Chapter 3 – Web Forms and PHP

ICT24A Web 2 Programming

Introduction

in this chapter we are going to investigate the way in which we can use PHP in association with Web forms to allow interaction between users of the website and PHP scripts.

You will learn how to create an XHTML form and learn how to combine PHP scripts and XHTML forms and why it is good practice. You will learn how the PHP form can invoke itself and come to an understanding of how the data provided by the form elements can be accessed inside a PHP script.

We will consider the difference between the POST and GET methods and will develop a number of scripts that provide data validation for the form.

As you work through this document creates and runs scripts that are displayed in blue text and save them for later submission.

Chapter 3 – Web Forms and PHP

ICT24A Web 2 Programming

We considered the diagram below at the commencement of the subject and it worth reviewing it once again. It gives an overview of how a PHP document is processed in a web-based application.

The system described below includes a web user on hardware that is running a browser and a server which is based at another location which is running the Web server software.

(1) The user requests the document that includes PHP statements.
(2) The Web server software determines that the web page to be processed include some PHP statements. This would have been determined by the file having a file extension of .php.
(3) The PHP interpreter then processes the file and creates an output file that includes HTML statements that were originally in the script along with the results of the script itself which will also now be HTML statements.
(4) The script output is then handed back to the Web server
(5) and handed further on to the web user's browser which will display the page based on the markup in HTML.

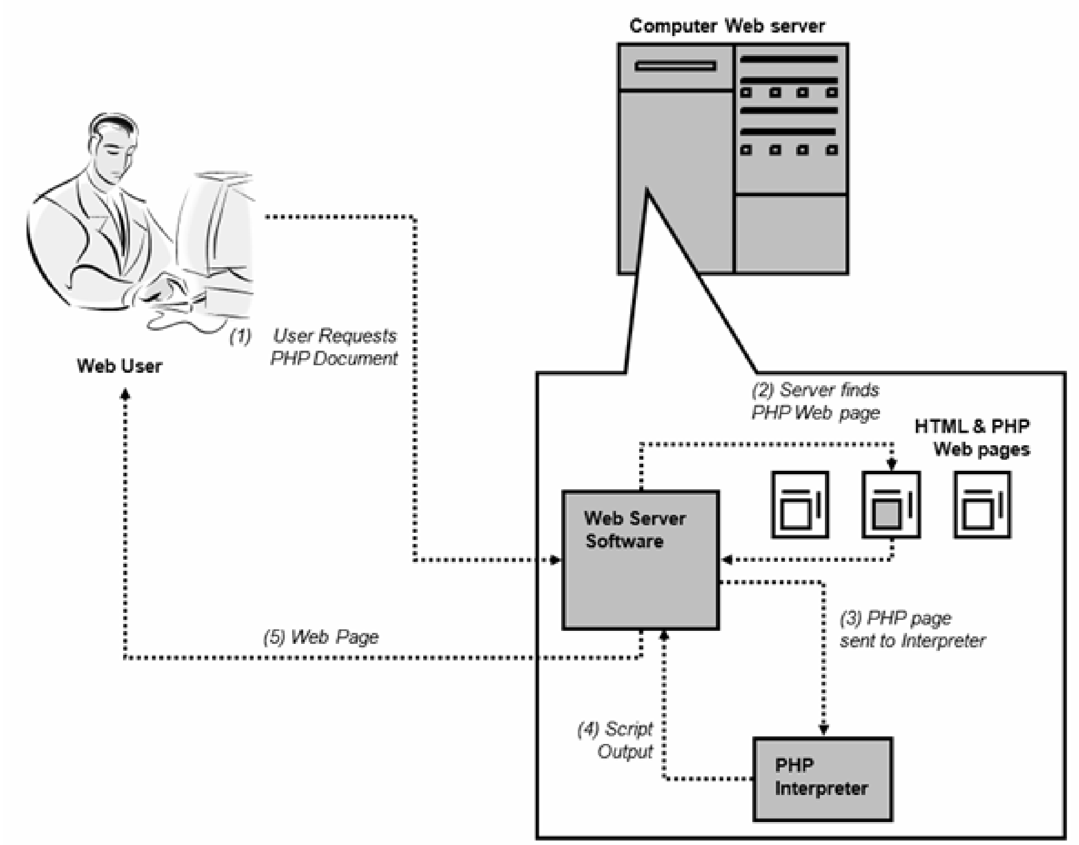


Diagram Source: Stobart, Simon and Parsons, David *Dynamic Web Application Development using PHP and MySQL,* Course Technology Cengage Learning 2008, ISBN-13: 978-1-84480-753-6 Page 226

