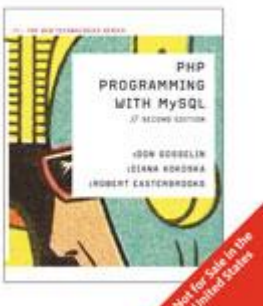




Institiúid Teicneolaíochta Chorcaí
Cork Institute of Technology

SOFT7008

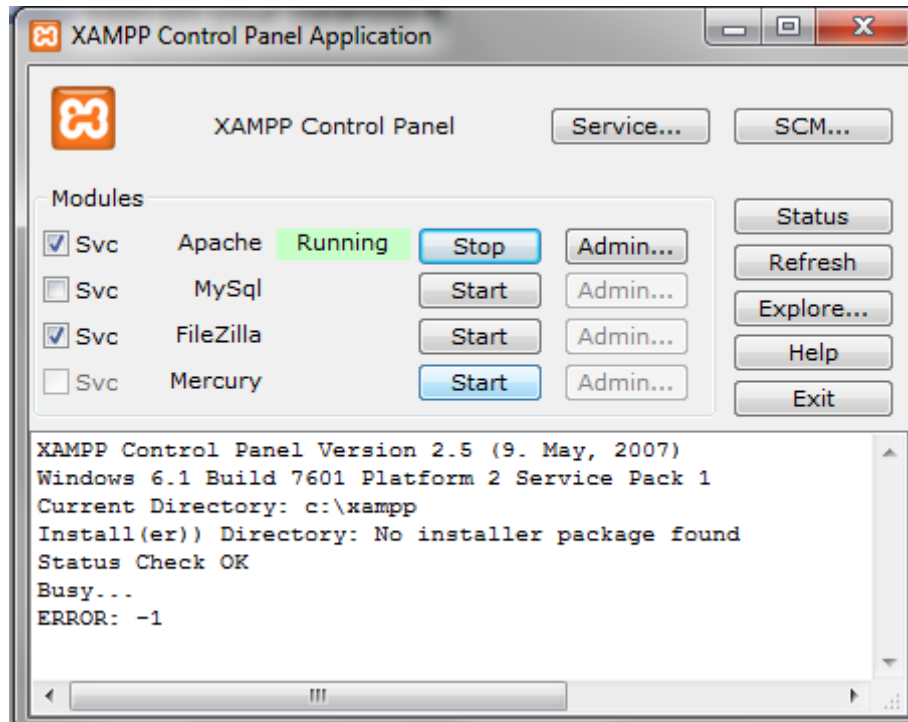
Lab03 – String and Form Processing

	<p>PHP Programming with MySQL</p> <p>AUTHORS: Gosselin/Kokoska/Easterbrooks - ©2011 ISBN10: 0-538-46814-9 ISBN13: 978-0-538-46814-5</p>
---	---

How to run your PHP programs:



1. Double click the XAMPP icon on your desktop. The XAMPP control panel will be displayed:



2. Click the **Start** button beside the **Apache** module. The status should display **Running**.
3. Edit and save your PHP code file.
4. Copy your PHP code file to the `\xampp\htdocs` folder.
5. Launch your browser.
6. Enter `localhost/myCode.php` into the address bar to execute the program.

PHP String Procssing

In this example you will perform basic validation and sorting on a list of email addresses.

