

Chapter 7

How to work with form data

Objectives

Applied

1. Use text boxes, password boxes, radio buttons, check boxes, drop-down lists, list boxes, and text areas to get input from the user.
2. Use hidden fields to pass data to the web application when a form is submitted.
3. Use the `htmlspecialchars` and `nl2br` functions to display user entries the way you want them displayed.
4. Use echo statements to display data in a web page.

Objectives (continued)

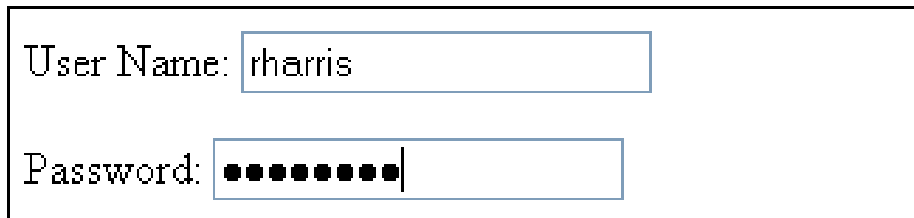
Knowledge

1. Describe the way a PHP application gets data from text boxes, password boxes, hidden fields, radio buttons, check boxes, drop-down lists, list boxes, and text areas.
2. Describe the use of the htmlspecialchars and nl2br functions.
3. Describe the use of the echo and print statements.

Text input: The HTML for three types of fields

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

The text and password fields in the browser



User Name:

Password:

Hidden fields:

- not displayed on the web page
- value attribute must be set
- not secure since the value can be viewed in the html source code

The URL when using the GET method

`process_data.php?user_name=rharris&password=s3cr3t72&action=login`

The PHP for the GET method

```
<?php
    $user_name = $_GET['user_name'];
    $password = $_GET['password'];
    $action = $_GET['action'];
?>
```

The URL when using the POST method

`process_data.php`

The PHP for the POST method

```
<?php
    $user_name = $_POST['user_name'];
    $password = $_POST['password'];
    $action = $_POST['action'];
?>
```

The HTML for three radio buttons in a group

```
<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
```

The radio buttons in the browser



☒ Visa
☐ MasterCard
☐ Discover

Note:

- the name attribute for each radio button must be the same
- the value attribute for each radio button must be different
- the checked attribute sets the default so should only be used on one of the options

PHP to access a radio button group with a default button

```
<?php
    $card_type = $_POST['card_type'];
?>
```

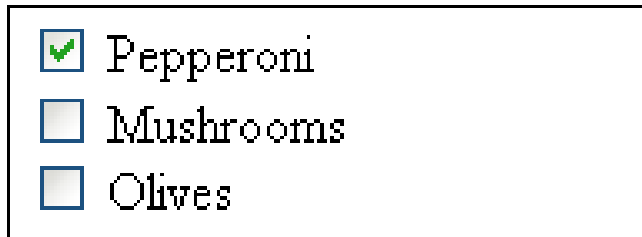
PHP to access a radio button group that doesn't have a default button

```
<?php
    if (isset($_POST['card_type'])) {
        $card_type = $_POST['card_type'];
    } else {
        $card_type = "unknown";
    }
?>
```

The HTML for three check boxes

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

The check boxes in the browser



☒ Pepperoni
☐ Mushrooms
☐ Olives

The PHP to access the check box data

```
<?php
    $pepperoni = isset($_POST['pep']);
    $mushrooms = isset($_POST['msh']);
    $olives = isset($_POST['olv']);
?>
```

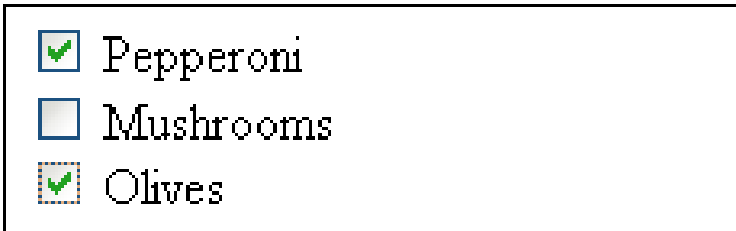

Using an array of check boxes

- If the items in a checkbox list are related, use an array to simplify the processing rather than creating separate identifiers for each one.

Three related check boxes in an array

```
<input type="checkbox" name="top[]" value="pep"/>  
    Pepperoni<br />  
<input type="checkbox" name="top[]" value="msh"/>  
    Mushrooms<br />  
<input type="checkbox" name="top[]" value="olv"/> Olives
```

The check boxes in the browser



A screenshot of a web browser window showing a list of three items, each with a checked checkbox and a label: "Pepperoni", "Mushrooms", and "Olives". The checkboxes are small squares with a green checkmark inside. The labels are in a standard sans-serif font.

