

CSE485 – Công nghệ Web

dungkt@tlu.edu.vn





QUI ĐỊNH

- Lớp học KHÔNG DÙNG ĐIỆN THOẠI
- Lớp học KHÔNG MẶC QUẦN ĐÙI đến lớp
- Lớp học CẦN MANG LAPTOP
- Lớp học KHÔNG ĐI MUỘN (*chậm nhất sau 5 phút không vào lớp*)
- Lớp học CẦN LÀM BÀI TẬP ĐẦY ĐỦ, NỘP ĐÚNG HẠN
- Lớp học BỊ PHẠT THEO NHÓM nếu ngồi cùng BÀN với thành viên vi phạm các QUI ĐỊNH
- Máy tính cần dán biển tên để định danh gọi tên SV



Back-end Tech Stack for Web Development

Programming languages



Web servers



Frameworks

django

for Python



for PHP



for JavaScript

Operating systems



Database languages



Bài 1. Cơ bản về PHP

NỘI DUNG

1. Giới thiệu về PHP
2. Cài đặt môi trường làm việc
3. Nhúng PHP vào HTML
4. Khai báo biến
5. Kiểu dữ liệu chuỗi và số
6. Quy tắc đặt tên biến trong PHP
7. Giới thiệu về Hàm
8. Phạm vi hoạt động của biến
9. Hằng số



NỘI DUNG

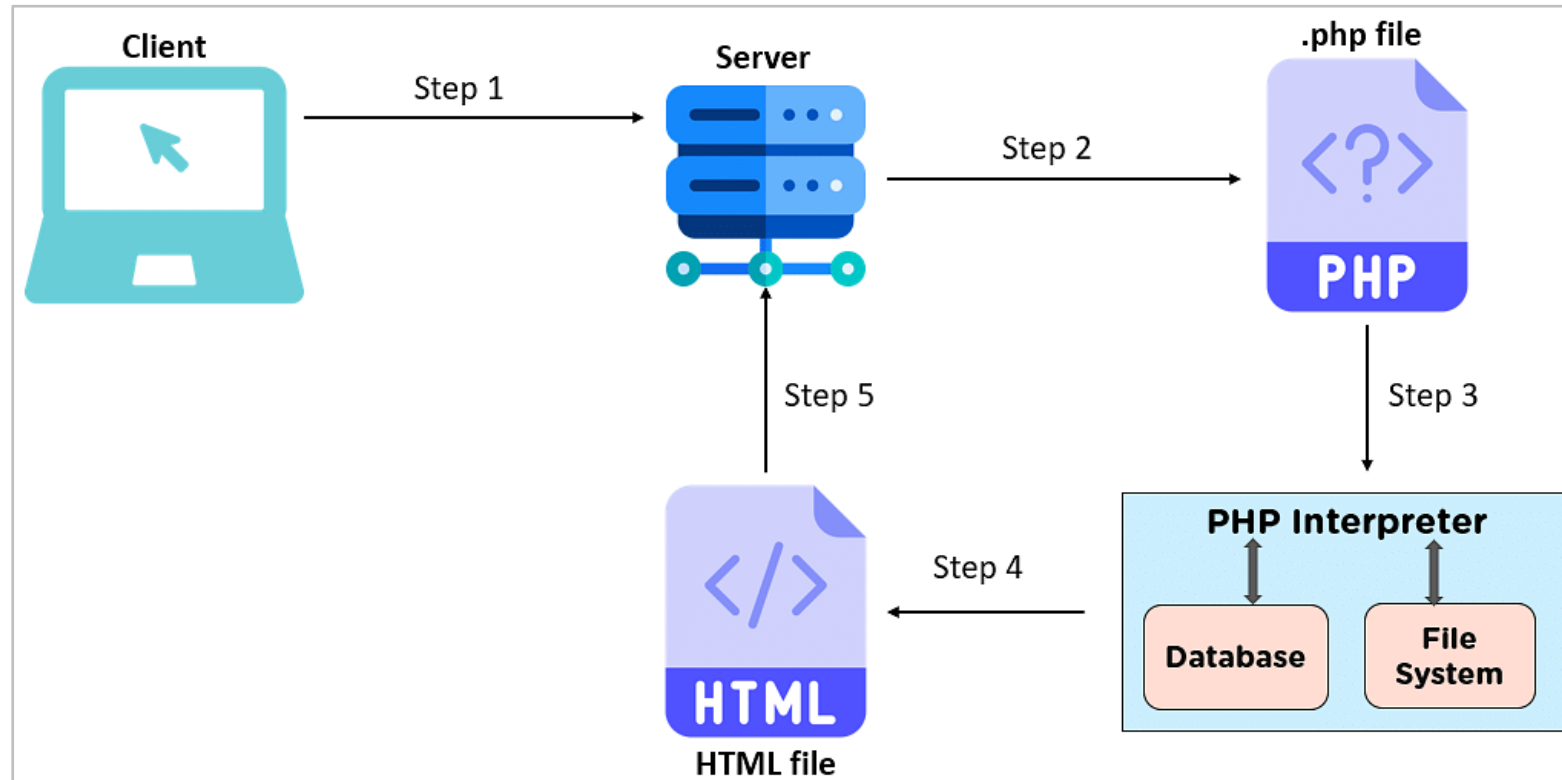
- 10. Các loại toán tử
- 11. Các kiểu dữ liệu
- 12. Các cấu trúc điều khiển
- 13. namespace
- 14. Các câu lệnh include
- 15. Mảng
- 16. Hàm
- 17. Xử lý ngoại lệ
- 18. Làm việc với thư mục
- 19. Đọc và ghi tệp





