**Exercise 02\_03\_01 – Step 1**

In this Exercise, we will learn how to manipulate strings and use regular expressions.



1. Create a folder named Exercise 02\_03\_01 and open it with your IDE. Create a new file called ***MusicalScale.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 ***Musical Scale***:  
   ***<!DOCTYPE html>  
   <html lang="en">  
   <head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width">  
    <title> Musical Scale </title>  
    <script src="modernizr.custom.65897.js"></script>  
   </head>  
   <body>  
   </body>  
   </html>***
2. In the <body>, create an ***<h2>*** element with content ***Musical Scale***. Create a set of PHP standard script delimiters.  
    ***<h2> Musical Scale </h2>  
    <?php  
     
    ?>***Copy the project folder into the appropriate spot on your Web Server and test it.
3. Insert the following ***array*** declaration and initialization into the script section:  
    <?php  
    ***$musicalScale = array("do", "re", "me", "fa", "so", "la",   
    "ti");*** ?>
4. Build an output string from the array using a foreach statement:  
    ***$outputString = "The notes of the musical scale are: ";  
    foreach ($musicalScale as $currentNote) {  
    $outputString .= " " . $currentNote;  
    }***  
   Give this a server test for syntax.
5. Add an echo statement to display the results:  
    ***function returnMessage() {  
    return "<p>This message was displayed from within a   
    function.</p>";  
    }***  
   Give this a server test.

**Exercise 02\_03\_01 – Step 2**



1. Copy the file ***MusicalScale.php*** to a file named ***FormattedText.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Formatted Text.*** In the <body>, create an ***<h2>*** element with content ***Formatted Text***. Create a set of PHP standard script delimiters.  
    ***<h2> Formatted Text </h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Declare and initialize a variable called ***$displayValue***:  
    ***$displayValue = 9.876;***
3. Add the following PHP code to display some ***unformatted*** text. Notice the use of the ***<pre>*** tags to tell the interpreter not to convert whitespace to spaces: ***echo "<pre>\n";  
    echo "Unformatted text line 1. ";  
    echo "Unformatted text line 2. ";  
    echo "$displayValue = $displayValue";  
    echo "</pre>\n";***Give this a server/browser test. Notice the lack of formatting and the variable names are expanded to values.
4. Add the following PHP code to display some ***formatted*** text:  
    echo "<pre>\n";  
    echo "Formatted text line 1. ***\r\n***";  
    echo "***\t***Formatted text line 2. ***\r\n***";  
    echo "**\$**displayValue = $displayValue";  
    echo "</pre>\n";  
   Give this a server/browser test. Notice the formatting and that escaped variable names are not expanded to values.

**Exercise 02\_03\_01 – Step 3**



1. Copy the file ***MusicalScale.php*** to a file named ***BooksAndAuthors.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Books And Authors.*** In the <body>, create an ***<h2>*** element with content ***Books And Authors***. Create a set of PHP standard script delimiters.  
    ***<h2>Books And Authors</h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Declare and initialize an ***array*** called ***$books*** as follows:  
    ***$books = array("the Adventures of Huckleberry Finn",   
    "Nineteen Eighty-Four", "Alice's Adventures in   
    Wonderland", "The Cat in the Hat");***
3. Declare and initialize an ***array*** called ***$realNames*** as follows:  
    ***$authors = array("Mark Twain", "George Orwell", "Lewis   
    Carroll", "Dr. Seuss");***
4. Declare and initialize an ***array*** called ***$authors*** as follows:  
    ***$realNames = array("Samuel Clemens", "Eric Blair",   
    "Charles Dodson", "Theodor Geisel");***Give this a server/browser test for syntax.
5. Create a ***for*** loop to display a string that combines the values of the three arrays. Notice the ***complex string syntax*** to ensure the elements are handled correctly:  
    ***for ($i = 0; $i < count($books); $i++) {  
    echo "<p>The real name of {$authors[$i]}, " . "the author of   
    \"{$books[$i]}\", " . "is {$realNames[$i]}.</p>";  
    }***  
   Give this a server/browser test.

