



Formation PHP - Symfony

D01 - Bases du Langage PHP

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

Table des matières

I	Préambule	2
II	Consignes	3
III	Exercice 00	4
IV	Exercice 01	5
V	Exercice 02	6
VI	Exercice 03	7
VII	Exercice 04	8
VIII	Exercice 05	9
IX	Exercice 06	10

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


Consignes

Sauf contradiction explicite, les consignes suivantes seront valables pour tous les jours de cette Piscine.

- Seul ce sujet sert de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- 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.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices. L'url de votre dépôt GIT pour cette journée est disponible sur votre intranet.
- Vos exercices seront évalués par vos camarades de Piscine.
- En plus de vos camarades, vous pouvez être évalués par un programme appelé la Moulinette. La Moulinette est très stricte dans sa notation car elle est totalement automatisée. Il est donc impossible de discuter de sa note avec elle. Soyez d'une rigueur irréprochable pour éviter les mauvaises surprises.
- Les exercices shell doivent s'exécuter avec `/bin/sh`.
- Vous ne devez laisser aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices dans votre dépôt de rendu.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Toutes les réponses à vos questions techniques se trouvent dans les `man` ou sur Internet.
- Pensez à discuter sur le forum Piscine de votre Intra et sur Slack !
- Lisez attentivement les exemples car ils peuvent vous permettre d'identifier un travail à réaliser qui n'est pas précisé dans le sujet à première vue.
- Réfléchissez. Par pitié, par Thor, par Odin !

Chapitre III

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 quatres 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 IV

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 V

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 VI

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
(
    [Nelly] => 22
    [Mary] => 28
    [Pierre] => 30
)
```


Chapitre VII

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 donnés 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 VIII

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 IX

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, formaté 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 ;)