1. GIỚI THIỆU VỀ PHP

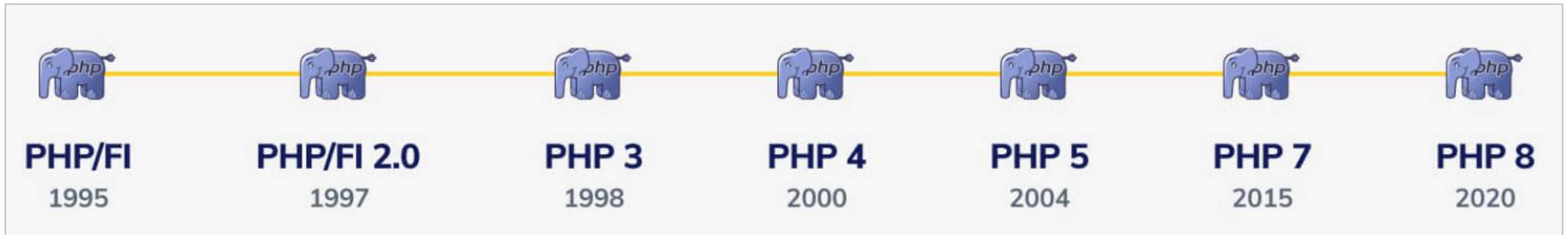
- **PHP** (Hypertext Preprocessor) là ngôn ngữ kịch bản phía máy chủ phổ biến được sử dụng để tạo ra các trang web động, tương tác với cơ sở dữ liệu và thực hiện nhiều tác vụ khác. PHP được nhúng trực tiếp vào mã HTML, giúp quá trình phát triển trở nên dễ dàng hơn.





1. GIỚI THIỆU VỀ PHP

- PHP ra đời năm 1994 bởi Rasmus Lerdorf
- PHP đã trải qua một quá trình phát triển lâu dài và không ngừng cải tiến. Mỗi phiên bản mới đều mang đến những tính năng mới, hiệu năng được nâng cao và hỗ trợ tốt hơn cho các tiêu chuẩn hiện đại.



- Ưu điểm của PHP:
 - Mã nguồn mở: Miễn phí sử dụng và tùy chỉnh
 - Dễ học: Cú pháp đơn giản, gần gũi với ngôn ngữ tự nhiên
 - Cộng đồng lớn: Nhiều tài liệu, thư viện và hỗ trợ
 - Tương thích đa nền tảng: Hoạt động tốt trên nhiều hệ điều hành và máy chủ



2. CÀI ĐẶT MÔI TRƯỜNG LÀM VIỆC

- **Stack:** Cài đặt một gói phần mềm bao gồm PHP và các thành phần liên quan như web server (Apache, Nginx), cơ sở dữ liệu (MySQL, PostgreSQL), và các công cụ khác.
 - XAMPP, WAMP, LAMP, MAMP, Vertrigo, OpenServer, **Laragon** ...
- **Standalone:** Cài đặt riêng lẻ từng thành phần, cho phép bạn có nhiều tùy biến hơn
 - PHP: Tải về bản cài đặt zip từ trang chủ của PHP (<https://www.php.net/downloads.php>). Chọn phiên bản phù hợp với hệ thống của bạn (32-bit hoặc 64-bit).
 - Apache: Tải về bản cài đặt MSI từ trang chủ của Apache (<https://httpd.apache.org/download.cgi>).
 - MySQL: Tải về bản cài đặt MSI từ trang chủ của MySQL hoặc MariaDB

Edition

Download Laragon - Full (173 MB)

- **Laragon Full (64-bit):** Apache 2.4, Nginx, MySQL 8, PHP 8, Redis, Memcached, Node.js 18, npm, git

Download Laragon - Portable (38 MB)

- **Laragon Portable:** PHP 5.4, MySQL 5.1, bitmana - Good for getting started with PHP, then you can add newer versions of PHP/MySQL easily later using “Tools > Quick add”





2. CÀI ĐẶT MÔI TRƯỜNG LÀM VIỆC

- Tổng quan về cơ chế làm việc của PHP stack khi tiếp nhận một request
 - **Client gửi yêu cầu (Request):**
 - Người dùng nhập URL
 - Trình duyệt đóng gói yêu cầu thành một gói tin HTTP, bao gồm:
 - Phương thức HTTP (GET, POST, PUT, DELETE,...)
 - URL của tài nguyênHeader chứa thông tin về trình duyệt, ngôn ngữ,...
 - Thân yêu cầu (body) có thể chứa dữ liệu gửi đi từ các form.
 - **Máy chủ web nhận yêu cầu:**
 - Apache/Nginx: Máy chủ web (như Apache hoặc Nginx) sẽ lắng nghe các kết nối từ trình duyệt.
 - Xác định file: Khi nhận được yêu cầu, máy chủ sẽ tìm kiếm file tương ứng với URL đó trên ổ cứng.
 - Kiểm tra phần mở rộng: Nếu file có phần mở rộng là .php, máy chủ sẽ chuyển yêu cầu đến trình thông dịch PHP.

