



THỰC HỌC – THỰC NGHIỆP



Conceive Design Implement Operate

## LẬP TRÌNH PHP1

PHP FILE

- ⦿ File inclusion
- ⦿ Làm việc với hệ thống tập tin (working with file system)
- ⦿ PHP send email
- ⦿ PHP upload files
- ⦿ Error & Exception





PHẦN 1

- ❑ Mã(code) giống nhau cần được gọi trên nhiều trang. Điều này có thể thực hiện bằng cách đặt các mã giống nhau trong một tập tin. Sau đó chèn tập tin đó bằng lệnh `include`
- ❑ `include` lấy tất cả văn bản trong tập tin được chỉ định chèn đến vị trí đã `include`
- ❑ `include` là một cấu trúc đặc biệt và không phải là hàm, vì vậy không nên sử dụng dấu ngoặc đơn khi `include` tập tin.

```
<?php  
include 'myfile.php';  
?>
```

```
<?php  
// myfile.php  
?>
```

- ❑ **Include path:** tập tin cần include có thể được chỉ định bằng đường dẫn tương đối (*relative path*), đường dẫn tuyệt đối (*absolute path*) hoặc không đường dẫn (*Without a path*).
- ❑ *relative path*: liên quan đến thư mục của tập tin
- ❑ *absolute path*: đường dẫn đầy đủ của tập tin

```
// Relative path  
include 'myfolder/myfile.php';  
  
// Absolute path  
include 'C:/xampp/htdocs/myfile.php';
```

## ❑ Include path

- ❖ Window: không phân biệt chữ hoa và chữ thường
- ❖ Linux: có phân biệt chữ hoa và chữ thường

## ❑ Relative path và no path:

- ❖ Tìm kiếm thư mục làm việc hiện tại cho tập tin
- ❖ Nếu không tìm thấy, xác định include\_path mặc định trong php.ini trước khi báo failing

```
// No path
```

```
include 'myfile.php';
```

## ❑ Ngoài include còn có require, include\_once và required\_once

## ❑ require

- ❖ Khi quá trình import tập tin chỉ định không thành công > tạm dừng các tập lệnh có lỗi bằng cách phát ra các warning.
- ❖ Quá trình import không thành công do không tìm thấy tập tin hoặc người dung đang chạy máy chủ web không có quyền đọc đối với tập tin chỉ định.

## ❑ include\_once

- ❖ Hoạt động giống include.
- ❖ Chèn tập tin chỉ định 1 lần (nếu đã đưa tập tin vào rồi thì sẽ không được đưa vào nữa)

## ❑ require\_once

- ❖ Hoạt động giống require.
- ❖ Chèn tập tin chỉ định 1 lần (nếu đã đưa tập tin vào rồi thì sẽ không được đưa vào nữa)

❑ Return: có thể thực hiện lệnh return trong tập tin import

```
<?php
// myimport.php
return 'OK';
?>
```

```
<?php
// myfile.php
if ((include 'myimport.php') == 'OK')
    echo 'OK';
?>
```



## ❑ Magic constants:

- ❖ `__FILE__` : Cho biết đường dẫn tuyệt đối của tập tin đang chạy
- ❖ `__DIR__` : Thư mục chứa file PHP đang chạy (thực thi)

## ❑ `Mkdir`: tạo thư mục

## ❑ `Rename`: đổi tên thư mục

## ❑ `Rmdir`: xóa thư mục

```
// Create directory
```

```
mkdir('test');
```

```
// Rename directory
```

```
rename('test', 'test2');
```

```
// Delete directory
```

```
rmdir('test2');
```

## ❑ Đọc tập tin và folders bên trong thư mục

```
$files = scandir('./');  
echo '<pre>';  
var_dump($files);  
echo '</pre>';
```

## ❑ file\_get\_contents: lấy nội dung của tập tin

## ❑ file\_put\_contents: ghi nội dung vào tập tin. Nếu tập tin không tồn tại, hàm sẽ tạo file và ghi nội dung vào đó.

```
// file_get_contents, file_put_contents  
$lorem = file_get_contents('lorem.txt');  
echo $lorem;  
echo '<br>';  
file_put_contents('lorem.txt', "First line" . PHP_EOL . $lorem);
```

Cú pháp: `file_input_contents($filename, $data, $flag)`

`$filename`: đường dẫn đến file

`$data`: nội dung của file

`$flag`: là tham số mang 1 trong 2 giá trị sau:

`FILE_USE_INCLUDE_PATH`: hàm sẽ tìm kiếm `$filename` trong `include directory`

`FILE_APPEND`: nếu file đã tồn tại, hàm sẽ không ghi `$data` vào nội dung sẵn có mà sẽ nối `$data` vào cuối file

- ❑ `File_exists`: kiểm tra tập tin có tồn tại hay không. Nếu có trả về `true`
- ❑ `Filesize`: lấy kích thước của tập tin
- ❑ `Unlink`: xóa tập tin

```
// Check if file exists or not  
file_exists('lorem.txt'); // true
```

```
// Get file size  
filesize('lorem.txt');
```

```
// Delete file  
unlink('lorem.txt');
```

- ❑ HTML form có input type cho phép tải tập tin lên máy chủ.
- ❑ Để tải tập tin lên có thể làm việc, thuộc tính enctype của form phải được thiết lập là "multipart/form-data"
- ❑ Ví dụ

```
<form action="mypage.php" method="post"
      enctype="multipart/form-data">
  <input name="myfile" type="file">
  <input type="submit" value="Upload">
</form>
```

- ❑ Thông tin về tập tin được upload được lưu trữ trong mảng `$_FILES`

Tên	Mô tả
<code>name</code>	Tên gốc của tập tin upload
<code>tmp_name</code>	Đường dẫn đến bản copy tạm thời máy chủ
<code>type</code>	Mime type của tập tin
<code>size</code>	Kích thước bytes của tập tin
<code>error</code>	Error code

❑ Tập tin nhận chỉ được lưu trữ tạm thời trên máy chủ. Nếu không được lưu bởi script, tập tin này sẽ bị xoá.

❑ Ví dụ

❖ Kiểm tra những mã lỗi để đảm bảo rằng tập tin đã nhận thành công và nếu có, di chuyển tập tin ra khỏi thư mục tạm thời để lưu.

```
$dest = 'upload/' . basename($_FILES['myfile']['name']);  
$file = $_FILES['myfile']['tmp_name'];  
$err  = $_FILES['myfile']['error'];  
  
if($err == 0 && move_uploaded_file($file, $dest))  
    echo 'File successfully uploaded';
```

❖ `move_uploaded_file`: đảm bảo rằng đối số đầu tiên chứa tập tin tải lên hợp lệ và nếu có, sẽ di chuyển tập tin đó đến đường dẫn và đổi tên thành tập tin được chỉ định bởi đối số thứ hai.

❖ Thư mục chỉ định phải tồn tại và nếu hàm thành công trong việc di chuyển, sẽ trả về true.

demo





PHẦN 2

- ❑ PHP mail là hàm được xây dựng sẵn được sử dụng để gửi email từ mã PHP
- ❑ Hàm mail chấp nhận các tham số sau

```
<?php  
mail($to_email_address,$subject,$message,[$headers],  
[$parameters]);  
?>
```

- ❖ \$to\_email\_address: địa chỉ email của người nhận thư.
- ❖ \$subject: chủ đề của email
- ❖ \$message: nội dung cần gửi
- ❖ [\$headers]: là tham số tùy chọn, có thể được sử dụng gồm CC, BCC

## □ Ví dụ

```
<?php
$to_email = 'name @ company . com';
$subject = 'Testing PHP Mail';
$message = 'This mail is sent using the PHP mail function';
$headers = 'From: noreply @ company . com';
mail($to_email,$subject,$message,$headers);
?>
```

## ❑ Sanitizing email user input

- ❖ Khi tạo form liên hệ và để người dùng điền vào các chi tiết. Người dùng có thể vô tình hoặc cố ý chèn code vào header, dẫn đến việc gửi thư rác. Để bảo vệ hệ thống khỏi các tấn công như vậy, chúng ta có thể tạo một chức năng tùy chỉnh để cải thiện và xác thực các giá trị trước khi thư được gửi đi.
- ❖ `Filter_var`: được sử dụng để làm sạch và xác thực dữ liệu đầu vào của người dùng.