Chapter 3 – Web Forms and PHP

ICT24A Web 2 Programming

Creating a Form whether created in a simple HTML page, or assembled via output from a program such as PHP, all web forms must have the following:

- Opening and closing <form> and </form> tags.
- A submission method of either Post or Get.
- One or more input fields (although you can omit them, but you won't be able to send any data if you do).
- A destination URL of a program to receive the form data.

Before we start creating any of our scripts complete the following tasks:

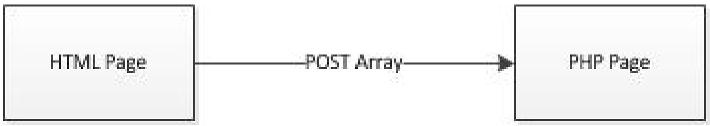
1. In the www.folder of your Uni server installation at a folder named "W4" and as we create new scripts place in this location.
2. Set up a new site in Dreamweaver and set up a new test server that points to this location.

See Over…

Approach one: Separate forms and PHP scripts

In previous study you have created HTML forms and some of the following content is simply a refresher of the process. In this chapter however we will also look at how Web forms and PHP interact .

In the first exercise we're going to create two related webpages. The first contains a form that will collect information from the user and the second is a PHP form that will display that information on the screen. The diagram below shows what's happening when the webpages they used.



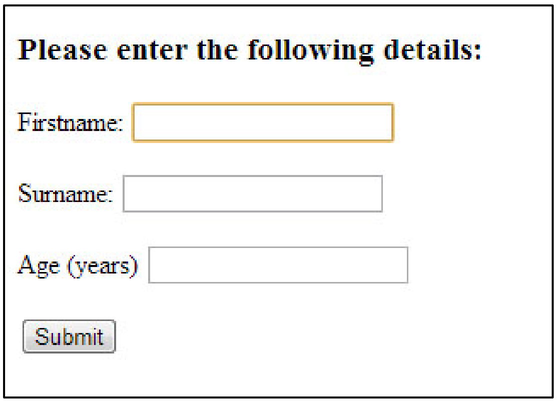
The following example illustrates how to build a simple form. The HTML to create the form is shown below and let's just make a few observations about various parts of the code.

The opening and closing form tags surround all of the other elements of the form. The opening form tags shown on the third line of code includes two attributes. The first of these specify a script that will be run when the form is submitted. The second attribute is the method in this case the method is post. The post method facilitates the creation of an array that will hold all the details collected by the form and the Array will be passed to the PHP form for processing.

There are a various input elements within the form which are given a name and an ID so they can be uniquely identified and referenced. Enter the following code between the body tags of your new webpage.

```
<!-- File: example3_1.htm -->
<h3>Please enter the following details:</h3>

 <form action="example3_1.php" method="post">
   <p>
     <label for="strFirstname">Firstname: </label>
     <input type="text" name="strFirstname" id="strFirstname"/>
   </p><p>
     <label for="strSurname">Surname: </label>
     <input type="text" name="strSurname" id="strSurname"/>
   </p><p>
     <label for="intAge">Age (years)</label>
     <input type="text" name="intAge" id="intAge"/>
   </p><p>
     <input type="submit"/>
   </p>
 </form>
```

It will produce the following output …



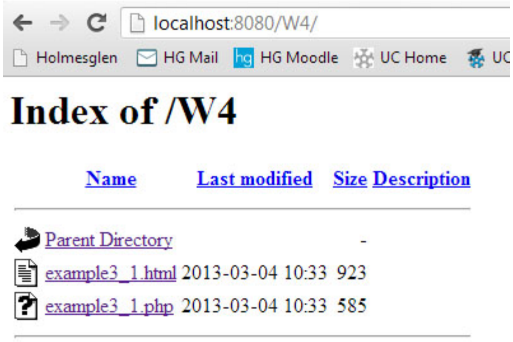
Please enter the following details:

The next example demonstrates one way of developing a PHP script to process the information collected by the form. In this instance we're going to create a separate PHP form to process the details.

