



TRƯỜNG ĐẠI HỌC VINH
VIỆN KỸ THUẬT VÀ CÔNG NGHỆ

CHƯƠNG 3

NGÔN NGỮ LẬP TRÌNH PHP

Nội dung giảng dạy

- 3.1. Giới thiệu về PHP
- 3.2. PHP cơ bản
- 3.3. Các phép toán
- 3.4. Cấu trúc điều khiển
- 3.5. Hàm
- 3.6. Xử lý Form
- 3.7. Mảng
- 3.8. Chuỗi
- 3.9. Include và Require
- 3.10. Cookie và Session
- 3.11. Xử lý File
- 3.12. Xử lý ngoại lệ

3.1. Giới thiệu về PHP

- 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



3.1. Giới thiệu về PHP

- 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, ...)

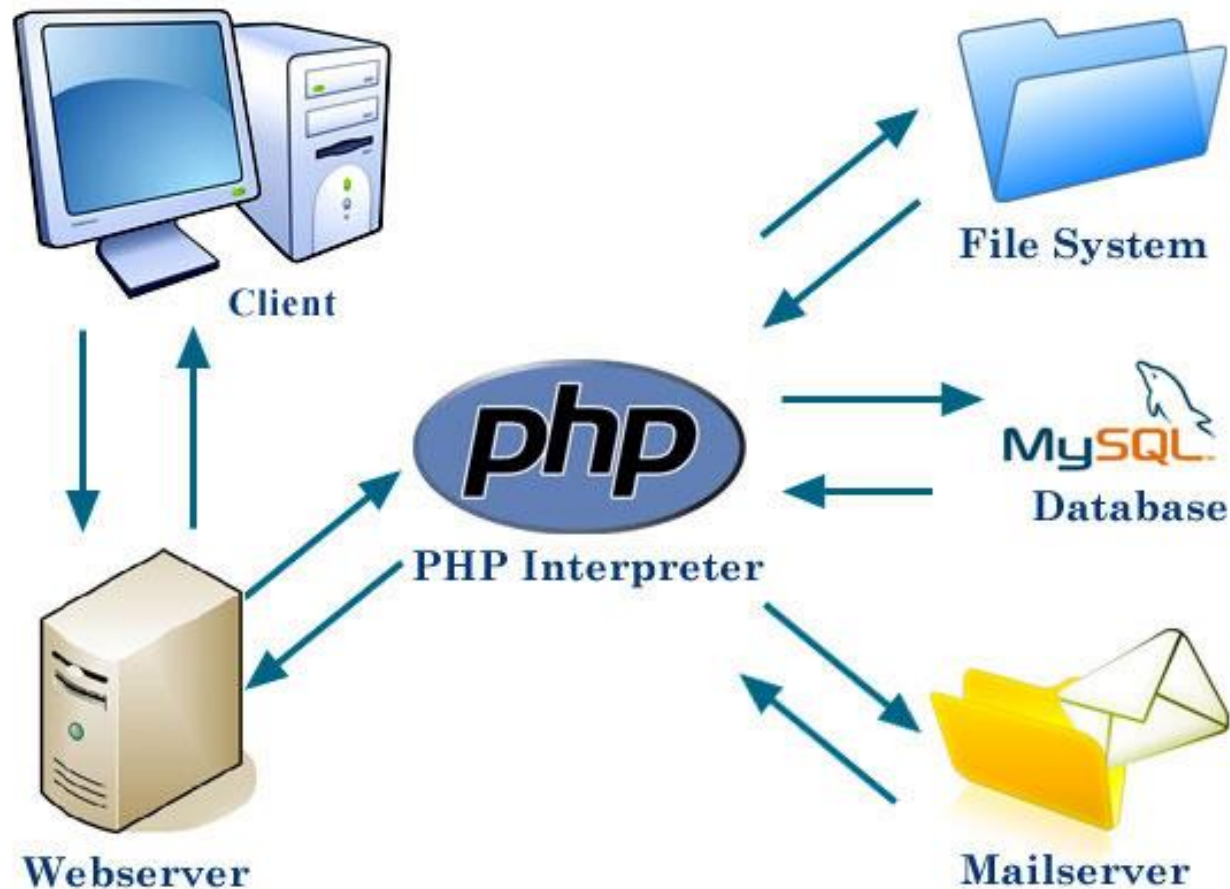
3.1. Giới thiệu về PHP

- 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

3.1. Giới thiệu về PHP

- Mô hình hoạt động





3.1. Giới thiệu về PHP

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

3.2. PHP 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



3.2. PHP cơ bản

- 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}";

