

PW 1 : PHP Working Environment and Tools PHP Basics

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 working environment

You should be able to work with at least 2 environments: an Apache / MySQL localhost server and a remote server, namely Alwaysdata.

- Reactivate your account access to Alwaysdata web hosting, or create a new account, possibly with a new email address. Check for FTP remote access parameters, in particular the host name and user management. Who is the main user, and his password? **Don't forget to access the administration panel each month!**
- A localhost environment must be installed and, above all, **functional!** Choose either Wampserver for Windows, Mamp for MacOS, Lamp for Linux, or Xampp (multiple OS).
- You should also be able to use an online tool to parse and to execute PHP script, like <https://ideone.com/>
- Typically, you will develop a localhost website (development step), and then export it to a remote server, as Alwaysdata (production release).

Objective N2: PHP Basics

The following exercises should be prepared before the PW session, and discussed during the session if you have any difficulties.

Remember that PHP instructions must be written between a starting and closing PHP tag : `<?php ... ?>`. Each PHP instruction ends with a semi-colon ;

The PHP script must be run under a localhost environment!

The **echo** "..." statement (or **echo()** which is less used) allows to get an output (displays a combination of strings and variables), from the server side running PHP to the browser client. The **print** "..." (or **print()**) statement is less used. **echo()** and **print()** are not functions.

A **PHP variable** starts with the \$ sign, followed by the name of the variable, starting with a letter or `_`. Allowed alpha numeric characters for variables are A..Za...z0...9. The variable name is case sensitive.

The **+** operator is used to add two variables, the **.** (dot) is a string concatenation operator.

Variables, Constants, Operators (arithmetic and assignment, concatenation and modulo operators).

1. From 2 or more numbers, either variables or constants, perform the most common arithmetic operations.
2. Apply the modulo operator (%) to a number, which returns the rest of the division.
3. From a given number n, calculate the square, the cube.
4. Let be a rectangle of length L and width l, calculate the perimeter, the area.
5. From a circle of radius R, calculate the diameter, perimeter, area. We will take for π the constant M_PI.
6. From an amount excluding VAT, calculate the VAT (we will take a VAT rate of 20%), then the amount including VAT.
7. Convert a number of days to hours, minutes, then seconds. For example, 5 days is 120h, or 7200mn or 432000s.
8. Convert a number of seconds into days, hours, minutes and seconds. Use the modulo % operator. 1 day is 86400s. Typically a timestamp is called the number of seconds elapsed since January 1, 1970.

Loops for doing repetitive tasks

1. Display a sequence of numbers, from a starting value to an end value. Use a **for** loop, then a **while** loop.
2. From the previous example, display the sequence of even numbers.
3. For a given number, display its multiplication table.
4. For a given number n, calculate the sum of 1 to n, e. g. for the number 4, calculate $1+2+3+4 = 10$. Then calculate the product (the factorial). For 4, calculate $1 * 2 * 3 * 4 = 24=4!$.
5. The value of π can be approached with the following formula : $\frac{\pi^2}{6} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \frac{1}{6^2} + \dots$ Calculate π with 1000 first terms of the arithmetic progression, then 10000 first terms.

Making decision: conditional testing

1. For a given number, display a message if it is larger or smaller than 100.
2. For a given age, determine the sport category in France, e. g. poussin, senior,... Choose a sport with few different categories (Kayak) :
[https://fr.wikipedia.org/wiki/Cat%C3%A9gorie_\(sports\)#Cat.C3.A9gories_en_cano.C3.AB-kayak_-_VA.27A](https://fr.wikipedia.org/wiki/Cat%C3%A9gorie_(sports)#Cat.C3.A9gories_en_cano.C3.AB-kayak_-_VA.27A)
3. Calculate the discount rate for a given amount excluding VAT. For example: 10% discount rate for an amount ≥ 2000 €, 5% for an amount ≥ 1000 € but < 2000 €, and 0% if < 1000 €. Calculate the discount rate, then the reduced amount excluding VAT, then the VAT (take a VAT rate of 20%), and finally the amount including VAT.

Objective N3: Advance algorithmic PHP programming

These exercises have already been coded in Javascript, during Web Programming 1.

1. **Equality.** Let A, B and C be three digits between 1 and 9. Note the number ABC, which will therefore have a value between 111 and 999, where A represents the hundreds, B the tens and C the units, namely $ABC = A * 100 + B * 10 + C * 1$. Find A, B and C, such as : $ABC = A^3 + B^3 + C^3$, with $A^3 = A * A * A$, Use an iterative method, i. e. a test of all possible values for A, B and C . The program will display the right solutions found as the iterations progress. Example, if A=2,B=5,C=3, ABC will have the value of 253. Is it equal to $2^3+5^3+3^3$?
2. **Give change back.** This consists in giving change in the form of notes for a given amount (the Manat will be chosen, with banknotes of 1, 5, 10, 20, 50, 100 and 200 €). For example, for an amount of 369 €, the program will suggest the optimized rendering (as few banknotes as possible) in the form : a 200 Manat banknote, a 100 Manat banknote, a 50 Manat banknote, a 10 Manat banknote, a 5 Manat banknote, and four 1 Manat

banknotes. Display the suggested change back. You could use the modulo operator, that finds the remainder after division of one number by another (sometimes called modulus).

3. **Luhn algorithm.** Using Luhn's algorithm (https://en.wikipedia.org/wiki/Luhn_algorithm), check the validity of the credit card number. You will find fictitious numbers here (<http://www.getcreditcardnumbers.com/>), which you will test with your algorithm. Display the credit card number, the calculated check digit, and the keyword Valid / Not Valid.

Objective N4: PHP processing and HTML displayed results

PHP is used to write dynamic content in web pages. Therefore, not only can PHP perform calculations, as seen above, but it can also write HTML code to display the calculation results, in an HTML table for example. It is proposed to display a trigonometric data table, with a column for angles, in degrees, from 0 to 90, a column with the conversion of the angle into radian, then in the following columns, the sinus, the cosine, etc...

Trigonometric table

Deg	Rad	sin	cos	tan	cotan
0	0.000	0.000	1.000	0.000	∞
1	0.017	0.017	1.000	0.017	57.290
2	0.035	0.035	0.999	0.035	28.636
3	0.052	0.052	0.999	0.052	19.081
4	0.070	0.070	0.998	0.070	14.301
5	0.087	0.087	0.996	0.087	11.430

Write the common structure of an HTML web page, the structure of an HTML table with the header, and then, in PHP, generate the table body content (data calculation and table rows, table cells HTML code).

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.