1. Create a new PHP page and enter the following script between the body tags

```
<?php
   // File: example3_1.php

   $strFirstname = $_POST["strFirstname"];
   $strSurname = $_POST["strSurname"];
   $intAge = $_POST["intAge"];

   echo "<p>Hello $strFirstname $strSurname</p>";
   echo "<p>Your age is $intAge</p>";
?>
```

2. Save the PHP script and this time we going to run the webpages from Uni server as at times like this LiveView sometimes misbehaves and doesn't run this type of script properly. Use your browser to access your scripts. In this instance I have them in a folder named W4 that resides in the WWW folder of my Uni server.
3. Click on the entry for Example3_1.html to view the page with the form, enter your details and click submit to run the associated PHP script. It will display the details on the screen.



Index of /W4

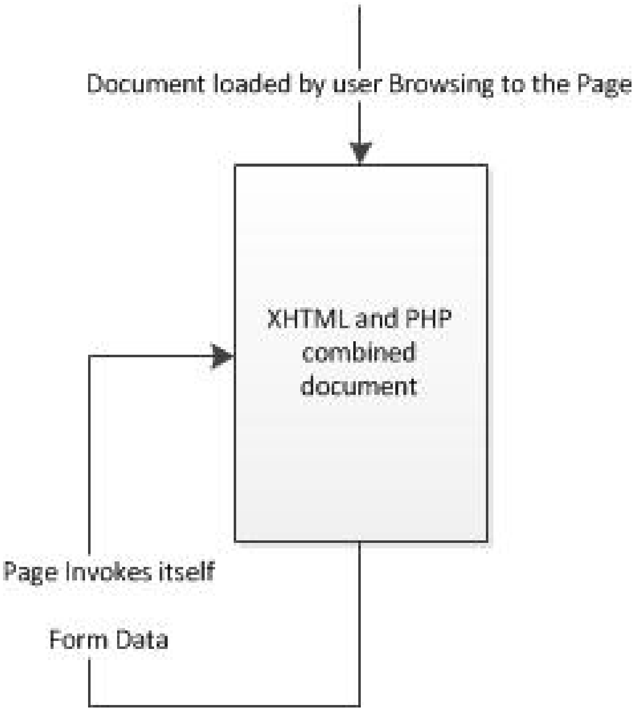
Name	Last modified	Size	Description
Parent Directory		-	
example3_1.html	2013-03-04 10:33	923	
example3_1.php	2013-03-04 10:33	585	

Significant problem with using this method of processing the data arises when we want to use a PHP script to validate the form entries. This next two examples show a second method of combining forms with PHP scripts.

See Over….

Approach two: Combining forms and PHP scripts

if we combine our XHTML form and our PHP script into a single document the user initially loads the page and when they enter the data and then click on the submit button the form invokes itself and processes the PHP script within the document.


Document loaded by user Browsing to the Page
XHTML and PHP combined document
Page Invokes itself
Form Data

See Over….

If we simply combine the HTML and PHP code from the two forms that we had previously we will have the following.

1. Combine the files as shown below and save the work as Example3_2

```
<!-- File: example3_2.php -->

<h3>Please enter the following details:</h3>

 <form action="example3_2.php" method="post">
   <p>
     <label for="strFirstname">Firstname: </label>
     <input type="text" name="strFirstname" id="strFirstname"/>
   </p><p>
     <label for="strSurname">Surname: </label>
     <input type="text" name="strSurname" id="strSurname"/>
   </p><p>
     <label for="intAge">Age (years)</label>
     <input type="text" name="intAge" id="intAge"/>
   </p><p>
     <input type="submit" />
   </p>
 </form>

 <?php
  // File: example3_2.php

  $strFirstname = $_POST["strFirstname"];
  $strSurname = $_POST["strSurname"];
  $intAge = $_POST["intAge"];

  echo "<p>Hello $strFirstname $strSurname</p>";
  echo "<p>Your age is $intAge</p>";

?>
```

2. Use The browser to run the script from uniserver. What happens when we view the webpage?

When we view the webpage initially it is displaying output from the 2 echo statements towards the end of the PHP script. If you enter the details and press submit you will notice that the output at this point is as we expected. It is also possible depending on the settings of your Uni server installation that error messages may appear at this time if they are not suppressed.

