

Assignment 1

Practical PHP Programming Tasks (15%)

DUE DATE

Session 5

OBJECTIVE

- Perform basic operations about PHP environment
- Write basic PHP scripts
- Solve algorithmic problems
- Manipulate strings
- Perform basic operations on arrays in PHP
- Convert between arrays and strings in PHP

DESCRIPTION

In this assignment, you are expected to write common tasks in PHP. You are given 25 short task in PHP. For each task, you must create an independent .php file to show your solution. In addition to displaying your source code in a `<pre>` `</pre>` element, you are also expected to generate a sample input and output for your program.

Once you have complete all of the tasks, create an index.php file in which you will print all of the tasks in a numbered list format. For each task you must provide a hyperlink to the .php file that contains the solution. If you are unable to complete a task, you must provide a .php file for that task that explains the challenge you have faced in finding a solution for that task.

Your final work must be in a directory structure so that it can be zipped together for submission purposes.

INSTRUCTIONS

1. Write a PHP script to get the PHP version and configuration information.
2. Create a simple HTML form and accept the user name and display the name through PHP echo statement.
3. Write a PHP script to get the client IP address.
4. Write a PHP script, which changes the color of the first character of a word.
5. Write a PHP script, to check whether the page is called from 'https' or 'http'.
6. Write a PHP script to redirect a user to a different page.

7. Write a simple PHP program to check that emails are valid.
Hints : Use FILTER_VALIDATE_EMAIL filter that validates value as an e-mail address.
Note : The PHP documentation does not say that FILTER_VALIDATE_EMAIL should pass the RFC5321.
8. Write a PHP script to count number of lines in a file.
9. Write a PHP script to get the document root directory under which the current script is executing, as defined in the server's configuration file.
10. Write a PHP script to get the time of the last modification of the current page.
11. Write a PHP program to check whether a number is an Armstrong number or not. Return true if the number is Armstrong otherwise return false.
Note: An Armstrong number of three digits is an integer so that the sum of the cubes of its digits is equal to the number itself. For example, 153 is an Armstrong number since $1^3 + 5^3 + 3^3 = 153$
12. Write a PHP program to remove duplicates from a sorted list.
13. Write a PHP program to print out the multiplication table up to $12 * 12$.
14. Write a PHP program that multiplies corresponding elements of two given lists.
15. Write a PHP program to compute the sum of the digits of a number.
16. Write a PHP program to compute the amount of the debt in n months. The borrowing amount is \$100,000 and the loan adds 5% interest of the debt and rounds it to the nearest 1,000 above month by month.
17. Write a PHP program to sort an array of positive integers using the Bead-Sort Algorithm.
18. Write a PHP function to change the following array's all values to upper or lower case.
19. Write a PHP script to generate simple random password [do not use rand() function] from a given string.
20. Write a PHP script to convert a string into an arbitrary array.
21. Write a PHP script to extract the filename component of a URL path.
22. Write a PHP script to insert a string at the specified position in a given string.
23. Write a PHP script to get the characters after the last '/' in an url.
24. Write a PHP class that calculates the factorial of an integer.
25. Write a PHP Calculator class (called MyCalculator), which will accept two values as arguments, then add them, subtract them, multiply them together, or divide them on request.

SUBMISSION INSTRUCTIONS

Your submission must include:

1. One index.php file in which the list of all questions are printed. For each question place a link to a PHP file corresponding to the source code of the solution.
2. Every solution must have a sample input and output displayed on the page.
3. Place the entire directory structure in a folder.
4. Zip the folder and submit a single zip file with all your work in the submission page for this assignment.

Work must be submitted in the correct file type and be properly labelled as per the College naming convention:

NAME_COURSE_ASSIGNMENT. E.g. XuXiaLing_FM50D_A01.

GRADING CRITERIA

Assignment Value: **15%**

Grading Criteria	Grading
Question 1	/4
Question 2	/4
Question 3	/4
Question 4	/4
Question 5	/4
Question 6	/4
Question 7	/4
Question 8	/4
Question 9	/4
Question 10	/4
Question 11	/4
Question 12	/4
Question 13	/4
Question 14	/4
Question 15	/4
Question 16	/4
Question 17	/4
Question 18	/4
Question 19	/4
Question 20	/4
Question 21	/4

Question 22	/4
Question 23	/4
Question 24	/4
Question 25	/4
TOTAL	/100