PHP that accesses the array and its values

```
$toppings = $_POST['top']; // get the toppings array  
$top1 = $toppings[0];      // $top1 is pep  
$top2 = $toppings[1];      // $top2 is olv  
$top3 = $toppings[2];      // $top3 is not set
```

PHP that uses a loop to process the array

```
<?php
    if (isset($_POST['top'])) {
        $toppings = $_POST['top'];
        foreach($toppings as $key => $value) {
            echo $key. ' = ' . $value . '<br />';
        }
    } else {
        echo 'No toppings selected.';
    }
?>
```

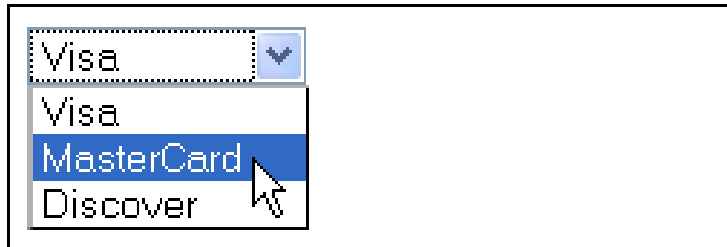
The message displayed by the browser

```
0 = pep
1 = olv
```

The HTML for a drop-down list

```
<select name="card_type">
  <option value="visa">Visa</option>
  <option value="mastercard">MasterCard</option>
  <option value="discover">Discover</option>
</select>
```

The drop-down list in a browser



The PHP to access the drop-down list data

```
<?php
    $card_type = $_POST['card_type'];
?>
```

A list box that doesn't allow multiple options

```
<select name="card_type" size="3">  
  <option value="visa">Visa</option>  
  <option value="mastercard">MasterCard</option>  
  <option value="discover">Discover</option>  
</select>
```

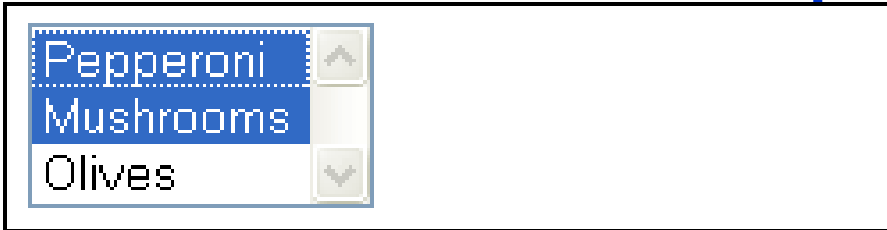
A list box that doesn't allow multiple options



A list box that allows multiple options

```
<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>
```

A list box that allows multiple options



PHP for a list box that allows multiple options

```
<?php
    if (isset($_POST['top'])) {
        $toppings = $_POST['top'];
        foreach ($toppings as $key => $value) {
            echo $key. ' = ' . $value . '<br />';
        }
    } else {
        echo 'No toppings selected.';
    }
?>
```

The HTML for a text area

```
<textarea name="comment" rows="4" cols="50">  
Welcome to PHP and MySQL!</textarea>
```

A text area in the browser



Welcome to PHP and MySQL!

If the user types past the end of the text area, a *soft return* is created

If the user presses Enter or Return, a *hard return* is created.

The URL when using the GET method

When the user includes spaces in the text area

```
process_data.php?comment=Welc+ome+to+PHP+and+MySQL!
```

When the user presses the Enter or Return key to start a new line

```
process_data.php?comment=Welc+ome+to%0D%0APHP+and+MySQL!
```

When the user doesn't enter any text

```
process_data.php?comment=
```

The PHP to get the data from the text area

```
<?php
    $comment = $_POST['comment'];
?>
```

`htmlspecialchars()` converts certain characters into their character entity so that they will show in the browser and not be interpreted.

Common HTML character entities

Character	Character entity
&	&
<	<
>	>
"	"
'	'
Non-breaking space	

Syntax of the htmlspecialchars function

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

The available quote styles are:

- ENT_COMPAT - Default. Encodes only double quotes
- ENT_QUOTES - Encodes double and single quotes
- ENT_NOQUOTES - Does not encode any quotes

A double-encoded less-than entity

`<`

The text entered by the user



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

Converting special characters to entities

```
<?php
    $comment = $_POST['comment'];
    $comment = htmlspecialchars($comment, ENT_COMPAT,
                                'ISO-8859-1', false);
?>
<p><?php echo $comment; ?></p>
```

The data displayed in the browser

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

Syntax of the nl2br function

```
nl2br($string[, $is_html])
```

The text entered into the text area

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

Converting line break characters to line break tags

```
<?php  
    $comment = $_POST['comment'];  
    $comment = nl2br($comment, false);  
?>  
<p><?php echo $comment; ?></p>
```

The data displayed in the browser

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

The echo statement

Syntax

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

Examples

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

The print statement

Syntax

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

Examples

```
print 'Welcome to PHP and MySQL!';  
print 'Name: ' . $name;  
print('Name: ' . $name);
```

Using print in an expression

```
<?php  
    ($age >= 18) ? print('Can vote.') :  
                  print('Cannot vote.');
```

?>