If you think about the situation here what we really want is to only display the echo statements once a submit button is pressed rather than initially when the page is first displayed so we need to add a couple extra lines of code to implement this functionality.

3. Save the example 3_2.php script and then save it again as example 3_3.php
4. Then make the couple of changes to the of code to the PHP script as shown below. The changes are highlighted in black text.

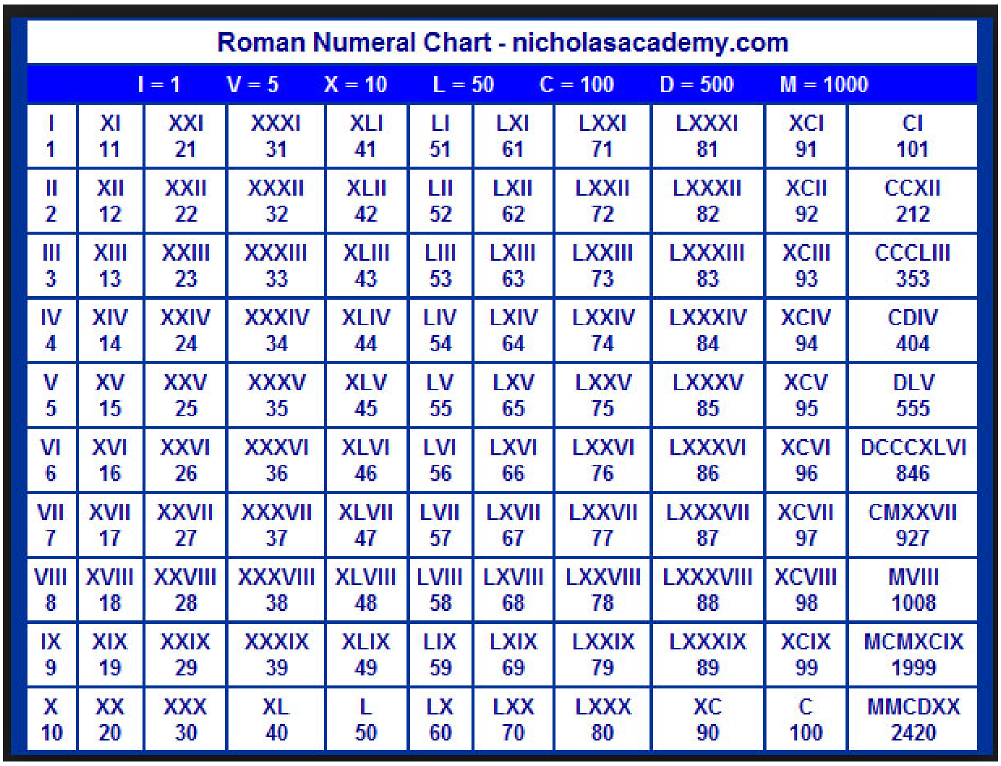
```
<!-- File: example3_3.php -->

<h3>Please enter the following details:</h3>

 <form action="example3_2.php" method="post">
   <p>
     <label for="strFirstname">Firstname: </label>
     <input type="text" name="strFirstname" id="strFirstname"/>
   </p><p>
     <label for="strSurname">Surname: </label>
     <input type="text" name="strSurname" id="strSurname"/>
   </p><p>
     <label for="intAge">Age (years)</label>
     <input type="text" name="intAge" id="intAge"/>
   </p><p>
     <input type="submit" name="submit"/>
   </p>
 </form>

 <?php
  // File: example3_3.php

  if (isset($_POST["submit"])) {
            $strFirstname = $_POST["strFirstname"];
            $strSurname = $_POST["strSurname"];
            $intAge = $_POST["intAge"];

            echo "<p>Hello $strFirstname $strSurname</p>";
            echo "<p>Your age is $intAge</p>";
  }
?>
```

- The first change that has been the age is to add a name attribute to the submit button. This allows us to refer to specifically in the PHP script.
- The second change is to introduce a selection structure so that we can display information from the echo statements only are the submit button has been pressed. The condition in the if statement contains an inbuilt PHP function isset() which checks to see if variables have

been set or not. If the variable has not been set (or created) then isset() returns a false, if the variable has been set then it returns true. In this instance it is checking to see whether the predefined $_POST array for the submit button has been set. This will ensure that the echo statements will only be run up to the submit button is pressed.

