Bài 2. PHP form và PHP nâng cao

Giảng viên: Trần Thị Thu Phương
Email: tttphuong2@hnmu.edu.vn
Mobile: 0966 224784

Phần 1. PHP form

Nội dung

1. Xử lý biểu mẫu
2. Validate biểu mẫu
3. Thiết lập các trường bắt buộc
4. Validate E-mail và URL
5. Ví dụ

Xử lý biểu mẫu: get và post

- GET và POST là hai phương thức chính được sử dụng để truyền dữ liệu từ client (trình duyệt) đến server (trang web). Đây là cách để trình duyệt gửi dữ liệu và yêu cầu từ server

Phương thức post

- Phương thức POST truyền thông tin bằng cách gửi dữ liệu trong phần body của yêu cầu HTTP. Dữ liệu không hiển thị trực tiếp trên URL và thường được sử dụng khi gửi dữ liệu lớn hoặc nhạy cảm như mật khẩu.

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

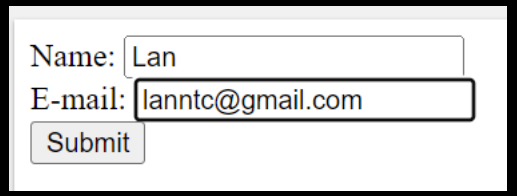
```
<html>
<body>

Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>

</body>
</html>
```

The output could be something like this:

```
Welcome John
Your email address is john.doe@example.com
```

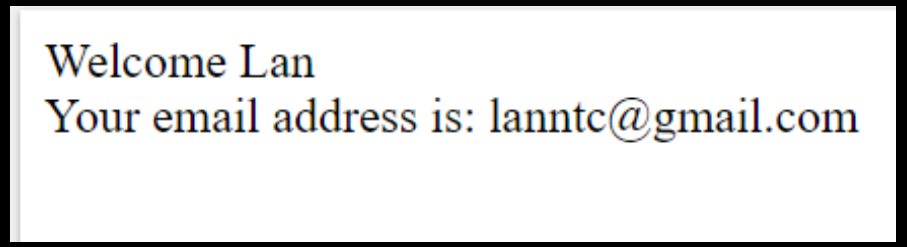
Ví dụ post

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```




```
<html>
<body>

Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>

</body>
</html>
```

Welcome.php



Phương thức get

- Phương thức GET truyền thông tin bằng cách đính kèm dữ liệu vào URL của trang. Dữ liệu được gửi dưới dạng các cặp key-value (tên tham số - giá trị).
- Phương thức GET thích hợp khi muốn truyền dữ liệu nhỏ, không nhạy cảm và có thể hiển thị trực tiếp trên URL.

```
<html>
<body>

<form action="welcome_get.php" method="get">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

```
<html>
<body>

Welcome <?php echo $_GET["name"]; ?><br>
Your email address is: <?php echo $_GET["email"]; ?>

