



# W1- Piscine PHP

---

W-WEB-024

## Jour 06

---

Sérialisation et algorithme

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 :



- Etre 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 06

## Sérialisation et algorithme

Nom du répertoire: Piscine\_PHP\_Jour\_06

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 possible de la semaine.



# Etape 1

---

2 points

**Nom de rendu:** Piscine\_PHP\_Jour\_06/ex\_01.php

**Restrictions:** Aucune

Créer une fonction nommée `my_generate_file`. Cette fonction prend en paramètre une chaîne de caractères et elle crée un fichier vide nommé "**\$name.txt**" (remplacer \$name par sa valeur). La fonction retourne `true` en cas de succès ou `false` en cas d'erreur.

**Prototype:** `bool my_generate_file (string $name);`



# Etape 2

2 points

**Nom de rendu:** Piscine\_PHP\_Jour\_06/ex\_02.php

**Restrictions:** Aucune

Écrire une fonction qui retourne le contenu des fichiers dont les chemins sont passés en paramètre. Il y aura une ligne de 5 caractères "\_" entre le contenu de chaque fichier.

**Prototype:** string my\_cat\_files(string \$path, ...);



# Etape 3

3 points

**Nom de rendu:** Piscine\_PHP\_Jour\_06/ex\_03.php

**Restrictions:** Aucune

Écrire une fonction qui écrit la chaîne de caractère qui lui est passée en premier paramètre dans un fichier passé en second paramètre.

Cette fonction a pour troisième paramètre un booléen qui, s'il est vrai, concatène la chaîne de caractère à celle déjà présente dans le fichier.

Dans le cas contraire le fichier ne doit contenir que la chaîne de caractère envoyée. Par défaut ce paramètre doit être faux.

La fonction renvoie true en cas de succès ou false si une erreur survient.

**Prototype:** bool write\_into\_my\_file(string \$str, string \$file, [bool \$append = false]);



# Etape 4

3 points

**Nom de rendu:** Piscine\_PHP\_Jour\_06/ex\_04.php

**Restrictions:** Aucune

Écrire une fonction qui affiche les informations contenues dans un noeud XML suivies d'un retour à la ligne. La fonction renvoie true en cas de succès ou false si une erreur survient.

**Prototype:** bool display\_xml\_nodes(SimpleXMLElement \$xmlstr, string \$node);

**Exemple:**

```
$xmlstr = "<webacademie>
<staff>
<title>Staff Samung Campus</title>
<personnes>
<personne>
<name>Francois—Afif Benthane</name>
<poste>Fondateur</poste>
</personne>
<personne>
<name>Sophie Viger</name>
<poste>Directrice</poste>
</personne>
<personne>
<name>Michel Girard</name>
<poste>Responsable Pedagogique</poste>
</personne>
<personne>
<name>Sylvain Peigne</name>
<poste>Pangolin</poste>
</personne>
</personnes>
</staff>
</webacademie>"
display_xml_nodes($xmlstr, "name");

/* Affiche :
Francois—Afif Benthane
Sophie Viger
Michel Girard
Sylvain Peigne
*/
```



# Etape 5

2 points

**Nom de rendu:** Piscine\_PHP\_Jour\_06/ex\_05.php

**Restrictions:** Aucune

Ecrire une fonction qui transforme une chaîne de caractères contenant du JSON en objet et le retourne.

**Prototype:** object json\_to\_object(string \$json);

**Exemple:**

```
$json = ' {
  "title" : "Staff WAC Ambition Feminine",
  "personnes" : [
    {
      "name" : "Francois-Afif Benthane",
      "poste" : "Fondateur"
    },
    {
      "name" : "Sophie Viger",
      "poste" : "Directrice"
    },
    {
      "name" : "Lilas Merbouche",
      "poste" : "Responsable Pedagogique"
    },
    {
      "name" : "toto",
      "poste" : "Mascotte"
    }
  ]
}';
$object = json_to_object ($json);
```





# Etape 6

4 points

**Nom de rendu:** Piscine\_PHP\_Jour\_06/ex\_06.php

**Restrictions:** Aucune

Écrire une fonction qui affiche toutes les différentes combinaisons de  $n$  chiffre(s) ( $n$  étant compris entre 0 et 9 inclus), dans l'ordre croissant, suivies d'un retour à la ligne.

**Prototype:** void my\_comb\_n(int \$n);

**Exemple:**

```
/*  
Si n = 3, cela donne quelque chose comme ça :  
123  
124  
...  
359  
367  
...  
689  
789  
*/
```



# Etape 7

---

4 points

**Nom de rendu:** Piscine\_PHP\_Jour\_06/ex\_07.php

**Restrictions:** Toutes les fonctions de tri automatique sont interdites (sort, asort, usort, ...)

Écrire une fonction qui trie un tableau d'entiers, ordre croissant alphanumérique.

**Prototype:** void my\_sort\_int\_tab (array &\$stab);