?>
```

3.2. PHP cơ bản

- 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;
?>
```



3.2. PHP cơ bản

- Echo và print: 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\"";
?>
```



3.2. PHP cơ bản

- 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 mảng



3.2. PHP cơ bản

- Phạm vi của biến: 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



3.2. PHP cơ bản

- 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();
?>
```



3.2. PHP cơ bản

- 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`

3.2. PHP cơ bản

- 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'];
```




3.2. PHP cơ bản

- 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;
    }
}
?>
```



localhost:8080/code/hello.php?subject=web&location=day

Học lập trình web tại đây

```
<?php
echo "Học lập trình " . $_GET["subject"] . " tại " . $_GET["location"];
?>
```



3.3. 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`



3.4. 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";
}
?>
```



3.4. 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");
}
?>
```



3.4. 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>';  
}  
?>
```



3.4. 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);
?>
```



3.4. 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>";
}
?>
```

3.5. 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");
?>
```


3.5. 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;
?>
```

3.5. 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);
?>
```

3.5. Hàm

- 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"

3.6. Xử lý Form

- 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'] ?>
```

3.7. Mảng

- 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'];
?>
```

3.7. Mảng

- 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
?>
```

- Duyệt phần tử trong mảng:

```
<?php
$list = array('A', 'B', 'C');
$list_length = count($list);
for($i=0;$i<$list_length;$i++){
    echo $list[$i];
    echo '<br>';
}
?>
```

```
foreach($list as $i){
    echo $i;
    echo '<br>';
}
```



3.7. Mảng

- Các hàm xử lý mảng:
 - Count: đếm số phần tử
 - Sort/rsort: sắp xếp tăng dần/giảm dần
 - Asort/arsort: sắp xếp tăng dần/giảm dần theo giá trị
 - Ksort/ksort: tăng dần/giảm dần theo khóa
 - 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

3.8. Chuỗi

- Khai báo chuỗi:
 - Chuỗi có dấu nháy đơn: các biến sẽ không được biên dịch
 - Chuỗi có dấu nháy đôi: các biến sẽ được biên dịch

```
<?php
$bien_chuoi = "name";
$bien_chuoi_hang = 'Chuỗi $bien_chuoi sẽ không được in!\\n';

print($bien_chuoi_hang);
print "<br />";

$bien_chuoi_hang = "Chuỗi $bien_chuoi sẽ được in!\\n";

print($bien_chuoi_hang);
?>
```

- Nếu muốn dùng dấu nháy " trong chuỗi chứa nháy đôi thì thêm dấu \ trước dấu nháy đôi

3.8. Chuỗi

- Các hàm xử lý chuỗi:
 - Strlen: độ dài chuỗi
 - Str_word_count: đếm số từ
 - Strpos: tìm ký tự hoặc chuỗi
 - Strrev: đảo ngược chuỗi
 - Str_replace: thay thế chuỗi ký tự
 - Md5/sha1: mã hóa thành chuỗi md5/sha1 (32/40 ký tự)
- Tham khảo thêm các hàm xử lý chuỗi tại [đây](#)

3.9. 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>';
?>
```

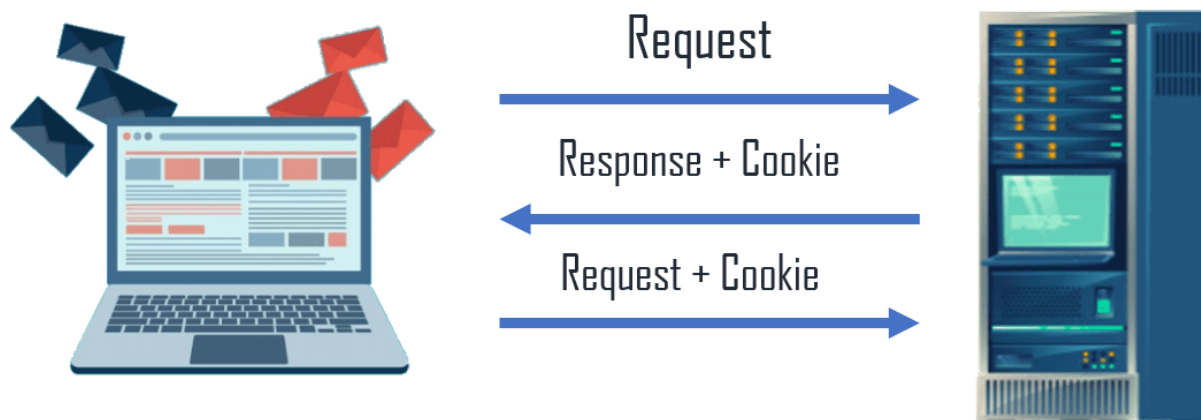
```
<!-- include_footer.php -->
<?php
echo "<p>Copyright -
" . date("Y") . "</p>";
?>
```