</body>
</html>
```

BÀI TẬP

- Tạo form gồm các thuộc tính:

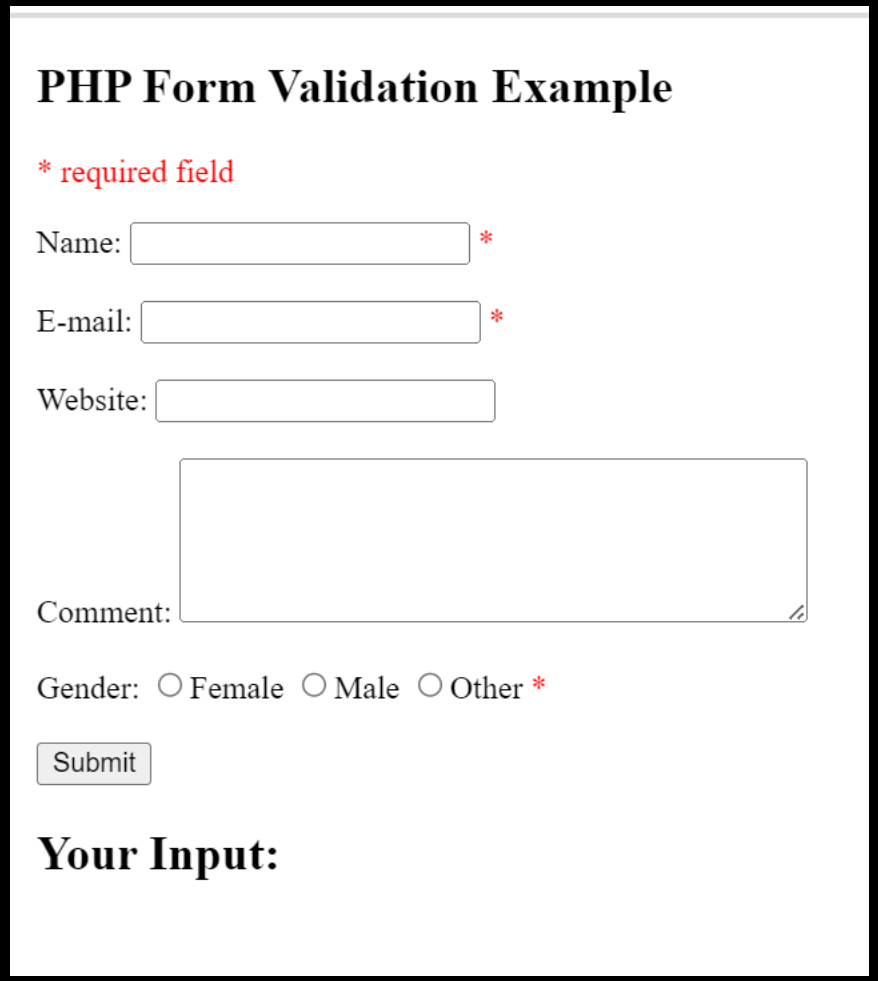
  tên sách, tác giả, nhà xuất bản, năm xuất bản

Sử dụng Phương thức get và post đề gửi và hiển thị lại dữ liệu.

Demo get và post

- Kết luận:
- Hai ví dụ trên không chứa bất kỳ xác thực biểu mẫu nào, nó chỉ hiển thị cách có thể gửi và truy xuất dữ liệu biểu mẫu.
- Việc xác thực đúng dữ liệu biểu mẫu là rất quan trọng để bảo vệ biểu mẫu khỏi tin tặc và kẻ gửi thư rác!

Form validate



PHP Form Validation Example

* required field

Name: *

E-mail: *

Website:

Comment:

Gender: Female Male Other *

Submit

Your Input:

Field	Validation Rules
Name	Required. + Must only contain letters and whitespace
E-mail	Required. + Must contain a valid email address (with @ and .)
Website	Optional. If present, it must contain a valid URL
Comment	Optional. Multi-line input field (textarea)
Gender	Required. Must select one

Form validate

- Việc xác thực đúng dữ liệu biểu mẫu là rất quan trọng để bảo vệ biểu mẫu khỏi tấn công.
- <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"> trong đó:
- $_SERVER["PHP_SELF"] là một biến siêu toàn cầu trả về tên tệp của tập lệnh hiện đang thực thi
- Vì vậy, $_SERVER["PHP_SELF"] sẽ gửi dữ liệu biểu mẫu đã gửi đến chính trang đó, thay vì chuyển sang một trang khác. Bằng cách này, người dùng sẽ nhận được thông báo lỗi trên cùng một trang với biểu mẫu.

Bảo mật biểu mẫu PHP

- Biến $_SERVER["PHP_SELF"] có thể bị tin tặc sử dụng!
- Nếu PHP_SELF được sử dụng trong trang web thì người dùng có thể nhập dấu gạch chéo (/) và sau đó nhập một số lệnh Cross Site Scripting (XSS) để thực thi
- Cross-site scripting (XSS) là một loại lỗ hổng bảo mật máy tính thường thấy trong các ứng dụng Web. XSS cho phép kẻ tấn công đưa tập lệnh phía máy khách vào các trang Web được người dùng khác xem.

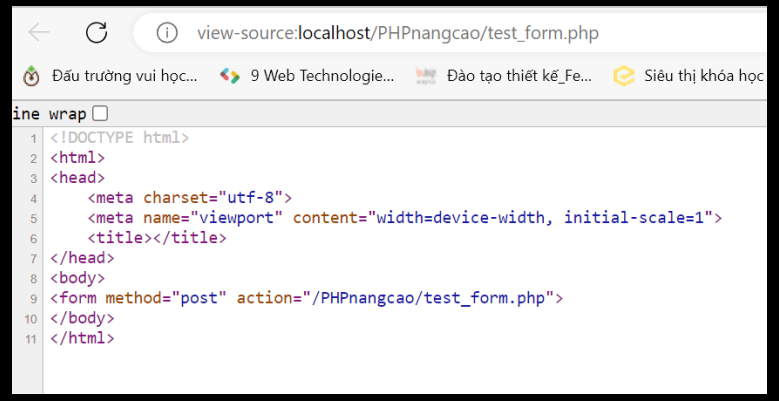
Bảo mật biểu mẫu

- Giả sử chúng ta có biểu mẫu sau trong trang có tên "test_form.php":

```
<form method="post" action="<?php echo $_SERVER["PHP_SELF"];?>">
```

- Bây giờ, nếu người dùng nhập URL bình thường vào thanh địa chỉ như "http://www.example.com/test_form.php", đoạn mã trên sẽ được dịch thành:

```
<form method="post" action="test_form.php">
```



Bảo mật biểu mẫu (tiếp)

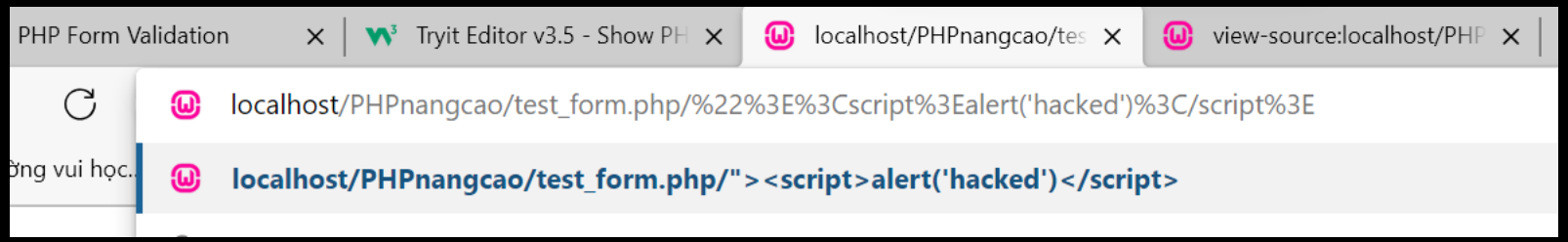
Tuy nhiên, hãy xem xét rằng người dùng nhập URL sau vào thanh địa chỉ:

```
http://www.example.com/test_form.php/%22%3E%3Cscript%3Ealert('hacked')%3C/script%3E
```

Trong trường hợp này, đoạn mã trên sẽ được dịch sang:

```
<form method="post" action="test_form.php/"><script>alert('hacked')</script>
```

Mã này thêm thẻ tập lệnh và lệnh cảnh báo. Và khi tải trang, mã JavaScript sẽ được thực thi (người dùng sẽ thấy một hộp cảnh báo). Đây chỉ là một ví dụ đơn giản và vô hại về cách khai thác biến PHP_SELF.



Tránh lỗ hổng

Có thể tránh khai thác $_SERVER["PHP_SELF"] bằng cách sử dụng hàm htmlspecialchars().

Mã biểu mẫu sẽ trông như thế này:

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

Hàm htmlspecialchars() chuyển đổi các ký tự đặc biệt thành các thực thể HTML. Bây giờ nếu người dùng cố gắng khai thác biến PHP_SELF, nó sẽ dẫn đến kết quả như sau:

```
<form method="post"
action="test_form.php/&quot;&gt;&lt;script&gt;alert('hacked')&lt;/script&gt;">
```

- Không có bất cứ điều gì xảy ra cả!

Form validate

- Hàm htmlspecialchars() chuyển đổi các ký tự đặc biệt thành các thực thể HTML. Điều này có nghĩa là nó sẽ thay thế các ký tự HTML như < và > bằng < và >. Điều này ngăn những kẻ tấn công khai thác mã bằng cách chèn mã HTML hoặc Javascript vào các biểu mẫu.

Các bước xác thực biểu mẫu

1. chuyển tất cả các biến thông qua hàm htmlspecialchars() của PHP.
2. Loại bỏ các ký tự không cần thiết (dấu cách thừa, tab, dòng mới) khỏi dữ liệu đầu vào của người dùng (với hàm trim() của PHP)
3. Xóa dấu gạch chéo ngược (\) khỏi dữ liệu đầu vào của người dùng (với chức năng dải dấu gạch chéo () của PHP)
4. Tạo một hàm sẽ thực hiện tất cả việc kiểm tra cho chúng ta (điều này thuận tiện hơn nhiều so với việc viết đi viết lại cùng một đoạn mã).

Các bước xác thực biểu mẫu

```php
<?php
// define variables and set to empty values
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $name = test_input($_POST["name"]);
  $email = test_input($_POST["email"]);
  $website = test_input($_POST["website"]);
  $comment = test_input($_POST["comment"]);
  $gender = test_input($_POST["gender"]);
}

function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
?>
```

Thiết lập các trường bắt buộc

Field	Validation Rules
Name	Required. + Must only contain letters and whitespace
E-mail	Required. + Must contain a valid email address (with @ and .)
Website	Optional. If present, it must contain a valid URL
Comment	Optional. Multi-line input field (textarea)
Gender	Required. Must select one

Php: thiết lập trường bắt buộc

```
<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
 if (empty($_POST["name"])) {
   $nameErr = "Name is required";
 } else {
   $name = test_input($_POST["name"]);
 }
 if (empty($_POST["email"])) {
   $emailErr = "Email is required";
 } else {
   $email = test_input($_POST["email"]);
 }
```

if it is not empty, it sends the user input data through the `test_input()` function:

```
if (empty($_POST["website"])) {
   $website = "";
 } else {
   $website = test_input($_POST["website"]);
 }
 if (empty($_POST["comment"])) {
   $comment = "";
 } else {
   $comment = test_input($_POST["comment"]);
 }
 if (empty($_POST["gender"])) {
   $genderErr = "Gender is required";
 } else {
   $gender = test_input($_POST["gender"]);
 }
}
?>
```

Xác thực tên

```
$name = test_input($_POST["name"]);
if (!preg_match("/^[a-zA-Z-' ]*$/",$name)) {
  $nameErr = "Only letters and white space allowed";
}
```

Hàm preg_match() tìm kiếm một chuỗi để tìm mẫu, trả về true nếu mẫu tồn tại và trả về false nếu ngược lại.

Xác thực email

```
$email = test_input($_POST["email"]);
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
  $emailErr = "Invalid email format";
}
```

Xác thực url

```
$website = test_input($_POST["website"]);
if (!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~_|!:,.;]*[-a-z0-9+&@#\/%=~_|]/i",$website)) {
  $websiteErr = "Invalid URL";
}
```

Ví dụ

- https://tryphp.w3schools.com/showphp.php?filename=demo_form_validation_special

Hiển thị lỗi

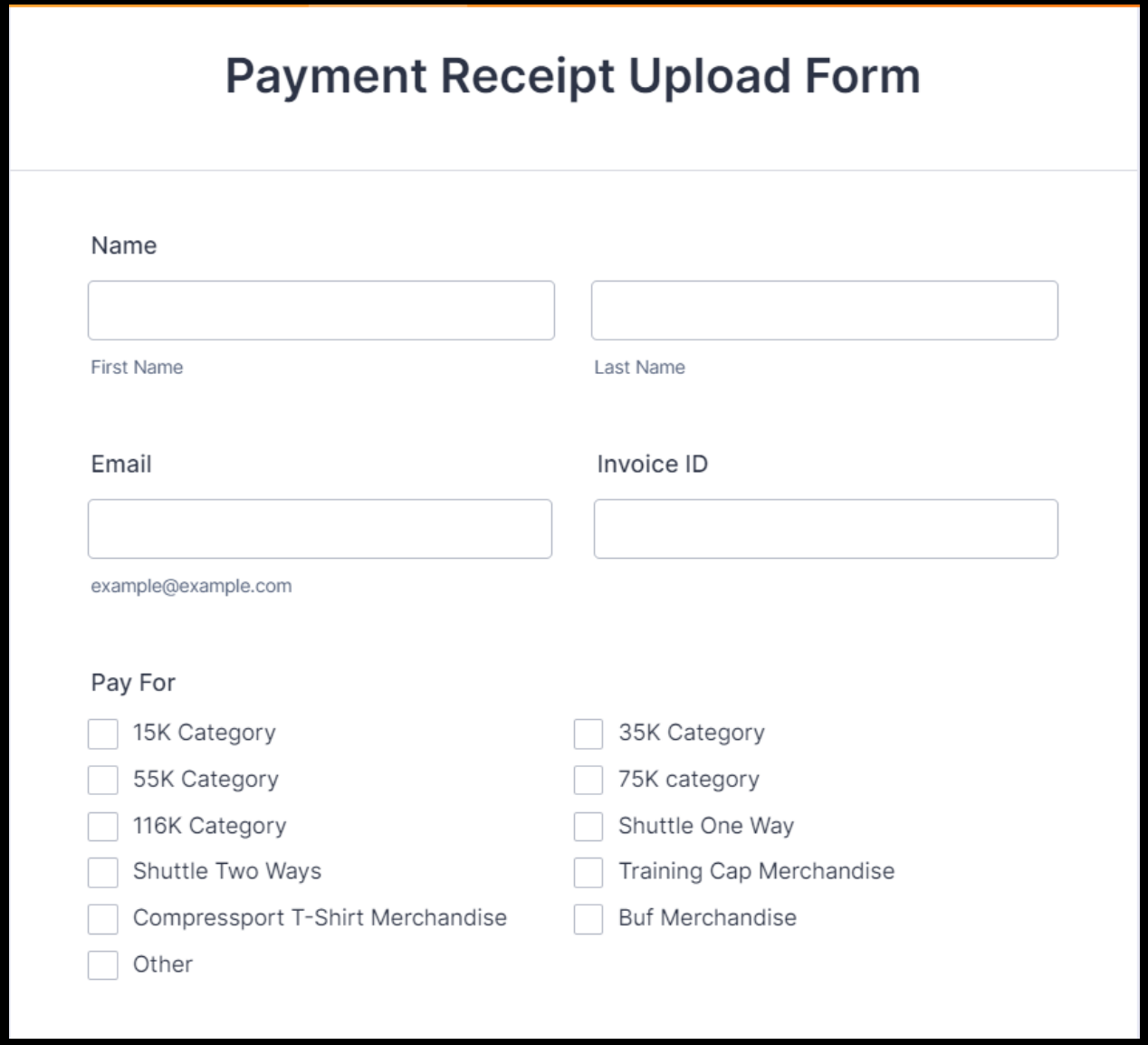
```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">

