Lab 2 – Working with Data Types and Operators

Task 1: Arrays (2 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.

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

Step 5:
Re-save the document as `daysarray.php`, test in the browser.

Example output:

The days of the week in English are:
Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday.

The days of the week in French are:
Dimanche, Lundi, Mardi, Mercredi, Jeudi, Vendredi, Samedi.

Task 2: 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 an integer
and whether the integer is even.

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

Step 2:
View in the browser.

Step 3:
Test the script by modifying the variable value, re-saving the script to the server each time, and refreshing the page in your browser.

Extra Challenge:

Create an XHTML 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

Task 3: Using if and while statements (2 marks)

Step 1:
Create a file **`mathfunctions.php`** to contain a function called `factorial` that accepts a positive integer and returns its factorial value. A factorial of a non-negative integer n, denoted by n!, is the product of all positive integers less than or equal to n. For example,

5! = 5 x 4 x 3 x 2 x 1 = 120

```
<?php
        function factorial ($n) {          // declare the factorial function
                $result = 1;                       // declare and initialise the result variable
                $factor = $n;                      // declare and initialise the factor variable
                while ($factor > 1) {  // loop to multiple all factors until 1
                  $result = $result * $factor;
                  $factor--;                       // next factor
                }                                  // Note that the factor 1 is not multiplied
                return $result;
        }
?>
```

Step 2:
Create a file **`factorial.php`** that will include the file **`mathfunctions.php`** in order to access the defined functions in the file. It should also receive an input from **`factorialform.php`** from Step 3 via GET method, and check if the input is a positive integer then output its factorial value. Otherwise, it should generate an appropriate error message.

Step 3:
Create a file **`factorialform.html`** that contains a form with a single text box that allows a user to enter a number, and submit it to **`factorial.php`**.

```
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title>Using if and while statements</title>
</head>
<body>
    <h1>Web Programming Form – Lab 3</h1>
    <form action = _____________ method = _____________ >
       ____________________________________
       ____________________________________
    </form>
</body></html>
```

Test in the browser.

Task 4: Using `if` statement (3 Marks)

Step 1:
Create a file **`leapyear.php`** with a script that tests if a variable value is a number (relevant function: `is_numeric()`, please check PHP manual for its usage), and if it is a leap year, and prints a message stating whether the year is a *standard year* or a *leap year*.

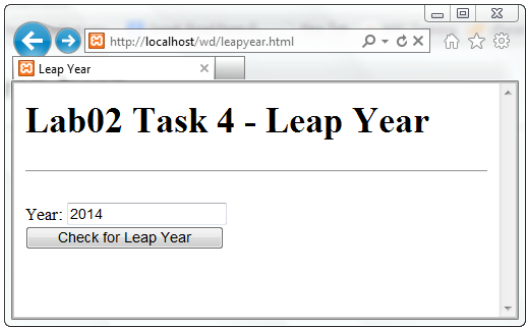
Algorithm: if the numerical value for a year is divisible by 4, it is a leap year. However, if the year is also divisible by 100 it is not a leap year, unless the year is also divisible by 400, in which case it is a leap year (http://stackoverflow.com/questions/725098/leap-year-calculation).

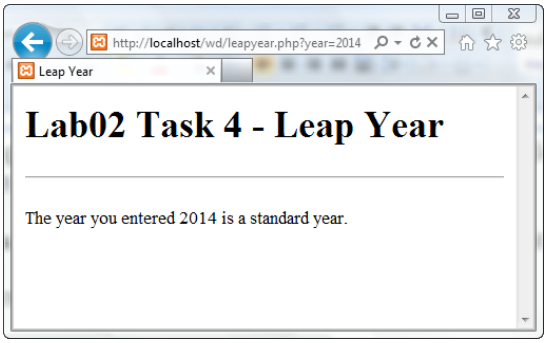
Test in the browser.

Step 2:
Create a file **`leapyearform.html`** that contains a form with a single text box that allows a user to enter a year, and submit it to **`leapyear.php`**.

Change **`leapyear.php`** to receive the year entered and determine if it is a leap year.

Test in the browser, and check that the pages are valid Strict XHTML.





Step 3:
Modify the script in **`leapyear.php`** to contain a function **`is_leapyear`** that accepts a single parameter representing the year. The function returns true if the year is a leap year otherwise false.

Test in the browser.