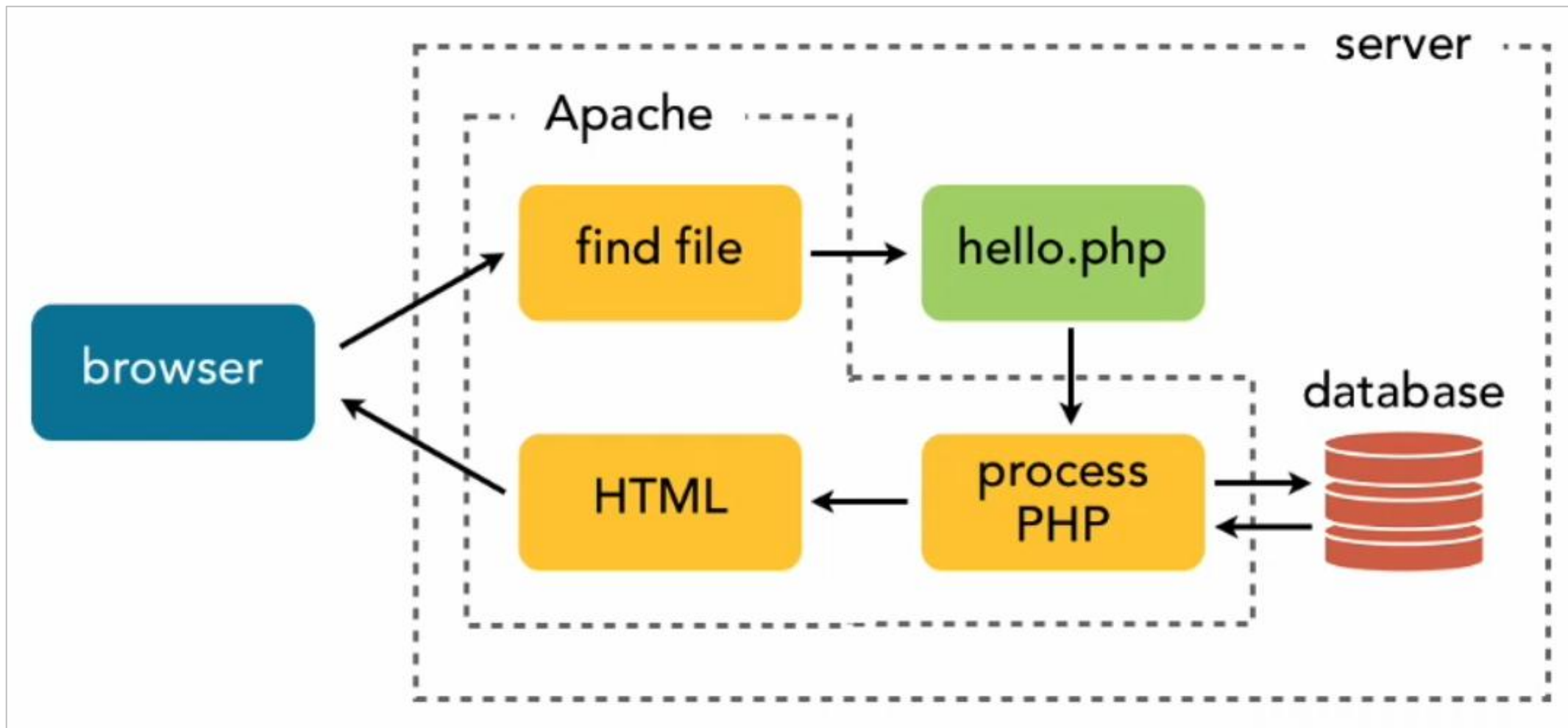


2. CÀI ĐẶT MÔI TRƯỜNG LÀM VIỆC

- Tổng quan về cơ chế làm việc của PHP stack khi tiếp nhận một request
 - **Trình thông dịch PHP:**
 - Phân tích mã: Trình thông dịch PHP (PHP engine) sẽ phân tích mã PHP trong file đó.
 - Thực thi mã: Nó sẽ thực hiện từng dòng lệnh PHP, bao gồm:
 - Kết nối cơ sở dữ liệu (nếu có)
 - Xử lý dữ liệu
 - Tạo ra HTML động
 - Tạo chuỗi HTML: Kết quả của việc thực thi mã PHP sẽ là một chuỗi HTML hoàn chỉnh.
 - **Máy chủ web trả về kết quả:**
 - Gửi response: Máy chủ web sẽ gửi chuỗi HTML này trở lại trình duyệt dưới dạng một response HTTP.
 - Hiển thị nội dung: Trình duyệt nhận được response và hiển thị nội dung HTML lên màn hình.



2. CÀI ĐẶT MÔI TRƯỜNG LÀM VIỆC





3. NHÚNG PHP VÀO HTML

- **Mã PHP** có thể được nhúng trực tiếp vào các file HTML.
- Các đoạn mã PHP thường được đặt trong các thẻ `<?php` và `?>`. Phần còn lại của file là HTML, định dạng giao diện cho trang web.

```
<!DOCTYPE html>
<html>
<head>
  <title>Ví dụ đơn giản</title>
</head>
<body>
  <h1>Xin chào, <?php echo "Thế giới!"; ?></h1>
  <p>Đây là một đoạn văn bản động.</p>
</body>
</html>
```



4. KHAI BÁO BIẾN

- Để khai báo một biến trong PHP, bạn chỉ cần đặt dấu \$ trước tên biến. Sau đó, gán giá trị cho biến bằng dấu =.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Variable</title>
</head>
<body>
    <?php
        $aboutme = "I am Me.";
        echo "<h1>About me: </h1>";
        echo $aboutme;
    ?>

    <?php
        $aboutme = "Text is changed.";
        echo $aboutme;
    ?>
</body>
</html>
```



5. KIỂU DỮ LIỆU CHUỖI VÀ SỐ

- Để chứa chuỗi, có thể sử dụng cả dấu nháy đơn (') và dấu nháy kép (")
 - Dấu nháy đơn: Không thể nhúng biến vào trong chuỗi.
 - Dấu nháy kép: Có thể nhúng biến vào trong chuỗi bằng cách sử dụng dấu {}. Ví dụ:

```
<h1>Define number</h1>
<?php
    $ver = 8.3;
    echo "PHP version is $ver";
?>

<h1>Combine string and number</h1>
<?php
    $lang = "PHP";
    echo $lang . " version is " . $ver;
    echo "<br />";
    echo "$lang version is $ver";
?>
```

```
<h1>Define string</h1>
<?php
    $name = "Me";
    $message = "Hello World.";
    echo $name . ": " . $message;
    echo "<br />";
    echo "$name: $message";
    echo "<br />";
    echo '$name: $message';
?>
```

```
<h1>Add two value</h1>
<?php
    $val1 = 10;
    $val2 = 20;
    $total = $val1 + $val2;
    echo "Total: $total";
?>
```



5. KIỂU DỮ LIỆU CHUỖI VÀ SỐ

- Chuyển đổi giữa các kiểu dữ liệu: PHP có tính năng chuyển đổi kiểu dữ liệu tự động. Tuy nhiên, đôi khi lập trình viên cần kiểm soát quá trình chuyển đổi này.

