**Exercise 02\_06\_01 – Step 1**

In this Exercise, we will learn how to use advanced techniques on both indexed and associative arrays.



1. Create a folder named Exercise 02\_06\_01 and open it with your IDE. Create a new file called ***PostMessage.php***. Scaffold a basic HTML code layout into it. Complete our standard opening documentation in the ***<head>*** element. Make sure to have the ***modernizr*** <script> linked in. Set the <title> content to ***View Files***:  
   ***<!DOCTYPE html>  
   <html lang="en">  
   <head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width">  
    <title>Post New Message</title>  
    <script src="modernizr.custom.65897.js"></script>  
   </head>  
   <body>  
   </body>  
   </html>***
2. In the <body>, create a set of PHP standard script delimiters. After the code block, add an ***<h1>*** element with content ***Post New Message***.   
    ***<?php  
     
    ?>  
    <h1>Post New Message</h1>***Copy the project folder into the appropriate spot on your Web Server and test it.
3. Add the following HTML code below the ***<h1>*** element to create the form to collect the messages:  
    <h1>Post New Message</h1>  
    ***<hr>  
    <form action="PostMessage.php" method="post">  
    <span style="font-weight: bold">Subject: <input   
    type="text"   
    name="subject"></span>  
    <span style="font-weight: bold">Name: <input   
    type="text"   
    name="name"></span><br>  
    <textarea name="message" rows="6" cols="80"   
    style="margin: 10px 5px 5px"></textarea><br>  
    <input type="submit" name="submit" value="Post   
    Message">  
    <input type="reset" name="reset" value="Reset Form">  
    </form>  
    <hr>  
    <p>  
    <a href="MessageBoard.php">View Message</a>  
    </p>***Give this a browser/server test.
4. Add the following code into the script section to create a message that will be submitted to a text file. Notice that we will remove any ***tilde*** (***~***) characters from the input, as we will use them as ***delimiters*** for the fields of the message records. Add some ***debug*** code to check your work:  
    <?php  
    ***if (isset($\_POST['submit'])) {  
    $subject = stripslashes($\_POST['subject']);  
    $name = stripslashes($\_POST['name']);  
    $message = stripslashes($\_POST['message']);  
    $subject = str\_replace("~", "-", $subject);  
    $name = str\_replace("~", "-", $name);  
    $message = str\_replace("~", "-", $message);  
    $messageRecord = "$subject~$name~$message\n";  
    echo $messageRecord; // debug  
    }*** ?>  
   Give this a browser/server test.
5. Remove the debug code. Add the following code at the bottom of the script section to create or just append to a file named ***messages.txt***:  
    $messageRecord = "$subject~$name~$message\n";  
    ***$messageFile = fopen("messages.txt", "ab");  
    if ($messageFile === false) {  
    echo "There was an error saving your message!\n";  
    }  
    else {  
    fwrite($messageFile, $messageRecord);  
    fclose($messageFile);  
    echo "Your message has been saved.\n";  
    }***Give this a browser/server test. Add the following messages one at a time, open the messages.txt file to check them:  
     
   **String Question  
   Mark Buckler  
   Which string function would you use to determine if an essay keyed in a textarea box exceeds the maximum number of words allowed?  
     
   Sending Mail with PHP  
   Mark Buckler  
   What are the three required arguments of the mail() function?  
     
   PHP and Directories  
   Mark Buckler  
   What function returns an indexed array containing the names of files and directories in the specified directory?**

**Exercise 02\_06\_01 – Step 2**



1. Copy the file ***PostMessage.php*** to a file named ***MessageBoard.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***MessageBoard.*** In the <body>, create an ***<h1>*** element with content ***View Files 2***. Create a set of PHP standard script delimiters. Below the delimiters add a ***hyperlink*** to take you to the PostMessage.php app:  
    ***<h1>Message Board</h1>  
    <?php  
     
