



Formation PHP Symfony - 0

Starting

Résumé: Ce document rassemble les exercices visant à faire pratiquer les bases du langage PHP.

Version: 1.1

Table des matières

I	Préambule	2
II	Règles communes	3
III	Consignes spécifiques à cette journée	4
IV	Exercice 00	5
V	Exercice 01	6
VI	Exercice 02	7
VII	Exercice 03	8
VIII	Exercice 04	9
IX	Exercice 05	10
X	Exercice 06	11
XI	Rendu et peer-évaluation	13

Chapitre I

Préambule

Quelques sages citations du passé :



640K ought to be enough for anybody.



Computers in the future may weigh no more than 1.5 tons.



We will never make a 32-bit operating system.



Spam will be a thing of the past in two years' time.

Chapitre II

Règles communes

- Votre projet doit être réalisé dans une machine virtuelle.
- Votre machine virtuelle doit avoir tout les logiciels necessaire pour réaliser votre projet. Ces logiciels doivent être configurés et installés.
- Vous êtes libre sur le choix du systmème d'exploitation à utiliser pour votre machine virtuelle.
- Vous devez pouvoir utiliser votre machine virtuelle depuis un ordinateur en cluster.
- Vous devez utiliser un dossier partagé entre votre machine virtuelle et votre machine hôte.
- Lors de vos évaluations vous allez utiliser ce dossier partager avec votre dépôt de rendu.
- Vos fonctions de doivent pas s'arrêter de manière inattendue (segmentation fault, bus error, double free, etc) mis à part dans le cas d'un comportement indéfini. Si cela arrive, votre projet sera considéré non fonctionnel et vous aurez 0 au projet.
- Nous vous recommandons de créer des programmes de test pour votre projet, bien que ce travail **ne sera pas rendu ni noté**. Cela vous donnera une chance de tester facilement votre travail ainsi que celui de vos pairs.
- Vous devez rendre votre travail sur le git qui vous est assigné. Seul le travail déposé sur git sera évalué. Si Deepthought doit corriger votre travail, cela sera fait à la fin des peer-evaluations. Si une erreur se produit pendant l'évaluation Deepthought, celle-ci s'arrête.


Chapitre III

Consignes spécifiques à cette journée

- Si aucune information contraire n'est explicitement présente, vous devez assumer les versions de langages suivantes :
 - PHP - Symfony version LTS
 - HTML 5
 - CSS 3

Chapitre IV

Exercice 00

	Exercice : 00
Exercice 00 : Var	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : var.php	
Fonctions Autorisées :	

Créez le fichier `var.php` qui contient quatre variables `a`, `b`, `c` et `d`. À l'exécution, vous devrez initialiser ces variables et faire en sorte de produire la sortie suivante :

```
# > php var.php
My first variables:
a contains : 10 and has type : integer
b contains : 10 and has type : string
c contains : ten and has type : string
d contains : 10 and has type : double
# >
```


Vous ne devez en aucun cas spécifier en dur les types de vos variables.



Changer uniquement les valeurs de `a`, `b`, `c` ou `d` doit avoir un impact significatif sur la sortie de votre programme.

Chapitre V

Exercice 01

	Exercice : 01
Exercice 01 : CSV	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : csv.php	
Fonctions Autorisées :	


Créez un fichier `csv.php` qui va lire un fichier `ex01.txt` présent dans le même répertoire (présent dans les ressources du sujet).

Ce fichier `txt` contient des valeurs séparées par des virgules. Votre programme devra lire le contenu de ce fichier et afficher ces valeurs, une par ligne.

```
# > cat ex01.txt
first,second,third,fourth
# > php csv.php
first
second
third
fourth
# >
```

Chapitre VI

Exercice 02

	Exercice : 02
Exercice 02 : Anciens temps	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : array2hash.php	
Fonctions Autorisées :	

Créer une fonction **array2hash** qui prend en argument un tableau contenant un ou plusieurs autres tableaux. Ces tableaux contiennent chacun une chaîne de caractères *name* et un entier *age*.


array2hash convertit ce tableau de tableau en **hash** dont les clés représentent l'âge et les valeurs correspondantes, le nom.

Exemple :

```
#> cat test02.php
<?php
include('./array2hash.php');
$array = array(array("Pierre","30"), array("Mary","28"));
print_r ( array2hash($array) );
#> php test02.php
Array
(
    [30] => Pierre
    [28] => Mary
)
```


Chapitre VII

Exercice 03

	Exercice : 03
Exercice 03 : Temps du tri	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : <code>array2hash_sorted.php</code>	
Fonctions Autorisées :	

Créer une fonction `array2hash_sorted` qui prend en argument un tableau contenant un ou plusieurs autres tableaux. Ces tableaux contiennent chacun une chaîne de caractères *name* et un entier *age*.


