



Initiation à la programmation PHP

Staff 42 pedago@42.fr

*Résumé:
Ce document est le sujet de la journée d'initiation au PHP.*

Table des matières

I	Consignes	2
II	Procédure de rendu : Git	3
III	Préambule	5
IV	Exercice 00 : HW	7
V	Exercice 01 : XXX !	8
VI	Exercice 02 : Odd-Even	9
VII	Exercice 03 : ft_split	10
VIII	Exercice 04 : aff_param	11
IX	Exercice 05 : epur_str	12
X	Exercice 06 : ssap	13
XI	Exercice 07 : rostring	14
XII	Exercice 08 : ft_is_sort	15
XIII	Exercice 09 : ssap - le retour -	16
XIV	Exercice 10 : do_op	17
XV	Exercice 11 : do_op_2	18
XVI	Exercice 12 : search_it !	19

Chapitre I

Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices.
- Vous serez corrigés par les autres participants du programme d'initiation. C'est le "peer-correcting" de la pédagogie 42 !
- 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 / man / Internet /`
- Lisez attentivement les exemples. Les exercices pourraient bien requérir des choses qui y sont précisées, et non dans le sujet...

Chapitre II

Procédure de rendu : Git

- Votre rendu est collecté via un serveur distant. Cela signifie que vous devrez envoyer sur ce serveur votre repertoire de rendu avec vos exercices suivant une arborescence précise.
- Le logiciel utilisé pour ce faire s'appelle **git**.
- Git permet de faire beaucoup de choses, vous trouverez ci-dessous les principales commandes :
- **Git clone** vous permet de **cloner** un repertoire présent sur un serveur directement dans votre repertoire courant. Cela signifie que tous les fichiers présents que vous y aviez mis depuis un autre poste se retrouveront sur votre ordinateur et pourront être modifiés et réenvoyés avec les commandes git suivantes. Cette commande est aussi importante quand vous souhaitez vérifier ce que vous avez "pushé" sur le serveur, n'hésitez pas à en abuser.

```
?> git clone vogsphere@vogsphere.42.fr:piscine/truc/machin/creme
Cloning into 'piscine'[...]
?>
```

- **Git add** vous permet d'ajouter à une liste de fichiers surveillés un ou plusieurs fichiers/repertoires. Ils seront ajoutés en l'état ce qui veut dire que si vous les modifiez par la suite, il faudra les ajouter à nouveau pour mettre à jour la liste.

```
?> git add file directory
?>
```

- **Git commit -m** vous permet de "fixer" votre liste de fichiers surveillés afin de préparer leur envoi sur le serveur. Le message est obligatoire et permet d'avoir un historique des créations et modifications que vous avez effectué au fur et à mesure de votre travail.

```
?> git commit -m "Ajout des exercices X Y Z"
[master (root-commit) 4e8b2aa] Ajout des exercices X Y Z
3 file changed, 500 insertions(+)
create mode 100755 X Y Z
?>
```

- **Git push origin master**, la commande ultime, celle qui vous permet d'envoyer sur le serveur vos exercices qui y seront collectés par la Moulinette. Vérifiez bien son retour, le moindre message d'erreur signifie que vous n'avez rien pu envoyer et que par conséquent votre répertoire sur le serveur est vide (ce qui serait dommage pour votre note). Si vous avez le moindre doute, contactez un membre du staff 42.

```
?> git push origin master
[...]  
X files written.  
?>
```



L'adresse git de votre dépôt se trouve sur l'intranet et dans le fichier config du répertoire `.git` de votre dossier de rendu, c'est avec elle que vous pourrez "git clone".

Chapitre III

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 IV

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 est un langage très approchable. Les variables, ces mots avec un dollar qui permettent de stocker des valeurs, sont faciles d'emploi, et comme tout langage de script, il suffit de lancer son fichier avec des commandes pour qu'elles s'exécutent.

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
$>
```


Chapitre V

Exercice 01 : XXX !

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

Réalisez un programme PHP qui affiche 1000 fois la lettre X et un retour à la ligne.

Facile ? Oui ! Ajoutons donc la contrainte suivante : votre programme ne doit pas dépasser les 100 chars.


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

[TL;DP]

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
$>
```

Chapitre VI

Exercice 02 : Odd-Even

	Exercice : 02
Odd-Even	
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>	


Faites un programme PHP qui vous demande gentiment un numéro, et qui vous répond s'il est pair 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 (oui, vous devez gérer les cas d'erreur!). 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 VII

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 VIII

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 IX

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 X

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 XI

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 XII

Exercice 08 : ft_is_sort

	Exercice : 08
	ft_is_sort
	Dossier de rendu : <i>ex08/</i>
	Fichiers à rendre : ft_is_sort.php
	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 XIII

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 XIV

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 XV

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 *nombre opérateur nombre*. 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 "six*7sept"
Syntax Error
$> ./do_op_2.php '`rm -rf ~/`;'
Syntax Error
```

Chapitre XVI

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
$>
```