

BÀI 5:

Xử lý tập tin trong PHP



Mục tiêu bài học

- Include nội dung trong PHP
- Upload file lên server bằng PHP
- Đọc – ghi tập tin văn bản với PHP

PHP Include Files

- Ta có thể nhúng nội dung của một file PHP vào một file PHP khác trước khi server thực thi file đó. Để thực hiện việc này, ta sử dụng 2 hàm của PHP.
 - The include() Function
 - The require() Function
- Việc sử dụng 2 hàm này sẽ cho phép chèn những vùng phổ biến có thể được sử dụng ở nhiều trang khác nhau, VD như: banner, footer, menu. Điều này sẽ giúp các nhà phát triển dễ dàng thay đổi bố cục (layout) của toàn bộ website mà không cần phải chỉnh sửa từng trang.
- Nếu có bất kỳ thay đổi nào, ta không cần phải sửa ở nhiều trang, mà ta chỉ cần chỉnh sửa ở file được included.

PHP Include Files

- Hai hàm include và require đều giống nhau, ngoại trừ việc xử lý lỗi:
 - Hàm require sẽ trả về một fatal error (E_COMPILE_ERROR) và dừng thực thi đoạn mã kịch bản khi có lỗi ở một dòng lệnh nào đó.
 - Hàm include sẽ chỉ trả về một cảnh báo (E_WARNING) khi có lỗi ở một dòng lệnh nào đó, và đoạn mã kịch bản sẽ tiếp tục.

PHP Include Files

➤ Cú pháp:

```
include 'filename';
```

or

```
require 'filename';
```

➤ Ví dụ:

```
<html>
<body>

<h1>Welcome to my home page!</h1>
<p>Some text.</p>
<p>Some more text.</p>
<?php include 'footer.php';?>

</body>
</html>
```

- Sự khác nhau lớn nhất giữa include và require là: khi một file được included với hàm include và PHP không tìm thấy file, đoạn mã kịch bản vẫn sẽ tiếp tục thực thi, còn hàm require thì sẽ trả về một fatal error

```
<html>
<body>

<h1>Welcome to my home page!</h1>
<?php require 'noFileExists.php';
echo "I have a $color $car.";
?>

</body>
</html>
```


- Giáo viên demo thiết kế bố cục trang web bằng cách include file

- Một đoạn mã kịch bản PHP có thể được sử dụng với một form HTML để cho phép người dùng upload file lên server. Đầu tiên file sẽ được upload lên một thư mục tạm thời trên server (temporary directory) và sau đó được chuyển đến một thư mục đích trên server.
- File được uploaded có thể là file văn bản hoặc hình ảnh, hoặc bất kỳ tài liệu nào.
- Quá trình upload file bao gồm các bước sau:
 - Người dùng mở trang web có chứa form, trong form có phần tử upload file.
 - Người dùng click lên browse button của phần tử upload file, rồi chọn một file để upload từ máy tính cá nhân.
 - Sau khi đường dẫn của file được chọn hiển thị trên text filed, người dùng click lên nút submit.
 - File được chọn sẽ được chuyển đến thư mục tạm thời (temporary directory) trên server.
 - Sau đó file sẽ được chuyển đến thư mục đích được chỉ định trong đoạn mã kịch bản PHP.

- Để phục vụ cho quá trình upload file, PHP cung cấp một biến toàn cục có tên là **\$_FILES**. Biến này là một mảng associate hai chiều, nó lưu trữ tất cả thông tin liên quan đến file được upload.
- Biến **\$_FILES** bao gồm 5 thuộc tính sau (nếu phần tử input được đặt tên là file):
 - **\$_FILES['file']['tmp_name']** – tên của file được đặt ở trong thư mục tạm thời (temporary directory) trên web server.
 - **\$_FILES['file']['name']** – tên thực tế của file được upload lên server.
 - **\$_FILES['file']['size']** – kích thước (được tính bằng bytes) của file được upload.
 - **\$_FILES['file']['type']** – kiểu MIME type của file được upload.
 - **\$_FILES['file']['error']** – Mã lỗi được kết hợp với file được upload.

PHP File Uploading

➤ Ví dụ:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Demo upload file</title>
</head>
<body>
<form action="xulyuploadfile.php" method="post" enctype="multipart/form-data">
Mời bạn chọn hồ sơ:
<input type="file" name="myfile"/><BR>
<input type="submit" value="Upload file" />
</form>
</body>
</html>
```

← → ↻ ⓘ localhost:8888/demophpdn37/formupload.php

Mời bạn chọn hồ sơ: No file chosen

➤ File xử lý:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<?php
    //lay ve ten file
    $temp_name = $_FILES["myfile"]["tmp_name"];
    $filename = $_FILES['myfile']['name'];
    echo 'Temp name: ' . $_FILES["myfile"]["tmp_name"];
    echo '<BR>Name: ' . $_FILES['myfile']['name'];
    echo '<BR>Size: ' . $_FILES['myfile']['size'];

    //upload va luu file vao thu muc images tren server
    move_uploaded_file($temp_name, 'images/' . $filename);
    echo 'Upload thành công !';
?>
</body>
</html>
```

← → ↻ ⓘ localhost:8888/demophp37/xulyuploadfile.php

Temp name: C:\xampp\tmp\php2A47.tmp
Name: Huong dan insert du lieu voi PHP MVC.pdf
Size: 1026087Upload thành công !

