

## PHP PART 1: INTRODUCTION TO SERVER-SIDE PROGRAMMING WITH PHP

**Note:** This exercise **cover lecture eight**.

## AIMS:

- To develop an understanding of the basic use of **variables, arrays and expressions** in PHP
- To be able to use various control structures and develop your own functions.
- Use PHP predefined ‘superglobal’ variables to get data from a form.
- Gain some of the knowledge and skills needed to complete Assignments.

Note: PHP script generates HTML on the **server**. You **must** place your PHP files to localhost i.e. xampp so they can be processed by the PHP engine that is attached to the HTTP server.

## TASK 1: USING VARIABLES, ARRAYS AND IF STATEMENTS

What we going to do:

- create an **array** and initialise it with values
- use an **if** statement to process the values
- use PHP to generate some HTML

**Step1: Create a PHP script**

Create a file **myfirst.php** with a PHP script. The script does the following:

1. Declares and initialises an **array** named **\$marks[]** and with three integer elements 85, 85 and 95.
2. Modifies the value of the second element to contain 90.
3. Computes the average score of the values from the three elements and stores the result in **\$ave**.
4. Uses an “**if**” statement to assign a value of “**PASSED**” to variable **\$status** if the average is at least 50, otherwise assigns a value of “**FAILED**”.
5. Uses output statements to display “**The average score is** ” along with the averaged value and “ **You** ” followed by the status.
6. Use any text editor on your local computer (e.g. Notepad++) and **code** the following: (**don't copy and paste!!!**)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8"/>
    <title>Using PHP variables, arrays and operators</title>
</head>
<body>
<h1>Creating Web Applications - Lab 8</h1>
<?php
    $marks = array (85, 85, 95);    // declare and initialise array
    $marks[1] = 90;                // modify second element
    $ave = ($marks[0] + $marks[1] + $marks[2]) / 3;    // compute average
    if($ave >= 50)                  // check status
        $status = "PASSED";
    else
        $status = "FAILED";
```

```
    echo "<p>The average score is $ave. You $status.</p>";  
?>  
</body>  
</html>
```

**Step 2: Test**

Test in the browser, and check that the page is valid HTML5.

1. Open the page in browser.
2. View the page source code
3. Copy the source codes
4. Go to W3C HTML validator site and choose "validate by direct input"
5. Paste the source code and check.

*Remember that PHP only works on a server - you can no longer validate your HTML from the local drive.*



---

**TASK 2: EXPERIMENTING ON ARRAYS**

In this task, we will apply the approach covered in Task 1 to a similar problem.

**Step1: Create the PHP script**

Create a file `daysarray.php` with a PHP script that declares and initialises an **array** named `$days[]` and with the days of the week Sunday, Monday, etc.

Use output statements to display “The Days of the week in English are:” along with the values in the `$days[]` array.

**Step 2: Load and test on the server**

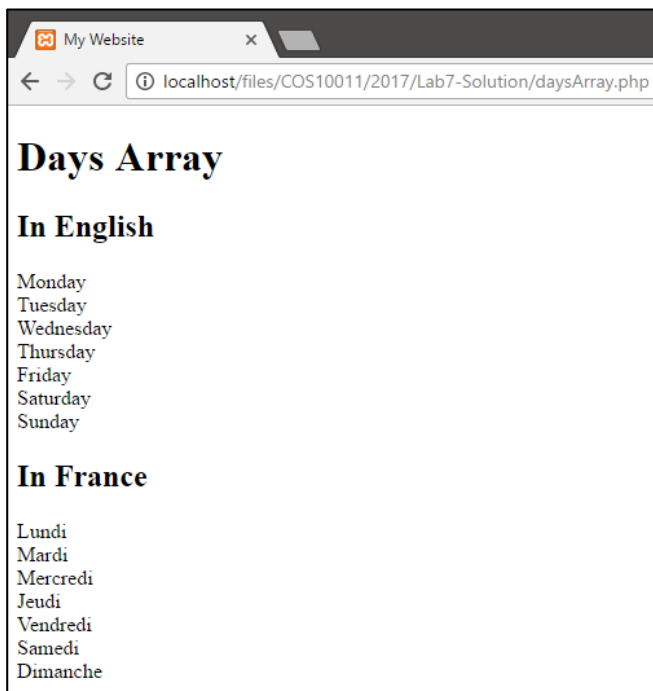
Copy to the server, test in the browser, and check that the page is valid HTML5.

**Step 3: Change the script**

Reassign the values in the `$days[]` array with the days of the week in French, Sunday is *Dimanche*, Monday is *Lundi*, Tuesday is *Mardi*, Wednesday is *Mercredi*, Thursday is *Jeudi*, Friday is *Vendredi*, and Saturday is *Samedi*.

**Step 4: Load and test again**

Re-save the document as `daysArray.php`, copy to the server, test in the browser, and again check that the page is valid HTML5.

**Step 5: Task Submission**

1. Only submit **Task1** and **Task2**. **Task3** and **Task4** is for your own reading.

---

**TASK3: FUNTIONS, GET, AND IF STATEMENTS (FOR YOUR OWN READING)**

In this task we will apply

- User-defined functions
- In-built functions
- `if` selection statements and loops

---

**CREATING PHP FUNCTIONS EXAMPLE**