1. Open Notepad++ and create an HTML file with embedded PHP as follows:

```
<!DOCTYPE html>
<head>
<title>E-Mail Validator</title>
<meta charset="utf-8" />
</head>
<body>
<?php
$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");

function validateAddress($Address)
{
    // check if $Address argument contains both and @ and . character
    if (strpos($Address, '@') !== FALSE && strpos($Address, '.') !== FALSE)
        return true;
    else
        return false;
}
```

```

function sortAddresses($Addresses)
{
    $SortedAddresses = array();
    // Set the upper limit for the outer loop
    $iLimit = count($Addresses)-1;
    // Set the upper limit for the inner loop
    $jLimit = count($Addresses);
    for ($i = 0; $i<$iLimit; ++$i)
    {
        $CurrentAddress = $Addresses[$i];
        for ($j = $i+1; $j<$jLimit; ++$j)
        {
            if (strcasecmp($CurrentAddress,$Addresses[$j]) > 0)
            {
                // swap
                $TempVal = $Addresses[$j];
                $Addresses[$j] = $CurrentAddress;
                $CurrentAddress = $TempVal;
            }
        }
        $SortedAddresses[] = $CurrentAddress;
    }
    return($SortedAddresses);
}

// sort array of email addresses in alphabetical order
$SortedAddresses = sortAddresses($EmailAddresses);
// convert $SortedAddresses array to a , delimited list string
$SortedAddressList = implode(", ", $SortedAddresses);
echo "<p>Sorted Addresses: $SortedAddressList</p>\n";

foreach ($SortedAddresses as $Address) {
    if (validateAddress($Address) == false)
        echo "<p>The e-mail address <em>$Address</em> does not appear to be
        valid.</p>\n";
}

?>
</body>
</html>

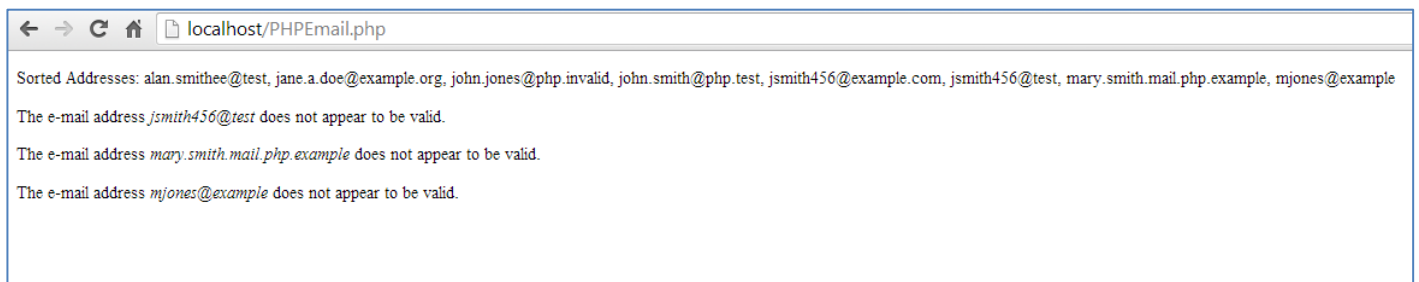
```

2. Save the file as `PHPEmail.php` Copy the file into your **XAMP\htdocs** folder.

To run your program:

3. Open your default browser.
4. Enter `http://localhost/PHPEmail.php` into the address bar and hit Enter.

The browser should display:



PHP String Validation

In this example you will perform remove dashes and spaces from a credit card string and check it is numeric.

1. Open Notepad++ and create an HTML file with embedded PHP as follows:

```
<!DOCTYPE>
<html>
<head>
<title>Validate Credit Card</title>
<meta charset="utf-8" />
</head>
<body>
<h1>Validate Credit Card</h1><hr />
<?php
$CreditCard = array(
    "",
    "8910-1234-5678-6543",
    "0000-9123-4567-0123");
foreach ($CreditCard as $CardNumber)
{
    if (empty($CardNumber))
        echo "<p>This Credit Card Number is invalid because it
            contains an empty string.</p>";
    else
    {
        // store current iteration credit card number in $CreditCardNumber
        $CreditCardNumber = $CardNumber;
        // remove any dash characters
        $CreditCardNumber = str_replace("-", "", $CreditCardNumber);
        // remove any spaces
        $CreditCardNumber = str_replace(" ", "", $CreditCardNumber);
        if (!is_numeric($CreditCardNumber))
            echo "<p>Credit Card Number " .
                $CreditCardNumber .
                " is not a valid Credit " .
                " Card number because it contains " .
                " a non-numeric character.</p>";
        else
            echo "<p>Credit Card Number " .
                $CreditCardNumber .
                " is a valid Credit Card number.</p>";
    }
}
?>
</body>
</html>
```

2. Save the file as ValidateCreditCard.php.. Copy the file into your XAMP\htdocs folder.

To run your program:

3. Open your default browser.
4. Enter `http://localhost/ValidateCreditCard.php` into the address bar and hit Enter.

PHP Form Processing

1. Create this HTML form in Notepad++.

```
<!DOCTYPE html PUBLIC>
<head>
<title>Scholarship Form</title>
<meta charset="utf-8" />
</head>
<body>
<h2 style = "text-align:center">Scholarship Form</h2>
<form name = "scholarship" action = "process_Scholarship.php" method = "post">
<p>First Name: <input type="text" name="fName" /></p>
<p>Last Name: <input type="text" name="lName" /></p>
<p><input type="reset" value="Clear Form" />&nbsp;&nbsp;&nbsp;<input type="submit" name="Submit" value="Send Form" />
</form>
</body>
</html>
```

