



Piscine PHP

Jour 01

Staff 42 piscine@42.fr

Résumé:

Ce document est le sujet du jour 01 de la piscine PHP de 42.

Table des matières

I	Consignes	2
II	Préambule	3
III	Exercice 00 : HW	5
IV	Exercice 01 : mlX	6
V	Exercice 02 : Au divin	7
VI	Exercice 03 : ft_split	8
VII	Exercice 04 : aff_param	9
VIII	Exercice 05 : epur_str	10
IX	Exercice 06 : ssap	11
X	Exercice 07 : rostring	12
XI	Exercice 08 : ft_is_sort	13
XII	Exercice 09 : ssap - le retour -	14
XIII	Exercice 10 : do_op	15
XIV	Exercice 11 : do_op_2	16
XV	Exercice 12 : search_it !	17
XVI	Exercice 13 : sing_it !	18
XVII	Exercice 14 : L'agent se tâte	19

Chapitre I

Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- Seul le travail rendu sur votre dépôt sera pris en compte par la correction.
- Comme lors de la piscine C, vos exercices seront corrigés par vos camarades de piscine ET/OU par une moulinette.
- La Moulinette est très stricte dans sa notation. Elle est totalement automatisée. Il est impossible de discuter de sa note avec elle. Soyez d'une rigueur irréprochable pour éviter les surprises.
- L'utilisation d'une fonction interdite est un cas de triche. Toute triche est sanctionnée par la note de -42.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas nous ne porterons attention ni ne prendrons en compte un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- Vous ne devez laisser dans votre répertoire aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Votre manuel de référence s'appelle **Google**.
- Pensez à discuter sur le forum. La solution à votre problème s'y trouve probablement déjà. Sinon, vous en serez l'instigateur.
- Lisez attentivement les exemples. Ils pourraient bien requérir des choses qui ne sont pas autrement précisées dans le sujet ...
- Réfléchissez. Par pitié, par Thor, par Odin ! Nom d'une pipe !

Chapitre II

Préambule

Un poisson au fond d'un étang
Qui faisait des bulles
Qui faisait des bulles
Un poisson au fond d'un étang
Qui faisait des bulles
Pour passer le temps

Un oiseau vint près de l'étang
Regarder les bulles
Regarder les bulles
Un oiseau vint près de l'étang
Regarder les bulles
Pour passer le temps

Que fais-tu joli poisson blanc ?
Moi je fais des bulles
Moi je fais des bulles
Que fais-tu joli poisson blanc ?
Moi je fais des bulles
Pour passer le temps !

Plus j'en fais, plus je suis content
Plus je fais des bulles (bis)
Plus j'en fais, plus je suis content
Des rouges et des bleues
Selon le courant.

Le poisson tout en discutant
A fait une bulle (bis)
Le poisson tout en discutant
A fait une bulle
Pour monter dedans.

Et la bulle porté par le vent
Ah ! la belle bulle (bis)

Et la bulle porté par le vent
A pris son envol
Le poisson dedans.


L'oiseau est tombé dans l'étang
En voyant la bulle (bis)
L'oiseau est tombé dans l'étang
En voyant la bulle
Du poisson volant.

Maintenant au fond de l'étang
L'oiseau fait des bulles (bis)
Maintenant au fond de l'étang
L'oiseau fait des bulles
Pour monter dedans.

Pierre Chêne, chanté par Raymond Fau

Chapitre III

Exercice 00 : HW

	Exercice : 00
HW	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : hw.php	
Fonctions Autorisées : toute la lib standard PHP	
Remarques : n/a	

Rappel : le PHP c'est vraiment easy. C'est comme du C, sauf qu'on ne déclare pas les variables, on leur met un dollar devant, elles ne sont pas typées, et il n'y a pas de main. Le reste, c'est -presque- du détail.


