Université de Strasbourg

Web Programming 2 : Back End

PW 2 : PHP Functions

Practical work procedure

These Practical Works (PW1 to PW8) all contribute to the final evaluation procedure of the Practical Work exam, which will conclude with an individual Practical Work examination. Therefore, they all contribute to the final score of the PW exam, and represent a fraction of your final score through your activity report during PWs, i.e. a maximum of 5 points. The final PW examination will be marked out of 15.

The teacher must be able to get an idea of the work done during the PW session, your level of skills, your knowledge.

Show them throughout the PW session, the successive steps of your work, and not only at the end of the session, so that they can evaluate you (score out of 5).

You can consult various web resources to find out more. Be careful, don't just copy the code from examples: you will need to be able to explain your code queries to team members.

Evaluation criteria for rating your practical work

- **Participating / Involved**: Participation, motivation, student involvement, autonomy, ability to work in a group (if in pairs), manage time to do the work, positive attitude, ...
- **Knowledge and Experimental skills:** Course knowledge, apply content seen in the course, ability to write queries by yourself, use of various tools, ease of handling data files, developed skills, ability to understand and correct errors, …
- **Results / queries**: queries correspond to the work requested, consistency of the results, …

During the practical works, teachers and assistants will rate your work, with a mark up to 5 points.

For a good evaluation of your work, you have to show your results all along the PW session, not only at the end.

These evaluations will contribute to your final Practical Work mark for Database course.

Topic of the practical work: PHP working environment / PHP Basics and algorithmic

Objective N1: PHP Date and Time functions

As a beginner in PHP programming, it's easier and faster to develop with procedural PHP. Procedural programming is based on the concept of function call. Documentation on Date and Time procedural functions can be found on : https://www.php.net/manual/en/ref.datetime.php.

1. Display the current date in the dd/mm/YYYY format (French standard format), then in the format generally used in MySQL databases, i. e. YYYY-mm-dd .
2. The timestamp is the number of seconds since January 1, 1970. Display the timestamp of the present moment. Subtract from this timestamp a full day (in seconds: 24 x 3600 s). Then display the date and time corresponding to this timestamp.
3. Generate the timestamp of the date of your next birthday. What will be the week number of the year?
4. By an appropriate function, generate the timestamp of your birthday by manually writing this date in MySQL format. From this timestamp, always with the same function, generate the timestamp at + 1 month + 1 day. Then display the date.
5. Using a for loop, display dates in 1 month, 2 months, etc... up to 12 months, starting from this day.
6. Display the full date of the day, in full text, for the day of the week and month, in English, then in another language (e. g. French).

Objective N2: PHP String functions

Documentation on Date and Time procedural functions can be found on :
https://www.php.net/manual/en/ref.strings.php.

1. Assign a text string to a variable (from Lorem ipsum). Display this text in an HTML paragraph with the echo statement.
2. Displays the number of characters in the previous string.
3. Transform the string into capital letters, then the first letter of each word. Of course, display the result at each step with the echo statement.
4. Extract a sub-string from this Lorem Ipsum string, the first 10, the last 10, then from position 10 to 20. Check the results obtained.
5. After intentionally adding spaces at the beginning and end of the chain, delete the spaces on the left, right, then left and right.
6. In a database, it is better to save passwords in md5 hash. But security is no longer ensured today, with passwords that are too simple! From a simple password (like a dictionary word), generate the md5 hash. Can you trace the origin of this password back using a specialized site (http://reversemd5.com/), which uses a dictionary of words already hashed? Then test with a more complex password using letters, numbers and special characters. Conclusion?