Arabic to Roman Numeral Date Conversion.

The next Example applies techniques we learned in the last two chapters as well as introducing a few new aspects. In the example the script is going to accept input from a user and convert a year in the Arabic numerals (e.g. 2013) to its equivalent in Roman numerals (MMXIII). See the chart below for some other examples.



Roman Numeral Chart - nicholasacademy.com

I = 1 V = 5 X = 10 L = 50 C = 100 D = 500 M = 1000

I 1	XI 11	XXI 21	XXXI 31	XLI 41	LI 51	LXI 61	LXXI 71	LXXXI 81	XCI 91	CI 101
II 2	XII 12	XXII 22	XXXII 32	XLII 42	LII 52	LXII 62	LXXII 72	LXXXII 82	XCII 92	CCXII 212
III 3	XIII 13	XXIII 23	XXXIII 33	XLIII 43	LIII 53	LXIII 63	LXXIII 73	LXXXIII 83	XCIII 93	CCCLIII 353
IV 4	XIV 14	XXIV 24	XXXIV 34	XLIV 44	LIV 54	LXIV 64	LXXIV 74	LXXXIV 84	XCIV 94	CDIV 404
V 5	XV 15	XXV 25	XXXV 35	XLV 45	LV 55	LXV 65	LXXV 75	LXXXV 85	XCV 95	DLV 555
VI 6	XVI 16	XXVI 26	XXXVI 36	XLVI 46	LVI 56	LXVI 66	LXXVI 76	LXXXVI 86	XCVI 96	DCCCXLVI 846
VII 7	XVII 17	XXVII 27	XXXVII 37	XLVII 47	LVII 57	LXVII 67	LXXVII 77	LXXXVII 87	XCVII 97	CMXXVII 927
VIII 8	XVIII 18	XXVIII 28	XXXVIII 38	XLVIII 48	LVIII 58	LXVIII 68	LXXVIII 78	LXXXVIII 88	XCVIII 98	MVIII 1008
IX 9	XIX 19	XXIX 29	XXXIX 39	XLIX 49	LIX 59	LXIX 69	LXXIX 79	LXXXIX 89	XCIX 99	MCMXCIX 1999
X 10	XX 20	XXX 30	XL 40	L 50	LX 60	LXX 70	LXXX 80	XC 90	C 100	MMCDXX 2420

Source: http://nicholasacademy.com/printroman.html accessed 4/3/2013

In each of the previous examples we have developed we have hard coded the name of the script to be run when the form is submitted. Note the second line of code in this example uses a server variable which contains the name of the current script. **$_SERVER["PHP_SELF"]** If we echo its contents as part of the form elements usage tools the script would always call itself.

1. Enter the script on the next page between the body tags of a new web page, review its use of selection structures and loops, run the script and test it out.
2. For more discussion on how the Roman Numerals work see http://en.wikipedia.org/wiki/Roman_numerals

```
<h2>Please enter a date in Arabic numerals:</h2>

<form action='<?php echo $_SERVER["PHP_SELF"]; ?>' method='post'>
<p>
<label for="intDate">Date: </label>
<input type="text" name="intDate" id="intDate"/></p>
<p><input type="submit" name="submit"/></p>
</form>

<?php
// File: example3_4.php

if (isset($_POST["intDate"])) {
   $intDate = $_POST["intDate"];

   echo "<p>$intDate is written ";
   while ($intDate >= 1000) {
      echo "M";
      $intDate -= 1000;
   }
   if ($intDate >= 900) {
      echo "CM";
      $intDate -= 900;
   }
   while ($intDate >= 500) {
      echo "D";
      $intDate -= 500;
   }
   if ($intDate >= 400) {
      echo "CD";
      $intDate -= 400;
   }
   while ($intDate >= 100) {
      echo "C";
      $intDate -= 100;
   }
   if ($intDate >= 90) {
      echo "XC";
      $intDate -= 90;
   }
   while ($intDate >= 50) {
      echo "L";
      $intDate -= 50;
   }
   if ($intDate >= 40) {
      echo "XL";
      $intDate -= 40;
   }
   while ($intDate >= 10) {
      echo "X";
```

```
        $intDate -= 10;
    }
    if ($intDate >= 9) {
        echo "IX";
        $intDate -= 9;
    }
    while ($intDate >= 5) {
        echo "V";
        $intDate -= 5;
    }
    if ($intDate >= 4) {
        echo "IV";
        $intDate -= 4;
    }
    while ($intDate >= 1) {
        echo "I";
        $intDate -= 1;
    }
    echo " in Roman numerals</p>";
}
?>
```

See Over..

