Unit 3: Basic PHP programming

In this practical session, we will practise basic PHP scripting by writing codes about string, array, external files, and functions. You will also need to explore how to do some file I/O operations yourself. There are total six tasks which will help you to understand the PHP scripting syntax. You can refer to unit 3 on Blackboard and of course you could find more details in the PHP.net, W3schools websites and many other resources.

Firstly, make sure that you have installed a web server on your computer (which should have been done during our on-campus session). There are many ways to set up your local web server. We suggested you to install an Apache/MySQL/ PHP bundle:

– XAMPP: for all major OSs (Recommended for all)
– WAMP: for Windows
– MAMP: for Mac OS

After the installation, start the server. To test that server is running, you could point a browser at: http://localhost. You should be able to see the relevant information displayed in the web page.

Then, download **Unit 3 Lab.zip** from Blackboard and store all the extracted files preferably in a new directory (named it as **unit3)** within your web server's document root directory. The document root is normally a subfolder called **htdocs** for XAMPP and **www** for WAMP depending the installation in your computer. We need to put all our files there since the document root is the directory designated for holding web pages for a web server to access.

You will see a file called **unit3.php** (supposed in the **unit3** sub-folder within the web server's document root directory). You should be able to check your **unit3.php** web pages using:

http://localhost/unit3/unit3.php

If you could see a list of tasks displayed in your browser, you are fine to go.

Finally, use your preferred editor to open the downloaded **unit3.php**. You need to write code for the following tasks in the suitable places within **unit3.php** according to the comments there.

*Note: You should be used to using **git** and we encourage you to create a repository for your practical work for this module. Set up a repository for DC2410 and clone it to your local machine. Create a subfolder within this called something like unit3 to contain your work for this unit.*

Task 1: String

Define a string variable and assign "I love programming" to it. Write suitable PHP statements to print the following list.

- First letter is: I
- Length of string is: 18
- Last letter is: g
- First 6 letters are: I love
- In capital: I LOVE PROGRAMMING

***Hint**: you could find and use some suitable PHP string functions.*

Task 2: Array and images

Define an array to store names of four images which are stored in a directory called **images** after you extract the **unit3.zip** file. Then, display a randomly chosen image on the webpage.

Hint:

- *Function rand() generates a random integer number. Use rand()%n to generate a number between 0 and n-1.*
- *<img src="pic.gif" alt="pic name" width="20%">*

Task 3: Functions – days in month

Write a function called **`daysInMonth`** that takes a month (between 1 and 12) as an argument and returns the number of days in that month in a non-leap year. For example a call to **`daysInMonth(6)`** should return 30, because June has 30 days.

Call this function and display the days of each month using an unordered list.

SAMPLE OUTPUT

- Month 1 => 31 days.
- Month 2 => 28 days.
- Month 3 => 31 days.
- Month 4 => 30 days.
- Month 5 => 31 days.
- Month 6 => 30 days.
- Month 7 => 31 days.
- Month 8 => 31 days.
- Month 9 => 30 days.
- Month 10 => 31 days.
- Month 11 => 30 days.
- Month 12 => 31 days.

Task 4: Array and favourite Artists from a File

Define an array with the names of some favourite artists. The array contents are read them from a file called **favourite.txt** which you have downloaded.

- Read the artists from the file;
- Print out the artist names one per line;
- Make each artist's name a hyperlink.
    - The links have names similar to the artist's name but in lowercase with dashes.
    - Example: Justin Bieber links to `http://www.mtv.com/artists/justin-bieber/`

SAMPLE OUTPUT

1. Christina Aguilera
2. Justin Bieber
3. Lady Gaga

Hint:

- *PHP has some built-in functions that can be used to manipulate file I/O. For this task, you could use* `file()` *function which returns the lines of a file as an array of strings each ends with \n ; About this function: http://php.net/manual/en/function.file.php Then you could use* `foreach` *to loop through this array.*
- *For more file functions: http://www.php.net/manual/en/refs.fileprocess.file.php*
- *HTML hyperlink :* `<a href = "url">link text </a>`

Task 5: Including external file

Write a separate script called **footer.php** to print the following information:

© 2020 Aston University, Aston triangle, Birmingham, B4 7ET +44 (0) 121 204 3000

Now include footer.php file in end part of the unit3.php

Hint*: The code for © is* `&copy`

Task 6: Directory operations

List all you files (not directory) directly within your documents root folder in an ordered list.

Hint*:*

- *You can use scandir() function to returns an array of all file/directories names in a given directory (e.g. returns the file names, such as "myfile.txt"). You could learn this function by referencing the link: http://php.net/manual/en/function.scandir.php*
- *You could explore a super global array, $_SERVER, and use it to get the document root directory path.*
- *Find a function by yourself on how to check whether the filename is a directory.*

Optional*: can you list all files within your documents root folder and its subfolder? You may think to define the function which can recursively check the sub-directories and display the file names.*