



Piscine PHP

Jour 03

Staff 42 piscine@42.fr

Résumé:

Ce document est le sujet du jour 03 de la piscine PHP de 42.

Table des matières

I	Consignes	2
II	Préambule	3
III	Exercice ex00 : Installation	4
IV	Exercice ex01 : phpinfo	5
V	Exercice ex02 : print_get	6
VI	Exercice ex03 : cookie_crisp	7
VII	Exercice ex04 : raw_text	8
VIII	Exercice ex05 : read_img	9
IX	Exercice ex06 : members_only	10

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 ! Nom d'une pipe !

Chapitre II

Préambule

Voici ce que Wikipédia a à dire sur les Apaches :

Apache (de apachu, « ennemi » en langue zuñi) est un nom générique donné à différentes tribus indiennes d'Amérique du Nord vivant dans le sud-ouest des États-Unis et le Nord des États mexicains de Chihuahua et du Sonora, formant le territoire de l'Apacheria et partageant la même langue athapascane méridionale (proche des langues athapascanes septentrionales parlées par les indiens d'Alaska et de l'ouest du Canada). Les Navajos parlent une langue très proche et partagent la même culture, ils sont donc souvent considérés comme des Apaches.


Nomades et chasseurs dans un environnement semi-aride, les Apaches furent de farouches guerriers attaquant les peuples cultivateurs dont les Pueblos et d'autres tribus sédentaires de la région. Ils s'opposèrent plus tard aux colons espagnols, puis aux Mexicains et aux colons européens. Ils furent finalement vaincus et décimés par ces derniers à la fin du xix^e siècle et leurs quelques descendants vivent aujourd'hui dans des réserves.

Lorsqu'ils dansaient, les Apaches revêtaient des costumes symbolisant les esprits de la montagne. Cherchant à guérir les malades en éloignant le mauvais sort, ils se paraient de peintures corporelles, de jupes, de masques aux couleurs sombres. Les Apaches croyaient en de nombreux hôtes surnaturels mais surtout en une divinité suprême nommée Yasun.

Leurs chefs les plus célèbres sont Mangas Coloradas, Cochise et Geronimo.

Chapitre III


Exercice ex00 : Installation

	Exercice : 00
Intallation	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre :	
Fonctions Autorisées :	
Remarques : n/a	

Ce premier exercice consiste à réaliser l'installation de votre serveur web avec l'outil PAMP présent sur les dumps. Reportez-vous à la vidéo dans l'e-learning pour l'installation et la configuration. Cela est bien évidemment nécessaire pour tous les autres exercices de la journée.

Chapitre IV

Exercice ex01 : phpinfo


	Exercice : 01
phpinfo	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <code>phpinfo.php</code>	
Fonctions Autorisées : <code>phpinfo()</code>	
Remarques : n/a	

Créez une page `phpinfo.php` qui exécute et affiche le résultat de `phpinfo()` ;

```
$> curl 'http://eXrXpX.42.fr:8080/j03/ex01/phpinfo.php'
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<style type="text/css">
...
<title>phpinfo()</title><meta name="ROBOTS" content="NOINDEX,NOFOLLOW,NOARCHIVE" /></head>
...
$>
```

Chapitre V

Exercice ex02 : print_get


	Exercice : 02
print_get	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : print_get.php	
Fonctions Autorisées : echo	
Remarques : n/a	

Créez une page `print_get.php` qui affiche toutes les variables passées dans l'url
Exemple :

```
$> curl 'http://eXrXpX.42.fr:8080/j03/ex02/print_get.php?login=mmontinet'
login: mmontinet
$> curl 'http://eXrXpX.42.fr:8080/j03/ex02/print_get.php?gdb=pied2biche&barry=barreamine'
gdb: pied2biche
barry: barreamine
$>
```

Chapitre VI

Exercice ex03 : cookie_crisp


	Exercice : 03
print_crisp	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : <i>cookie_crisp.php</i>	
Fonctions Autorisées : <i>echo, setcookie()</i> et <i>time()</i>	
Remarques : <i>n/a</i>	

Créez une page *cookie_crisp.php* qui permet de créer, lire, et effacer un cookie
Exemple :

```
$> curl -c cook.txt 'http://eXrXpX.42.fr:8080/j03/ex03/cookie_crisp.php?action=set&name=plat&value=choucroute'
$> curl -b cook.txt 'http://eXrXpX.42.fr:8080/j03/ex03/cookie_crisp.php?action=get&name=plat'
choucroute
$> curl -c cook.txt 'http://eXrXpX.42.fr:8080/j03/ex03/cookie_crisp.php?action=del&name=plat'
$> curl -b cook.txt 'http://eXrXpX.42.fr:8080/j03/ex03/cookie_crisp.php?action=get&name=plat'
$>
```