Name: <input type="text" name="name">
<span class="error">* <?php echo $nameErr;?></span>
<br><br>
E-mail:
<input type="text" name="email">
<span class="error">* <?php echo $emailErr;?></span>
<br><br>
Website:
<input type="text" name="website">
<span class="error"><?php echo $websiteErr;?></span>
<br><br>
Comment: <textarea name="comment" rows="5" cols="40"></textarea>
<br><br>
Gender:
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
<input type="radio" name="gender" value="other">Other
<span class="error">* <?php echo $genderErr;?></span>
<br><br>
<input type="submit" name="submit" value="Submit">

</form>
```

Bài tập 1

- Hoàn thiện chương trình validate form của bài demo (bằng 2 cách, validate phía client (sử dụng javascript) và validate phía server)
- Sử dụng Phương thức post gửi và trả lại dữ liệu do người dùng nhập vào.

Bài tập 2

- Tạo form như hình bên (bổ sung thêm nút submit)
- Tiến hành validate dữ liệu do người dùng nhập vào (Name, LastName, Email, Invoice ID- là số), phải chọn một trong Pay For)
- Sử dụng Phương thức post gửi và trả lại dữ liệu do người dùng nhập vào.



Phần 2. PHP nâng cao

Nội dung

- File handing (Xử lý tệp)
- File Open/read
- File create/Write
- File Upload
- Cookies
- Sessions
- Filter
- Callback
- JSON

Xử lý tệp

- Hàm readfile()đọc một tệp và ghi nó vào bộ đệm đầu ra.

```php
<!DOCTYPE html>
<html>
<body>

<?php
echo readfile("webdictionary.txt");
?>

</body>
</html>
```

AJAX = Asynchronous JavaScript and XML CSS = Cascading Style Sheets HTML = Hyper Text Markup Language PHP = PHP Hypertext Preprocessor SQL = Structured Query Language SVG = Scalable Vector Graphics XML = EXtensible Markup Language236

```
AJAX = Asynchronous JavaScript and XML
CSS = Cascading Style Sheets
HTML = Hyper Text Markup Language
PHP = PHP Hypertext Preprocessor
SQL = Structured Query Language
SVG = Scalable Vector Graphics
XML = EXtensible Markup Language
```

Webdictionary.txt

Mở/đọc/đóng tệp

- Một phương pháp tốt hơn để mở tệp là dùng hàm fopen(). Chức năng này cung cấp nhiều tùy chọn hơn readfile().
- Ví dụ

```
<?php
$myfile = fopen("webdictionary.txt", "r") or
die("Unable to open file!");
echo fread($myfile,filesize("webdictionary.txt"));
fclose($myfile);
?>
```

Đọc thêm tại: Mở/Đọc/Đóng tệp PHP (w3schools.com)

Tạo/ghi tệp

- Chức fwrite()năng được sử dụng để ghi vào một tập tin.
- Tham số đầu tiên fwrite()chứa tên của tệp để ghi vào và tham số thứ hai là chuỗi được ghi.

```php
<?php
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$txt = "John Doe\n";
fwrite($myfile, $txt);
$txt = "Jane Doe\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

Nếu chúng ta mở tệp "newfile.txt", nó sẽ trông như thế này:

```
John Doe
Jane Doe
```

Upload file (Tải tệp)

1. Tạo biểu mẫu HTML
2. Tạo tập lệnh PHP tải lên tệp
3. Kiểm tra xem tệp đã tồn tại chưa
4. Kích thước tệp giới hạn
5. Loại tệp giới hạn
6. Hoàn thành tải lên tệp PHP Script
7. Xem chi tiết tại: Tải lên tệp PHP (w3schools.com)

Tạo biểu mẫu

```
<!DOCTYPE html>
<html>
<body>

<form action="upload.php" method="post" enctype="multipart/form-data">
  Select image to upload:
  <input type="file" name="fileToUpload" id="fileToUpload">
  <input type="submit" value="Upload Image" name="submit">
</form>

</body>
</html>
```

- Đảm bảo rằng biểu mẫu sử dụng method="post"
- Biểu mẫu cũng cần thuộc tính sau: enctype="multipart/form-data". Nó chỉ định loại nội dung nào sẽ được sử dụng khi gửi biểu mẫu
- Nếu không có các yêu cầu trên, tệp tải lên sẽ không hoạt động.

Tạo tập lệnh PHP tải lên tệp

```php
<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
  $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
  if($check !== false) {
    echo "File is an image - " . $check["mime"] . ".";
    $uploadOk = 1;
  } else {
    echo "File is not an image.";
    $uploadOk = 0;
  }
}
?>
```

- $ target_dir = "uploads/" - chỉ định thư mục nơi tệp sẽ được đặt
- $ target_file chỉ định đường dẫn của tệp sẽ được tải lên
- $uploadOk=1 chưa được sử dụng (sẽ sử dụng sau)
- $imageFileType giữ phần mở rộng tệp của tệp (ở dạng chữ thường)
- Tiếp theo, kiểm tra xem tệp hình ảnh là hình ảnh thật hay hình ảnh giả mạo

Kiểm tra xem tệp đã có chưa

```
// Check if file already exists
if (file_exists($target_file)) {
  echo "Sorry, file already exists.";
  $uploadOk = 0;
}
```

- Kiểm tra xem tệp đã tồn tại trong thư mục “upload" chưa. Nếu có, một thông báo lỗi sẽ hiển thị và $uploadOk được đặt thành 0

Kích thước tệp giới hạn

```
// Check file size
if ($_FILES["fileToUpload"]["size"] > 500000) {
  echo "Sorry, your file is too large.";
  $uploadOk = 0;
}
```

- Trường nhập tệp trong biểu mẫu HTML của chúng ta ở trên được đặt tên là "fileToUpload".
- Kiểm tra kích thước của tệp. Nếu tệp lớn hơn 500KB, thông báo lỗi sẽ hiển thị và $uploadOk được đặt thành 0

Loại tệp giới hạn

```
// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
&& $imageFileType != "gif" ) {
  echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
  $uploadOk = 0;
}
```

- Mã bên trên chỉ cho phép người dùng tải lên các tệp JPG, JPEG, PNG và GIF. Tất cả các loại tệp khác đưa ra thông báo lỗi trước khi đặt $uploadOk thành 0

File upload hoàn thiện

```php
<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
  $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
  if($check !== false) {
    echo "File is an image - " . $check["mime"] . ".";
    $uploadOk = 1;
  } else {
    echo "File is not an image.";
    $uploadOk = 0;
  }
}

// Check if file already exists
if (file_exists($target_file)) {
  echo "Sorry, file already exists.";
  $uploadOk = 0;
}

// Check file size
```

```php
// Check file size
if ($_FILES["fileToUpload"]["size"] > 500000) {
  echo "Sorry, your file is too large.";
  $uploadOk = 0;
}

// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
&& $imageFileType != "gif" ) {
  echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
  $uploadOk = 0;
}

// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
  echo "Sorry, your file was not uploaded.";
// if everything is ok, try to upload file
} else {
  if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
    echo "The file ". htmlspecialchars( basename( $_FILES["fileToUpload"]["name"])). " has been uploaded.";
  } else {
    echo "Sorry, there was an error uploading your file.";
  }
}
?>
```

Cookies

- Cookie là một tệp nhỏ mà máy chủ nhúng vào máy tính của người dùng. Mỗi khi cùng một máy tính yêu cầu một trang bằng trình duyệt, nó cũng sẽ gửi cookie.
- Cookie thường được sử dụng để xác định người dùng.
- Cookie sẽ không bị mất khi đóng ứng dụng, nó phụ thuộc vào thời gian sống mà bạn thiết lập cho nó.

Tạo cookies

- setcookie(name, value, expire, path, domain, secure, httponly);
- Chỉ có tham số tên là bắt buộc. Tất cả các tham số khác là tùy chọn.

Tạo cookies

```php
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 day
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
  echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
  echo "Cookie '" . $cookie_name . "' is set!<br>";
  echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>
```

- Ví dụ sau tạo cookie có tên “user" với giá trị "John Doe". Cookie sẽ hết hạn sau 30 ngày (86400 * 30). "/"
- Truy xuất giá trị của cookie “user" (sử dụng biến toàn cục $_COOKIE).
- Sử dụng isset() để tìm hiểu xem cookie có được đặt hay không
- Hàmsetcookie()phải xuất hiện TRƯỚC thẻ <html>

Sửa đổi cookies

```php
<?php
$cookie_name = "user";
$cookie_value = "Alex Porter";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
  echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
  echo "Cookie '" . $cookie_name . "' is set!<br>";
  echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>
```

đặt (lại) cookie bằng hàm setcookie()

Xóa cookies

```php
<?php
// set the expiration date to one hour ago
setcookie("user", "", time() - 3600);
?>
<html>
<body>

<?php
echo "Cookie 'user' is deleted.";
?>

</body>
</html>
```

- Để xóa cookie, sử dụng setcookie()chức năng có ngày hết hạn trong quá khứ

Kiểm tra cookies đã bật lên chưa

```php
<?php
setcookie("test_cookie", "test", time() + 3600, '/');
?>
<html>
<body>

<?php
if(count($_COOKIE) > 0) {
  echo "Cookies are enabled.";
} else {
  echo "Cookies are disabled.";
}
?>

</body>
</html>
```

- Ví dụ bên tạo một tập lệnh nhỏ để kiểm tra xem cookie có được bật hay không. Đầu tiên, hãy thử tạo một cookie thử nghiệm với setcookie(), sau đó đếm biến mảng $_COOKIE

Sessions (phiên)

- Phiên là một cách để lưu trữ thông tin (trong các biến) sẽ được sử dụng trên nhiều trang.
- Không giống như cookie, thông tin không được lưu trữ trên máy tính của người dùng.
- Biến phiên lưu trữ thông tin người dùng sẽ được sử dụng trên nhiều trang (ví dụ: tên người dùng.). Theo mặc định, các biến phiên kéo dài cho đến khi người dùng đóng trình duyệt hoặc sau một khoảng thời gian không hoạt động
- Vì thế; Các biến phiên chứa thông tin về một người dùng và có sẵn cho tất cả các trang trong một ứng dụng.

Bắt đầu phiên làm việc

```php
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```

demo_session1.php

- Một phiên được bắt đầu với hàm session_start().
- Các biến phiên được đặt bằng biến toàn cục PHP: $_SESSION.
- Chức năng session_start() phải là điều đầu tiên trong tài liệu của bạn. Trước bất kỳ thẻ HTML nào.

Nhận giá trị biến phiên

```php
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . ".<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>

</body>
</html>
```

- Tiếp theo, chúng ta tạo một trang khác gọi là "demo_session2.php". Từ trang này, chúng ta sẽ truy cập thông tin phiên mà chúng ta đã đặt trên trang đầu tiên ("demo_session1.php").
- Lưu ý rằng các biến phiên không được chuyển riêng lẻ cho từng trang mới, thay vào đó chúng được truy xuất từ phiên chúng ta mở ở đầu mỗi trang (session_start()).
- Cũng lưu ý rằng tất cả các giá trị của biến phiên được lưu trữ trong biến $_SESSION toàn cục

Sửa đổi biến phiên

```php
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// to change a session variable, just overwrite it
$_SESSION["favcolor"] = "yellow";
print_r($_SESSION);
?>

</body>
</html>
```

- Để thay đổi một biến phiên, chỉ cần ghi đè lên nó

Phá hủy phiên PHP

```php
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// remove all session variables
session_unset();

// destroy the session
session_destroy();
?>

</body>
</html>
```

- Để xóa tất cả các biến phiên toàn cầu và hủy phiên, hãy sử dụng session_unset() và session_destroy()

Khi nào dùng session và cookies

Session	Cookies
Xác thực và Quản lý Phiên Làm Việc: thông tin đăng nhập cho phép không cần đăng nhập lại khi chuyển sang trang khác cùng trình duyệt	Ghi nhớ tùy chọn của người dùng: tùy chọn giao diện, ngôn ngữ người dùng.. Nhằm nâng cao trải nghiệm
Dữ liệu tạm thời: kết quả tìm kiếm, giỏ hàng	Tự động đăng nhập: Nếu muốn cung cấp tính năng "Ghi nhớ đăng nhập", có thể sử dụng cookies để lưu trữ một mã thông báo (token) đăng nhập nhưng cần có cơ chế mã hóa và bảo mật.
Các thông báo trạng thái (ví dụ: "Cập nhật thành công") và hiển thị chúng sau khi trang tải lại.	Theo dõi hành vi người dùng: Ghi nhớ và nâng cao trải nghiệm

HỎI

- Nếu yêu cầu là trong 5 phút nếu người dùng không thao tác gì cả với hệ thống thì tự động đăng xuất và yêu cầu người dùng đăng nhập lại thì giải pháp là gì?
- Bài tập: Tạo form cho người dùng đăng nhập vào hệ thống

Bộ lọc

- Xác thực dữ liệu = Xác định xem dữ liệu có ở dạng phù hợp không.
- Dọn dẹp dữ liệu = Xóa mọi ký tự không hợp lệ khỏi dữ liệu
- Tham khảo: https://www.w3schools.com/php/php_filter.asp
- https://www.w3schools.com/php/php_filter_advanced.asp

JSON

- JSON là viết tắt của JavaScript Object Notation, và là một cú pháp để lưu trữ và trao đổi dữ liệu.
- Vì định dạng JSON là định dạng dựa trên văn bản nên nó có thể dễ dàng được gửi đến và đi từ máy chủ và được sử dụng làm định dạng dữ liệu bởi bất kỳ ngôn ngữ lập trình nào
- json_encode()
- json_decode()

PHP - json_encode()

- Hàm json_encode() được sử dụng để mã hóa một giá trị sang định dạng JSON.

```php
<!DOCTYPE html>
<html>
<body>

<?php
$age = array("Peter"=>35, "Ben"=>37, "Joe"=>43);

echo json_encode($age);
?>

</body>
</html>
```

```
{"Peter":35,"Ben":37,"Joe":43}
```

mã hóa một mảng kết hợp thành một đối tượng JSON

```php
<!DOCTYPE html>
<html>
<body>

<?php
$cars = array("Volvo", "BMW", "Toyota");

echo json_encode($cars);
?>

</body>
</html>
```

```
["Volvo","BMW","Toyota"]
```

mã hóa một mảng được lập chỉ mục thành một mảng JSON

PHP - json_decode()

- Hàm json_decode() giải mã một đối tượng JSON thành một đối tượng PHP hoặc một mảng kết hợp

```
<!DOCTYPE html>
<html>
<body>

<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

var_dump(json_decode($jsonobj));
?>

</body>
</html>
```

```
object(stdClass)#1 (3) { ["Peter"]=> int(35) ["Ben"]=> int(37) ["Joe"]=> int(43) }
```

giải mã dữ liệu JSON thành một đối tượng PHP

PHP - Truy cập các giá trị được giải mã

```php
<!DOCTYPE html>
<html>
<body>

<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

$obj = json_decode($jsonobj);

echo $obj->Peter;
echo $obj->Ben;
echo $obj->Joe;
?>

</body>
</html>
```

```
353743
```

cách truy cập các giá trị được giải mã từ một đối tượng

PHP - Truy cập các giá trị được giải mã

```php
<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

$arr = json_decode($jsonobj, true);

echo $arr["Peter"];
echo $arr["Ben"];
echo $arr["Joe"];
?>
```

```
353743
```

truy cập các giá trị từ một mảng kết hợp PHP

PHP - Lặp qua các giá trị

```php
<?php
$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';

$obj = json_decode($jsonobj);

foreach($obj as $key => $value) {
  echo $key . " => " . $value . "<br>";
}
?>
```

Peter => 35
Ben => 37
Joe => 43

Bài tập