```
<?php
/* Defining a PHP Function */
function writeMessage() {
    echo "You are really a nice person, Have a nice time!";
}

/* Calling a PHP Function */
writeMessage();
?>
```

**Result:**



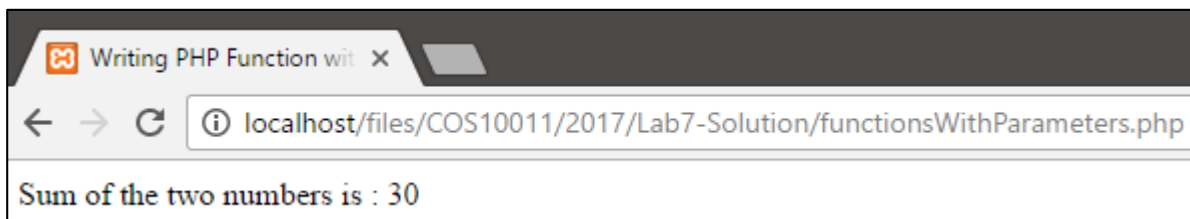
---

**PHP FUNCTIONS WITH PARAMETERS EXAMPLE**

```
<?php
function addFunction($num1, $num2) {
    $sum = $num1 + $num2;
    echo "Sum of the two numbers is : $sum";
}

addFunction(10, 20);
?>
```

**Result:**

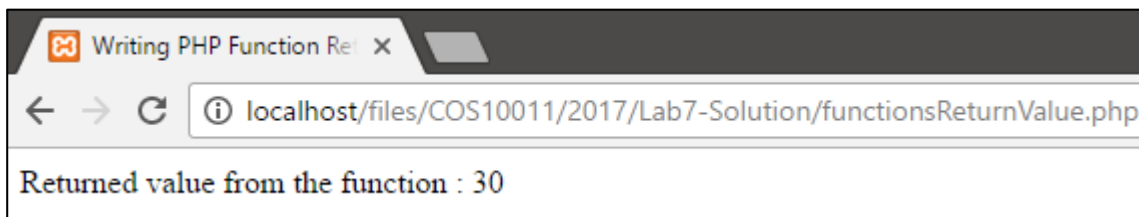


## PHP FUNCTIONS RETURNING VALUE EXAMPLE

