



# LẬP TRÌNH PHP

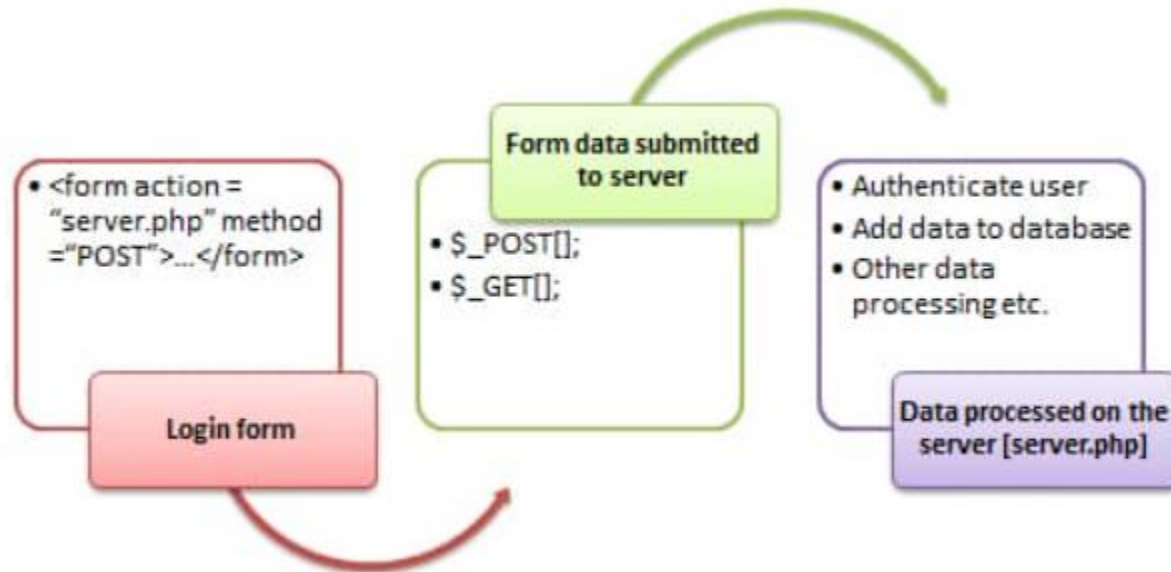
Th.S Nguyễn Đình Hoàng

- Thao tác với form
- Lấy dữ liệu từ Form
- Hiển thị dữ liệu trên PHP

## 1. Form là gì?

- Form được sử dụng để lấy dữ liệu từ phía người dùng và submit lên server để yêu cầu xử lý.

Sơ đồ dưới đây, cho bạn thấy qui trình xử lý đơn giản.



## 2. Tạo một Form để chuẩn bị xử lý dữ liệu

Chúng ta sẽ sử dụng các thẻ HTML để tạo ra một Form. Dưới đây là một thành phần tối thiểu các bạn cần phải có để tạo một Form.

- Form phải có thẻ mở và thẻ đóng `<form></form>`
- Phương thức submit dữ liệu POST hoặc GET
- URL xử lý dữ liệu sẽ submit
- Các trường nhận dữ liệu đầu vào như: Text, Button, Check box, text areas, ...

PHP sử dụng 2 hàm `$_POST` và `$_GET` để nhận dữ liệu từ Client gửi đến Server

## ❖ Phương thức GET trong PHP

Phương thức GET rất dễ nhận thấy đó là trên URL sẽ kèm theo dữ liệu mà chúng ta muốn gửi

- **Client gửi lên**

Phương thức GET là phương thức gửi dữ liệu thông qua đường dẫn URL nằm trên thanh địa chỉ của Browser. Server sẽ nhận đường dẫn đó và phân tích trả về kết quả cho bạn. Server sẽ phân tích tất cả những thông tin đằng sau dấu hỏi (?) chính là phần dữ liệu mà Client gửi lên.

**Ví dụ:** Với URL `http://localhost/demo/welcome.php?name=hello&email=infor@.vn` Server sẽ nhận được giá trị `name = hello & email=infor@.vn`

Để truyền **nhiều dữ liệu** lên Server ta dùng dấu & để phân cách giữa các cặp giá trị. Giả sử tôi muốn truyền `name = hello` và `email = 'infor@.vn'` thì URL sẽ có dạng `http://localhost/demo/welcome.php?name=hello&email=infor@.vn`

## ❖ Phương thức GET trong PHP

Phương thức GET rất dễ nhận thấy đó là trên URL sẽ kèm theo dữ liệu mà chúng ta muốn gửi

### Server nhận dữ liệu

Tất cả các dữ liệu mà Client gửi lên bằng phương thức GET đều được lưu trong một biến toàn cục mà PHP tự tạo ra đó là biến **\$\_GET**, biến này là kiểu **mảng kết hợp** lưu trữ danh sách dữ liệu từ client gửi lên theo quy luật key => value. Ví dụ với URL <http://localhost/demo/welcome.php?name=hello&email=infor@.vn> thì dữ liệu sẽ được lưu trong biến **\$\_GET** dưới dạng:

```
$_GET = array(  
    'name' => 'hello',  
    'email' => 'infor@.vn'  
);
```

## ❖ Phương thức GET trong PHP

Phương thức GET rất dễ nhận thấy đó là trên URL sẽ kèm theo dữ liệu mà chúng ta muốn gửi

### Server nhận dữ liệu

Tất cả các dữ liệu mà Client gửi lên bằng phương thức GET đều được lưu trong một biến toàn cục mà PHP tự tạo ra đó là biến **\$\_GET**, biến này là kiểu **mảng kết hợp** lưu trữ danh sách dữ liệu từ client gửi lên theo quy luật key => value. Ví dụ với URL `http://localhost/demo/welcome.php?name=hello&email=infor@.vn` thì dữ liệu sẽ được lưu trong biến **\$\_GET** dưới dạng:

Vì thế để lấy dữ liệu thì ta chỉ cần làm như sau:

```
// Lấy name
$id = $_GET['name'];
echo $name; // kết quả là hello

// Lấy email
$title = $_GET['email'];
echo $email; // kết quả là infor@.vn
```

## ❖ Phương thức GET trong PHP

### Lưu ý quan trọng:

Trước khi lấy một dữ liệu nào đó phải kiểm tra tồn tại không không mới lấy, vì nếu không kiểm tra thì giả sử người dùng không truyền dữ liệu qua thì sẽ bị báo lỗi ngay. Để kiểm tra ta dùng hàm **isset(\$tenbien)** trong php.

```
if (isset($_GET['name'])){\n    $id = $_GET['name'];\n}
```



## ❖ Phương thức GET trong PHP

Đoạn code dưới đây sẽ tạo một Form đăng ký:

```
<html>
<body>

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

</body>
</html>
```

## ❖ Phương thức GET trong PHP

Tạo file welcome.php với đoạn code sau đây:

```
<html>  
<body>
```

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

```
</body>  
</html>
```



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

## Lấy dữ liệu bằng Phương thức GET

- Đây cũng là một biến mảng Super Global. Lấy dữ liệu bằng phương thức HTTP GET.
- GET có thể được truy cập từ bất kỳ đâu, nó có phạm vi toàn cầu.
- Phương thức GET sẽ hiển thị các giá trị trong các địa chỉ URL
- Phương thức GET thường sử dụng khi dữ liệu không nhạy cảm và cần cho người dùng biết rằng có sự thay đổi trong các hành động.

## ❖ Phương thức POST trong PHP

Phương thức POST có tính bảo mật hơn vì dữ liệu gửi phải thông qua một form HTML nên nó bị ẩn, nghĩa là chúng ta không thể thấy các giá trị đó được.

### Client Gửi Lên

Với phương thức GET thì dữ liệu được thấy trên URL thì phương thức POST thì hoàn toàn ngược lại, POST sẽ gửi dữ liệu qua một cái form HTML và các giá trị sẽ được định nghĩa trong các input gồm các kiểu (*textbox*, *radio*, *checkbox*, *password*, *textarea*, *hidden*) và được nhận dạng thông qua tên (name) của các input đó.