    ?>  
    <p><a href="PostMessage.php">Post New   
    Message</a></p>***  
   Give this a server/browser test.
2. Create a new file named ***debug.php*** and add the following code to it. We will be using it often to debug our arrays:  
    ***echo "<pre>\n";  
    print\_r();  
    echo "</pre>\n";***
3. Add the following code to the script section to read the ***messages.txt*** file. Notice the use of the ***file()*** function to read the messages into an array. Copy in the ***debug*** code to test the array:  
    <?php ***if (!file\_exists("messages.txt") || filesize("messages.txt")   
    == 0) {  
    echo "<p>There are no messages posted.</p>\n";  
    }  
    else {  
    $messageArray = file("messages.txt");  
    echo "<pre>\n";  
    print\_r($messageArray);  
    echo "<pre>\n";  
    }*** ?>  
   Give this a server/browser test.
4. Remove the debug code and add the following code to the script section to deconstruct each data record into an array. Copy in the ***debug*** code to test the array:  
    $messageArray = file("messages.txt");  
    ***echo "<table style=\"background-color: lightgray\"   
    border=\"1\" width=\"100%\">\n";  
    $count = count($messageArray);  
    for ($i = 0; $i < $count; $i++) {  
    $currMsg = explode("~", $messageArray[$i]);  
    echo "<pre>\n";  
    print\_r($currMsg);  
    echo "</pre>\n";  
    }  
    echo "</table>\n";***Give this a server/browser test.
5. Remove the debug code and add the following code to the script section to display the data:  
    $currMsg = explode("~", $messageArray[$i]);  
    ***echo "<tr>\n";  
    echo "<td width=\"5%\" style=\"text-align: center;   
    font-weight: bold\">" . ($i + 1) . "</td>\n";  
    echo "<td width=\"95%\"><span style=\"font-weight:   
    bold\">Subject: </span>" .   
    htmlentities($currMsg[0]) . "<br>\n";  
    echo "<span style=\"font-weight: bold\">Name:   
    </span>" . htmlentities($currMsg[1]) . "<br>\n";  
    echo "<span style=\"text-decoration: underline; font-  
    weight: bold\">Message: </span><br>\n" .   
    htmlentities($currMsg[2]) . "</td>\n";  
    echo "</tr>\n";***Give this a server/browser test.