Capturing data from different form elements.

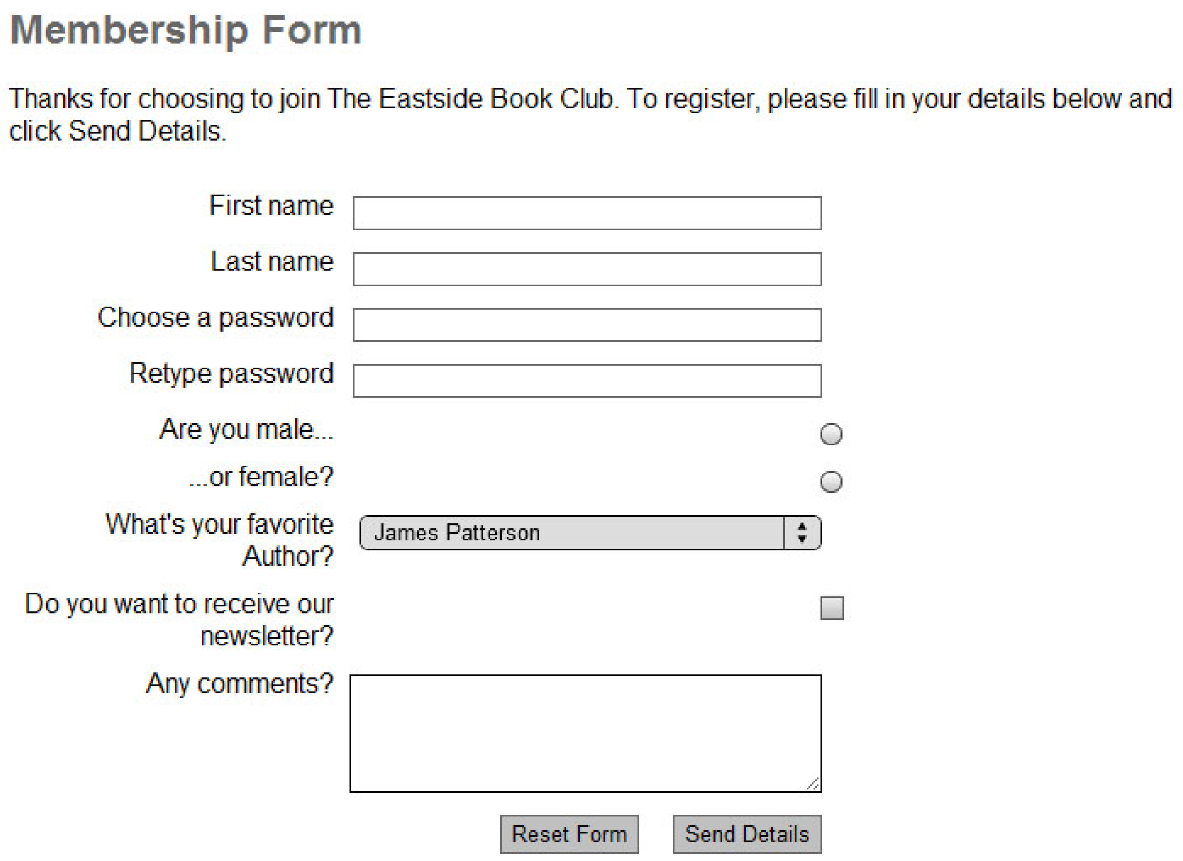
The example below demonstrates a HTML form that collects information from prospective members and passes the information to a PHP page which then displays some output to confirm with the user that the details are correct.

This exercise is based on scripts in the text “Doyle, Matt (2011-01-06). Beginning PHP 5.3 (Wrox Programmer to Programmer) (p. 232). John Wiley and Sons. Kindle Edition. You will need to file **common.css** to be placed in the same folder as your XHTML and PHP scripts.