Aujourd'hui, on va rester sur du PHP en ligne de commande. Commencez par réaliser un petit programme tout simple, appelé hw.php. Ce programme doit afficher un message de bonjour au monde bien connu.

```
$> ./hw.php
Hello World
$>
```

Note : journée MMORPG (Massive Moulinette Online Rules PHP Geniuses)

Chapitre IV

Exercice 01 : mlX

	Exercice : 01
mlX	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : mlx.php	
Fonctions Autorisées : toute la lib standard PHP	
Remarques : n/a	

Puisque vous êtes tous très à l'aise avec la **minilibX** désormais, je suis au regret de vous dire qu'il n'existe pas de binding **PHP** pour pouvoir l'utiliser ici. Cet exercice n'aura donc absolument rien à voir avec du **graphique** pas plus qu'avec des **maths**. Non, ce que vous devez réaliser, c'est un programme php qui affiche 1000 fois la lettre X, un retour à la ligne, et avec comme contrainte qu'il ne doit pas dépasser les 100 chars.


```
$> ls -la mlx.php
-rwxr-xr-x 1 boulon users 92 Mar 12 11:54 mlx.php
$> ./mlx.php
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

[TL;DP]

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
$>
```

Chapitre V

Exercice 02 : Au divin

	Exercice : 02
Au divin	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : <i>oddeven.php</i>	
Fonctions Autorisées : <i>toute la lib standard PHP</i>	
Remarques : <i>n/a</i>	


Comme le disait si bien le vénérable du sommet, c'est grâce à Olympe la lessive des dieux que le linge est plus souple, plus doux. Mais à bien y réfléchir, il n'y avait finalement qu'une chance sur deux de se tromper de pile de linge. Tout dépend s'il s'agit d'une pile paire ou d'une pile impaire. Faites donc un programme en php qui vous demande gentiment un numéro de pile de linge, et qui vous répond s'il est pair (donc lavé avec Olympe) ou s'il est impair.

```
$> ./oddeven.php
Entrez un nombre: 42
Le chiffre 42 est Pair
Entrez un nombre: 0
Le chiffre 0 est Pair
Entrez un nombre:
' ' n'est pas un chiffre
Entrez un nombre: toto
'toto' n'est pas un chiffre
Entrez un nombre: 21
Le chiffre 21 est Impair
Entrez un nombre: 99cosmos
'99cosmos' n'est pas un chiffre
Entrez un nombre: ^D
$>
```

Faites bien attention à l'exemple, notamment les espaces, les majuscules et les messages exacts. À la fin, c'est un 'CTRL-D' pour quitter. Et la librairie readline ne fait pas partie de la librairie standard de PHP.

Chapitre VI

Exercice 03 : ft_split

	Exercice : 03
ft_split	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : ft_split.php	
Fonctions Autorisées : toute la lib standard PHP	
Remarques : n/a	


Réalisez la fonction `ft_split`. Elle prend une chaîne de caractères en paramètre, et renvoie un tableau trié des différents mots, initialement séparés par un ou des espaces dans la chaîne originale. Votre rendu `ft_split.php` sera inclus dans un fichier php de test.

```
<?PHP
include("ft_split.php");
print_r(ft_split("Hello   World AAA"));
?>
```

```
$> ./main.php
Array
(
    [0] => AAA
    [1] => Hello
    [2] => World
)
$>
```

Chapitre VII

Exercice 04 : aff_param


	Exercice : 04
aff_param	
Dossier de rendu : <i>ex04/</i>	
Fichiers à rendre : aff_param.php	
Fonctions Autorisées : toute la lib standard PHP	
Remarques : n/a	

Très classique, ce programme affiche ses différents paramètres de la ligne de commande, dans l'ordre reçu. Le nom du programme ne s'affiche pas.

```
$> ./aff_param.php
$> ./aff_param.php toto ahah foo bar quax
toto
ahah
foo
bar
quax
$>
```

Chapitre VIII

Exercice 05 : epur_str


	Exercice : 05
epur_str	
Dossier de rendu : <i>ex05/</i>	
Fichiers à rendre : epur_str.php	
Fonctions Autorisées : toute la lib standard PHP	
Remarques : n/a	

Ce programme prend un unique paramètre, et réduit à un seul espace entre chaque mot, et à aucun en début et fin de chaîne. Il n'y a que des espaces, pas de tabulation ou autre.

```
$> ./epur_str.php
$> ./epur_str.php "Salut,      comment ca      va      ?"
Salut, comment ca va ?
$> ./epur_str.php "  Hello World  "
Hello World
$>
```

Chapitre IX

Exercice 06 : ssap


	Exercice : 06
ssap	
Dossier de rendu : <i>ex06/</i>	
Fichiers à rendre : ssap.php	
Fonctions Autorisées : toute la lib standard PHP	
Remarques : n/a	

À ne pas confondre avec le progiciel de gestion d'entreprise SAP, il s'agit ici pour vous de mixer les 2 exercices précédents. L'ensemble des mots contenus dans tous les paramètres (sauf nom du programme lui-même) sont triés et affichés.

```
$> ./ssap.php
$> ./ssap.php foo bar
bar
foo
$> ./ssap.php foo bar "yo man" "A moi compte, deux mots" Xibul
A
Xibul
bar
compte,
deux
foo
man
moi
mots
yo
$>
```