**Exercise 02\_03\_01 – Step 4**



1. Copy the file ***BooksAndAuthors.php*** to a file named ***BooksAndAuthors2.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Books And Authors 2.*** In the <body>, change the ***<h2>*** element content to ***Books And Authors 2***. Delete the ***$authors and $realNames*** arrays***.***
2. Modify the ***for*** loop to display the information about the book titles as follows:  
    ***for ($i = 0; $i < count($books); $i++) {  
    echo "<p>The title \"{$books[$i]}\" contains " .   
    strlen($books[$i]) . " characters and " .   
    str\_word\_count($books[$i]) . " words.</p>";  
    }***Give this a server/browser test.

**Exercise 02\_03\_01 – Step 5**



1. Copy the file ***MusicalScale.php*** to a file named ***WordPlay.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Word Play.*** In the <body>, create an ***<h2>*** element with content ***Word Play***. Create a set of PHP standard script delimiters.  
    ***<h2>Word Play</h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Declare and initialize a ***string*** called ***$startingText*** as follows:  
    ***$startingText = "mAdAm, i'M aDaM.";***
3. Add the following code to convert the string and display it with different mixes of uppercase and lowercase:  
    ***$uppercaseText = strtoupper($startingText);  
    $lowercaseText = strtolower($startingText);  
    echo "<p>$uppercaseText</p>\n";  
    echo "<p>$lowercaseText</p>\n";***Give this a server/browser test.
4. Add the following code to convert the string and display it in uppercase and lowercase:  
    ***echo "<p>" . ucfirst($lowercaseText) . "</p>\n";  
    echo "<p>" . lcfirst($uppercaseText) . "</p>\n";  
    $workingText = ucwords($lowercaseText);  
    echo "<p>$workingText</p>\n";***Give this a server/browser test.

**Exercise 02\_03\_01 – Step 6**



1. Copy the file ***WordPlay.php*** to a file named ***WordPlay2.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Word Play 2.*** In the <body>, create an ***<h2>*** element with content ***Word Play 2***. Create a set of PHP standard script delimiters.  
    ***<h2>Word Play 2</h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Add the following code before the end of the PHP block:  
    ***echo "<p>" . md5($workingText) . "</p>\n";  
    echo "<p>" . substr($workingText, 0, 6) . "</p>\n";  
    echo "<p>" . substr($workingText, 7) . "</p>\n";  
    echo "<p>" . strrev($workingText) . "</p>\n";  
    echo "<p>" . str\_shuffle($workingText) . "</p>\n";***Give this a server/browser test.

**Exercise 02\_03\_01 – Step 7**



1. Copy the file ***WordPlay.php*** to a file named ***PHPEmail.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***PHP Email.*** In the <body>, create an ***<h2>*** element with content ***PHP Email***. Create a set of PHP standard script delimiters.  
    ***<h2>PHP Email</h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Declare and populate an array named ***$emailAddresses***:  
    ***$emailAddresses = array("john.smith@php.test",  
    "mary.smith.mail@php.example",  
    "john.jones@php.invalid",  
    "alan.smithee@test",  
    "jsmith456@example.com",  
    "jsmith456@test",  
    "mjones@example",  
    "mjones@example.net",  
    "jane.a.doe@example.org");***Give this a server/browser test for syntax.
3. After the array declaration, add the following function which make use of the ***strpos()*** function:  
    ***function validateAddress($address) {  
    if (strpos($address, '@') !== false && strpos($address, '.') !==   
    false) {  
    return true;  
    }  
    else {  
    return false;  
    }  
    }***  
   Give this a server/browser test.
4. After the function, add a foreach loop which will test the validity of all of the emails in the array:  
    ***foreach ($emailAddresses as $address) {  
    if (validateAddress($address) === false) {  
    echo "<p>The e-mail address <em>$address</em>   
    does not appear to be valid.</p>";  
    }***Give this a server/browser test.

**Exercise 02\_03\_01 – Step 8**



1. Copy the file ***WordPlay.php*** to a file named ***Presidents.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Presidents.*** In the <body>, create an ***<h2>*** element with content ***Presidents***. Create a set of PHP standard script delimiters.  
    ***<h2>Presidents</h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Declare and populate an array named ***$Presidents***:  
    ***$presidents = array("George Washington",*** ***"John Adams",  
    "Thomas Jefferson",  
    "James Madison",  
    "James Monroe");***Give this a server/browser test for syntax.
3. After the first array, declare and populate another array named ***$yearsInOffice***:  
    ***$yearsInOffice = array("1789 to 1797",  
    "1797 to 1801",  
    "1801 to 1809",  
    "1809 to 1817",  
    "1817 to 1825");***Give this a server/browser test for syntax.
4. Declare a ***template*** string for the output. The purpose of this string is to provide placeholders that will be replaced by the real data as processing occurs:  
    ***$outputTemplate = "<p>President [NAME] served from   
    [TERM]</p>\n";***  
   Give this a server/browser test.
5. After the function, add a ***foreach*** loop which will use the template string to start to create an output string:  
    ***foreach ($presidents as $sequence => $name) {  
    $tempString = str\_replace("[NAME]", $name,   
    $outputTemplate);  
    echo $tempString;  
    }***Give this a server/browser test.
6. Complete the output string in the ***foreach*** loop:  
    ***foreach ($presidents as $sequence => $name) {  
    $tempString = str\_replace("[NAME]", $name,   
    $outputTemplate);  
    $outputString = str\_replace("[TERM]",   
    $yearsInOffice[$sequence], $tempString);  
    echo $outputString;  
    }***Give this a server/browser test.

**Exercise 02\_03\_01 – Step 9**



1. Copy the file ***Presidents.php*** to a file named ***Presidents2.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Presidents 2.*** In the <body>, create an ***<h2>*** element with content ***Presidents 2***. Create a set of PHP standard script delimiters.  
    ***<h2>Presidents 2</h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Create a string of presidents containing semicolons as delimiters named ***$Presidents***:  
    ***$presidents = "George Washington;John Adams;Thomas   
    Jefferson;James Madison;James Monroe";***Give this a server/browser test for syntax.
3. Make the first ***strtok()*** call, using both the ***string*** and a ***single*** ***delimiter*** string as parameters, to return the first token in the string:  
    ***$thisPresident = strtok($presidents, ";");***Give this a server/browser test for syntax.
4. Build a ***while*** loop to output this token, and continue getting tokens without using the string parameter, until there are no more tokens:  
    ***while ($thisPresident != NULL) {  
    echo "$thisPresident<br>";  
    $thisPresident = strtok(";");  
    }***  
   Give this a server/browser test.
5. Let’s change this up to use a ***multiple*** ***delimiter*** string. Copy all of the above code starting with the first ***strtok()*** call to below itself. Then just change the delimiter string to contain a ***semicolon*** followed by a ***space***.  
    ***$thisPresident = strtok($presidents, "; ");  
    while ($thisPresident != NULL) {  
    echo "$thisPresident<br>";  
    $thisPresident = strtok("; ");  
    }***Give this a server/browser test.

**Exercise 02\_03\_01 – Step 10**



1. Copy the file ***Presidents.php*** to a file named ***PasswordFields.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Password Fields.*** In the <body>, create an ***<h2>*** element with content ***Password Fields***. Create a set of PHP standard script delimiters.  
    ***<h2>Password Fields</h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Declare and initialize an array named ***$passwordFields*** as follows:  
    ***$passwordFields = array("login name",  
    "optional encrypted password",  
    "numerical user ID",  
    "numerical group ID",  
    "user name or comment field",  
    "user home directory",  
    "optional user command interpreter");***Give this a server/browser test for syntax.
3. Enter the following code to ***tokenize*** the string and display the output:  
    ***$fieldIndex = 0;*** ***$extraFields = 0;  
    $currField = strtok($record, ":");  
    while ($currField != NULL) {  
    if ($fieldIndex < count($passwordFields)) {  
    echo "<p>The {$passwordFields[$fieldIndex]} is   
    <em>$currField</em></p>\n";  
    }  
    else {  
    ++$extraFields;  
    echo "<p>Extra field # $extraFields is   
    <em>$currField</em></p>\n";  
    }  
    $currField = strtok(":");  
    ++$fieldIndex;  
    }***Give this a server/browser test for syntax.