2. Save it as scholarship.html.
3. Copy the file into your **XAMP\htdocs** folder.
4. Create a new file in Notepad++ and enter the following code:

```
<!DOCTYPE html>
<head>
<title>Scholarship Form</title>
<meta charset="utf-8" />
</head>
<body>
<?php
$firstName = htmlspecialchars($_POST['fName'], ENT_QUOTES);
$lastName = htmlspecialchars($_POST['lName'], ENT_QUOTES);
echo "Thank you for filling out the scholarship form,
 ".$firstName." ".$lastName . ".";
?>
</body>
</html>
```

5. Save the file as process_Scholarship.php

To run your program:

5. Open your default browser.
6. Enter `http://localhost/scholarship.html` into the address bar and hit Enter.
7. Enter your name in the form fields. Hit **Send Form**.

Your PHP script should display a personalised Thank You message.

PHP Form Validation

In this example you will add some simple form validation processing along with a message to the user indicating that field(s) require data.

1. Modify the `process_Scholarship.php` code file from the previous exercise in Notepad++ as follows:

```
<?php
function displayRequired($fieldName) {
    echo "The field \"\$fieldName\" is required.<br />\n";
}
function validateInput($data, $fieldName)
{
    global $errorCount;
    if (empty($data)) {
        displayRequired($fieldName);
        ++$errorCount;
        $retval = "";
    } else { // Only clean up the input if it isn't empty
        $retval = trim($data);
        $retval = htmlspecialchars($retval, ENT_QUOTES);
    }
    return($retval);
}
$errorCount = 0;
$firstName = validateInput($_POST['fName'], "First name");
$lastName = validateInput($_POST['lName'], "Last name");
```

```

if ($errorCount>0)
    echo "Please use the \"Back\" button to re-enter the
        data.<br />\n";
else
{
    echo "Thank you for filling out the scholarship form,
        ".$firstName." ".$lastName . ".";
}
?>
</body>
</html>

```

6. Save the file as `process_Scholarship.php`

To run your program:

8. Open your default browser.
9. Enter `http://localhost/scholarship.html` into the address bar and hit Enter.
10. Enter your first name only in the appropriate form field. Hit **Send Form**.

The browser should display a message indicating that the **Last name** field is required.

PHP Sticky Form

In this example you will create a 'sticky' form.

1. Modify the `process_Scholarship.php` code file from the previous exercise in Notepad++.
2. Insert the `redisplayForm ()` function after the `validateInput ()` function.

```
function redisplayForm($firstName, $lastName)
{
    ?> <!--advanced escape from HTML-->
    <!--Sticky Form Defintion-->
    <h2 style = "text-align:center">Scholarship Form</h2>
    <form name="scholarship" action="process_Scholarship.php"
        method = "post">
        <p>First Name: <input type="text" name="fName" value="<?php
            echo $firstName; ?>" /></p>
        <p>Last Name: <input type="text" name="lName" value="<?php
            echo $lastName; ?>" /></p>
        <p><input type="reset" value="Clear Form" />&nbsp;&nbsp;&nbsp;
            &nbsp;&nbsp;&nbsp;<input type="submit" name="Submit" value="Send Form" />
    </form>
```

3. Modify code below the form definition as follows:

```
<?php
}

$errorCount = 0;
$firstName = validateInput($_POST['fName'], "First name");
$lastName = validateInput($_POST['lName'], "Last name");

if ($errorCount>0)
{
    echo "Please re-enter the information below.<br />\n";
    redisplayForm($firstName, $lastName);
}
else
{
    echo "Thank you for filling out the scholarship form,
        ".$firstName." ".$lastName." . ".";
}
?>
</body>
</html>
```

7. Save the file as `process_Scholarship.php`

To run your program:

11. Open your default browser.
12. Enter `http://localhost/scholarship.html` into the address bar and hit Enter.
13. Enter your first name only in the appropriate form field. Hit **Send Form**.

The browser should display a message to re-enter the data below and display the form with your previously entered first name.