## ❑ Sanitizing email user input

```
<?php
filter_var($field, SANITIZATION TYPE);
?>
```

- ❑ `filter_var`: hàm xác thực và làm sạch
- ❑ `$field`: giá trị của field sẽ được lọc.
- ❑ `SANITIZATION TYPE`: loại sanitization
  - ❖ `FILTER_VALIDATE_EMAIL`: trả về true cho địa chỉ email có giá trị.
  - ❖ `FILTER_SANITIZE_EMAIL`: di chuyển những ký tự bất hợp lệ từ địa chỉ email
  - ❖ `FILTER_SANITIZE_URL`: di chuyển những ký tự bất lệ từ url
  - ❖ `FILTER_SANITIZE_STRING`: xoá các tag từ giá trị chuỗi

- ❑ Error: là một lỗi trong mã mà nhà phát triển cần sửa. Khi một lỗi xảy ra trong php, mặc định là hiển thị thông báo lỗi trong trình duyệt.
- ❑ Thông báo này bao gồm tên tập tin, số dòng và mô tả lỗi.
- ❑ Compile và parse error (lỗi biên dịch và cú pháp) thường dễ phát hiện và sửa chữa, nhưng run-time error có thể khó tìm hơn vì một số tình huống nhất định và những lý do ngoài tầm kiểm soát

## ❑ PHP có 4 loại lỗi cơ bản

- ❖ Parse error of syntax error
- ❖ Fatal error
- ❖ Warning error
- ❖ Notice error

## ❑ Các mức lỗi

Tên	Mô tả
E_ERROR	Fatal run-time error (nghiêm trọng). Việc thực thi sẽ bị tạm dừng
E_WARNING	Non-fatal run-time error (không nghiêm trọng)
E_NOTICE	Run-time notice (thông báo lỗi run-time) có thể xảy ra.

## ❑ Các mức lỗi

Tên	Mô tả
E_USER_ERROR	Lỗi nghiêm trọng do người dùng tạo
E_USER_WARNING	Cảnh báo lỗi do người dùng tạo
E_USER_NOTICE	Thông báo lỗi do người dùng tạo.
E_COMPILE_ERROR	Lỗi compile-time nghiêm trọng
E_PARSE	Lỗi thời gian biên dịch cú pháp
E_STRICT	Đề xuất thay đổi để đảm bảo khả năng tương thích với trước đó
E_ALL	Tất cả lỗi, ngoại trừ E_STRICT trước PHP 5.4



## ❑ Các mức lỗi

```
// E_NOTICE (<PHP8) - Use of undefined variable
```

```
$a = $x;
```

```
// E_WARNING - Missing file
```

```
$b = fopen('missing.txt', 'r');
```

```
// E_ERROR - Missing function
```

```
$c = missing();
```

- ❑ PHP5 đã giới thiệu các ngoại lệ, một cơ chế tích hợp để xử lý các lỗi chương trình trong ngữ cảnh mà chúng xảy ra. Không giống như các lỗi thường được chỉnh sửa bởi nhà phát triển, các ngoại lệ được xử lý bởi script.
- ❑ `Die()` và `exit()`
  - ❖ Hai hàm này tương đương nhau
  - ❖ Gọi hàm này để dừng chương trình ngay lập tức

```
include ('header.php');  
blah blah blah  
if (!$mysql_connect) {  
    echo "unable to connect";  
    include ('footer.php');  
    exit;  
}
```

## ❑ Ví dụ

in index.php:

```
<?php  
define ('INDEX', true);  
?>
```

in your included file:

```
<?php  
if (!defined('INDEX')) {  
    die('You cannot call this script directly !');  
}  
?>
```

## □ Try...catch

- ❖ Khối try chứa mã có thể gây ra exception (ngoại lệ)
- ❖ Nếu khối try thực thi thành công, chương trình sẽ tiếp tục chạy sau câu lệnh try-catch.
- ❖ Try: hàm sử dụng ngoại lệ. Nếu ngoại lệ không kích hoạt, mã sẽ tiếp tục như bình thường. Tuy nhiên, nếu ngoại lệ kích hoạt, một ngoại lệ sẽ bị "thrown"
- ❖ Thrown: Đây là cách kích hoạt exception. Mỗi lần thrown phải có ít nhất 1 catch
- ❖ Catch: khối catch truy xuất exception và tạo một đối tượng chứa thông tin exception.

```

<?php
//create function with an exception
function checkNum($number) {
    if($number>1) {
        throw new Exception("Value must be 1 or below");
    }
    return true;
}

//trigger exception in a "try" block
try {
    checkNum(2);
    //If the exception is thrown, this text will not be shown
    echo 'If you see this, the number is 1 or below';
}

//catch exception
catch(Exception $e) {
    echo 'Message: ' . $e->getMessage();
}

finally {
    echo "First finally.\n";
}

echo "hello";
  
```

demo

- ☑ File inclusion
- ☑ Làm việc với hệ thống tập tin (working with file system)
- ☑ PHP send email
- ☑ PHP upload files
- ☑ Error & Exception



thank  
you!