Objective N3: PHP Arrays

Documentation on Date and Time procedural functions can be found on :
https://www.php.net/manual/en/ref.array.php

To quickly visualize the content of an array, `$myArray` in PHP, use the following code:

```
echo "<pre>";
print_r($myArray);
echo "</pre>";
```

or simply `var_dump($myArray);`

1. Write a text string in a variable, containing words separated by white space , e. g. a sentence from the literature. Cut this text string according to the white space character using the appropriate function. An array is generated. Display the array (see above), then the number of elements (and therefore the number of words).
   Sort the elements in ascending and then descending order. Display the content of the array each time and rebuilt the text string using the appropriate function.
2. … to be continued…

Objective N4: PHP processing and HTML displayed results

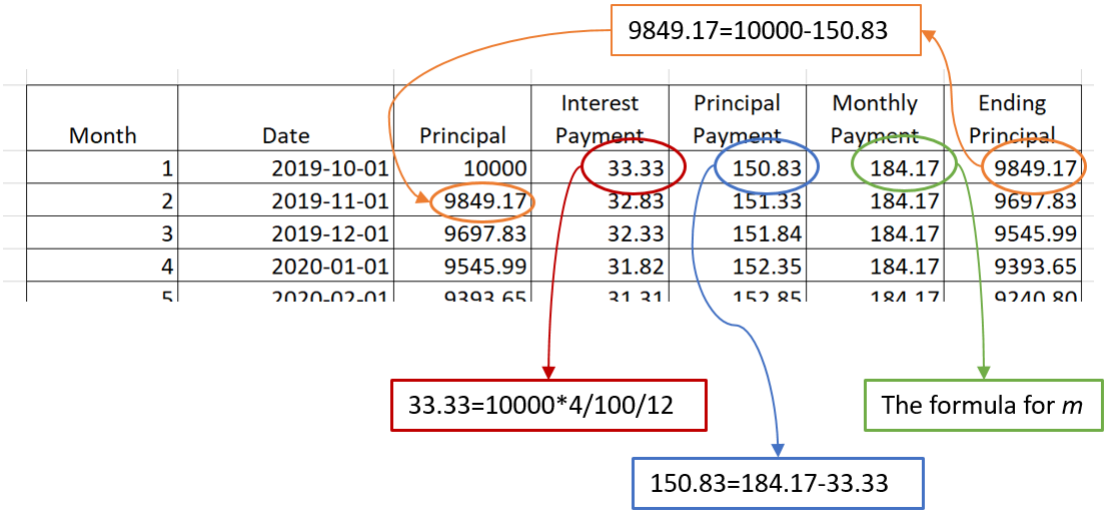
We propose to calculate an amortization schedule, namely a complete table of periodic monthly loan payment, showing, for each month, the amount of principal and the amount of interest that comprise each payment until the loan is paid off at the end of its term.

For a starting principal amount, e.g. *pa*=10 000 €, and an annual interest rate of 4% (or a monthly rate *r*= 4/100/12), and a term of the loan in 5 years (*n*=60 month), the constant monthly payment *m* is calculated with the formula :

$$m = \frac{pa * r}{1 - (1 + r)^{-n}}$$

The interest portion of each monthly payment is equal the principal balance time the monthly rate, and the principal portion of the payment for month is equal to the monthly payment minus the interest portion. The ending principal, which will be the base principal for the next month calculation is equal to the current principal balance minus the principal payment.

Generate in PHP the HTML table for the amortization schedule, also showing the date of payment:



Month	Date	Principal	Interest Payment	Principal Payment	Monthly Payment	Ending Principal
1	2019-10-01	10000	33.33	150.83	184.17	9849.17
2	2019-11-01	9849.17	32.83	151.33	184.17	9697.83
3	2019-12-01	9697.83	32.33	151.84	184.17	9545.99
4	2020-01-01	9545.99	31.82	152.35	184.17	9393.65
5	2020-02-01	9393.65	31.31	152.85	184.17	9240.80

Expected results & skill assessments:

Show your teachers the steps you have completed gradually, not just at the end of the PW session. In addition, you must be able to explain your PHP code.

PHP scripts must be uploaded to the remote web server Alwaysdata.