start();  
if (isset($_POST['save_session'])) {  
    $_SESSION['name'] = $_POST['username'];  
}  
?>
```

```
<?php  
if (isset($_SESSION['name'])) {  
    echo 'Tên Đăng Nhập Là: ' . $_SESSION['name'];  
}  
?>
```

```
<form method="POST" action="">  
    <input type="text" name="username" value="" /> <br />  
    <input type="submit" name="save_session" value="Lưu Session" />  
</form>
```

# Xử lý ngoại lệ

- Sử dụng try, throw, catch
  - Try: hàm sử dụng ngoại lệ phải nằm trong khối try
  - Throw: kích hoạt ngoại lệ
  - Catch: bắt ngoại lệ

```
<?php
function checkNum($number)
{
    if ($number >= 1) {
        throw new Exception("gia tri phai nho hon 1");
    }
    return true;
}
try {
    checkNum(2);
    echo 'so nho hon hoac bang 1';
} catch (Exception $e) {
    echo 'Loi: ' . $e->getMessage();
}
```





# Kết nối PHP với MySQL

- Các cách kết nối:
  - MySQLi (i là chữ viết tắt của improved)
    - Chỉ làm việc với cơ sở dữ liệu MySQL
    - Có 2 cách: object-oriented (hướng đối tượng) và Procedural (hướng thủ tục)
  - PDO (PHP Data Objects):
    - Làm việc trên nhiều hệ thống cơ sở dữ liệu khác nhau
- Các bước thao tác trên cơ sở dữ liệu:
  - Bước 1: Tạo kết nối
  - Bước 2: Mở kết nối dữ liệu
  - Bước 3: Tạo lệnh điều khiển truy vấn SQL
  - Bước 4: Thực thi lệnh
  - Bước 5: Đóng kết nối
  - Bước 6: in kết quả



# Kết nối PHP với MySQL

- Kết nối MySQL bằng MySQLi Object-Oriented:

```
<?php
$conn = new mysqli('localhost', 'root', '');
if ($conn->connect_error) {
    die('Ket noi that bai: ' . $conn->connect_error);
}
echo 'Ket noi thanh cong';
?>
```

- Kết nối MySQL bằng MySQLi Procedural:

```
<?php
$conn = mysqli_connect('localhost', 'root', '', 'quanlyhoc_db');
if (!$conn) {
    die('Ket noi that bai: ' . mysqli_connect_error());
}
echo 'Ket noi thanh cong';
?>
```



# Kết nối PHP với MySQL

- Kết nối MySQL bằng PDO:

```
<?php
try{
    $conn=new PDO('mysql:host=localhost;dbname=quanlyhoc_db','root','')
    );
    $conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
    echo 'Ket noi thanh cong';
}
catch(PDOException $e){
    echo 'Ket noi that bai: ' . $e->getMessage();
}
?>
```



# Kết nối PHP với MySQL

---

- Ngắt kết nối:
  - MySQLi Object-Oriented:
    - `$conn->close()`
  - MySQLi Procedural:
    - `mysqli_close($conn)`
  - PDO:
    - `$conn = null`

# Tạo cơ sở dữ liệu

- MySQLi Object-oriented:
  - Chọn cơ sở dữ liệu:
    - \$conn->select\_db(ten\_csdl)

```
<?php
$conn = new mysqli('localhost', 'root', '');
if ($conn->connect_error) {
    die("Kết nối thất bại: " . $conn->connect_error);
}
$sql = "CREATE DATABASE test";
if ($conn->query($sql)) {
    echo "Tạo database thành công";
} else {
    echo "Tạo database thất bại: " . $conn->error;
}
$conn->close();
?>
```

# Tạo cơ sở dữ liệu

- MySQLi Procedural:
  - Chọn cơ sở dữ liệu:
    - `mysql_select_db(db_name, connection)`

```
<?php
$conn = mysqli_connect('localhost', 'root', '');
if (!$conn) {
    die('Ket noi that bai: ' . mysqli_connect_error());
}
$sql = 'create database test';
if (mysqli_query($conn, $sql)) {
    echo 'Tao DB thanh cong';
} else {
    echo 'Tao DB that bai' . mysqli_error($conn);
}
mysqli_close($conn);
?>
```

# Tạo cơ sở dữ liệu

- PDO:
  - Chọn cơ sở dữ liệu:
    - \$conn->exec("use database\_name")

```
<?php
try {
    $conn = new PDO('mysql:host=localhost', 'root', '');
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = 'create database test4';
    $conn->exec($sql);
    echo 'Tao DB thanh cong';
} catch (PDOException $e) {
    echo 'Loi: ' . $e->getMessage();
}
$conn = null;
?>
```

# Tạo bảng

- MySQLi Object-oriented:

```
<?php
$conn = new mysqli('localhost', 'root', '', 'test');
if ($conn->connect_error) {
    die('ket noi that bai: ' . $conn->connect_error);
}
$sql = 'create table News(
id int auto_increment primary key,
title nvarchar(30) not null,
content text,
add_date timestamp)
';
if ($conn->query($sql)) {
    echo 'Tao bang thanh cong';
} else
    echo 'Tao bang khong thanh cong: ' . $conn->error;
$conn->close();
?>
```





# Tạo bảng

- MySQLi Procedural:

```
<?php
$conn = mysqli_connect('localhost', 'root', '', 'test');
if (!$conn) {
    die('ket noi that bai: ' . mysqli_connect_error());
}
$sql = 'create table News(
    id int auto_increment primary key,
    title nvarchar(30) not null,
    content text,
    add_date timestamp)
';
if (mysqli_query($conn, $sql)) {
    echo 'Tao bang thanh cong';
} else {
    echo 'Tao bang khong thanh cong: ' . mysqli_error($conn);
}
mysqli_close($conn);
?>
```

# Tạo bảng

- PDO:

```
<?php
try {
    $conn = new PDO('mysql:host=localhost;dbname=test', 'root', '');
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = 'create table News(
        id int auto_increment primary key,
        title nvarchar(30) not null,
        content text,
        add_date timestamp)
        ';
    $conn->exec($sql);
    echo 'Tao bang thanh cong';
} catch (PDOException $e) {
    echo $e->getMessage();
}
$conn = null;
?>
```



# Tạo cơ sở dữ liệu và bảng

- MySQLi Procedural:

```
<?php
$conn = mysqli_connect('localhost', 'root', '');
if (!$conn) {
    die('ket noi that bai: ' . mysqli_connect_error());
}
$sql = 'create database test';
if (mysqli_query($conn, $sql)) {
    echo 'Tao DB thanh cong';
    mysqli_select_db($conn, 'test5');
    $sql = 'create table News(id int auto_increment primary key, title nvarchar(30) not null, content text, add_date timestamp)';
    if (mysqli_query($conn, $sql)) {
        echo 'Tao bang thanh cong';
    } else {
        echo 'Tao bang khong thanh cong: ' . mysqli_error($conn);
    }
    mysqli_close($conn);
} else {
    echo 'Tao DB that bai: ' . mysqli_error($conn);
}
mysqli_close($conn);
?>
```

# Thêm dữ liệu vào bảng

- MySQLi Object-oriented:

```
<?php
$conn = new mysqli('localhost', 'root', '', 'test');
if ($conn->connect_error) {
    die('ket noi that bai: ' . $conn->connect_error);
}
$sql = 'insert into News(title, content, add_date) values("PL","Phap l
uat", NOW())';
if ($conn->query($sql) == true) {
    echo 'Them du lieu thanh cong';
} else
    echo 'Them du lieu khong thanh cong: ' . $conn->error;
$conn->close();
?>
```



# Lấy ID bản ghi vừa insert

- MySQLi Object-oriented:
  - \$conn->insert\_id
- MySQLi Procedural:
  - mysqli\_insert\_id(\$conn)
- PDO:
  - \$conn->lastInsertId()

```
.....  
$sql = 'insert into News(title, content, add_date) values("PL","Phap lu  
at", NOW())';  
if (mysqli_query($conn, $sql)) {  
    $last_ID=mysqli_insert_id($conn);  
    echo 'Them du lieu thanh cong, ID = ' . $last_ID;  
} else  
    echo 'Them du lieu khong thanh cong: ' . mysqli_error($conn);  
mysqli_close($conn);
```

# Thêm nhiều bản ghi vào bảng

- MySQLi Object-oriented:
  - \$conn->multi\_query(\$sql)
- MySQLi Procedural:
  - mysqli\_multi\_query(\$conn, \$sql)

```
$conn = new mysqli('localhost', 'root', '', 'test');
if ($conn->connect_error) {
    die('ket noi that bai: ' . $conn->connect_error);
}
$sql = 'insert into News(title, content, add_date) values("PL","Phap lu
at", NOW());';
$sql .= 'insert into News(title, content, add_date) values("PL","Phap l
uat", NOW());';
if ($conn->multi_query($sql) == true) {
    echo 'Them du lieu thanh cong';
} else
    echo 'Them du lieu khong thanh cong: ' . $conn->error;
$conn->close();
```



# Cập nhật dữ liệu

- MySQLi Object-oriented:

```
<?php
$conn = new mysqli('localhost', 'root', '', 'test');
if ($conn->connect_error) {
    die('ket noi that bai: ' . $conn->connect_error);
}
$sql = 'update News set content = "Phap luat 1" where id = 1';
if ($conn->query($sql) == true) {
    echo 'Cap nhat du lieu thanh cong';
} else
    echo 'Cap nhat du lieu khong thanh cong: ' . $conn->error;
$conn->close();
?>
```

# Lấy dữ liệu

- MySQLi Object-oriented:
  - Dùng vòng lặp while