**Exercise 02\_03\_01 – Step 11**



1. Copy the file ***PasswordFields.php*** to a file named ***PasswordFields2.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***Password Fields 2.*** In the <body>, create an ***<h2>*** element with content ***Password Fields 2***. Create a set of PHP standard script delimiters.  
    ***<h2>Password Fields 2</h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Replace the ***$currField*** declaration with a variable named ***$fields*** with an ***explode()*** function call::  
    ***$fields = explode(":", $record);***
3. Replace the ***while*** loop with a ***foreach*** loop, removing the ***$currField*** assignment and the ***$fieldIndex increment***:  
    ***foreach ($fields as $fieldIndex => $fieldValue) {  
    if ($fieldIndex < count($passwordFields)) {  
    echo "<p>The {$passwordFields[$fieldIndex]} is   
    <em>$fieldValue</em></p>\n";  
    }  
    else {  
    ++$extraFields;  
    echo "<p>Extra field # $extraFields is   
    <em>$fieldValue</em></p>\n";  
    }  
    }***Give this a server/browser test for syntax.

**Exercise 02\_03\_01 – Step 12**



1. Copy the file ***PHPEmail.php*** to a file named ***PHPEmail2.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***PHP Email 2.*** In the <body>, create an ***<h2>*** element with content ***Password Fields 2***. Create a set of PHP standard script delimiters.  
    ***<h2> PHP Email 2</h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Add the following function directly after the ***validateAddress()*** function. It will use nested for loops to sort the email address array, using the ***comparison*** operator:  
    ***function sortAddresses($addresses) {  
    $sortedAddresses = array();  
    $ilimit = count($addresses)-1;  
    $jlimit = count($addresses);  
    for ($i = 0; $i < $ilimit; $i++) {  
    $currentAddress = $addresses[$i];  
    for ($j = $i + 1; $j < $jlimit; $j++) {  
    if ($currentAddress > $addresses[$j]) {  
    $tempVal = $addresses[$j];  
    $addresses[$j] = $currentAddress;  
    $currentAddress = $tempVal;  
    }  
    }  
    $sortedAddresses[] = $currentAddress;  
    }  
    return($sortedAddresses);  
    }***Give this a server/browser test for syntax.
3. Add the following code directly below the new function. Notice the implode() function, which takes the sorted array and converts it into a string, using a ***comma*** and a ***space*** to delimit the tokens:  
    ***$sortedAddresses = sortAddresses($emailAddresses);  
    $sortedAddressList = implode(", ", $sortedAddresses);  
    echo "<p>Sorted Addresses: $sortedAddressList</p>\n";***Give this a server/browser test.
4. Change the ***foreach*** loop to use ***$sortedAddresses*** instead of ***$emailAddresses***:  
    foreach (***$sortedAddresses*** as $address) {  
   Give this a server/browser test.

**Exercise 02\_03\_01 – Step 13**



1. Copy the file ***PHPEmail2.php*** to a file named ***PHPEmail3.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***PHP Email 3.*** In the <body>, create an ***<h2>*** element with content ***Password Fields 3***. Create a set of PHP standard script delimiters.  
    ***<h2> PHP Email 3</h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Modify the ***if*** statement in the ***sortAddresses()*** function to use the ***strcasecmp()*** function instead of the ***comparison*** operator as follows:  
   ***if (strcasecmp($currentAddress, $addresses[$j]) > 0) {***Give this a server/browser test.

**Exercise 02\_03\_01 – Step 14**



1. Copy the file ***PHPEmail3.php*** to a file named ***PHPEmail4.php***, and open it with your IDE. Update the standard opening documentation in the ***<head>*** element. Set the <title> content to ***PHP Email 4.*** In the <body>, create an ***<h2>*** element with content ***Password Fields 4***. Create a set of PHP standard script delimiters.  
    ***<h2> PHP Email 4</h2>  
    <?php  
     
    ?>***Give this a server/browser test.
2. Modify the ***if*** statement in the ***validateAddress()*** function to use the ***preg\_match()*** Regex function instead of the ***strpos()*** function as follows:  
    ***if (preg\_match("/^[\w-]+(\.[\w-]+)\*@[\w-]+(\.[w-]+)\*(\.[A-Za-z]{2,})$/i" ,$address) == 1) {***Give this a server/browser test.