

PHP to access a radio button group

```
<?php
$card_type = filter_input(INPUT_POST, 'card_type');
?>
```

PHP to add a default value for a group with no default button

```
<?php
$card_type = filter_input(INPUT_POST, 'card_type');
if ($card_type == NULL) {
    $card_type = 'unknown';
}
?>
```

MURACH BOOKS
Murach's PHP and MySQL (3rd Ed.)
C7, Slide 7

PHP that accesses the array and its values

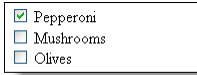
```
<?php
$toppings = filter_input(INPUT_POST, 'top',
    FILTER_SANITIZE_SPECIAL_CHARS, FILTER_REQUIRE_ARRAY);
if ($toppings != NULL) {
    $top1 = $toppings[0]; // $top1 is pep
    $top2 = $toppings[1]; // $top2 is olv
    $top3 = $toppings[2]; // $top3 is not set - error
}
?>
```

MURACH BOOKS
Murach's PHP and MySQL (3rd Ed.)
C7, Slide 10

The HTML for three check boxes

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

The check boxes in the browser



The PHP to access the check box data


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

MURACH BOOKS
Murach's PHP and MySQL (3rd Ed.)
C7, Slide 8

PHP that uses a loop to process the array

```
<?php
$toppings = filter_input(INPUT_POST, 'top',
    FILTER_SANITIZE_SPECIAL_CHARS, FILTER_REQUIRE_ARRAY);
if ($toppings != NULL) {
    foreach($toppings as $key => $value) {
        echo $key. ' = ' . $value . '<br>';
    }
} else {
    echo 'No toppings selected.';
}
?>
```

The message displayed by the browser

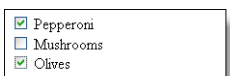


MURACH BOOKS
Murach's PHP and MySQL (3rd Ed.)
C7, Slide 11

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




MURACH BOOKS
Murach's PHP and MySQL (3rd Ed.)
C7, Slide 9

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 HTML to set a default option

```
<option value="mastercard" selected>MasterCard</option>
```

The PHP to access the drop-down list data

```
<?php
$card_type = filter_input(INPUT_POST, 'card_type');
?>
```

MURACH BOOKS
Murach's PHP and MySQL (3rd Ed.)
C7, Slide 12

A list box that doesn't allow multiple selections

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

The list box in a browser



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



A list box that allows multiple selections

```
<select name="top[]" size="3" multiple>
  <option value="pep" selected>Pepperoni</option>
  <option value="msh">Mushrooms</option>
  <option value="olv">Olives</option>
</select>
```

The list box in a browser



The URL when using the GET method

When the user includes spaces in the text area

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

When the user presses the Enter or Return key

```
process_data.php?comment=Welcome+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 = filter_input(INPUT_POST, 'comment');
?>
```

PHP for a list box that allows multiple selections

```
<?php
  $toppings = filter_input(INPUT_POST, 'top',
    FILTER_SANITIZE_SPECIAL_CHARS, FILTER_REQUIRE_ARRAY);
  if ($toppings !== NULL) {
    foreach ($toppings as $key => $value) {
      echo $key . ' = ' . $value . '<br>'; // 0 = pep, 1 = msh
    }
  } else {
    echo 'No toppings selected.';
  }
?>
```

Syntax of the htmlspecialchars() function

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

Common HTML character entities

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

An example that uses special characters

The text entered by the user

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

PHP that converts special characters to character entities

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

The data displayed in the browser

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

The text entered into the text area

Welcome to
PHP and MySQL!

Converting line break characters to line break tags

```
<?php
    $comment = filter_input(INPUT_POST, 'comment');
    $comment = nl2br($comment, false); // use <br> tags,
    not <br /> tags
?>
<p><?php echo $comment; ?></p>
```

The data displayed in the browser

Welcome to
PHP and MySQL!

A double-encoded less than character entity (<)

```
&amp;lt;
```

A statement that prevents double encoding

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

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;
```

Syntax of the nl2br() function

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

The print statement

Syntax

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

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.');
```