```
<?php
// Chuyển đổi từ chuỗi sang số
$so = "10";
$so_int = (int)$so; // $so_int sẽ là 10 (số nguyên)

// Chuyển đổi từ số sang chuỗi
$so = 10;
$so_str = (string)$so; // $so_str sẽ là "10" (chuỗi)

?>
```

- Nối chuỗi: Sử dụng dấu chấm (.)
- Kiểm tra kiểu dữ liệu: Sử dụng hàm `gettype()` để kiểm tra kiểu dữ liệu của một biến.
- Ép kiểu: Sử dụng hàm `settype()` để ép một biến sang kiểu dữ liệu khác.
- So sánh chuỗi và số: Khi so sánh chuỗi và số, PHP sẽ cố gắng chuyển đổi chuỗi thành số.



6. QUI TẮC ĐẶT TÊN BIẾN TRONG PHP

- Bắt đầu bằng dấu đô la (\$)
- Tên biến phân biệt chữ hoa và chữ thường
- Tên biến chỉ chứa các ký tự chữ cái, số và dấu gạch dưới
- Tên biến nên có ý nghĩa
- Nên sử dụng kiểu viết hoa chữ cái đầu mỗi từ (camelCase)
- Tránh sử dụng các từ khóa của PHP

```
<?php
// Rule 1
$message = 'Hello PHP';
echo $message;

//Rule 2
$name = 'PHP';
// $1name = 'PHP';
// $_1name = 'php';
// $-name = 'php';

// Rule 3
$first_name = "Me";

// Rule 4
$name1 = 'php';
$n1ame = 'php';

// Rule 5
$name = 'me';
$Name = 'Me';
echo "$Name $name";

?>
```




7. GIỚI THIỆU VỀ HÀM

- Tại sao sử dụng hàm?
 - Tái sử dụng mã: Tránh việc viết lại cùng một đoạn mã nhiều lần.
 - Tổ chức code: Chia nhỏ chương trình thành các khối chức năng nhỏ, giúp code dễ hiểu và dễ quản lý hơn.
 - Mở rộng tính năng: Dễ dàng thêm hoặc sửa đổi các chức năng của chương trình bằng cách thêm hoặc sửa đổi các hàm.
 - Tăng tính đọc: Code trở nên rõ ràng hơn khi mỗi hàm thực hiện một nhiệm vụ cụ thể.
- Cấu trúc một hàm trong PHP

```
<?php
function ten_ham(tham_so1, tham_so2, ...) {
    // Thực hiện các tác vụ ở đây
    return gia_tri_tra_ve; // Tùy chọn
}
```

```
<?php
function add() {
    $val1 = 10;
    $val2 = 20;
    $total = $val1 * $val2;

    echo "Total: $total";
}

add();

/*
function add() {
    echo "Hello";
}
*/
echo "<br />";

Welcome();
function welcome() {
    echo "Hello World !!!";
}
// function welCome() {
//     echo "Hello World !!!";
// }

?>
```



8. PHẠM VI HOẠT ĐỘNG CỦA BIẾN

- Phạm vi hoạt động của một biến trong PHP xác định nơi bạn có thể truy cập và sử dụng biến đó trong mã của mình
- Các loại phạm vi biến trong PHP
 - Biến cục bộ (Local variable)
 - Biến toàn cục (Global variable)
 - Biến siêu toàn cục (Super global variable)
 - Biến tĩnh (Static variable)

```
<?php
global $name;
$name = 'Nghia';

function print_name() {
    global $name;
    // $name = 'PHP';
    echo $name;
}

print_name();
echo $name;

?>
```

```
<?php
global $message;
$message = 'Welcome to PHP';
echo $GLOBALS['message'] . "<br />";

function print_welcome() {
    echo $GLOBALS['message'] . "<br />";
}

print_welcome();

@$val = 1/0;
print_r(error_get_last());

?>
```

```
<?php
function counter() {
    $count = 1;
    echo $count . '<br />';
    $count = $count + 1;
}

counter();
counter();
counter();
counter();
counter();

echo '<hr />';

function counter_static() {
    static $count = 1;
    echo $count . '<br />';
    $count = $count + 1;
}

counter_static();
counter_static();
counter_static();
counter_static();
counter_static();

?>
```

9. HẲNG SỐ



- Hằng số là một giá trị không thể thay đổi trong suốt quá trình thực thi chương trình
- Có hai cách định nghĩa hằng số trong PHP
 - Sử dụng hàm define()
 - Sử dụng từ khóa const (kể từ PHP 5.3)

```
<?php
define("LANGUAGE", "PHP");
$lang = LANGUAGE;
echo "This program is written using $lang <br />";
echo "This program is written using LANGUAGE <br />";
echo "This program is written using ". LANGUAGE . "<br />";

// define("LANGUAGE", "JAVA");
// echo "This program is written using ". LANGUAGE .
"<br />";

// define("7VERSION", "7.4");
// ECHO 7VERSION;

define("NAME", "Nghia");
echo NAME;

echo constant('NAME');
?>
```

9. HẲNG SỐ



- Hằng số là một giá trị không thể thay đổi trong suốt quá trình thực thi chương trình
- Có hai cách định nghĩa hằng số trong PHP
 - Sử dụng hàm define()
 - Sử dụng từ khóa const (kể từ PHP 5.3)

```
<?php
define("LANGUAGE", "PHP");
$lang = LANGUAGE;
echo "This program is written using $lang <br />";
echo "This program is written using LANGUAGE <br />";
echo "This program is written using ". LANGUAGE . "<br />";

// define("LANGUAGE", "JAVA");
// echo "This program is written using ". LANGUAGE .
"<br />";

// define("7VERSION", "7.4");
// ECHO 7VERSION;

define("NAME", "Nghia");
echo NAME;

echo constant('NAME');

?>
```



10. CÁC LOẠI TOÁN TỬ

- Toán tử số học

```
<?php  
  
$math = 7;  
$science = 3;  
$english = 10;  
$total = $math + $science + $english;  
  
echo "Total Mark: $total for Math($math) + Science($science) +  
English($english) <br />";  
  
echo "Total Mark: $math + $science + $english for Math($math) +  
Science($science) + English($english) <br />";  
echo "Total Mark: ", ($math + $science + $english), " for  
Math($math) + Science($science) + English($english) <br />";  
  
$length = 10;  
$breath = 20;  
$area = $length * $breath;  
echo $area . "<br />";  
  
$percent = ($total / 30) * 100;  
echo "Percentage score: $percent <br />";
```

```
$a = 100 % 5;  
echo $a . "<br />";  
$a = 101 % 5;  
echo $a . "<br />";  
$a = 102 % 5;  
echo $a . "<br />";  
$a = 103 % 5;  
echo $a . "<br />";  
  
$a = 10;  
$b = 20;  
$result = ($a ** 2) * ($b ** 2);  
echo $result;  
?>
```



10. CÁC LOẠI TOÁN TỬ

- Toán tử điều kiện và toán tử so sánh

```
<?php

$status = (empty($user)) ? "not logged in" : "logged in";
echo $status . "<br />";

$user = "Me";
$status = (empty($user)) ? "not logged in" : "logged in";
echo $status . "<br />";

echo $color = $color ?? "red";
echo "<br />";
// Toán tử so sánh
$x = 100;
$y = "100";
var_dump($x == $y);
echo "<br />";
var_dump($x === $y);
echo "<br />";

var_dump($x != $y);
echo "<br />";
var_dump($x <> $y);
echo "<br />";
var_dump($x !== $y);
echo "<br />";
```

```
$x = 100;
$y = 50;
var_dump($x > $y);
echo "<br />";
var_dump($x < $y);
echo "<br />";

$x = 50;
$y = 50;
var_dump($x >= $y);
echo "<br />";
var_dump($x <= $y);
echo "<br />";
?>
```



10. CÁC LOẠI TOÁN TỬ

- Toán tử logic

```
<?php
```

```
$value1 = true;  
$value2 = false;
```

```
$result1 = $value1 and $value2;  
$result2 = $value1 && $value2;  
var_dump($result1);  
echo "<br />";  
var_dump($result2);  
echo "<br />";  
echo "<br />";
```

```
$value1 = true;  
$value2 = false;  
$result1 = ($value1 or $value2);  
$result2 = $value1 || $value2;
```

```
var_dump($result1);  
echo "<br />";  
var_dump($result2);  
echo "<br />";  
echo "<br />";
```

```
$value1 = false;  
$value2 = false;  
$result1 = ($value1 xor $value2);  
var_dump($result1);  
echo "<br />";  
echo "<br />";
```

```
$value1 = false;  
$result1 = !$value1;  
var_dump($result1);  
echo "<br />";  
?>
```



10. CÁC LOẠI TOÁN TỬ

- Toán tử gán

```
<?php
$mark = 10;
$name = "Me";

echo $name . "<br />";

$counter = 1;
echo $counter . "<br />";
// $counter = $counter + 1;
$counter += 1;
echo $counter . "<br />";

$fullname = "I Am Me ";
$fullname .= $name;
echo $fullname . "<br />";

$counter = 10;
$counter -= 5;
echo $counter . "<br />";

$square = 10;
$square *= $square;
echo $square . "<br />";
```

```
$val1 = 50;
$val1 /= 2;
echo $val1 . "<br />";

$val2 = 50;
$val2 %= 2;
echo $val2 . "<br />";
echo "<br />";

$counter = 1;
echo $counter++ . "<br />";
echo $counter . "<br />";

$counter = 1;
echo ++$counter . "<br />";
echo $counter . "<br />";

$counter = 1;
echo $counter-- . "<br />";
echo $counter . "<br />";

$counter = 1;
echo --$counter . "<br />";
echo $counter . "<br />";
?>
```




11. CÁC KIỂU DỮ LIỆU

- Kiểu dữ liệu số

```
<?php
$mark = 8;
echo $mark . "<br />";

$mark = 0x01AB56;
echo $mark . "<br />";

$mark = 10;
$total = "1000" + $mark;
echo $total . "<br />";

$mark = -5;
echo $mark . "<br />";

echo PHP_INT_MAX . "<br />";

$aave = -7.87;
echo $ave . "<br />";

$price = 4.49;
echo round($price) . "<br />";
echo abs($ave) . "<br />";
?>
```



11. CÁC KIỂU DỮ LIỆU

- Kiểu dữ liệu chuỗi

```
<?php
header("Content-Type: text/plain");
$name1 = "Nghia";
$name2 = 'Trong';
echo "$name1 and $name2" . "\n";
echo '$name1 and $name2' . "\n";

$name4 = "This is a \"Special\" String";
echo $name4 . "\n";

$name5 = "\t\tThis is a \"Special\" String";
echo $name5 . "\n";

$name5 = "\T\h\i\s is a \"Special\" String";
echo $name5 . "\n";

$name5 = "\t\h\i\s is a \"Special\" String";
echo $name5 . "\n";

$name5 = '\t\h\i\s is a \"Special\" String';
echo $name5 . "\n";

echo strlen($name5);
?>
```



11. CÁC KIỂU DỮ LIỆU

- Kiểu dữ liệu luận lý

```
<?php
$b1 = True;
$b2 = false;

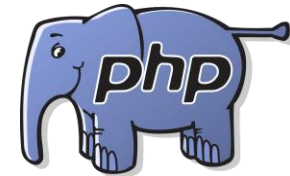
echo "True: " . $b1 . "<br />";
echo "False: " . $b2 . "<br />";

// string
$loggedin = true;
$username = "Nghia";

echo ( $loggedin && $username ) ? "Hello Me" : "Not Logged
in";
echo "<br />";
// int
$loggedin = true;
$userid = 0;

echo ( $loggedin && $userid ) ? "Hello Me" : "Not Logged
in";
echo "<br />";
?>
```

12. CÁC CẤU TRÚC ĐIỀU KHIỂN



- Cấu trúc rẽ nhánh

```
<?php
$number = 10;
$result = $number % 2;
if ($result) {
    echo 'Odd number <br />';
} else {
    echo 'Even number <br />';
}
?>
```

```
<!doctype html>
<html>
    <head>
        <title>if else statement</title>
    </head>
    <body>
        <?php
            $condition = false;
            if ($condition) {
                ?>
                <h1>This is a true Condition Block</h1>
                <?php
            }
            else {
                ?>
                <h1>This is a false Condition Block</h1>
                <?php
            }
        ?>
    </body>
</html>
```



12. CÁC CẤU TRÚC ĐIỀU KHIỂN

- Cấu trúc rẽ nhánh

```
<?php
if ($number > 10) {
    echo 'Number greater than 10 <br />';
} elseif ($number == 10) {
    echo 'Number equal 10. <br />';
} else {
    echo 'Number less than 10 <br />';
}
?>
```

```
<?php
switch ($result) {
    case 1:
        echo 'Odd number. <br />';
        break;
    case 0:
        echo 'Even number. <br />';
        break;
    default:
        echo 'Others .... <br />';
}
?>
```



12. CÁC CẤU TRÚC ĐIỀU KHIỂN

- Cấu trúc lặp

```
<?php
for ($value = 1; $value <= 10; $value++) {
    if ($value%2 == 1) {
        echo 'Odd Number: ' . $value . '<br />';
    }
}

echo '<br />';
$value = 1;
while ($value <= 10) {
    if ($value%2 == 1) {
        echo 'Odd Number: ' . $value . '<br />';
    }
    $value++;
}

echo '<br />';
$value = 1;
do {
    if ($value%2 == 1) {
        echo 'Odd Number: ' . $value . '<br />';
    }
    $value++;
} while ($value <= 10);
?>
```



12. CÁC CẤU TRÚC ĐIỀU KHIỂN

- break và continue

```
<?php
$value = 1;
do {
    echo "$value <br />";
    if ($value >= 10){
        break;
    }
    $value++;
} while (true);

echo "<br /><br />";
$value1 = 1;
$value2 = 1;
```

```
do {
    while (true) {
        if ($value2 >= 4){
            break;
        }
        echo "Value2: $value2 <br />";
        $value2++;
    }

    if ($value1 >= 5){
        break;
    }
    echo "Value1: $value1 <br />";
    $value1++;
} while (true);
echo "<br /><br />";
```

```
// continue
$value = 0;
while (true) {
    $value++;
    if ($value%2 == 1) {
        continue;
    }

    if ($value > 10){
        break;
    }
    echo "Value: $value <br />";
}
?>
```



13. namespace

- Namespace trong PHP là một cơ chế giúp tổ chức code một cách hiệu quả, đặc biệt khi làm việc với các dự án lớn hoặc các thư viện bên thứ ba
 - Nó tạo ra một không gian tên riêng biệt để các lớp, interface, function và các cấu trúc khác có thể được đặt vào đó, tránh xung đột tên với các cấu trúc khác có cùng tên trong các namespace khác.
- Ví dụ:
 - Giả sử bạn có hai lớp User khác nhau, một lớp dùng để quản lý người dùng trên website và một lớp khác dùng để quản lý người dùng trong một hệ thống khác. Nếu cả hai lớp này được đặt trong cùng một file hoặc không gian tên, sẽ xảy ra xung đột tên

```
<?php
namespace MyNamespace;

class MyClass {
    // ...
}
```

```
// Sử dụng lớp MyClass từ namespace MyNamespace
use MyNamespace\MyClass;

$obj = new MyClass();
```





14. Câu lệnh include

- Câu lệnh include trong PHP được sử dụng để nhúng nội dung của một file PHP vào một file PHP khác nhằm chia nhỏ code thành nhiều file nhỏ hơn, giúp code dễ quản lý, dễ bảo trì và tái sử dụng hơn.
 - include: Nếu file không tồn tại, PHP sẽ hiển thị một thông báo lỗi nhưng vẫn tiếp tục thực thi chương trình.
 - require: Nếu file không tồn tại, PHP sẽ tạo ra một lỗi fatal error và dừng việc thực thi chương trình.

```
<?php
include "inc/header.php";
?>

<body>

<?php
include "inc/menu.php";
require_once "inc/menu.php";
?>

<h1>Use include Statements</h1>

<?
include "inc/footer.html";
?>
```



15. Mảng

- Mảng (Array) trong PHP là một cấu trúc dữ liệu có khả năng lưu trữ nhiều giá trị khác nhau dưới một tên biến duy nhất. Các giá trị này có thể thuộc các kiểu dữ liệu khác nhau (số, chuỗi, boolean, thậm chí là các mảng khác), và chúng được truy cập thông qua một chỉ số (index).

```
<?php
$arr = [1, 2, 3];

var_dump($arr);
echo "<br />";
print_r($arr);
echo "<br />";

echo $arr[0] . "<br />";
echo $arr[1] . "<br />";
echo $arr[2] . "<br />";

echo "Array Length: " . count($arr) . "<br />";
for ($count = 0; $count < count($arr); $count++) {
    echo $arr[$count] . "<br />";
}

$arr[0] = 5;
$arr[1] = 6;
$arr[2] = 7;
var_dump($arr);
echo "<br />";

$arr[0] = "String";
$arr[1] = true;
$arr[2] = 6.5;
var_dump($arr);
echo "<br />";
?>
```



15. Mảng

- Duyệt mảng (ngoài cách thông thường như for, while ...)