```
<?php
function addFunction($num1, $num2) {
    $sum = $num1 + $num2;
    return $sum;
}
$return_value = addFunction(10, 20);

echo "Returned value from the function : $return_value";
?>
```

Result:



## GET VALUE FROM A FORM

webpage.html

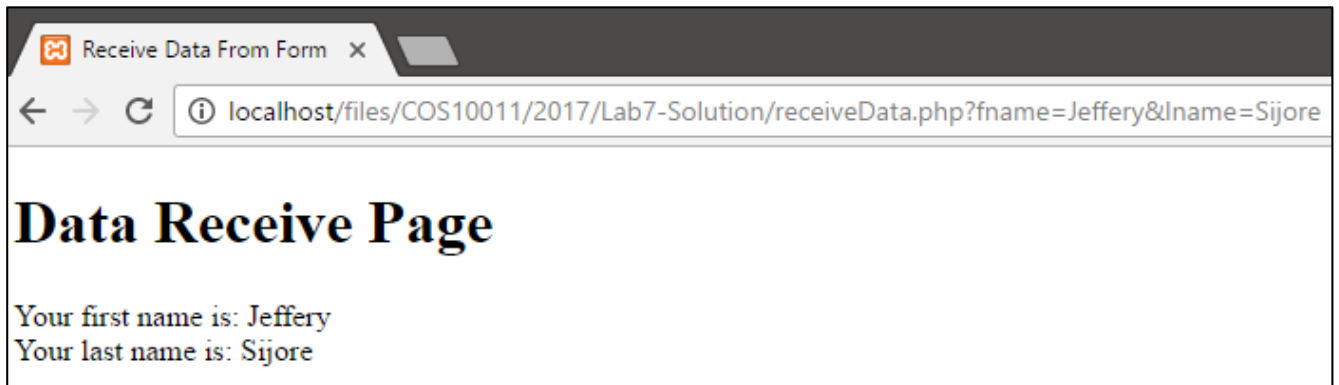
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <title>Simple Form</title>
</head>
<body>
<!-- form creation -->
    <form method="get" action="receiveData.php">
        <label>Enter your first name: </label><input type="text" name="fname" />
        <br />
        <label>Enter your last name: </label><input type="text" name="lname" />
        <br />
        <input type="submit" value="Submit" />
        <input type="reset" value="reset" />
    </form>
</body>
</html>
```

receiveData.php

```
<?php
//set the variable and use GET method to get the value
$firstname = $_GET["fname"];
$lastname = $_GET["lname"];

// print the value
echo "Your first name is: " . $firstname. "<br />";
echo "Your last name is: " . $lastname. "<br />";
?>
```

Result:



PHP IF ELSE STATEMENT EXAMPLE

```
<?php
// Alphabetical Comparison
$a="brian";
$b="zebra";
if ($a < $b){
    echo $a." is before ".$b." in the alphabet";
}
else{
    echo $a." is after ".$b." in the alphabet";
}
// Result : brian is before zebra in the alphabet
?>
```

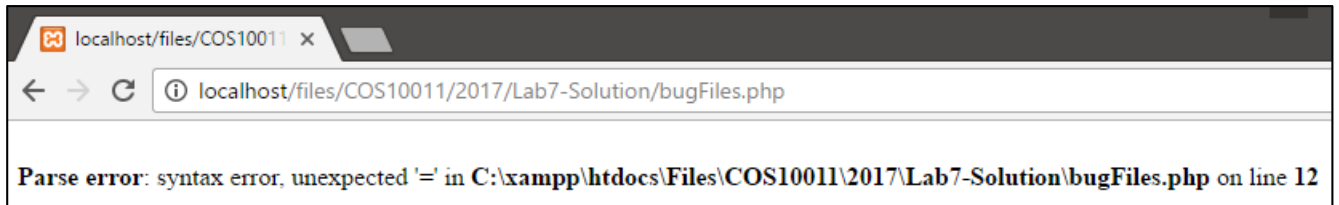
Result:



---

**TASK 4: DEBUGGING PHP (NO NEED TO SUBMIT)**

Open the files **bugFiles.php** in **xampp**. Once you open, you will see something like the picture below:



Your **task** is to **solve** the **problem**. Once you solve the problem, the **result** should be the same like the picture below:

