



# W1 - Piscine PHP

---

W-WEB-024

## Jour 08

---

Chaînes de caractères



# Jour 08

repository name: poolphpday08  
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](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.
- Tenir à jour régulièrement le dépôt.

- Ne laissez jamais votre session ouverte sans surveillance.

## JOUR 08

---



Afin de devenir de bons développeurs il vous faut être rigoureux et méthodique. Cela s'applique aussi à la qualité du code que vous allez produire. Pour cela, vous devez respecter des normes de code ! Pour le PHP, par exemple, vous devez respecter la norme [PSR-12](#).



Toute erreur de norme entraînera un malus conséquent sur votre note finale.



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 - (2 POINTS)

---

**Nom de rendu :** ex\_01.php

**Restrictions :** Aucune

écrire une fonction qui compare la taille de deux chaînes de caractères et qui renvoie la taille de la chaîne la plus grande.

Dans le cas où elles sont de même taille, renvoyer la valeur de la taille.

Si jamais l'une des deux est invalide, la fonction devra renvoyer "-1".

**Prototype :** `int which_is_longer(string $str1, string $str2);`



## ÉTAPE 2 - (2 POINTS)

---

Nom de rendu : ex\_O2.php

Restrictions : Aucune

écrire une fonction qui affiche : “**Results : [arr1] - [arr2] - [arr3]**” suivi d'un retour à la ligne.

- “[arr1]” devra être remplacé par l'arrondi de “nb” à l'entier le plus proche.
- “[arr2]” devra être remplacé par l'arrondi de “nb” à l'entier supérieur.
- “[arr3]” devra être remplacé par l'arrondi de “nb” à l'entier inférieur.

**Prototype** : void my\_rounds(float \$nb);



## ÉTAPE 3 - (2 POINTS)

---

Nom de rendu : ex\_O3.php

Restrictions : Aucune

écrire une fonction qui prend en paramètre une chaîne de caractères et qui affiche les caractères entre le premier arobase rencontré et le dernier point suivant celui-ci, suivi d'un retour à la ligne. En cas d'erreur, la fonction n'affichera rien.

**Prototype :** void my\_mail(string \$mail);

**Exemple :**

```
my_mail("fakemail@mon.mail.com");  
// Cet appel devra afficher "mon.mailn"
```

## ÉTAPE 4 - (2 POINTS)

---

Nom de rendu : ex\_O4.php

Restrictions : Aucune

écrire une fonction qui supprime tous les espaces au début et à la fin de la chaîne "\$string", puis qui remplace les 2 premières occurrences (répétitions) de la chaîne "\$word" présentes dans "\$string", par la chaîne "Pangolin" et qui renvoie le résultat.

**Prototype :** string clear\_and\_replace(string \$string, string \$word);

**Exemple:**

```
clear_and_replace("Salut le chat, le chien et le lapin" , "le");  
// Cet appel devra retourner "Salut Pangolin chat, Pangolin chien et le lapin"
```



## ÉTAPE 5 - (2 POINTS)

---

Nom de rendu : ex\_O5.php

Restrictions : Aucune

écrire une fonction qui prend en paramètre une chaîne de caractères.

Elle doit la retourner avec pas plus d'un espace consécutif entre chaque mot, ni aucun espace au début et à la fin, le tout avec une majuscule à chaque fin de mot uniquement.

En cas d'erreur, si les arguments ne sont pas du bon type ou si la fonction est appelée sans argument, la fonction devra retourner NULL.

**Prototype :** mixed str\_lower\_epur(string \$str);

**Exemple :**

```
str_lower_epur("Un PanGolin CacHe ");  
// retourne "uN pangoliN cachE"
```

## ÉTAPE 6 - (2 POINTS)

---

Nom de rendu : ex\_O6.php

Restrictions : Aucune

écrire une fonction qui prend en paramètres une chaîne de caractères et un délimiteur.

Elle doit retourner un tableau contenant les morceaux de la chaîne découpée grâce au délimiteur.

En cas d'erreur, si les arguments ne sont pas du bon type ou si la fonction est appelée sans argument, la fonction devra retourner NULL.

**Prototype :** mixed str\_to\_wordtab(string \$str, string \$delim);

**Exemple :**

```
str_to_wordtab("Mais tu es tout la ", ' ');  
// retourne array("Mais", "tu", "es", "tout", "la")
```





## ÉTAPE 7 - (2 POINTS)

---

**Nom de rendu :** ex\_07.php

**Restrictions :** Obligation d'utiliser le mot-clé **“foreach”**

écrire une fonction nommée **“count\_char”** qui prend une chaîne de caractères en paramètre et qui retourne un tableau qui contient le nombre d'occurrences de chaque caractère (sauf les espaces et les tabulations).

La fonction est sensible à la casse. Aussi, vous devez impérativement trier le tableau dans l'ordre croissant des clés avant d'être retourné.

**Prototype :** array count\_char(string \$str);

**Exemple :**

```
$chaine = "Hello World";  
count_char($chaine); // array( "H" => 1, "W" => 1, "d" => 1, "e" => 1, "l" => 3, "o",  
                             => 2, "r" => 1)  
$chaine = "H e l l o";  
count_char($chaine); // array( "H" => 1, "e" => 1, "l" => 2, "o"=> 1)
```

## ÉTAPE 8 - (2 POINTS)

---

**Nom de rendu :** ex\_08.php

**Restrictions :** Aucune

écrire une fonction nommée **“get\_median”** qui retourne la médiane d'un ensemble de valeurs passés en paramètre sous forme de tableau. Dans le cas d'une liste paire d'éléments, c'est la moyenne arithmétique des deux valeurs centrales qui doit être retourné.

**Prototype :** mixed get\_median(array \$vals);

**Exemple :**

```
$values = array (11 , 2, 23, 5, 9, 84, 25);  
get_median($values); // retourne 11  
  
$values = array(11, 2, 23, 5, 9, 84, 11, 3);  
get_median ($values); // retourne 10  
  
$values = array (999 , 0);  
get_median($values); // retourne 499.5
```



## ÉTAPE 9 - (2 POINTS)

---

Nom de rendu : ex\_09.php

Restrictions : Obligation d'utiliser le mot-clé “foreach”

écrire une fonction qui remplace toutes les valeurs du tableau passé en paramètre par la chaîne de caractères “pangolin”.

**Prototype** : void make\_pangolins(array &\$my\_array);

## ÉTAPE 10 - (2 POINTS)

---

Nom de rendu : ex\_10.php

Restrictions : Aucune

écrire une fonction qui ne prend aucun paramètre et qui retourne une fois sur deux le nombre de fois qu'elle est appelée.

**Prototype** : mixed return\_calls(void);

**Exemple :**

```
return_calls(); // NULL
return_calls(); // 2
return_calls(); // NULL
return_calls(); // 4
```