

### Task 1.

What is the output of the following php document?

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial 1 - Task 1</title>
    <meta charset="utf-8" />
  </head>

  <body>
    <h1> my first php task </h1>
  <?php
    $SingleFamilyHome = 399500;
    $SingleFamilyHome_Display = number_format($SingleFamilyHome,
2);
    echo "<p>The current median price",
      " of a single-family home in Australia",
      " is $$SingleFamilyHome_Display.</p>";
  ?>
</body>
</html>
```

\*Try to display the information on the web using multiple echo statements.

\*Try to display the information on the web using print statement(s).

Notes: Open new document in Notepad++. Copy the code in the document and save it as *file\_name.php*.

- For WAMP users - Upload the document into your localhost *www* (C:\wamp64\www) folder, or upload it on the web server.
- For XAMPP users - Upload the document into your localhost *htdocs* (C:\xampp\htdocs) folder, or upload it on the web server.
- For AMPPS users - Upload the document into your localhost *www* (C:\program files\ampps\www) folder, or upload it on the web server.

Open the php file in your Web browser by entering the following URL:  
[http://<yourserver>/file\\_name.php](http://<yourserver>/file_name.php) (e.g. [http://localhost/file\\_name.php](http://localhost/file_name.php) )

### Task 2.

Write a script that displays a list of the Celsius equivalents of zero degrees Fahrenheit through 100 degrees Fahrenheit. Display the output in a table format.

Add new table that will display a list of the Fahrenheit equivalents of zero degrees Celsius through 100 degrees Celsius.

To convert Fahrenheit to Celsius, subtract 32 from the Fahrenheit temperature, and then multiply the result by (5/9). To convert Celsius to Fahrenheit, multiply the Celsius temperature by (9/5), and then add 32.

Notes: Looping statement can be used for displaying the degrees. The round() function can be used for rounding the temperature to one decimal place.

\*Create a solution for Task 2 using functions.

Create two functions that will make the conversions (*functions FahtoCel and CelToFah*). The functions should return a float value and have input parameter of integer data type. The function call should use named arguments.

### Task 3.

Write a PHP function that has one input parameter (*number*). The function should return the value of the reversed number (reversed digits of the input parameter) divided by 3.

For example:

- if the input parameter is 2569, the reversed number is 9652, so the function should return the value 3217.3
- if the input parameter is 56984, the reversed number is 48965, so the function should return the value 16321.7

Write a PHP program that will test the function (with 3 different numbers) and display the results returned from the function.

\* The function should have an input parameter of integer data type and return a float value. The returned value should be rounded to 1 decimal place.

#### Some tips

- Use type declarations to restrict the type of information passed into and out of the function
- You can separate the digits of the number by  
e.g.  $256 \rightarrow 256\%10 = 6$ ;  $256/10 = 25 \rightarrow 25\%10 = 5$ ;  $25/10 = 2 \rightarrow 2\%10 = 2$ .  
\*please be careful  $256/10$  will return 25.6  $\rightarrow$  you will need to use casting
- You can also cast the number as a string and use some of the PHP (string) functions

### Task 4.

Write a PHP script that will create codes (ciphers) from a sentence. The script should include a function that has one input parameter (a sentence) and returns 3 ciphers. The script should display all the ciphers returned by the function.

The ciphers should be created as:

- (1) every first character of each word in the sentence
- (2) every first two characters of each word in the sentence
- (3) every last character of each word in the sentence

For example:

The sentence is “I need to study PHP very hard”

The 3 created ciphers should be: “intspvh”, “inetostphveha”, “idoypyd”.

\*If the word does not have 2 characters, only one character is added to the cipher.

\*We are assuming that the sentence does not have any punctuation marks (as '!' ';' '?' '!' ... )

Tip: The function can return multiple values within an array

### Task 5.

Write a PHP function that checks the elements of a string array named \$Passwords.

A strong password must have at least one number, one lowercase letter, one uppercase letter, no spaces, and at least one character that is not a letter or number (special character). The string should also be between 8 and 16 characters long.

Use regular expressions to test whether each element of the array is a strong password. The function should display a message with the details regarding the structure of each password in the array (e.g. the password is strong, the password is missing the special character, the password is missing uppercase letter, etc).

Write a PHP program that will test the function.

Note: Additional readings about regular expressions and lookahead (lookahead and lookbehind) assertions:

- <https://www.regular-expressions.info/quickstart.html>
- <https://www.regular-expressions.info/lookaround.html>
- <https://www.php.net/manual/en/regexp.reference.assertions.php>
- <https://www.regular-expressions.info/refadv.html>

### Task 6.

Design a web site for an imaginary finance company. Use different HTML elements to design the user interface. The user should enter the “Amount of Mortgage”, “Interest Rate”, as well as “Number of Years”.

Write a server-side script that will validate that the user did not leave empty fields, the entered data is in the right format (number), and calculate the “Monthly Payment” using the formula:

$$\text{Monthly Payment} = \text{loan} * \text{rate} / (1 - (1 / (1 + \text{rate})^{\text{PaymentsNumber}}))$$

In the above formula “loan” is the total amount of mortgage, “rate” is the interest rate divided by 12 and “PaymentsNumber” is the result of multiplying “Number of Years” by 12.

After pressing the “Calculate” button, the value of “Monthly Payment” is presented on the web.

*Create two versions of this task. In the first version, the solution should have a (display) form and a separate script (for calculations), and the other solution should be with All-in-One form.*

Hint: check PHP `pow(...)` function

Note: Do not use JavaScript to validate the input fields. Validation needs to be done on the server side.