# PHP / SQL intermediate evaluation

## Time allowed: 4h00

## Marking

At the end of the allocated time, you will have to deliver all of your files, and a mark out of 20 will be awarded according to the following scale:

- **2 points** for exercise 1
- **5 points** for exercise 2
- **5 points** for exercise 3
- **8 points** for exercise 4
- **2 points** on the following criteria:
  - Indentation of the code
  - Comments
  - Relevance of the name of the variables
  - Readability

**NB: it is not permitted to copy and use code obtained from other students of this training or previous students of this training - doing so is considered cheating and will be sanctioned.**

**Bonus** : bonus points may be added if the optional functions have time to be carried out. Do it only if you have the time!

## Tips

If the task seems overwhelming, don't worry! Each step is detailed to move forward little by little.

- Read the entire statement from the start, to know where you are going.
- Take the time to code, commenting on your code as soon as it is necessary, why not by copying the instructions in comment.
  The proofreader must be able to understand what you have done!
- Keep functions simple, which do only one thing, the better to navigate.
- Focus on PHP rather than CSS/HTML !
- The more your code will be indented and readable, the easier the rest will be!

Each exercise is independent, be sure to separate them into different files or folders.

## Exercise 1: Setting-up the project

Create a new github repository.
To submit your work at the end of the evaluation, you will have to upload your sources to a GitHub repository.

**!! Set your repository as private.**

If you don’t have time or cannot make it, zip your work and send it by mail.

## Exercise 2: "Celebrities"

We have gathered information about a few celebrities. We managed to retrieve their bank account information among other things.
You can find this information in the ‘exercise2_array.php’ file.

Given this information, write a PHP script that displays the celebrity name AND the number of houses for each person, as well as the bank account balance (credit - debit).
If the bank account balance is below 0, display it in red.

This list should be displayed in descending alphabetical order (based on the name).

Example :

'Dwayne Johnson own 21 houses and has a bank balance of 871399200 $'

'Kylie Jenner own 5 houses and has a bank balance of -15000 $'

'Jeff Bezos own 15 houses and has a bank balance of 812014041 $'

**Bonus :**

Display the person with the largest bank balance.

## Exercise 3: "Numbers"

Write a function that counts and displays how many times a number (absolute value) appears in an array of integers (see example below).

The function has 2 arguments:

- The array (array type)
- The number to search (integer type)

The function must return the result as a String.
You also have to perform the necessary verifications in order to validate the arguments types.
You can't use `array_count_values` function.

Example :
Given the following array $array = [5, 19, 23, -5, -2, 5, 7, 10];
And the number to search for: 5

The expected result is : 'The number of occurrences is 3'.

## Exercise 4: "It’s almost Christmas !"

### Part 1 :

Create a database and name it "christmas_shop". Inside it, you must create a table that you will call "toys" with the following :

Table : toys

Attributes :
id (int PrimaryKey)
name (varchar)
price (double)
photo (varchar)
type (enum: dolls, mechanic and puzzle)
description (varchar)

You have to export the DB and include the SQL file in your project folder.

### Part 2 :

Create a page 'add-toys.php'.

Inside it, there should be a form to add a toy in the table 'toys'.

Prerequisites:

- The field name, price and type are required.
- The field 'type' will be a drop-down menu (**bonus**).
- The price must contain numbers only.
- Error messages will be displayed in red.

Once you submit the form, the toys must be added to the database.
A success message will be displayed.

**Part 3 :**

Create a page 'edit-toys.php'.

Inside it, there should be a form to edit a toy in the table 'toys'. It is the same as part 2.
You also have to perform the same validations.

The difference from part 2 are :

- You need to use the GET method to retrieve the toy's id from the URL.
- Pre-fill the form with the toy's information when arriving on the page.
- You have to UPDATE and not INSERT

Once you submit the form, the toys must be updated in the database.
A success message will be displayed.

**Part 4 :**

Create a page 'toys.php'.

This page will display all the toys with their respective information.
You have to :

- Display the toys in a HTML table.
- Display the name of each toy in capital case.
- Display the price with a € sign.

If the description is more than 30 characters long, cut the text by adding "...".
Add a link ‘Edit’ for each toy.