```
<?php
$arr = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'];
var_dump($arr);
echo "<br />";

foreach ($arr as $value) {
    echo $value . "<br />";
}

$count = 0;
foreach ($arr as $value) {
    echo "[".$count."] => $value <br />";
    $count++;
}
?>
```



15. Mảng

- Mảng rỗng

```
<?php
$arr = [];
var_dump($arr);
echo "<br />";

if (empty($arr)) {
    echo "Array is empty. <br />";
}
```

```
$arr = 10;
var_dump($arr);
echo "<br />";
```

```
$arr = [10];
var_dump($arr);
echo "<br />";
```

```
$arr[0] = [10];
var_dump($arr);
echo "<br />";
```

```
$arr = [];
$arr[0] = 1;
$arr[2] = "string";
$arr[4] = true;
$arr[10] = 14.5;
var_dump($arr);
echo "<br />";
```

```
$arr[] = "new1";
$arr[] = "new2";
var_dump($arr);
echo "<br />";
```

```
foreach ($arr as $value) {
    echo $value . "<br/ >";
}
?>
```



15. Mảng

- Mảng kết hợp

```
<?php
$arr = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'];

$arr = [2 => 'Mon', 3 => 'Tue', 4 => 'Wed', 5 => 'Thu', 6 =>
'Fri', 7 => 'Sat', 8 => 'Sun'];
var_dump($arr);
echo "<br />";

foreach ($arr as $value) {
    echo $value . "<br />";
}

foreach ($arr as $key => $value) {
    echo "$key => $value" . "<br />";
}

foreach (array_keys($arr) as $key) {
    echo $key . "<br />";
}

var_dump(array_keys($arr));
echo "<br />";

$userEmails = [
    "John" => "john@gmail.com",
    "Jenn" => "jenn@gmail.com",
    "Mary" => "mary@gmail.com"
```

```
var_dump($userEmails);
echo "<br />";
echo $userEmails["John"] . "<br />";

$userEmails[] = ["Jane" => "jane@gmail.com"];
var_dump($userEmails);
echo "<br />";
echo $userEmails["Jane"] . "<br />";

$userEmails = [
    "John" => "john@gmail.com",
    "Jenn" => "jenn@gmail.com",
    "Mary" => "mary@gmail.com"
];

$userEmails["Jane"] = "jane@gmail.com";
var_dump($userEmails);
echo "<br />";

unset($userEmails["Jane"]);
var_dump($userEmails);
echo "<br />";

unset($userEmails["John"]);
var_dump($userEmails);
echo "<br />";
?>
```





15. Mảng

- Mảng đa chiều

```
<?php
$arr = [1, 2, 3, 4, 5];

$arr1 = [
    "John" => "john@gmail.com",
    "Mary" => "mary@gmail.com"
];

// Multi dimension Array
$arr2 = [
    [0, 1, 2, 3, 4, 5],
    [6, 7, 8],
    [9, 10]
];
var_dump($arr2);
echo "<br />";
foreach ($arr2 as $singleArr) {
    var_dump($singleArr);
    echo "<br />";
}

foreach ($arr2 as $singleArr) {
    echo '[';
    foreach ($singleArr as $value) {
        echo $value . ", ";
    }
    echo " ]<br />";
}
```

```
$arr3 = [
    "emaillist1" => [
        "John1" => "john1@gmail.com",
        "Mary1" => "mary1@gmail.com"
    ],
    "emaillist2" => [
        "John2" => "john2@gmail.com",
        "Mary2" => "mary2@gmail.com"
    ]
];
var_dump($arr3);
echo "<br />";

foreach ($arr3 as $singleArr) {
    echo '[';
    foreach ($singleArr as $value) {
        echo $value . ", ";
    }
    echo " ]<br />";
}

foreach ($arr3 as $key => $singleArr) {
    echo "Value for key: $key [";
    foreach ($singleArr as $value) {
        echo $value . ", ";
    }
    echo " ]<br />";
}
?>
```





15. Mảng

- Sắp xếp mảng

```
<?php
// Indexed array
$arr = [ 5, 2, 4, 3, 1, 0];
foreach ($arr as $val) {
    echo "$val ";
}
echo "<br />";
sort($arr);
foreach ($arr as $val) {
    echo "$val ";
}
echo "<br />";
rsort($arr);
foreach ($arr as $val) {
    echo "$val ";
}
echo "<br />";
echo "<br />";

// Associative Array
$arr = [
    "3" => "Nghia",
    "1" => "Trong",
    "2" => "Dinh"
];
```

```
foreach ($arr as $key => $val) {
    echo "$key => $val, ";
}
echo "<br />";
asort($arr);
foreach ($arr as $key => $val) {
    echo "$key => $val, ";
}
echo "<br />";
arsort($arr);
foreach ($arr as $key => $val) {
    echo "$key => $val, ";
}
echo "<br />";

ksort($arr);
foreach ($arr as $key => $val) {
    echo "$key => $val, ";
}
echo "<br />";
krsort($arr);
foreach ($arr as $key => $val) {
    echo "$key => $val, ";
}
echo "<br />";
?>
```



15. Mảng

- Phân tách mảng

```
<?php
$students = "Đai, Hoc, Thuy, Loi";
$stuArr = explode(", ", $students);
var_dump($stuArr);
echo "<br />";
$stuList = implode(", ", $stuArr);
echo $stuList . "<br />";

$stuArr = explode(", ", $students, 2);
var_dump($stuArr);
echo "<br />";
?>
```




16. Hàm

- Định nghĩa hàm

```
<?php
declare(strict_types = 1);
function printNumbers(int $limit) {
    for ($i=1; $i <= $limit; $i++) {
        echo $i . " ";
    }
    echo "<br />";
}

printNumbers(10);
//printNumbers("10");
?>
```



16. Hàm

- Hàm với giá trị trả về

```
<?php
function add($a, $b) : int {
    $result = $a + $b;
    return $result;
}

echo "Result: " . add(2, 3) . "<br />";

function getEvenNumbers($limit) : array {
    $retArr = [];
    for ($i = 0; $i <= $limit; $i++) {
        if ($i%2==0) {
            $retArr[] = $i;
        }
    }
    return $retArr;
}

$evenNums = getEvenNumbers(10);
print_r($evenNums);
?>
```



16. Hàm

- Truyền tham số cho hàm