## ❖ Phương thức POST trong PHP

Phương thức POST có tính bảo mật hơn vì dữ liệu gửi phải thông qua một form HTML nên nó bị ẩn, nghĩa là chúng ta không thể thấy các giá trị đó được.

### Server nhận dữ liệu

Tất cả các dữ liệu gửi bằng phương thức POST đều được lưu trong một biến toàn cục **\$\_POST** do PHP tự tạo ra, vì thế để lấy dữ liệu thì bạn chỉ cần lấy trong biến này là được. Cũng như lưu ý với các bạn là trước khi lấy phải dùng hàm `isset($bien)` để kiểm tra có hay không.

```
1 | if (isset($_POST['id'])) {  
2 |     $id = $_POST['id'];  
3 | }
```

## ❖ Ví dụ POST trong PHP

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Post php</title>
8  </head>
9  <body>
10     <form action="" method="post">
11         Username: <input type="text" name="username" value=""> <br>
12         Password: <input type="password" name="pass" value=""> <br>
13         <input type="submit" name="form_submit" value="Gửi dữ liệu">
14         <?php
15             if(isset($_POST['form_submit']))
16             {
17                 echo "<br>Họ và tên:" . $_POST['username'];
18                 echo "<br>";
19                 echo "Mật khẩu:". $_POST['pass'];
20             }
21         ?>
22     </form>
23 </body>
24 </html>
```

## Gửi dữ liệu bằng phương thức POST trong PHP

- Đây là biến mảng supper global được PHP cung cấp sẵn. Sử dụng để lấy dữ liệu submit thông qua phương thức HTTP POST
- Biến mảng này có thể được truy cập từ bất kỳ chỗ nào. Đây là phạm vi global.
- Sử dụng phương thức POST khi bạn không muốn hiển thị các thông tin cho người dùng biết.
- Phương thức POST thường được sử dụng khi có các thông tin nhạy cảm như mật khẩu, số thẻ ngân hàng...

## Phương thức GET và POST khác nhau cụ thể như thế nào?

Phương thức POST	Phương thức GET
Không hiển thị giá trị trên URL	Có hiển thị giá trị trên URL
Không giới hạn độ dài của giá trị được submit	Thông thường giới hạn độ dài là 255 ký tự. Tuy nhiên phụ thuộc vào hỗ trợ của trình duyệt
Mất nhiều thời gian để thực thi hơn là Phương thức GET	Thực thi nhanh hơn phương thức POST
Hỗ trợ nhiều kiểu dữ liệu khác nhau: String, Numeric, Binary, ...	Chỉ hỗ trợ dữ liệu kiểu String bởi vì nó hiển thị chúng trên URL
Không làm lộ dữ liệu	Thấy được dữ liệu trên thanh URL



## Khi nào dùng GET - POST

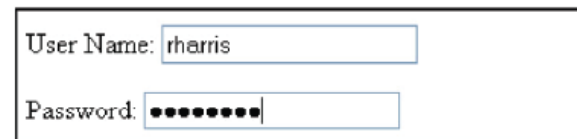
- ❑ Khi dữ liệu bạn muốn SEO thì phải sử dụng phương thức GET.
- ❑ Khi dữ liệu bạn không cần bảo mật thì dùng phương thức GET, ngược lại dữ liệu bảo mật thì dùng phương thức POST.
- ❑ Ví dụ khi đăng nhập, Comment, đăng tin dùng phương thức POST. Còn khi lấy tin ra thì dùng phương thức GET...
- ❑ Khi request sử dụng câu lệnh select thì dùng GET, khi request có sử dụng lệnh insert update, delete thì nên dùng POST.

# NHẬN DỮ LIỆU TỪ TEXTBOX, PASSWORD BOX, HIDDEN FIELD

- Thuộc tính của thẻ HTML input:

Tên	Mô tả
type	Đặt là “text” nếu là trường văn bản, “password” nếu là trường mật khẩu và “hidden” nếu là hidden field. Mặc định là “text”.
name	Tên của trường (text box) khi form được đệ trình lên server.
value	Giá trị của trường. Đối với hidden field, thuộc tính này phải được thiết lập bằng mã thì mới hoạt động đúng.

- Textbox cho phép người dùng nhập dữ liệu vào ô
- Password box cho phép người dùng nhập dữ liệu vào ô nhưng dữ liệu được mã hóa thành các ký tự đặc biệt
- Hidden field cho phép lập trình viên bổ sung thêm cặp tên/giá trị vào form.
- Người dùng không thấy được hidden field khi vào trang web
- Nếu form sử dụng phương thức GET, dữ liệu cho trường được hiển thị trong URL



A screenshot of a web form with two input fields. The first field is labeled "User Name:" and contains the text "rharris". The second field is labeled "Password:" and contains a series of dots, indicating that the password is masked.

## NHẬN DỮ LIỆU TỪ TEXTBOX, PASSWORD BOX, HIDDEN FIELD

---

### □ Ví dụ: Mã HTML cho ba kiểu dữ liệu:

```
<input type="text" name="user_name" value="rharris">  
<input type="password" name="password">  
<input type="hidden" name="action" value="login">
```

```
//URL khi sử dụng phương thức GET  
process_data.php?user_name=rharrises&password=s3sc3t72
```

```
//Mã PHP cho phương thức GET  
$user_name = $_GET['user_name'];  
$password = $_GET['password'];  
$action = $_GET['action'];
```

```
//URL khi sử dụng phương thức POST  
process_data.php
```

```
//Mã PHP cho phương thức POST  
$user_name = $_POST['user_name'];  
$password = $_POST['password'];  
$action = $_POST['action'];
```

## LẤY DỮ LIỆU TỪ RADIO BUTTON

---

- ❑ Radio button cho phép người dùng chọn một tùy chọn từ một nhóm các tùy chọn cho trước
- ❑ Thuộc tính thẻ HTML `<input>` cho radio button:

Thuộc tính	Mô tả
<code>type</code>	Đặt là “radio” nếu muốn hiển thị radio button.
<code>name</code>	Tên được dùng chung cho tất cả các radio button trong nhóm.
<code>value</code>	Giá trị của radio button. Thuộc tính này phải có thì radio button mới làm việc đúng và dữ liệu đẩy lên mới có giá trị.
<code>checked</code>	Thuộc tính tùy chọn được dùng để xác định radio button mặc định.

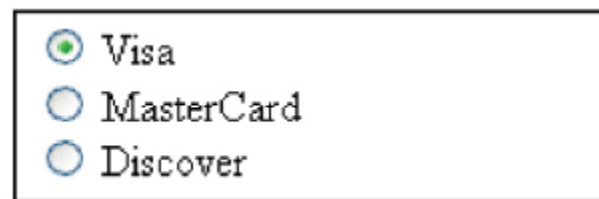
- ❑ Nếu không có nút nào trong nhóm radio button được chọn, cặp tên/giá trị của nhóm sẽ không được đẩy lên server như một phần dữ liệu của form.
- ❑ Nếu mã PHP đặt thuộc tính `checked` cho nhiều hơn một nút trong nhóm, trình duyệt sẽ chọn nút cuối cùng khi trang được dựng.

## LẤY DỮ LIỆU TỪ RADIO BUTTON

---

### Ví dụ:

```
//Mã HTML của ba radio button trong một nhóm
<input type="radio" name="card_type" value="visa" checked="checked">Visa
</br>
<input type="radio" name="card_type" value="mastercard">Mastercard
</br>
<input type="radio" name="card_type" value="discover">Discover
```



```
//Mã PHP lấy dữ liệu của nhóm radio button có giá trị mặc định
$card_type=$_POST['card_type'] ;