Ví dụ 2:

```
<?php
    if(isset($_FILES['image'])){
        $errors= array();
        $file_name = $_FILES['image']['name'];
        $file_size = $_FILES['image']['size'];
        $file_tmp = $_FILES['image']['tmp_name'];
        $file_type=$_FILES['image']['type'];
        $file_ext=strtolower(end(explode('.',$_FILES['image']['name'])));

        $extensions= array("jpeg","jpg","png");

        if(in_array($file_ext,$extensions)=== false){
            $errors[]="extension not allowed, please choose a JPEG or PNG file.";
        }

        if($file_size > 2097152){
            $errors[]='File size must be excately 2 MB';
        }

        if(empty($errors)==true){
            move_uploaded_file($file_tmp,"images/".$file_name);
            echo "Success";
        }else{
            print_r($errors);
        }
    }
?>
<html>
    <body>

        <form action="" method="POST" enctype="multipart/form-data">
            <input type="file" name="image" />
            <input type="submit"/>
        </form>

    </body>
</html>
```


- Một số quy tắc cần chú ý với HTML form khi upload file:
 - Kiểu request của form phải là `method="post"`
 - Form cũng cần khai báo thuộc tính sau:
`enctype="multipart/form-data"`. Thuộc tính này cho phép thiết lập content-type khi submit form.
 - Nếu không khai báo các thuộc tính trên, file sẽ không upload được thành công.
 - Cần chuẩn bị sẵn thư mục chứa file trên server, cũng như thiết lập quyền tạo tập tin trong thư mục đó trên server.

- Để mở tập tin trong PHP, ta sử dụng hàm **fopen()**. Hàm này bao gồm 2 tham số:
 - Tham số thứ nhất là tên file cần mở.
 - Tham số thứ hai là chế độ (mode) thao tác với tập tin.
- Nếu quá trình mở file thất bại, hàm **fopen()** sẽ trả về một giá trị **false**, nếu mở file thành công hàm này sẽ trả về một con trỏ file (**file pointer**) được sử dụng để phục vụ cho quá trình đọc/ghi file về sau.
- Sau khi đã thao tác với file, ta nên đóng file bằng cách gọi hàm **fclose()**. Hàm **fclose()** yêu cầu một tham số là con trỏ file và trả về **true** nếu đóng file thành công, hoặc **false** nếu đóng file thất bại.

- Ngay khi file được mở bằng hàm **fopen()**, ta có thể đọc file bằng cách gọi hàm **fread()**. Hàm này sẽ nhận 2 đối số: con trỏ file và độ dài của file được tính theo byte.
- Chiều dài hay kích thước của file có thể nhận được bằng hàm **filesize()**. Hàm này sẽ nhận tên file làm đối số, và sẽ trả về kích thước của file tính theo byte.
- Dưới đây là các bước để đọc file trong PHP.
 - Mở file bằng cách gọi hàm **fopen()**.
 - Lấy về kích thước file bằng hàm **filesize()**.
 - Đọc nội dung của file bằng hàm **fread()**.
 - Đóng file bằng hàm **fclose()**.

➤ Ví dụ:

```
<?php
    $filename = "tmp.txt";
    $file = fopen( $filename, "r" );

    if( $file == false ) {
        echo ( "Error in opening file" );
        exit();
    }

    $filesize = filesize( $filename );
    $filetext = fread( $file, $filesize );
    fclose( $file );

    echo ( "File size : $filesize bytes" );
    echo ( "<pre>$filetext</pre>" );
?>
```

- Để ghi file mới, hoặc ghi thêm nội dung vào một file đã tồn tại, ta gọi hàm **fwrite()**. Hàm này nhận 2 đối số đầu vào:
 - Một con trỏ file (**file pointer**)
 - Một chuỗi dữ liệu dùng để ghi lên file.
- Sau khi đã ghi xong file, có thể dùng hàm **file_exist()** để kiểm tra xem file có tồn tại hay không.
- Ví dụ:

```
<?php
$filename = "/home/user/guest/newfile.txt";
$file = fopen( $filename, "w" );

if( $file == false ) {
    echo ( "Error in opening new file" );
    exit();
}
fwrite( $file, "This is a simple test\n" );
fclose( $file );
?>
```

➤ Một số chế độ mở file trong PHP

Chế độ	Ý nghĩa
r	Mở file theo chế độ read-only (chỉ đọc).
r+	Mở file theo chế độ đọc và ghi
w	Mở file theo chế độ chỉ ghi
w+	Mở file theo chế độ đọc và ghi
a	Mở file theo chế độ ghi nối (chỉ ghi)
a+	Mở file theo chế độ đọc và ghi (ghi nối)

- Hàm `print_r()`: Hiển thị thông tin mô tả có thể đọc được về một biến.

- Ví dụ:

```
<pre>
<?php
$a = array ('a' => 'apple', 'b' => 'banana', 'c' => array ('x', 'y', 'z'));
print_r ($a);
?>
</pre>
```

- Kết quả:

```
<pre>
Array
(
    [a] => apple
    [b] => banana
    [c] => Array
        (
            [0] => x
            [1] => y
            [2] => z
        )
)
</pre>
```


Tổng kết

- Include nội dung trong PHP
- Upload file lên server bằng PHP
- Đọc – ghi tập tin văn bản với PHP