W1 - Piscine PHP

W-WEB-024

Jour 03

Notions fondamentales de PHP

2.1

Jour 03

repository name: poolphpday03
repository rights: ramassage-tek
language: PHP

- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.

Informations

Avant de commencer

- Lisez attentivement toutes les consignes.
- Consultez vos mails plusieurs fois par jour, tous les jours.

Commencez par lire vos mails tout de suite à l'adresse : mail.office365.com.

- C'est une pangolinette (un programme) qui corrige vos exercices. Vérifiez le travail que vous allez rendre afin qu'il respecte scrupuleusement les consignes.
- Vous devez respecter les restrictions qui sont imposées dans chaque exercice. Le cas contraire, la pangolinette va considérer comme **triche** en attribuant la note de **-42**.
- Vous êtes susceptibles à tout moment de recevoir des corrections intermédiaires

Pour bénéficier de corrections intermédiaires, vous devez chaque jour :
- Etre inscrit au projet et aux activités dans l'intranet.
- Tenir à jour régulièrement le dépôt.

- Ne laissez jamais votre session ouverte sans surveillance.

JOUR 03

Votre répertoire ne doit pas contenir de fichiers inutiles (fichiers temporaires, …)
N'oubliez pas de push régulièrement vos fichiers, sans cela, pas de correction

Pensez à créer votre répertoire en début de journée et à envoyer votre travail via **git**!
Le nom du répertoire est spécifié dans les instructions pour chaque étape/exercice.
Pour garder votre répertoire propre, regardez du côté de `gitignore`.

N'oubliez pas de vous inscrire à toutes les activités possibles de la semaine.

ÉTAPE 1 - (1 POINT)

Nom de rendu : ex_01.php
Restrictions : Aucune

Faites un fichier PHP qui affichera le message suivant : **Welcome to this pool** suivi d'un retour à la ligne.

Exemple :

```
Terminal
~/W-WEB-024> php ex_01.php
Welcome to this pool
~/W-WEB-024>
```

echo

ÉTAPE 2 - (1 POINT)

Nom de rendu : ex_02.php
Restrictions : Aucune

Créez une variable "helpers" à laquelle vous assignerez la valeur "Pangolins" suivi d'un retour à la ligne, et affichez-la.

Exemple :

```
Terminal
~/W-WEB-024> php ex_02.php
Pangolins
~/W-WEB-024>
```

variables

ÉTAPE 3 - (1 POINT)

Nom de rendu : ex_03.php
Restrictions : Aucune

Créez les variables :

- **integer**
- **float**
- **string**
- **bool**
- **null**
- **array**

Affectez les valeurs suivantes aux variables dont le nom correspond au type de la valeur :

- `true`
- `[]`
- `quarante-deux`
- `42`
- `NULL`
- `42.42`

ÉTAPE 4 - (1 POINT)

Nom de rendu : ex_04.php
Restrictions : Aucune

Détruisez la variable “myvar”.

La variable “myvar” sera créée par la moulinette, n’oubliez pas de la retirer après vos tests.

unset

ÉTAPE 5 - (1 POINT)

Nom de rendu : ex_05.php
Restrictions : Aucune

Créez une constante que vous appellerez **“CONSTANTE”** et à laquelle vous assignerez la valeur **“Je suis une constante”**.

constants

ÉTAPE 6 - (1 POINT)

Nom de rendu : ex_06.php
Restrictions : Aucune

Créez un tableau que vous appellerez “my_array” qui contiendra 6 éléments qui seront de type (dans cet ordre) **“string”, “integer”, “string”, “float”, “string” et “bool”** et qui auront respectivement pour valeur : **“aux”, “42”, “Gloire”, “42.42”, “Pangolins” et “true”**.

array

ÉTAPE 7 - (1 POINT)

Nom de rendu : ex_07.php
Restrictions : Aucune

Créez une fonction nommée **“print_something”** qui affichera la chaîne de caractères **“Something”** suivi d’un retour à la ligne, ceci à chaque fois qu’elle sera appelée.

Prototype : void print_something(void);
Exemple :

Terminal

```
print_something();
// Un appel a cette fonction doit afficher : Something
```

function
print

ÉTAPE 8 - (1 POINT)

Nom de rendu : ex_08.php
Restrictions : Vous devrez choisir entre “echo” et “print” pour cette fonction et vous n’avez le droit qu’à une seule utilisation de la fonction d’affichage que vous aurez choisi.

Créez une fonction que vous appellerez **“my_concat”** qui prend deux paramètres. La fonction devra afficher le premier paramètre suivi d’un espace suivi du second paramètre.

Prototype : void my_concat(mixed $str1, mixed $str2); **Exemple :**

Terminal

```
my_concat("Hello", "world");
// Un appel a cette fonction doit afficher : "Hello world"
```

variables scope

Toute tentative de triche ou non-respect des restrictions est sanctionné par un -42 sur la journée.

ÉTAPE 9 - (1 POINT)

Nom de rendu : ex_09.php
Restrictions : Aucune

Créez une fonction **“print_variable”**. Cette fonction devra afficher la chaîne de caractères suivante : **“variable = [val]" où "[val]"** est remplacé par la valeur de la variable passée en paramètre.