//Mã PHP lấy dữ liệu nhóm radio button không có giá trị mặc định
if (isset($_POST['card_type'])) {
    $card_type = $_POST['card_type'];
}
else {
    $card_type = 'unknown';
}
```

## LẤY DỮ LIỆU TỪ CHECKBOX

---

- ❑ Check box cho phép người dùng chọn một tùy chọn
- ❑ Thuộc tính của thẻ HTML `<input>` cho check box:

Thuộc tính	Mô tả
<code>type</code>	Thiết lập “checkbox” nếu cần hiển thị check box.
<code>name</code>	Mỗi check box sẽ được đặt tên khác nhau. Tuy vậy, nếu muốn định nghĩa một mảng check box thì bạn có thể đặt cùng tên như minh họa trong hình tiếp theo.
<code>value</code>	Giá trị khi check box được chọn. Thuộc tính này chỉ có ý nghĩa khi bạn làm việc với mảng check box như minh họa ở hình tiếp theo.
<code>checked</code>	Thuộc tính tùy chọn cho check box được dùng để xác định hộp mặc định có được chọn không.

- ❑ Hàm `isset` trả về `TRUE` nếu check box được chọn và `FALSE` nếu không được chọn

## LẤY DỮ LIỆU TỪ CHECKBOX

---

### Ví dụ:

//Mã HTML của 3 check box

```
<input type="checkbox" name="pep" checked="checked"> Pepperonis<br>
<input type="checkbox" name="msh"> Mushrooms<br>
<input type="checkbox" name="olv"> Olives
```

### Ba check box hiển thị trên trình duyệt

☒ Pepperoni  
☐ Mushrooms  
☐ Olives

//Mã PHP lấy dữ liệu của check box

```
$pepperonis = isset($_POST['pep']);
$mushrooms = isset($_POST['msh']);
$olives = isset($_POST['olv']);
```

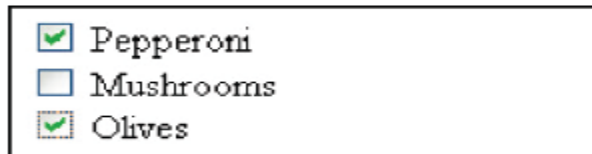
## LẤY DỮ LIỆU TỪ MẢNG CHECKBOX

---

- ❑ Nếu tên check box kết thúc bằng ký tự mảng [], PHP sẽ thêm check box vào mảng và mảng này được lồng vào mảng toàn cục \$\_GET hoặc \$\_POST => Cho phép nhiều giá trị được gửi lên server với cùng một tên
- ❑ Nếu không có check box nào trong mảng được chọn thì tên mảng sẽ không có trong mảng \$\_GET hay \$\_POST
- ❑ Ví dụ:

```
//Mã HTML lưu các check box liên quan vào mảng  
<input type="checkbox" name="top[]" value="pep"> Pepperonis <br>  
<input type="checkbox" name="top[]" value="msh"> Mushrooms <br>  
<input type="checkbox" name="top[]" value="olv"> Olives
```

### Check box hiển thị trên trình duyệt



A screenshot of a web browser displaying a form with three checkboxes. The first checkbox, labeled 'Pepperoni', is checked. The second checkbox, labeled 'Mushrooms', is also checked. The third checkbox, labeled 'Olives', is checked. The checkboxes are arranged vertically within a rectangular box.



## LẤY DỮ LIỆU TỪ MẢNG CHECKBOX

---

```
//PHP lấy dữ liệu từ mảng check box và giá trị của nó
$toppings = $_POST['top']; //lấy mảng lớp phủ cho pizza
$top1 = $toppings[0]; // $top1 = pep
$top2 = $toppings[1]; // $top2 = olv
$top3 = $toppings[2]; // $top3 không được thiết lập