`array2hash_sorted` convertit ce tableau de tableau en `hash` dont les clés représentent cette fois le nom et les valeurs correspondantes, l'âge. Ce hash doit être trié ensuite par ordre alphabétique inverse.

Exemple :

```
#> cat test03.php
<?php
    include('./array2hash_sorted.php');
    $array = array(array("Pierre","30"), array("Mary","28"), array("Nelly", "22"));
    print_r ( array2hash_sorted($array) );
#> php test03.php
Array
(
    [Pierre] => 30
    [Nelly] => 22
    [Mary] => 28
)
```

Chapitre VIII

Exercice 04


	Exercice : 04
Exercice 04 : Etats & Capitales	
Dossier de rendu : <i>ex04/</i>	
Fichiers à rendre : <code>capital_city_from.php</code>	
Fonctions Autorisées :	

Etant donné les tableaux suivants, écrire une fonction `capital_city_from` qui prend comme argument le nom d'un état et retourne sa capitale. Si la capitale n'existe pas, la fonction doit renvoyer "Unknown".

```
#> cat capital_city_from.php
<?php
[...]
$states = [
    'Oregon' => 'OR',
    'Alabama' => 'AL',
    'New Jersey' => 'NJ',
    'Colorado' => 'CO',
];
$capitals = [
    'OR' => 'Salem',
    'AL' => 'Montgomery',
    'NJ' => 'trenton',
    'KS' => 'Topeka',
];
[...]
#> cat test04.php
<?php
include('./capital_city_from.php');
echo capital_city_from('Oregon');
echo capital_city_from('Origan');
#> php test04.php
Salem
Unknown
```

Chapitre IX

Exercice 05

	Exercice : 05
Exercice 05 : Recherche d'états ou capitales	
Dossier de rendu : <i>ex05/</i>	
Fichiers à rendre : search_by_states.php	
Fonctions Autorisées :	


En reprenant les mêmes tableaux que dans l'exercice précédent, créer une fonction **search_by_states** qui prendra en argument une chaîne de caractères, composée de un ou plusieurs états.

La fonction retournera alors un ensemble de chaînes de caractères formatées comme dans cet exemple :

```
# > cat test05.php
<?php
    include('./search_by_states.php');
    search_by_states("Oregon, trenton, Topeka, NewJersey");
# > php test05.php
Salem is the capital of Oregon.
trenton is the capital of New Jersey.
Topeka is neither a capital nor a state.
NewJersey is neither a capital nor a state.
# >
```

Chapitre X

Exercice 06

	Exercice : 06
Exercice 06 : Mendeleiev table	
Dossier de rendu : <i>ex06/</i>	
Fichiers à rendre : mendeleiev.php	
Fonctions Autorisées :	

Créez un programme qui va ouvrir le fichier **ex06.txt** contenant le tableau de Mendeleiev, formatté d'une certaine manière. Ce programme devra créer le fichier **mendeleiev.html** qui contiendra le code HTML de ce tableau de Mendeleiev en fonction des données du fichier **ex06.txt**, et ce tel que :

- Chaque élément du tableau soit une cellule dans un **tableau HTML**
- Le titre soit dans une balise **h4**
- Les attributs de chaque élément soient dans une **liste HTML**
- Le format général de la table de Mendeleiev soit respecté, notamment les cellules vides, nouvelles lignes, etc.

Des images du Tableau de Mendeleiev peuvent être trouvées facilement sur le web.

Exemple d’affichage HTML :

```
[...]
<table>
  <tr>
    <td style="border: 1px solid black; padding:10px">
      <h4>Hydrogen</h4>
      <ul>
        <li>No 42</li>
        <li>H</li>
        <li> 1.00794 </ li>
        <li>1 electron</li>
      <ul>
    </td>
  </tr>
</table>
[...]
```

Libre à vous de le personnaliser ensuite à souhait pour le rendre plus cool ;)

Chapitre XI

Rendu et peer-évaluation

Rendez votre travail dans votre dépôt Git comme d'habitude. Seul le travail présent dans votre dépôt sera évalué en soutenance. Vérifiez bien les noms de vos dossiers et de vos fichiers afin que ces derniers soient conformes aux demandes du sujet.



L'évaluation se déroulera sur l'ordinateur du groupe évalué.