Prototype : void print_variable(mixed $variable);

ÉTAPE 10 - (1 POINT)

Nom de rendu : ex_10.php
Restrictions : Aucune

Créez une fonction **“print_calls”** qui ne prend aucun paramètre et qui affiche le nombre de fois où elle est appelée.

Prototype : void print_calls(void);
Exemple :

Terminal

```
print_calls () ; // 1
print_calls () ; // 2
print_calls () ; // 3
```

static

ÉTAPE 11 - (1 POINT)

Nom de rendu : ex_11.php
Restrictions : Aucune

Créez une fonction **“my_sub”** qui ne prend aucun paramètre. Cette fonction devra soustraire deux variables globales nommées **“nb_a”** et **“nb_b”** (**“nb_a” – “nb_b”**) et devra assigner le résultat à la variable globale **“nb_a”**, puis retourner cette valeur.

Prototype : mixed my_sub(void);

ÉTAPE 12 - (1 POINT)

Nom de rendu : ex_12.php
Restrictions : Aucune

Créez une fonction **"my_increment"** qui prendra en paramètre une variable par référence. Cette fonction devra incrémenter la variable et ne rien retourner.

Prototype : void my_increment(int &$nb);

variables références

ÉTAPE 13 - (1 POINT)

Nom de rendu : ex_13.php
Restrictions : Aucune

Écrire une fonction qui échange le contenu de deux variables dont les références sont données en paramètres.

Prototype : void my_swap_vars(mixed &$a, mixed &$b);

ÉTAPE 14 - (1 POINT)

Nom de rendu : ex_14.php
Restrictions : Aucune

Créez une fonction **“say_my_name”** qui prend en paramètre une chaîne de caractères et qui affiche **“My name is [name] !”** où **“[name]”** est remplacé par la variable passée en paramètre. Il doit être possible d’appeler la fonction sans paramètre auquel cas elle affichera **“My name is Toto !”**.

arguments fonction

ÉTAPE 15 - (1 POINT)

Nom de rendu : ex_15.php
Restrictions : Aucune

Créez une fonction **“teacher”** qui affiche le message **“I am a Teacher”**.
Créez une fonction **“student”** qui affiche le message **“I am a student and my name is [name]”** où **“[name]”** est rem- placé par la valeur de la variable passée en paramètre.
Créez également les variables **“func_teacher”** et **“func_student”** et faites en sorte qu’il soit possible d’appeler la fonction **“teacher”** avec la variable **“func_teacher”** et la fonction **“student”** avec la variable **“func_student”**.

Prototypes :

- void teacher(void);
- void student(string$name);

Exemple :

Terminal

```
$func_teacher();
// Appelle la fonction teacher()
$func_student('Manu');
// Appelle la fonction student('Manu')
```

callable

ÉTAPE 16 - (1 POINT)

Nom de rendu : ex_16.php
Restrictions : Vous ne devez pas créer de fonctions qui ne soient pas anonymes.

Créez une fonction anonyme qui prend en paramètre une variable de type string et qui retourne son équivalent avec la première lettre en majuscule. Vous devrez assigner cette fonction anonyme à une variable **"func"**.

fonctions anonymes

Pas de création de fonctions non anonymes… Vous pouvez néanmoins en utiliser des déjà exitantes **dans** votre fonction anonyme.

ÉTAPE 17 - (1 POINT)

Nom de rendu : ex_17.php
Restrictions : Aucune

Créez une fonction **"array_key"** qui devra retourner la valeur de l'élément du tableau située à l'index **"key"**.

Prototype : mixed array_key(array $arr, int $key);

ÉTAPE 18 - (1 POINT)

Nom de rendu : ex_18.php
Restrictions : Aucune

Créez les fonctions **"get_args"** et **"get_last_arg"** :
- **"get_args"** devra retourner tous les arguments passés en paramètre de la fonction dans un tableau.
- **"get_last_arg"** devra renvoyer le dernier argument passé en paramètre.

Prototypes :
- array get_args(…);
- mixed get_last_arg(…);

ÉTAPE 19 - (1 POINT)

Nom de rendu : ex_19.php
Restrictions : Aucune

Créez une fonction **"calc"** qui prend en paramètre un type d'opération ("+", "*","/","%","-") et deux entiers. La fonction retourne le résultat de l'opération en respectant l'ordre des paramètres.

Prototype : mixed calc(string $operation, int $nb1, int $nb2);
Exemple :

Terminal

```
echo calc("%", 5, 2);
// Affiche : 1
```

if
switch

ÉTAPE 20 - (1 POINT)

Nom de rendu : ex_20.php
Restrictions : Aucune

Créez une fonction **“spupof”** qui prend en paramètre une chaîne de caractères et qui affiche cette chaîne en rem- plaçant chacun des caractères par le suivant dans l’ordre alphabétique, suivi d’un **“\n”**. Les majuscules deviennent des minuscules. Les minuscules restent des minuscules. Le **“z”** devient **“a”**.

Prototype : void spupof(string $str);
Exemple :

Terminal

```
spupof("CoUcOu lEs gEnS");
// Affiche : dpvdpv mft hfot
```

chr
ord