

LẬP TRÌNH PHP (PHẦN 3) LÀM VIỆC VỚI DỮ LIỆU CỦA **FORM**



Hệ thống bài cũ

- Làm việc với chuỗi và số
 - Tạo ra chuỗi
 - Thêm các ký tự đặc biệt vào chuỗi
 - Làm việc với độ dài chuỗi và chuỗi con
 - Tìm kiếm trong chuỗi
 - Thay thế một bộ phận trong chuỗi
 - Sửa chuỗi
 - Chuyển đổi giữa chuỗi và mảng
- Làm việc với ngày tháng
 - Sử dụng nhãn thời gian
 - Sử dụng đối tượng
- Khởi tạo và sử dụng mảng
 - Khởi tạo và sử dụng mảng & mảng liên kết
 - Làm việc với hàng đợi và ngăn xếp
 - Làm việc với mảng hai chiều

Nội dung bài học

- Lấy dữ liệu từ form
- Hiển thị dữ liệu trên trang web

1. Lấy dữ liệu từ form

Trong phần này có các nội dung:

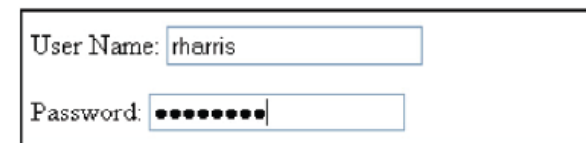
- 1.1. Nhận dữ liệu từ textbox, hộp password, hidden field
- 1.2. Lấy dữ liệu từ radio button
- 1.3. Lấy dữ liệu từ checkbox
- 1.4. Lấy dữ liệu từ mảng checkbox
- 1.5. Lấy dữ liệu từ dropdown list
- 1.6. Lấy dữ liệu từ list
- 1.7. Lấy dữ liệu từ text area

1.1. 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'];
```

1.2. 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
type	Đặt là “radio” nếu muốn hiển thị radio button.
name	Tên được dùng chung cho tất cả các radio button trong nhóm.
value	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.
checked	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
```

☒ Visa
☐ MasterCard
☐ 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';  
}
```


1.3. 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
type	Thiết lập “checkbox” nếu cần hiển thị check box.
name	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.
value	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.
checked	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="pep"> 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']);
```

1.4. 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

☒ Pepperoni
☐ Mushrooms
☒ Olives

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'; }
```

1.5. 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.

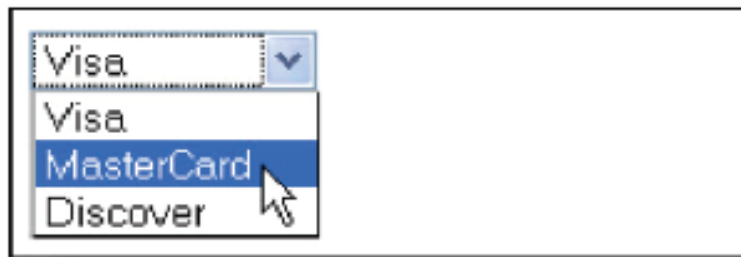
- Thuộc tính thẻ HTML <option>:

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'];
```

1.6. 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'])) {
    $toppings = $_POST['top'];
    foreach ($toppings as $key => $value) {
        echo $key. ' = ' . $value. '<br />'; //'0 = pep' và '1 = msh'
    }
}
else { echo 'No topping selected';}
```


1.7. Lấy dữ liệu từ text area

- 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 text area

Ví dụ:

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



```
//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'];
```

2. Hiển thị dữ liệu trên trang web

Trong phần này có các nội dung:

2.1. Định dạng các ký tự đặc biệt

2.2. Định dạng ngắt dòng

2.3. Hiển thị dữ liệu với câu lệnh echo và print

2.1. Định dạng các ký tự đặc biệt

- Thực thể ký tự HTML (HTML character entity) cho phép bạn hiển thị các ký tự đặc biệt trên trang web
- Hàm htmlspecialchars chuyển đổi các ký tự đặc biệt thành thực thể ký tự
- Cú pháp của hàm htmlspecialchars:

```
htmlspecialchars($string [, $quote_style[, $charset[, $double_encode]]]);
```

- Tham số của hàm htmlspecialchars:

Thuộc tính	Mô tả
\$string	Chuỗi cần chuyển đổi. Tham số này là bắt buộc.
\$quote_style	Chỉ định cách chuyển nháy đơn và nháy kép sang thực thể HTML. Hằng mặc định ENT_COMPAT chỉ chuyển dấu nháy kép. Hằng ENT_QUOTES chuyển cả dấu nháy đơn lẫn nháy kép. Hằng ENT_NOQUOTES sẽ không chuyển bất kỳ dấu nháy đơn hay nháy kép nào.
\$charset	Chỉ định tập ký tự (character set) của tham số \$string. Mặc định là mã “ISO-8859-1”. Muốn chuyển sang Unicode thì dùng mã “UTF8”.
\$double_encode	Giá trị Boolean xác định có mã đúp thực thể ký tự không. Mặc định là TRUE, vì vậy các thực thể ký tự sẽ luôn được mã đúp.

Định dạng các ký tự đặc biệt

- Một số thực thể HTML cơ bản:

Ký tự thực	Thực thể ký tự	Ký tự thực	Thực thể ký tự
&	&	"	"
<	<	'	'
>	>	Dấu cách không ngắt dòng	

- Ví dụ:
 - Văn bản do người dùng nhập:

```
Welcome to <i>PHP</i> and MySQL!
```

- Mã PHP chuyển đổi ký tự đặc biệt sang thực thể ký tự:

```
comment = $_POST['comment'];  
comment = htmlspecialchars($comment, ENT_COMPAT, 'ISO-8859-1', false);  
cho $comment;
```

- Dữ liệu được hiển thị trên trình duyệt:

```
Welcome to <i>PHP</i> and MySQL!
```

2.2. Định dạng ngắt dòng

- Hàm nl2br chuyển đổi ký tự dòng mới trong chuỗi thành thẻ HTML
. Hàm này cho phép hiển thị ngắt dòng HTML trên trang web
- Cú pháp của hàm nl2br: `nl2br($string[, $is_xhtml]);`
- Tham số của hàm nl2br:

Thuộc tính	Mô tả
\$string	Chuỗi cần chuyển đổi. Tham số này là bắt buộc.
\$is_xhtml	Giá trị boolean được dùng để xác định nên sử dụng cú pháp XHTML () hay cú pháp HTML () để ngắt dòng. Giá trị mặc định là TRUE.

Định dạng ngắt dòng

- Ví dụ:
 - Văn bản được nhập vào vùng văn bản:

```
Welcome to  
PHP and MySQL!
```

- Mã PHP chuyển đổi ký tự ngắt dòng thành thẻ ngắt dòng HTML:

```
$comment = $_POST['comment'];  
$comment = nl2br($comment, false); //sử dụng thẻ <br> thay vì thẻ <br />  
echo $comment;
```

- Dữ liệu hiển thị trên trình duyệt:

```
Welcome to  
PHP and MySQL!
```

2.3. Hiển thị dữ liệu với câu lệnh echo và print

- Lệnh echo và print gửi dữ liệu kiểu chuỗi ra trang web. Các ký tự không phải chuỗi sẽ được chuyển thành chuỗi
- Lệnh echo có thể chấp nhận nhiều giá trị kiểu chuỗi, trong khi lệnh print chỉ chấp nhận một giá trị
- Ngoặc đơn là ký tự tùy chọn cho cả lệnh echo và print. Để in ra nhiều giá trị với lệnh echo thì bỏ ngoặc đơn đi
- Lệnh echo không trả về giá trị và không thể là thành phần cấu tạo nên biểu thức. Lệnh print trả về 1 và có thể dùng như một thành phần của biểu thức
- Lệnh echo và print không phải là hàm mà là bộ phận của tập định nghĩa ngôn ngữ PHP

Hiển thị dữ liệu với câu lệnh echo và print

- Cú pháp lệnh echo:

```
echo $var1  
echo ($var1)  
echo $var1 [, $var2 ...]
```

- Ví dụ:

```
echo 'Welcome to PHP and MySQL!';  
echo 'Name: ' . $name;  
echo ('Name: ' . $name);  
echo 'Cost: $', $cost;
```

- Cú pháp lệnh print:

```
print $var1  
print ($var1)
```

- Ví dụ:

```
print 'Welcome to PHP and MySQL!';  
print 'Name: ' . $name;  
print ('Name: ' . $name);  
//Sử dụng lệnh print trong biểu thức  
$age >= 18)? print ('Can vote.'): print ('Cannot vote.');
```

Tổng kết bài học

- Dữ liệu nhập vào các điều khiển trên form được gửi lên server dưới dạng chuỗi trong một mảng các cặp tên/giá trị
- Text box được dùng để lấy các dữ liệu đơn giản
- Password box lấy các dữ liệu nhạy cảm và được hiển thị bằng các ký tự đặc biệt
- Hidden field được dùng để truyền những dữ liệu không do người dùng nhập lên máy chủ
- Radio button cho phép người dùng chọn một trong số các tùy chọn, ngược lại check box cho phép chọn nhiều tùy chọn khác nhau.
- Để lấy dữ liệu từ nhiều check box liên quan thì thiết lập sao cho dữ liệu được trả về trong mảng

Tổng kết bài học

- Dropdown list cho phép người dùng chọn một tùy chọn từ nhóm tùy chọn. Trong khi đó, list box cho phép chọn một hoặc nhiều tùy chọn
- Text area cho phép người dùng nhập văn bản trên nhiều dòng. Văn bản có thể chứa xuống dòng mềm và xuống dòng cứng
- Sử dụng hàm htmlspecialchars để kiểm soát cách hiển thị các ký tự đặc biệt do người dùng nhập trên trang web.
- Dùng hàm nl2br để chuyển ký tự xuống dòng theo định dạng HTML
- Sử dụng lệnh PHP echo hoặc print để gửi dữ liệu ra trình duyệt

XIN CẢM ƠN!