**Exercise 02\_04\_01 – Step 3**



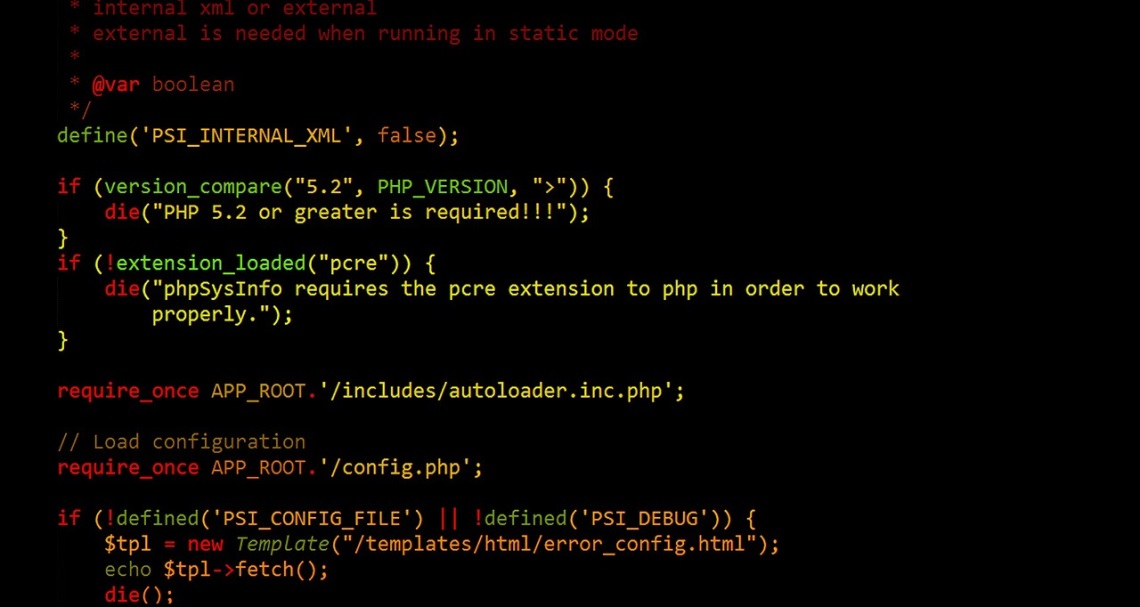
1. Make a backup copy of ***messages.txt***. Reopen the file ***MessageBoard.php*** with your IDE. Add the following ***hyperlink*** to the ending ***<p>*** element, below the first hyperlink. Note that when clicked it will create a URL with an ***action*** name/value pair. The ***%20*** in the URL is hexadecimal for a space character:  
    <p>  
    <a href="PostMessage.php">Post New Message</a>***<br>  
    <a href="MessageBoard.php?action=Delete%20First">  
    Delete First Message</a>*** </p>Give this a server/browser test. Click the Delete First Message link and check the URL for correctness.
2. Add the following code to the start of the PHP code block, above the existing code. It will check for the ***action*** name/value pair on the URL. It then checks for the existence of the ***messages.txt*** file, and reads its contents into an array. Copy in the ***debug*** code to check the array contents:  
    <?php ***if (isset($\_GET['action'])) {  
    if (file\_exists("messages.txt") &&   
    filesize("messages.txt") != 0) {  
    $messageArray = file("messages.txt");  
    echo "<pre>\n";  
    print\_r($messageArray);  
    echo "</pre>\n";  
    }  
    }***Give this a server/browser test. Click the ***Delete First Message*** link and check the array contents for correctness.
3. Add a ***switch*** statement into the ***if*** statement directly after the ***file()*** call, leaving the debug code. We will use a switch because we will be adding more cases later:  
    $messageArray = file("messages.txt");  
    ***switch ($\_GET['action']) {  
    case 'Delete First':  
    array\_shift($messageArray);  
    break;  
    }*** echo "<pre>\n";  
    print\_r($messageArray);  
    echo "</pre>\n";  
   Give this a server/browser test. Make sure the first message is gone from the array by looking at the debug. You will see that it has not yet been removed from the ***messages.txt*** file, because it remains in the <table> display.
4. Remove the ***print\_r()*** debug code. Add an ***if…else*** construct below the ***switch***. It will check for any messages left in the array. If none, we will use an ***unlink()*** call to delete ***messages.txt***. For the case of remaining messages, place a debug statement for now:   
    ***if (count($messageArray) > 0) {  
    echo "There are remaining messages in the   
    array.<br>"; // debug  
    }  
    else {  
    unlink("messages.txt");  
    }***Give this a server/browser test. The debug in the ***if*** should work, but we cannot yet test the ***unlink()*** as no messages are removed from the file yet.
5. Remove the ***debug*** code in the ***if*** statement and add the following code. We will first create a variable to which we will ***implode()*** the remaining ***$messageArray*** as a single text string. We will then open the messages.txt file in ***wb*** mode, open for writing and deleting the current contents. We then use ***fwrite()*** to replace the contents, and then ***fclose()*** the file:  
    if (count($messageArray) > 0) {  
    ***$newMessages = implode($messageArray);  
    $messageFile = fopen("messages.txt", "wb");  
    if ($messageFile === false) {  
    echo "There was an error updating the message   
    file.\n";  
    }  
    else {  
    fwrite($messageFile, $newMessages);  
    fclose($messageFile);  
    }  
    }***Give this a server/browser test. Repeat the deletes until the <table> is empty. Check to see that the file has been unlinked and is gone. When the file is empty, copy the messages from your backup to ***messages.txt***.