Chapitre X

Exercice 07 : rostring


	Exercice : 07
rostring	
Dossier de rendu : <i>ex07/</i>	
Fichiers à rendre : rostring.php	
Fonctions Autorisées : toute la lib standard PHP	
Remarques : n/a	

Votre programme prend en paramètre une chaîne de caractère, et met le premier mot (séparateur espace) en dernière place. Le tout est ré-affiché, avec uniquement un seul espace entre 2 mots.

```
$> ./rostring.php  
$> ./rostring.php sdfkjsdkl sdkjfskljdf  
sdfkjsdkl  
$> ./rostring.php "hello world aaa" fslkdjf  
world aaa hello  
$>
```

Chapitre XI

Exercice 08 : ft_is_sort

	Exercice : 08
	ft_is_sort
	Dossier de rendu : <i>ex08/</i>
	Fichiers à rendre : <code>ft_is_sort.php</code>
	Fonctions Autorisées : toute la lib standard PHP
	Remarques : n/a

Il s'agit pour vous de réaliser une petite fonction qui renvoie vrai ou faux selon que le tableau passé en paramètres est trié ou non.

```
<?PHP
include("ft_is_sort.php");


$tab = array("!/@#;^", "42", "Hello World", "salut", "zZzZzZz");
$tab[] = "Et qu'est-ce qu'on fait maintenant ?";

if (ft_is_sort($tab))
    echo "Le tableau est trie\n";
else
    echo "Le tableau n'est pas trie\n";
?>
```

```
$> ./main.php
Le tableau n'est pas trie
$>
```

Chapitre XII

Exercice 09 : ssap - le retour -


	Exercice : 09
ssap - le retour -	
Dossier de rendu : <i>ex09/</i>	
Fichiers à rendre : ssap2.php	
Fonctions Autorisées : toute la lib standard PHP	
Remarques : n/a	

Reprenez votre ssap.php. Il s'agit ici de faire la même chose (prendre tous les mots de tous les paramètres et de les trier), mais de changer les modalités du tri : celui-ci devra être case insensitive, et mettre les caractères alphabétiques dans un premier temps, puis les chiffres, enfin tous les autres caractères, chacun des 3 groupes suivant l'ordre ASCII.

```
$> ./ssap2.php
$> ./ssap2.php toto tutu 4234 "_hop XXX" ## "1948372 AhAhAh"
AhAhAh
toto
tutu
XXX
1948372
4234
##
_hop
$>
```

Chapitre XIII

Exercice 10 : do_op

	Exercice : 10
do_op	
Dossier de rendu : <i>ex10/</i>	
Fichiers à rendre : do_op.php	
Fonctions Autorisées : toute la lib standard PHP	
Remarques : n/a	


Ce programme php prend 3 paramètres. Le second est une opération arithmétique parmi les suivantes : '+', '-', '*', '/', '%'. Les premiers et troisièmes sont des nombres. Il s'agit de réaliser l'opération et d'afficher le résultat. Le programme ne doit pas effectuer de gestion d'erreur, hormis le nombre de paramètres fournis. Espaces et tabulations peuvent être présents dans les 3 paramètres.

```
$> ./do_op.php
Incorrect Parameters
$> ./do_op.php 1 + 3
4
$> ./do_op.php " 1 " + " 3"
4
$> ./do_op.php " 1 " * " 3"
3
$> ./do_op.php 42 "% " 3
0
```

Note : respectez le message d'erreur.

Chapitre XIV

Exercice 11 : do_op_2


	Exercice : 11
do_op_2	
Dossier de rendu : <i>ex11/</i>	
Fichiers à rendre : do_op_2.php	
Fonctions Autorisées : toute la lib standard PHP	
Remarques : n/a	

Cette fois-ci, un seul paramètre au menu. Celui-ci contient l'intégralité du calcul à effectuer. Ce calcul sera toujours de la forme *chiffre opérateur chiffre*. Un nouveau message d'erreur "Syntax Error" vient compléter le précédent dans le cas où la syntaxe ne serait pas correcte. Il peut ne pas y avoir d'espace entre les chiffres et l'opérateur, ou bien plusieurs. Le résultat attendu reste le même.

```
$> ./do_op_2.php
Incorrect Parameters
$> ./do_op_2.php toto
Syntax Error
$> ./do_op_2.php "42*2"
84
$> ./do_op_2.php " 42 / 2 "
21
$> ./do_op_2.php "six6*7sept"
Syntax Error
$> ./do_op_2.php '`rm -rf ~/`;'
Syntax Error
```

Chapitre XV

Exercice 12 : search_it !