```
<?php
$conn = new mysqli('localhost', 'root', '', 'test');
if ($conn->connect_error) {
    die('ket noi that bai: ' . $conn->connect_error);
}
$sql = 'select * from News';
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo 'Title: ' . $row['title'] . ', Content: ' . $row['content']
    ] . ', Date: ' . $row['add_date'] . '<br>';
    }
} else {
    echo 'Khong co ban ghi nao';
}
$conn->close();
?>
```



# Lấy dữ liệu

- MySQLi Object-oriented:
  - Dùng vòng lặp foreach

```
<?php
$conn = new mysqli('localhost', 'root', '', 'test');
if ($conn->connect_error) {
    die('ket noi that bai: ' . $conn->connect_error);
}
$sql = 'select * from News';
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    foreach($result as $row){
        echo 'Title: ' . $row['title'] . ', Content: ' . $row['content']
    ] . ', Date: ' . $row['add_date'] . '<br>';
    }
} else {
    echo 'Khong co ban ghi nao';
}
$conn->close();
?>
```

# Lấy dữ liệu

- MySQLi Procedural:
  - Dùng vòng lặp while

```
<?php
$conn = mysqli_connect('localhost', 'root', '', 'test');
if (!$conn) {
    die('ket noi that bai: ' . mysqli_connect_error());
}
$sql = 'select * from News';
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        echo 'Title: ' . $row['title'] . ', Content: ' . $row['content']
    ] . ', Date: ' . $row['add_date'] . '<br>';
    }
} else
    echo 'Khong co ban ghi nao: ' . mysqli_error($conn);
mysqli_close($conn);
?>
```

# Lấy dữ liệu

- PDO:
  - Dùng vòng lặp foreach

```
<?php
try {
    $conn = new PDO('mysql:host=localhost;dbname=test', 'root', '');
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = 'select * from News';
    $result = $conn->query($sql);
    if ($result->rowCount() > 0) {
        foreach ($result as $row) {
            echo 'Title: ' . $row['title'] . ', Content: ' . $row['content'] . ', Date: ' . $row['add_date'] . '<br>';
        }
    } else {
        echo 'Khong co ban ghi nao';
    }
} catch (PDOException $e) {
    echo $e->getMessage();
}
$conn = null;
?>
```

# Xóa dữ liệu

- MySQLi Object-oriented:

```
<?php
$conn = new mysqli('localhost', 'root', '', 'test');
if ($conn->connect_error) {
    die('ket noi that bai: ' . $conn->connect_error);
}
$sql = 'delete from News where id = 1';
if ($conn->query($sql) == true) {
    echo 'Xoa du lieu thanh cong';
} else
    echo 'Xoa du lieu khong thanh cong: ' . $conn->error;
$conn->close();
?>
```



# Cơ chế prepared câu SQL

- Mục đích: tránh lỗi SQL Injection
- Cơ chế: truyền tham số bằng biến ẩn danh, sau đó truyền giá trị cho các biến ẩn danh để chạy câu truy vấn  
→ an toàn hơn truyền trực tiếp tham số
- Ví dụ về lỗi SQL Injection:
  - Câu lệnh PHP:

```
$variable = $_POST['user_input'];  
mysql_query("INSERT INTO table('column') VALUES ('$unsafe_variable')");
```

- Truyền giá trị cho \$variable: `value'); DROP TABLE table;--`
- Câu SQL chuyển thành:

```
INSERT INTO table('column') VALUES('value'); DROP TABLE table;--')
```



# Cơ chế prepared câu SQL

- MySQLi Object-oriented:
  - i: integer; d: double; s: string

```
<?php
$conn = new mysqli('localhost', 'root', '', 'test');
if ($conn->connect_error) {
    die('ket noi that bai: ' . $conn->connect_error);
}
$sql = 'insert into News(title, content) values(?, ?)';
$tmp = $conn->prepare($sql);
$tmp->bind_param('ss', $title, $content);
$title = "PL";
$content = "Phap luat";
$tmp->execute();
if ($tmp == true) {
    echo 'Them du lieu thanh cong';
} else
    echo 'Them du lieu khong thanh cong: ' . $conn->error;
$tmp->close();
$conn->close();
?>
```



# Cơ chế prepared câu SQL

- PDO:

```
<?php
try {
    $conn = new PDO('mysql:host=localhost;dbname=test', 'root', '');
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = 'insert into News(title, content) values(:title, :content)';
    $tmp=$conn->prepare($sql);
    $tmp->bindParam(':title', $title);
    $tmp->bindParam(':content', $content);
    $title = "PL";
    $content = "Phap luat";
    $tmp->execute();
    if ($tmp == true) {
        echo 'Them du lieu thanh cong';
    } else
        echo 'Them du lieu khong thanh cong: ' . $conn->error;
} catch (PDOException $e) {
    echo $e->getMessage();
}
$tmp = null; $conn = null;
?>
```

# PHP framework

---

- Một số framework PHP phổ biến:
  - Laravel
  - Codeigniter
  - Zend
  - ...





# Câu hỏi thảo luận

---