1. The first task is to create the XHTML document from the statements below. Place the statements between the body tags of a new XHTML document.

```
<!-- File: example3_5.htm -->
<h1>Membership Form</h1>
<p>Thanks for choosing to join The Eastside Book Club. To register, please fill in your details below
and click Send Details.</p>
<form action="example3_5.php" method="post">

  <div style="width: 30em;">
    <label for="firstName">First name</label>
    <input type="text" name="firstName" id="firstName" value="" />
    <label for="lastName">Last name</label>
    <input type="text" name="lastName" id="lastName" value="" />
    <label for="password1">Choose a password</label>
    <input type="password" name="password1" id="password1" value="" />
    <label for="password2">Retype password</label>
    <input type="password" name="password2" id="password2" value="" />
    <label for="genderMale">Are you male...</label>
    <input type="radio" name="gender" id="genderMale" value="M" />
    <label for="genderFemale">...or female?</label>
    <input type="radio" name="gender" id="genderFemale" value="F" />
    <label for="favoriteAuthor">What's your favorite Author?</label>

    <select name="favoriteAuthor" id="favoriteAuthor" size="1">
      <option value="James Patterson">James Patterson</option>
      <option value="Tess Gerritson">Tess Gerritson</option>
      <option value="Tom Clancy">Tom Clancy</option>
    </select>

    <label for="newsletter">Do you want to receive our newsletter?</label>
    <input type="checkbox" name="newsletter" id="newsletter" value="yes" />
    <label for="comments">Any comments?</label>
    <textarea name="comments" id="comments" rows="4" cols="50"> </textarea>

    <div style="clear: both;">
      <input type="submit" name="submitButton" id="submitButton" value="Send Details" />
     <input type="reset" name="resetButton" id="resetButton" value="Reset Form" style="margin-right: 20px;" />
    </div>
  </div>
</form>
```

Your webpage should look like:



Membership Form

Thanks for choosing to join The Eastside Book Club. To register, please fill in your details below and click Send Details.

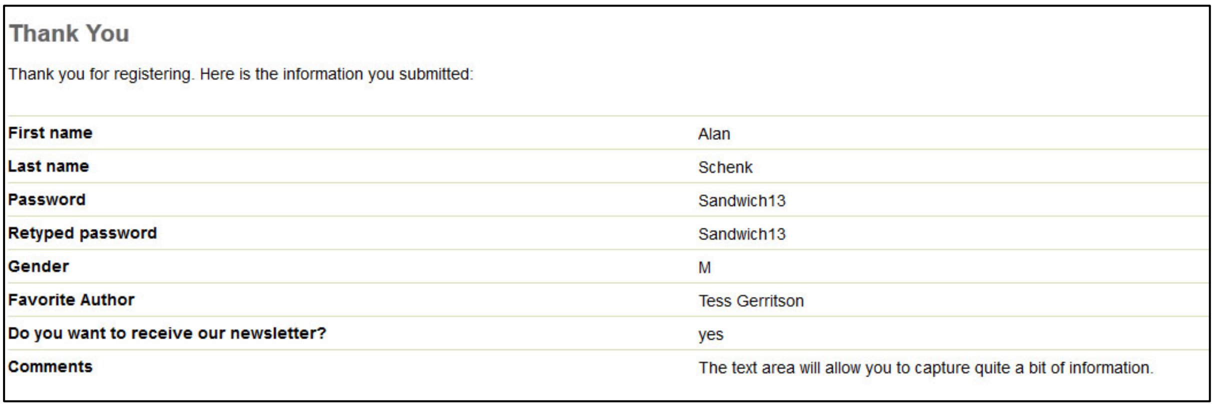
First name	
Last name	
Choose a password	
Retype password	
Are you male...	○
...or female?	○
What's your favorite Author?	James Patterson
Do you want to receive our newsletter?	□
Any comments?	

2. The second task is to create the PHP document from the statements below. Place the statements between the body tags of a new XHTML document.