**Exercise 02\_04\_01 – Step 4**



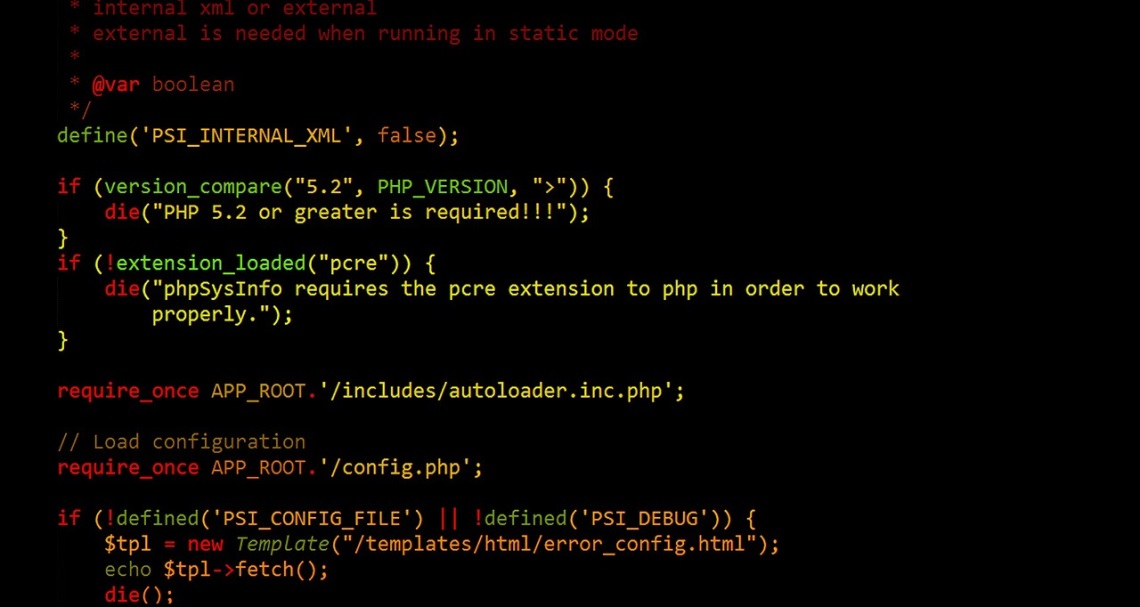
1. Make sure you keep a backup copy of ***messages.txt***. Reopen the file ***MessageBoard.php*** with your IDE. Add the following ***hyperlink*** to the ending ***<p>*** element, below the last hyperlink. Note that when clicked it will create a URL with an ***action*** name/value pair. The ***%20*** in the URL is hexadecimal for a space character:  
    <p>  
    <a href="PostMessage.php">Post New Message</a><br> <a href="MessageBoard.php?action=Delete%20First">  
    Delete First Message</a>***<br>***  
    ***<a href="MessageBoard.php?action=Delete%20Last">  
    Delete Last Message</a>*** </p>Give this a server/browser test. Click the Delete Last Message link and check the URL for correctness.
2. Add the following ***case*** to the ***switch*** statement. This will delete the last message from the array. The existing file writing code will take care of the ***messages.txt*** file:  
    ***case 'Delete Last':  
    array\_pop($messageArray);  
    break;***Give this a server/browser test. Repeat the deletes until the <table> is empty. Check to see that the file has been unlinked and is gone. When the file is empty, copy the messages from your backup to ***messages.txt***.

**Exercise 02\_04\_01 – Step 5**



1. Make sure you keep a backup copy of ***messages.txt***. Reopen the file ***MessageBoard.php*** with your IDE. Add the following ***case*** to the ***switch*** statement. This will delete one of the messages from the array. We will use the message board itself to generate a URL from a hyperlink:  
    ***case 'Delete Message':  
    if (isset($\_GET['message'])) {  
    array\_splice($messageArray, $\_GET['message'],   
    1);  
    }  
    break;***Give this a server/browser test for syntax.
2. Now let’s modify the page presentation to handle more columns. Modify the second column ***<td>*** as follows:  
    echo "<td width=\"***85%***\"><span style=\"font-weight:   
    bold\">Subject: </span>" .   
    htmlentities($currMsg[0]) . "<br>\n";Give this a server/browser test for presentation, there will probably be no noticeable change.
3. Now let’s modify the page presentation to add another table column to the right. Add another ***<td>*** before the closing ***</tr>*** as follows:  
    ***echo "<td width=\"10%\" style=\"text-align:   
    center\">" .   
    "<a href='MessageBoard.php?" .   
    "action=Delete%20Message&" .   
    "message=$i'>" .   
    "Delete This Message</a></td>\n";***Give this a server/browser test for syntax. Select a message and delete it. Repeat the deletes until the <table> is empty. Check to see that the file has been unlinked and is gone. When the file is empty, copy the messages from your backup to ***messages.txt***.

**Exercise 02\_04\_01 – Step 6**



1. Make sure you keep a backup copy of ***messages.txt***. Reopen the file ***MessageBoard.php*** with your IDE. Modify the Delete Message ***case*** to use ***unset()***. This is a preferable way to delete an element from the array, because it deletes by index number and can delete multiple elements per call if they are comma-separated. However, it has a problem of renumbering the indexes, as we will see with our debug code:  
    case 'Delete Message':  
    if (isset($\_GET['message'])) {  
    ***$index = $\_GET['message'];  
    unset($messageArray[$index]);  
    echo "<pre>\n";  
    print\_r($messageArray);  
    echo "</pre>\n";*** }  
    break;  
   Delete the middle element, and you can see that there is now an index missing. This is called a ***sparse*** array. Copy the messages from the backup back into the file.
2. We will use ***array\_values()*** to renumber the indexes of the sparse array after removing an element:  
    case 'Delete Message':  
    if (isset($\_GET['message'])) {  
    unset($messageArray[$index]);  
    ***$messageArray =   
    array\_values($messageArray);  
    echo "<pre>\n";  
    print\_r($messageArray);  
    echo "</pre>\n";*** }  
    break;  
   Give this a server/browser test for syntax. Select a message and delete it. Look at the debug to make sure the indexes are fixed. Repeat the deletes until the file is empty. When the file is empty, remove the debug and copy the messages from your backup to ***messages.txt***.

**Exercise 02\_04\_01 – Step 7**



1. Make sure you keep a backup copy of ***messages.txt***. Open the file and ***duplicate*** one of the messages and save it. Give this a server/browser test to ensure there is a duplicate message.
2. Reopen the file ***MessageBoard.php*** with your IDE. Add the following ***hyperlink*** to the ending ***<p>*** element, below the last hyperlink. Note that when clicked it will create a URL with an ***action*** name/value pair. The ***%20*** in the URL is hexadecimal for a space character:  
    <p>  
    <a href="PostMessage.php">Post New Message</a><br> <a href="MessageBoard.php?action=Delete%20First">  
    Delete First Message</a><br>  
    <a href="MessageBoard.php?action=Delete%20Last">Delete Last   
    Message</a>***<br>  
    <a href="MessageBoard.php?action=  
    Remove%20Duplicates">Remove Duplicates</a>*** </p>Give this a server/browser test. Click the Delete First Message link and check the URL for correctness.
3. Add the following ***case*** to the ***switch*** statement. The ***array\_unique()*** function will remove any duplicate messages from the file. We then have to make sure the array is properly indexed:  
    ***case 'Remove Duplicates':  
    $messageArray = array\_unique($messageArray);  
    $messageArray = array\_values($messageArray);  
    break;***Give this a server/browser test.

**Exercise 02\_04\_01 – Step 8**



1. Reopen the file ***MessageBoard.php*** with your IDE. Comment out the opening ***<table>*** statement and entire ***for*** loop and closing ***</table>*** tag. Leave the **$count** assignment. We will come back to it and modify it later:  
    **/\*  
    echo "<table style=\"background-color: lightgray\"   
    border = \"1\" width=\"100%\">\n";  
    \*/** $count = count($messageArray);  
    ***/\*  
    for ($i = 0; $i < $count; $i++) {  
    . . .  
    echo "</table>\n";  
    \*/***
2. Add the following ***for*** loop and ***debug*** directly after the ***$count*** assignment. Notice that the explode() again turns the data into a new array. We then convert that array into an ***associative*** array by assigning the first field as the ***key*** and the rest as ***values***:  
    $count = count($messageArray);  
    ***for ($i = 0; $i < $count; $i++) {  
    $currMsg = explode("~", $messageArray[$i]);  
    $keyMessageArray[$currMsg[0]] = $currMsg[1] . "~" .   
    $currMsg[2];  
    }  
    echo "<pre>\n";  
    print\_r($keyMessageArray);  
    echo "</pre>\n";***Give this a server/browser test. We should see that we have created an ***associative*** array with the ***subject*** field used as the array keys. Now we can remove the ***debug*** code.

**Exercise 02\_04\_01 – Step 9**



1. Reopen the file ***MessageBoard.php*** with your IDE. Remove the two sets of block comments and we will begin to modify the ***for*** loop. First let’s set up a couple of variables between the for loops to track the ***current*** ***file*** ***pointer*** and the ***current*** ***key***:  
    ***$index = 1;  
    $key = key($keyMessageArray);*** for ($i = 0; $i < $count; $i++) {
2. Next we will modify the ***for*** loop to a ***foreach*** loop and make use of our new variables and some ***internal array pointer*** functions:  
    $key = key($keyMessageArray);  
    ***foreach ($keyMessageArray as $message) {***  
    $currMsg = explode("~", ***$message***);  
    echo "<tr>\n";  
    echo "<td width=\"5%\" style=\"text-align: center; font-  
    weight: bold\">" . ***$index*** . "</td>\n";  
    echo "<td width=\"85%\"><span style=\"font-weight:   
    bold\">Subject: </span>" . htmlentities(***$key***) . "<br>\n";  
    echo "<span style=\"font-weight: bold\">Name: </span>" .   
    htmlentities($currMsg[***0***]) . "<br>\n";  
    echo "<span style=\"text-decoration: underline; font-weight:   
    bold\">Message </span><br>\n" .   
    htmlentities($currMsg[***1***]) . "</td>\n";  
    echo "<td width=\"10%\" style=\"text-align: center\">" .   
    "<a href='MessageBoard.php?" .   
    "action=Delete%20Message&" .   
    ***"message=" . ($index - 1) .***   
    "***'>***Delete This Message</a></td>\n";  
    echo "</tr>\n";  
    ***++$index;  
    next($keyMessageArray);  
    $key = key($keyMessageArray);*** }Give this a server/browser test. We should see that we have created an ***associative*** array with the ***subject*** field used as the array keys. Now we can remove the ***debug*** code.

**Exercise 02\_04\_01 – Step 10**



1. Open the file ***PostMessage.php*** with your IDE. We will put in code to insure the user can only enter ***unique*** subjects. Firstly, we must get the ***messages.txt*** file if it exists and is not empty. Then we will put the messages into a ***$messageArray[]*** array as we have done before. Directly before the initialization of ***$messageRecord*** enter the following code with some debug.  
    $message = str\_replace("~", "-",$message);  
    ***$existingSubjects = array();  
    if (file\_exists("messages.txt") &&   
    filesize("messages.txt") != 0) {  
    $messageArray = file("messages.txt");  
    $count = count($messageArray);  
    echo "$count<br>";  
    }***
2. Let’s comment out the file operations so the file does not get messed up on submit:   
   ***//*** $messageFile = fopen("messages.txt", "ab");  
   ***//*** if ($messageFile === false) {  
   ***//*** echo "There was an error saving your message!\n";  
   ***//*** }  
   ***//*** else {  
   ***//*** fwrite($messageFile, $messageRecord);  
   ***//*** fclose($messageFile);  
   ***//*** echo "Your message has been saved.\n";  
   Give this a server/browser test. ***Submit*** with ***Post Message*** and we should see that we have created an array with the proper number of messages.
3. Remove the debug code. Now let’s ***explode()*** the existing subjects into a ***$existingSubjects*** array. Let’s use our saved debug code to examine its contents.  
    $count = count($messageArray);  
    ***for ($i = 0; $i < $count; $i++) {  
    $currMsg = explode("~", $messageArray[$i]);  
    $existingSubjects[] = $currMsg[0];  
    }  
    echo "<pre>\n";  
    print\_r($existingSubjects);  
    echo "</pre>\n";***Give this a server/browser test. ***Submit*** with ***Post Message*** and we should see that we have created an array with the ***subject*** fields of the messages.
4. Remove the debug code and let’s see if we can stop the user from entering a duplicate. Directly after the previous if statement, enter the following code:  
    ***if (in\_array($subject, $existingSubjects)) {  
    echo "<p>The subject <em>\"$subject\"</em> you   
    entered already exists!<br>\n";  
    echo "Please enter a new subject and try   
    again.<br>\n";  
    echo "Your message was not saved.</p>";  
    $subject = "";  
    }***Give this a server/browser test. Enter an existing subject and ***submit*** with ***Post Message***. We should get an error message. Enter a new subject and we should not.
5. Now let’s enclose our ***commented*** file write code in an ***else*** clause, and empty the ***$subject*** and ***$message*** variables. ***Uncomment*** the commented code as follows:  
    ***else {*** $messageRecord = "$subject~$name~$message\n";  
    $messageFile = fopen("messages.txt", "ab");  
    if ($messageFile === false) {  
    echo "There was an error saving your message!\n";  
    }  
    else {  
    fwrite($messageFile, $messageRecord);  
    fclose($messageFile);  
    echo "Your message has been saved.\n";  
    ***$subject = "";  
    $message = "";*** }  
    ***}***
6. Just before the closing ***?>*** PHP delimiter, let’s clean up the variables as follows:  
    ***else {  
    $subject = "";  
    $name = "";  
    $message = "";  
    }*** ?>  
   Give this a server/browser test. Enter an existing subject and ***submit*** with ***Post Message***. We should get an error message. Enter a new subject and it should be saved.
7. Convert the form into a ***sticky*** form with the following changes:  
    <span style="font-weight: bold">Subject: <input type="text"   
    name="subject"   
    ***value="<?php echo $subject ?>"***></span>  
    <span style="font-weight: bold">Name: <input type="text"   
    name="name"   
    ***value="<?php echo $name ?>"***></span><br>  
    <textarea name="message" rows="6" cols="80"   
    style="margin: 10px 5px 5px">  
    ***<?php echo $message ?>***</textarea><br>  
   Give this a server/browser test.

**Exercise 02\_04\_01 – Step 11**



1. Reopen the file ***MessageBoard.php*** with your IDE. Add the following ***hyperlink*** to the ending ***<p>*** element, below the ***first*** hyperlink:  
    <p>  
    <a href="PostMessage.php">Post New Message</a><br> ***<a href="MessageBoard.php?action=Sort%20Ascending">  
    Sort Subjects A-Z</a><br>*** <a href="MessageBoard.php?action=Delete%20First">  
    Delete First Message</a><br>  
    <a href="MessageBoard.php?action=Delete%20Last">Delete Last   
    Message</a><br> <a href="MessageBoard.php?action=  
    Remove%20Duplicates">Remove Duplicates</a> </p>Give this a server/browser test.
2. Add the following ***case*** to the ***switch*** statement. The ***sort()*** function will sort the messages ascending:  
    ***case 'Sort Ascending':  
    sort($messageArray);  
    break;***Give this a server/browser test.

**Exercise 02\_04\_01 – Step 12**



1. Reopen the file ***MessageBoard.php*** with your IDE. Add the following ***hyperlink*** to the ending ***<p>*** element, below the first ***sort*** hyperlink:  
    <p>  
    <a href="PostMessage.php">Post New Message</a><br> <a href="MessageBoard.php?action=Sort%20Ascending">  
    Sort Subjects A-Z</a><br>  
    ***<a href="MessageBoard.php?action=Sort%20Descending">  
    Sort Subjects Z-A</a><br>*** <a href="MessageBoard.php?action=Delete%20First">  
    Delete First Message</a><br>  
    <a href="MessageBoard.php?action=Delete%20Last">Delete Last   
    Message</a><br> <a href="MessageBoard.php?action=  
    Remove%20Duplicates">Remove Duplicates</a> </p>Give this a server/browser test.
2. Add the following ***case*** to the ***switch*** statement. The ***rsort()*** function will sort the messages descending:  
    ***case 'Sort Descending':  
    rsort($messageArray);  
    break;***Give this a server/browser test.