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



3.10. Cookie và Session

- Cookie:
 - 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





3.10. Cookie và Session

- 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)`



3.10. Cookie và Session

- Ví dụ cookie:

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

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

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



3.10. Cookie và Session

- Session:
 - 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



3.10. Cookie và 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)`



3.10. Cookie và Session

- Ví dụ session:

```
<?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(); ?>
```




3.10. Cookie và Session

- Ví dụ session:

```
<?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>
```

3.11. Xử lý File

- Mở File: `fopen($path_to_file, $mode)`
 - `r`: mở file cho phép đọc
 - `r+`: mở file để đọc và ghi. Ghi đè
 - `w`: mở file để ghi, file chưa tồn tại thì tạo file mới. Ghi đè lên. Nội dung file cũ bị mất
 - `w+`: tương tự như chế độ `w` nhưng có thể đọc file được
 - `a`: ghi file, nhưng không đè lên file cũ mà sẽ ghi tiếp vào file cũ. Nếu file chưa tồn tại thì tạo file mới
 - `a+`: tương tự `a`, nhưng có thể đọc file được
 - `x`: ghi file, nhưng ngược lại với chế độ `a`. Nếu file tồn tại thì báo lỗi, ngược lại chưa tồn tại thì tạo file mới
 - `x+`: tương tự `x` nhưng có thể đọc file được
- Đóng File: `fclose(resource $handle)`

3.11. Xử lý File

- Mở File:

```
<?php
$fp = @fopen('demo.txt', "r");
if (!$fp) {
    echo 'Mở file không thành công';
    fclose($fp);
} else {
    echo 'Mở file thành công';
}
?>
```

- Đọc File:

- Đọc từng kí tự: fgetc(\$file)
- Đọc từng dòng: fgets(\$file)
- Đọc hết file: fread(\$file, \$size)

3.11. Xử lý File

- Đọc File:

```
<?php
$fp = @fopen('demo.txt', "r");
if (!$fp) {
    echo 'Mở file không thành công';
} else {
    while (!feof($fp)) {
        //Đọc từng ký tự
        echo fgetc($fp);

        //Đọc từng dòng
        echo fgets($fp);

        //Đọc cả file
        $data = fread($fp, filesize('demo.txt'));
        echo $data;
    }
}
?>
```

3.11. Xử lý File

- Ghi File: fwrite(\$file, \$content)

```
<?php
$fp = @fopen('demo.txt', "r");
if (!$fp) {
    echo 'Mở file không thành công';
} else {
    while (!feof($fp)) {
        //Đọc từng ký tự
        echo fgetc($fp);

        //Đọc từng dòng
        echo fgets($fp);

        //Đọc cả file
        $data = fread($fp, filesize('demo.txt'));
        echo $data;
    }
}
?>
```

3.11. Xử lý File

- Một số hàm xử lý File:
 - `file_exists($path)`: kiểm tra file có tồn tại không
 - `is_writable ($path)`: kiểm tra file có được cấp quyền ghi không
 - `file_get_contents($path)`: lấy nội dung file (có thể lấy nội dung từ đường link trên internet)
 - `file_put_contents($path, $content)`: ghi nội dung
 - `rename($oldname, $newname)`: đổi tên file
 - `copy($source, $dest)`: sao chép file
 - `unlink($path)`: xóa file
 - `is_dir($filename)`: kiểm tra đường dẫn có tồn tại không
 - `mkdir($path)`: tạo folder mới



3.12. 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();
}
```



Câu hỏi thảo luận