```
<?php
function getEvenNumbers($limit, $skip = 0) : array {
    $retArr = [];
    for ($i = 0; $i <= $limit; $i++) {
        if ($i == $skip) {
            continue;
        }
        if ($i%2==0) {
            $retArr[] = $i;
        }
    }
    return $retArr;
}

$evenNums = getEvenNumbers(10);
print_r($evenNums);
echo "<br /><br />";

$result = 0;
function add($a, $b, &$result) {
    $result = $a + $b;
}
add(1, 2, $result);
echo $result . "<br />";
?>
```



17. Xử lý ngoại lệ

- try --- catch

```
<?php
// try {
//     // code
//     throw new Exception("Error Example");

//     echo "This line not excuted";
// } catch (Exception $e) {
//     echo "Caught Exeption: " . $e->getMessage() . "<br
//>";
// }
```



```
function div($x, $y) : int
{
    if ($y == 0) {
        throw new Exception("Divide by Zero.");
    }
    $result = $x / $y;
    return $result;
}

try {
    $result = div(1, 0);
} catch (Exception $e) {
    echo "Caught Exeption: " . $e->getMessage() . "<br />";
}
?>
```



17. Xử lý ngoại lệ

- Ngoại lệ tùy chỉnh

```
<?php
function custom_exception_handler($exception) {
    echo "Uncaught Exception: " . $exception->getMessage() .
"<br />";
}
```

```
set_exception_handler('custom_exception_handler');
```

```
// throw new Exception("Raise Exception");
```

```
function add($a, $b) : int {
    if ($a <= 0) {
        throw new Exception("Input param a <= 0");
    }
    $result = $a + $b;
    return $result;
}
```

```
echo add(2,3) . "<br />";
```

```
// Multiple Exception
```

```
function div($x, $y) : int
{
    try {
        if ($y == 0) {
            throw new Exception("Divide by Zero.");
        }
    } catch (Exception $e) {
        throw new Exception("Input param y = 0");
    }

    $result = $x / $y;
    return $result;
}
```

```
echo div(4, 0) . "<br />";
?>
```





17. Xử lý ngoại lệ

- finally

```
<?php
function custom_exception_handler($exception) {
    echo "Uncaught Exception: " . $exception->getMessage() .
"<br />";
}

set_exception_handler('custom_exception_handler');

// throw new Exception("Raise Exception");
// echo "Hello";

try {
    throw new Exception("Raise Exception");
}
catch (Exception $e) {
    echo "Caught Exception: " . $e->getMessage() . "<br />";
}
finally {
    echo "Hello <br />";
}

echo "End file";
?>
```



17. Xử lý ngoại lệ

- error

```
<?php
function myErrorHandler($errno, $errstr, $errfile, $errline)
{
    echo "<b>Error: </b> [$errno] - $errstr <br />";
    echo "Error on line $errline in $errfile. <br />";
}

set_error_handler('myErrorHandler');

$test = 2;

if ($test > 1) {
    trigger_error("test > 1");
}
?>
```



18. Làm việc với thư mục

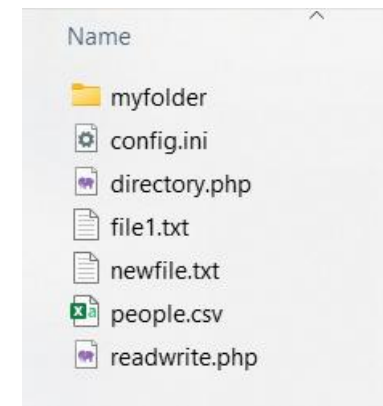
- Các hàm thao tác với thư mục

```
<?php
$path = ".";
$result = scandir($path);
$result = array_diff($result, ['.', '..']);
foreach ($result as $file) {
    if (is_file($file)) {
        echo $file . "<br />";
    }
}
echo "<br />";

$result = glob("*.php");
foreach ($result as $file) {
    if (is_file($file)) {
        echo $file . "<br />";
    }
}

if ( !file_exists("myfolder")) {
    mkdir("myfolder");
}

copy("file1.txt", "myfolder/file2.txt");
?>
```





19. Đọc và ghi tệp

- Đọc và ghi tệp tin

```
<?php
// Read txt file
$filename = "file1.txt";
$content = file_get_contents($filename);
echo $content . "<br />";
echo "<br />";

//2
$fileHandler = fopen($filename, "r");
$fileSize = filesize($filename);
$content = fread($fileHandler, $fileSize);
fclose($fileHandler);
echo $content . "<br />";
echo "<br />";

// Write txt file
$filename = "newfile.txt";
$fileHandler = fopen($filename, "w+") or die("Unable
to write the file");
fwrite($fileHandler, "Line 1. \n");
fwrite($fileHandler, "Line 2. \n");
fclose($fileHandler);
```

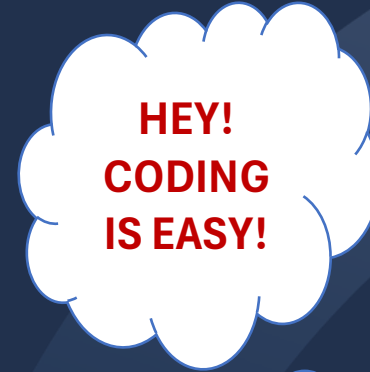
```
// Read ini file
$setting = parse_ini_file("config.ini");
print_r($setting);
echo "<br />";
foreach ($setting as $key => $value) {
    echo "$key => $value <br />";
}
echo "<br />";

// Read csv file
$filename = "people.csv";
$content = file_get_contents($filename);
var_dump(str_getcsv($content));
echo "<br />";

$csvfile = file($filename);
foreach ($csvfile as $line) {
    $data[] = str_getcsv($line);
}
print_r($data);
echo "<br /><br />";

$csv = array_map('str_getcsv', file($filename));
print_r($csv);
?>
```

“Câu hỏi & Thảo luận”



THE END!

