C1- Code & Go

pool_php_d01

Day 01

Time To Play Again !?

CODING ACADEMY
BY EPITECH

Foreword

Welcome to this first day of PHP pool.

Today, you will learn the first bricks of PHP programming. You will see, it's easier than C programming. No more types but DO NOT forget semi-colons.

Please note that prototypes are just jere as information. They are deprecated in PHP5 and more.

Exercise 01

1pt

Turn in: pool_php_d01/ex_01/ex_01.php
Restrictions: None.

Create the variables "integer", "float", "string", "bool", "null". Give the following values to the variables whose name correspond to the type of value: "true", "forty two", "42", "NULL" and "42.42".

Exercise 02

1pt

Turn in: pool_php_d01/ex_02/ex_02.php
Restrictions: None.

Delete the variable "var".

The "var" variable will be created by moulinette, don't forget to delete it after your tests.

Exercise 03

1pt

Turn in: pool_php_d01/ex_03/ex_03.php
Restrictions: None.

Create an array named "my_array" which will contain 5 elements of type (in that order): "string", "integer", "string", "float" and "string". These elements will have the following values, respectively: "to", 42, "Glory", "42.42" and "Geckos".

Exercise 04

1pt

Turn in: pool_php_d01/ex_04/ex_04.php
Prototype: void my_concat(mixed $str1, mixed $str2);
Restrictions: You will have to choose between "echo" and "print" for this function, and you will use only once the display function that you have chosen.

Create a function named "my_concat" that takes two parameters. The function must display the first parameter followed by a space followed by the second parameter.
Example :

```
<?php
require("my_concat.php");
my_concat("Hello", "world");
```

```
Terminal
~/pool_php_d01> php index.php
Hello world
~/pool_php_d01>
```

Exercise 05

2pts

Turn in: pool_php_d01/ex_05/ex_05.php
Prototype: void my_swap(mixed &$a, mixed &$b);
Restrictions: None.

Write a function that exchanges the contant of two variables whose **references** are given as parameters.

Exercise 06

1pt

Turn in: pool_php_d01/ex_06/ex_06.php
Restrictions: All functions that are not anonymous are forbidden.

Create an anonymous function that takes as parameter a variable of type string and returns its equivalent in upper case. You will have to assign this anonymous function to a variable "func".

Exercise 07

2pt

Turn in: pool_php_d01/ex_07/ex_07.php
Prototype: string get_angry_dog(int $nbr);
Restrictions: Obligation to use the keyword "for".

Write a function that returns a string composed of as many "woof" as the value of the variable passed as parameter, followed by a new line. *Example :*

```
<?php
require("ex_07.php");
get_angry_dog(3);
```

```
Terminal
~/pool_php_d01> php index.php
woofwoofwoof
~/pool_php_d01>
```

Exercise 08

2pts

Turn in: pool_php_d01/ex_08/ex_08.php
Prototype: void print_array(array $my_array);
Restrictions: Obligation to use the keyword "foreach".

Write a function that displays all the values of the array passed as parameter, each value will be followed bu a new line.

Exercise 09

2pts

Turn in: pool_php_d01/ex_09/ex_09.php
Prototype: void print_movie_from_nbr(int $nbr); a
Restrictions: Obligation to use the keyword "switch".

Write a function that takes an integer as parameter. If the value is 3, it displays "The Three Brothers". If the value is 6, it displaus "The Sixth Sense". For 23, it displays "The Number 23". And for 28, it displays "28 Days Later". For other values, it will displays "I don't know". A new line will be called after each display.

Exercise 10

2pts

Turn in: pool_php_d01/ex_10/ex_10.php
Prototype: array my_get_args([mixed $var, ...]);
Restrictions: Use of func_get_args forbidden.

Write a function my_get_args that takes as parameter a variable number of arguments and returns these arguments in an array.

Exercise 11

2pts

Turn in: pool_php_d01/ex_11/ex_11.php
Prototype: void my_print_args([mixed $var, ...]);

Create a function that outputs a variable list of the arguments followed by a new line.

```
Terminal
~/pool_php_d01> php ex_11.php test "Hello world" "php rocks" 42
test
Hello world
php rocks
42 ~/pool_php_d01>
```

Exercise 12

3pts

Turn in: pool_php_d01/ex_12/ex_12.php
Prototype: void sequence(int $nbr);

Write a function that outputs this sequence to the nth iteration.
Caution: first iteration is 0.
Your function should not print anything if you pass it a negative number.
Example :

```
<?php
require("ex_12.php");
sequence(5);
```

Produce the following output :

```
1
11
21
1211
111221
312211
```