Lab 2 – Working with Data Types and Operators

Reference: Chapter 1 of the "PHP Programming with MySQL" textbook.

Aims:

- To develop an understanding of the basic use of *variables, arrays and expressions.*

Getting Started:

Create a new folder '**lab02**' under the unit folder on the mercury server *~/cos30020/www/htdocs* folder on mercury.
Save today's work in this lab02 folder.

You could create and link an external stylesheet, to the pages, and this should be valid CSS3.

Task 1: Using variables, arrays and operators (9 marks)

Step 1:
Create a file **module02.php** with a PHP script that declares and initialises an **array** named `$marks[]` and with the three integer elements 85, 85 and 95. Modify the value of the second element to contain 90, and compute for the average score of the values from the three elements and store the result in `$ave`. Finally, using the `?:` operator assign a value of "PASSED" to variable `$status` is the average is at least 50, otherwise assign a value of "FAILED".

Use output statements to display "The average score is " along with the averaged value and " You " followed by the status.

Step 2:
Use any text editor on your local computer (e.g. Notepad++) and code the following:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" >
    <meta name="description" content="Web Programming :: Lab 2" >
    <meta name="keywords" content="Web,programming" >
    <title>Using variables, arrays and operators</title>
</head>
<body>
    <h1>Web Programming - Lab 2</h1>
    <?php
      $marks = array (85, 85, 95);      // declare and initialise array
      $marks[1] = 90;                   // modify second element
      $ave = ($marks[0] + $marks[1] + $marks[2]) / 3;     // compute average
      ($ave >= 50)                      // checks status
          ? $status = "PASSED"
          : $status = "FAILED" ;

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

Step 3:
Test in the browser, and check that the page validates.

Task 2: Experimenting on arrays (3 marks)

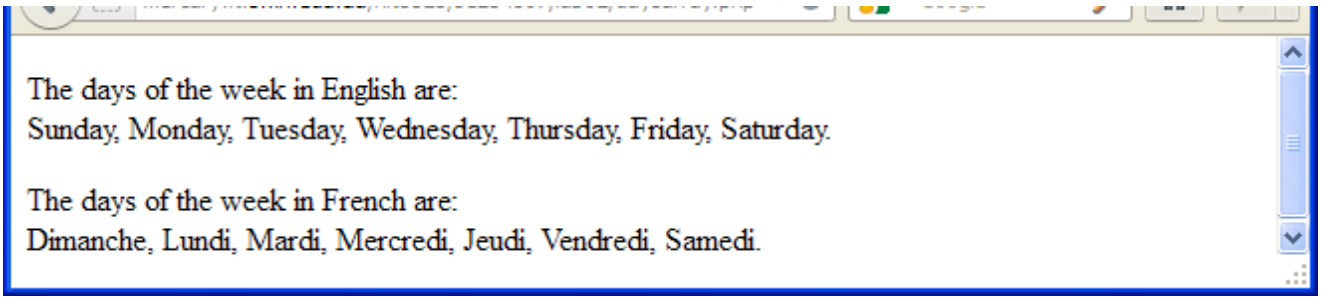
Step 1:
Create a file **`daysarray.php`** with a PHP script that declares and initialises an **array** named `$days[]` and with the days of the week Sunday, Monday, etc.

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

Step 2:
Test in the browser, and check that the page is valid.

Step 3:
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*.

Then use output statements to display "The days of the week in French are:" along with the French values in the `$days[]` array



Task 3: Using expression and looking up built-in functions (3 marks)

Step 1:
Create another file **`iseven.php`** with a script that declares a variable with a value.

Use a **conditional operator** to determine whether the variable contains a number **and** whether the number is even.

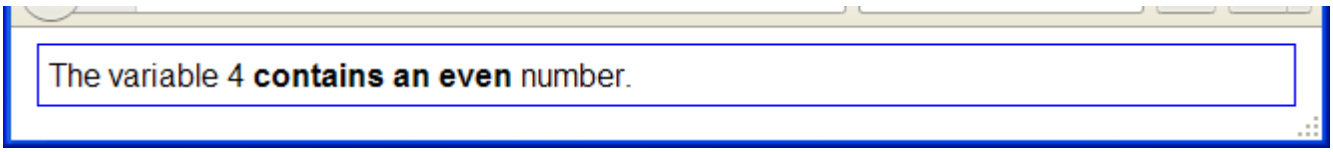
Set a message, and use an output statement to display the message.

Hint: Use the **`is_numeric()`** function to check whether the variable is a number.
For floating-point numbers, you need to use the **`round()`** function to convert the value to the nearest whole number.

Step 2:
Test the script by modifying the variable value, re-saving the script to the server each time, and refreshing the page in your browser.

Function Description and Examples

`is_numeric()`: http://php.net/manual/en/function.is-numeric.php
`round()`: http://php.net/manual/en/function.round.php



Extra Challenge:

Create a form in another page that passes the variable using `action="iseven.php"` and `method="get"`

Add code in `iseven.php` to receive the variable.

Hint: Use `$_GET[...]`

See ***Predefined Variables***, Superglobals and examples: http://php.net/manual/en/reserved.variables.php