//PHP sử dụng vòng lặp foreach để xử lý mảng
if (isset($_POST['top'])) {
    $toppings = $_POST['top'];
    foreach ($toppings as $key => $value) {
        echo $key. ' = ' . $value. '<br />';
    }
}
else { echo 'No topping selected'; }
```

## LẤY DỮ LIỆU TỪ DROPDOWN LIST

---

- ❑ Dropdown list (danh sách thả xuống) cho phép người dùng chọn một tùy chọn trong nhóm các tùy chọn
- ❑ Nếu thuộc tính selected không được viết thì tùy chọn đầu tiên trong danh sách sẽ mặc định được chọn
- ❑ Thuộc tính thẻ HTML <select> của dropdown list:

Thuộc tính	Mô tả
name	Tên của danh sách thả xuống.

- ❑ Thuộc tính thẻ HTML <option>:

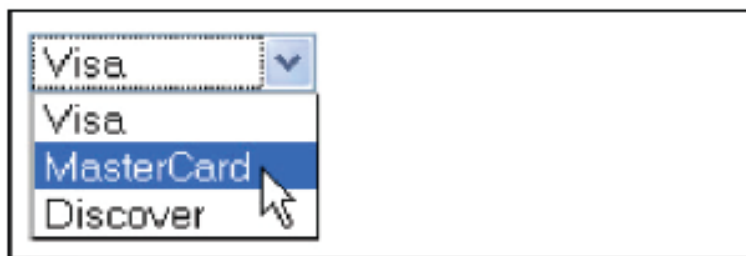
Thuộc tính	Mô tả
value	Giá trị của tùy chọn.
selected	Thuộc tính tùy chọn để đặt một tùy chọn là mặc định.

## LẤY DỮ LIỆU TỪ DROPDOWN LIST

---

Ví dụ:

```
//Mã HTML của dropdown list
<select name="card_type">
    <option value="visa">Visa</option>
    <option value="mastercard">Mastercard</option>
    <option value="discover">Discover</option>
</select>
```



```
//Mã PHP lấy dữ liệu từ dropdown list
$card_type = $_POST['card_type'];
```

## LẤY DỮ LIỆU TỪ LISTBOX

---

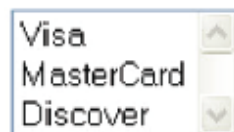
- ❑ Listbox (hộp danh sách) cho phép người dùng không chọn hoặc chọn nhiều từ danh sách tùy chọn
- ❑ Nếu danh sách cho phép đa chọn, tên của danh sách phải kết thúc bằng ký tự mảng []
- ❑ Thuộc tính thẻ HTML <option>:

Thuộc tính	Mô tả
size	Khi được thiết lập giá trị là 1 hoặc bị bỏ qua, ta có danh sách thả xuống hiển thị các tùy chọn. Nhưng khi đặt giá trị là 2 hoặc lớn hơn, ta có hộp danh sách hiển thị các tùy chọn theo số lượng đã thiết lập.
multiple	Thuộc tính tùy chọn cho phép người dùng cùng lúc chọn nhiều tùy chọn trong hộp danh sách. Bạn nên sử dụng thuộc tính này khi size được thiết lập giá trị bằng 2 hoặc hơn.

## LẤY DỮ LIỆU TỪ LISTBOX

### Ví dụ:

```
//Mã HTML của hộp danh sách không cho phép đa chọn
<select name="card_type" size="3">
    <option value="visa">Visa</option>
    <option value="mastercard">Mastercard</option>
    <option value="discover">Discover</option>
</select>
```



```
//Mã HTML của hộp danh sách cho phép đa chọn
<select name="top[]" size="3" multiple="multiple">
    <option value="pep" selected="selected">Pepperoni</option>
    <option value="msh">Mushrooms</option>
    <option value="olv">Olives</option>
</select>
```



```
//Mã PHP lấy dữ liệu từ hộp danh sách đa chọn
if (isset($_POST['top'])) {
    $stoppings = $_POST['top'];
    foreach ($stoppings as $key => $value) {
        echo $key. ' = ' . $value. '<br />'; //'0 = pep' và '1 = msh'
    }
}
else { echo 'No topping selected';}
```

## LẤY DỮ LIỆU TỪ TEXTAREA

---

- ❑ Text area (vùng văn bản) cho phép người dùng nhập nhiều dòng văn bản
- ❑ Để thiết lập văn bản mặc định cho vùng, viết văn bản này giữa cặp thẻ `<textarea>`

- ❑ Thuộc tính của thẻ HTML `<textarea>`:

Thuộc tính	Mô tả
Name	Tên của vùng văn bản.
rows	Số hàng xấp xỉ trong vùng văn bản.
cols	Số ký tự xấp xỉ trên mỗi dòng.

- ❑ Nếu người dùng nhập liệu đến cuối dòng, vùng văn bản sử dụng xuống dòng mềm để bắt đầu dòng mới
- ❑ Nếu người dùng nhấn phím Enter hoặc Return, vùng văn bản sử dụng xuống dòng cứng để bắt đầu dòng
- ❑ Nếu người dùng không nhập ký tự nào, cặp tên/giá trị của vùng văn bản vẫn được gửi lên với giá trị được gán là chuỗi rỗng

## LẤY DỮ LIỆU TỪ TEXTAREA

---

### Ví dụ:

```
//Mã HTML của vùng văn bản
<textarea name="comment" rows="4" cols="50">Welcome to PHP and MySQL!
</textarea>
```

A screenshot of a web browser showing a text area with the text "Welcome to PHP and MySQL!". The text area is a simple rectangular box with a thin border, and the text is displayed in a monospaced font.

```
//URL khi sử dụng phương thức GET
//Khi người dùng sử dụng dấu cách trong vùng văn bản
process_data.php?comment=Welcome+to+PHP+and+MySQL!
//Khi người dùng nhấn phím Enter hoặc Return để thêm dòng mới
process_data.php?comment=Welcome+to%0D%0A+PHP+and+MySQL!
//Khi người dùng không nhập gì
process_data.php?comment=
```

```
//Mã PHP lấy dữ liệu từ vùng văn bản
$comment=$ POST['comment'];
```

# PHP FORM VALIDATION

---

Ngoài validate form bằng Javascript ra thì chúng ta nên sử dụng PHP để validate nữa bởi chỉ validate bằng Javascript thôi thì chưa đủ độ bảo mật. Nếu người dùng cố tình tắt Javascript thì họ hoàn toàn có thể lách qua được lớp bảo vệ đó của website, nên trong bài viết này sẽ hướng dẫn các bạn cách validate form bằng PHP trước khi lưu trữ vào cơ sở dữ liệu.

## 1. Các hàm validate trong PHP

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



# PHP FORM VALIDATION

---

## 1. Các hàm validate trong PHP

```
// validate mail
```

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

```
// Validate URL
```

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

# PHP FORM VALIDATION

---

## 2. Validate form bằng PHP

Để validate form bằng PHP thì chúng ta phải thông qua hai công đoạn, **công đoạn thứ nhất** là lấy dữ liệu và kiểm tra dữ liệu, **công đoạn thứ hai** là hiển thị thông báo lỗi và hiển thị dữ liệu mà người dùng đã nhập để họ khỏi mất công nhập lại.

### Công đoạn 1: Kiểm tra dữ liệu

Ví dụ về code kiểm tra dữ liệu đầu vào PHP validate trong thẻ `<?php ?>` sau đây:

# PHP FORM VALIDATION

---

```
<?
// define variables and set to empty values
$nameErr = $emailErr = $websiteErr = "";
$name = $email = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
        // check if name only contains letters and whitespace
        if (!preg_match("/^[a-zA-Z-' ]*$/", $name)) {
            $nameErr = "Only letters and white space allowed";
        }
    }
}

if (empty($_POST["email"])) {
    $emailErr = "Email is required";
} else {
    $email = test_input($_POST["email"]);
    // check if e-mail address is well-formed
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $emailErr = "Invalid email format";
    }
}
}
}?>
```

## PHP FORM VALIDATION

---

```
if (empty($_POST["website"])) {
    $website = "";
} else {
    $website = test_input($_POST["website"]);
    // check if URL address syntax is valid (this regular
expression also allows dashes in the URL)
    if (!preg_match("/\b(?:(:https?|ftp):\/\/|www\.|
    [-a-z0-9+&@#\/%?=_|!:,.;]*[-a-z0-
9+&@#\/%=~_|])/i",$website)) {
        $websiteErr = "Invalid URL";
    }
}
}
?>
```

# PHP FORM VALIDATION

---

## Công đoạn 2: Hiện thị lỗi và dữ liệu

Chúng ta có hai biến cục bộ đó là biến \$name, \$email, \$website và \$nameErr, \$emailErr, \$websiteErr dựa vào các biến này chúng ta show thông tin ra như sau:

```
<form method="post" action="">
  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>
  <input type="submit" name="submit" value="Submit">
</form>
```

**KẾT THÚC**