See over for script statements…

```
<!-- File: example3_5.php -->
<h1>Thank You</h1>

  <p>Thank you for registering. Here is the information you submitted:</p>

  <dl>
   <dt>First name</dt><dd><?php echo $_POST["firstName"]?></dd>
   <dt>Last name</dt><dd><?php echo $_POST["lastName"]?></dd>
   <dt>Password</dt><dd><?php echo $_POST["password1"]?></dd>
   <dt>Retyped password</dt><dd><?php echo $_POST["password2"]?></dd>
   <dt>Gender</dt><dd><?php echo $_POST["gender"]?></dd>
   <dt>Favorite Author</dt><dd><?php echo $_POST["favoriteAuthor"]?></dd>
   <dt>Do you want to receive our newsletter?</dt><dd><?php echo $_POST["newsletter"]?></dd>
   <dt>Comments</dt><dd><?php echo $_POST["comments"]?></dd>
  </dl>
```

When you run the XHTML document it will collect the data and then invoke the PHP script and you should see output in the following format.



Thank You

Thank you for registering. Here is the information you submitted:

First name	Alan
Last name	Schenk
Password	Sandwich13
Retyped password	Sandwich13
Gender	M
Favorite Author	Tess Gerritson
Do you want to receive our newsletter?	yes
Comments	The text area will allow you to capture quite a bit of information.

See Over…

Validating forms.

The example below demonstrates a PHP form that collects the title first name and surname of the user submitted for processing. Once the submit button is pressed the PHP script checks to ensure that there is an entry in each input element and if there is not pushing the message next to the element that has missing data. If the validation confirms that all the data is present in a message is printed out to the user.

1. Create a PHP document that includes the following code between the body tags. The statements continue on to a second page.
2. Once it is completed and tested out to ensure that its operating properly.
3. A number of comments on different sections of code appear the text of the script.

```
<?php
// File: example10_6.php
$booTitle = 0;
$booFirstname = 0;
$booSurname = 0;
$strTitle = "";
$strFirstname = "";
$strSurname = "";


if (isset($_POST["submit"])) {
  if($_POST["strTitle"] == "Select...")
    $booTitle = 1;
  else
        $strTitle = $_POST["strTitle"];
  if($_POST["strFirstname"] == NULL)
    $booFirstname = 1;
  else
        $strFirstname = $_POST["strFirstname"];
  if($_POST["strSurname"] == NULL)
    $booSurname = 1;
  else
        $strSurname = $_POST["strSurname"];
}
?>
<h2>Please select your title and name:</h2>
<form action='<?php echo $_SERVER["PHP_SELF"]; ?>' method='post'>
  <p>
    <label for="strTitle">Title: </label>
    <select name='strTitle' id='strTitle'>
      <option>Select...</option>
      <option <?php if($strTitle == "Mr") echo "Selected" ?>>Mr</option>
      <option <?php if($strTitle == "Miss") echo "Selected" ?>>Miss</option>
      <option <?php if($strTitle == "Ms") echo "Selected" ?>>Ms</option>
      <option <?php if($strTitle == "Mrs") echo "Selected" ?>>Mrs</option>
      <option <?php if($strTitle == "Dr") echo "Selected" ?>>Dr</option>
    </select>
    <?php if ($booTitle) echo "Please select a title!" ?>
  </p>

<p>
    <label for="strFirstname">Firstname: </label>
    <input type='text' name='strFirstname' value='<?php echo $strFirstname ?>' id='strFirstname'/>
    <?php if ($booFirstname) echo "Please enter a firstname!" ?>
  </p>
  <p>
    <label for="strSurname">Surname: </label>
    <input type='text' name='strSurname' value='<?php echo $strSurname ?>' id='strSurname'/>
    <?php if ($booSurname) echo "Please enter a surname!" ?>
  </p>
  <p>
    <input type='submit' name='submit'/>
  </p>
</form>
<?php
if (!($booTitle + $booFirstname + $booSurname) && isset($_POST["submit"]))
  echo "<p>All is well, you are " . $_POST["strTitle"] . " "  . $_POST["strFirstname"] . " " .
$_POST["strSurname"] . "</p>";
?>
```

Comments on code:

- The statements in this section of the code that is in **green** are testing to see if the input items each have a value.
- The initial if Statement is ensuring that this code is only run if the submit button has been pressed.
- The inner if statements are testing to see if values exists in the$_POST array for each of the items and if it is not since the variable that simulates a Boolean.
- The statements in the beige color display an error message beside any element that is missing a value.
- The statements in the orange text is testing a condition in which the variables for each item is true(all have data) and that the submit button has been pressed displaying the data and message to say that everything was correct.

That's all for now!!!!