Chapitre VII

Exercice ex04 : raw_text

	Exercice : 04
	raw_text
	Dossier de rendu : <i>ex04/</i>
	Fichiers à rendre : raw_text.php
	Fonctions Autorisées : header()
	Remarques : n/a

Créez une page `raw_text.php` qui affiche la même chose à l'écran que l'on regarde son code source avec `curl` ou son rendu html dans Chrome.

```
$> curl 'http://eXrXpX.42.fr:8080/j03/ex04/raw_text.php'
<html><body>Hello</body></html>
$>
```


Si vous aviez `lynx` vous pourriez tester de cette façon (au saut de ligne prêt) :

```
$> lynx -dump 'http://eXrXpX.42.fr:8080/j03/ex04/raw_text.php'
<html><body>Hello</body></html>

$> lynx -source 'http://eXrXpX.42.fr:8080/j03/ex04/raw_text.php'
<html><body>Hello</body></html>
$>
```

Chapitre VIII

Exercice ex05 : read_img

	Exercice : 05
read_img	
Dossier de rendu : <i>ex05/</i>	
Fichiers à rendre : read_img.php	
Fonctions Autorisées : header() , readfile()	
Remarques : n/a	


Créez une page `read_img.php` qui renvoie au navigateur le fichier `42.png` avec le bon Content-Type. Vous trouverez ce fichier dans le sujet sur l'intranet. Vous devez le mettre dans votre dépôt dans `/img/42.png` pour que nous puissions le réutiliser dans plusieurs exercices.



```
$> curl --head http://eXrXpX.42.fr:8080/j03/ex05/read_img.php
HTTP/1.1 200 OK
Date: Tue, 26 Mar 2013 09:42:42 GMT
Server: Apache
X-Powered-By: PHP/5.4.26
Content-Type: image/png
$>
```

Chapitre IX

Exercice ex06 : members_only

	Exercice : 06
members_only	
Dossier de rendu : <i>ex06/</i>	
Fichiers à rendre : members_only.php	
Fonctions Autorisées : header() , echo , \$_SERVER , file_get_contents , base64_encode	
Remarques : n/a	

Créez une page `members_only.php` qui réclame un login/pass au niveau du protocole http.

Si le login est “zaz” et le pass “jaimelespetitsponeys” la réponse doit être une page html contenant un tag `img` dont la source est directement les datas de l’image “/img/42.png” et non son url. (attention nous allons sûrement changer le contenu de 42.png pour la correction, donc pas de valeur de contenu en dur)

Vous devrez reproduire l’exemple suivant :

```
$> curl --user zaz:jaimelespetitsponeys http://eXrXpX.42.fr:8080/j03/ex06/members_only.php
<html><body>
Bonjour Zaz<br />
<img src='data:image/png;base64,iVBORwOKGgoAAAAA...
...
...
...6MIHnr2t+ee04Fr+v/H80AmcVvzqAfAAAAAE1FTkSuQmCC'>
</body></html>
$>
```

Si le login/pass ne correspond pas à “zaz” / “jaimelespetitsponeys”, renvoyer un message d’erreur exactement comme dans l’exemple suivant

```
$> curl -v --user root:root http://eXrXpX.42.fr:8080/j03/ex06/members_only.php
* About to connect() to j03.local.42.fr port 8080 (#0)
*   Trying 127.0.0.1...
* connected
* Connected to j03.local.42.fr (127.0.0.1) port 8080 (#0)
* Server auth using Basic with user 'root'
> GET /ex05/members_only.php HTTP/1.1
> Authorization: Basic cm9vdDpyb290
> User-Agent: curl/7.24.0 (x86_64-apple-darwin12.0) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.5
> Host: j03.local.42.fr:8080
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 401 Unauthorized
< Date: Tue, 26 Mar 2013 09:42:42 GMT
< Server: Apache
< X-Powered-By: PHP/5.4.26
< WWW-Authenticate: Basic realm='Espace membres'
< Content-Length: 72
< Connection: close
< Content-Type: text/html
<
<html><body>Cette zone est accessible uniquement aux membres du site</body></html>
* Closing connection #0
$>
```