	Exercice : 12
	search_it !
	Dossier de rendu : <i>ex12/</i>
	Fichiers à rendre : search_it!.php
	Fonctions Autorisées : toute la lib standard PHP
	Remarques : n/a

Votre but est de réaliser un programme qui affiche la valeur correspondante à une clef donnée comme premier paramètre, parmi les couples “clef :valeur” présents en nombre indéterminé passés en paramètres suivants.

```
$> ./search_it!.php
$> ./search_it!.php toto
$> ./search_it!.php toto "key1:val1" "key2:val2" "toto:42"
42
$> ./search_it!.php toto "toto:val1" "key2:val2" "toto:42"
42
$> ./search_it!.php "toto" "key1:val1" "key2:val2" "0:hop"
$> ./search_it!.php "0" "key1:val1" "key2:val2" "0:hop"
hop
$>
```

Chapitre XVI

Exercice 13 : sing_it !


	Exercice : 13
sing_it !	
Dossier de rendu : <i>ex13/</i>	
Fichiers à rendre : sing_it!.php	
Fonctions Autorisées : toute la lib standard PHP	
Remarques : n/a	

Comme vous l'avez peut-être remarqué, la pointe d'humour qui caractérisait les épreuves d'hier et les premières de ce sujet, a quelque peu disparue. Qu'à cela ne tienne, nous allons y remédier. Votre objectif est d'être parmi les tous premiers à aller chanter à Kwame la chanson complète dont les paroles forment le préambule. Kwame doit retenir votre login et le communiquer à la pedago pour obtenir les points de cet exercice (cela ne sera pas bloquant pour la suite). N'attendez pas, à un moment il en aura marre, et ne voudra plus écouter personne.

```
$> ./sing_it!.php
$> ./sing_it!.php "mais pourquoi cette demo ?"
Tout simplement pour qu'en feuilletant le sujet
on ne s'aperçoive pas de la nature de l'exo
$> ./sing_it!.php "mais pourquoi cette chanson ?"
Parce que Kwame a des enfants
$> ./sing_it!.php "vraiment ?"
Nan c'est parce que c'est le premier avril
$> ./sing_it!.php "vraiment ?"
Oui il a vraiment des enfants
$>
```

Chapitre XVII

Exercice 14 : L'agent se tâte

	Exercice : 14
	L'agent se tâte
	Dossier de rendu : <i>ex14/</i>
	Fichiers à rendre : agent_stats.php
	Fonctions Autorisées : toute la lib standard PHP
	Remarques : n/a

Après l'agent d'hier (celui qui se tasse), voici celui qui hésite et qui a des choix à faire. Parmi les ressources de la journée, vous avez quelques fichiers de peer-notes. L'idée est choisir justement comment les utiliser. Plusieurs solutions :

- option “moyenne” qui calcule la moyenne de toutes les notes hors moulinette
- option “moyenne_user” qui calcule la moyenne par user et par ordre alphabétique
- option “ecart_moulinette” qui calcule la moyenne par user des écarts entre une note reçue et celle de la moulinette

Le fichier de note se lit sur l'entrée standard, et l'option est passée en paramètre du programme. Dans les cas 2 et 3, si un user n'a aucune note, il n'est pas affiché.

```
$> cat peer_notes_1.csv | ./agent_stats.php
$> cat peer_notes_1.csv | ./agent_stats.php moyenne
9.8621262458472
$> cat peer_notes_1.csv | ./agent_stats.php moyenne_user
adam_e:9.05555555555556
bertrand_y:7.9473684210526
bruce_w:9.0434782608696
clark_k:10.464285714286
david_a:8.68
dexter_m:8.9

[....]

sandra_n:11.181818181818
steve_j:11.5
trevor_r:6.1052631578947
$>
```

```
$> cat peer_notes_1.csv | ./agent_stats.php ecart_moulinette
adam_e:3.05555555555556
bertrand_y:-1.0526315789474
bruce_w:-9.9565217391304
clark_k:0.46428571428571

[.....]

steve_j:10.5
trevor_r:-12.894736842105
$>
```