



# W1- Piscine PHP

---

W-WEB-024

## Jour 03

---

Notions fondamentales de PHP

v1.0



# 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](mailto: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 :



- Être inscrit au projet et aux activités dans l'intranet.
- Avoir créé le dépôt avec BLIH.
- Tenir à jour régulièrement le dépôt.

- Ne laissez jamais votre session ouverte sans surveillance.



# Jour 03

## Notions fondamentales de PHP

Nom du répertoire: Piscine\_PHP\_Jour\_03

Droits de ramassage: ramassage-tek

langage: php

Taille du groupe: 1



- Votre repertoire ne doit pas contenir de fichiers inutiles (fichiers temporaires, ...)
- Vous ne devez pas oublier votre fichier *auteur*, si vous l'oubliez, la moulinette ne pourra pas vous corriger.
- 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.



# Etape 1

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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>
```



# Etape 2

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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>
```



# Etape 3

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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" et "42.42".



# Etape 4

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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.



# Etape 5

---

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/ex\_05.php

**Restrictions:** Aucune

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





# Etape 6

---

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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"**.



# Etape 7

---

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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:**

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



# Etape 8

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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:**

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



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



# Etape 9

---

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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);



# Etape 10

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/ex\_10.php

**Restrictions:** Aucune

Créez une fonction "**print\_calls**" qui ne prend aucun paramètre et qui affiche le nombre de fois qu'elle est appelée.

**Prototype:** void print\_calls(void);

**Exemple:**

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



static



# Etape 11

---

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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);



# Etape 12

---

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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);



# Etape 13

---

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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);





# Etape 14

---

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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] !" ou "[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 !"**.



# Etape 15

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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 remplacé 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:**

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



# Etape 16

---

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/ex\_16.php

**Restrictions:** Toutes les fonctions qui ne sont pas anonymes sont interdites.

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".



# Etape 17

---

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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);



# Etape 18

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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(...);



# Etape 19

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/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:**

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



# Etape 20

1 point

**Nom de rendu:** Piscine\_PHP\_Jour\_03/ex\_20.php

**Restrictions:** Aucune

Créez une fonction "**spupof**" qui prend en paramètre une chaîne de caractères et qui affiche cette chaîne en remplaçant 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:**

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