

## 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

-	Consignes	_
II	Préambule	3
III	Exercice ex00 : vhost	4
IV	Exercice ex01 : phpinfo	5
$\mathbf{V}$	Exercice ex02 : print_get	6
VI	${\bf Exercice}{\bf ex03:cookie\_crisp}$	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 dépot 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 <u>ne devez</u> laisser dans votre répertoire <u>aucun</u> autre fichier que ceux explicitement specifiés 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 déja. 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 xixe 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: vhost

Exercice: 00	
vhost	
Dossier de rendu : $ex00/$	
Fichiers à rendre : httpd-app.conf, httpd-vhosts.conf et	
bitnami-apps-vhosts.conf	
Fonctions Autorisées : aucune	/
Remarques: n/a	

Aujourd'hui, on va faire du PHP sur le serveur web.

On va pour être à l'aise installer MAMP :

- $\bullet$ installez "Bitnami MAMP" dans ~/mamp/ (ne décochez rien pendant l'installation, installez tout)
- configurez un vhost "j03.local.42.fr" dans  $\sim$ /mamp/apps/j03/ (utilisez la même structure pour votre application que  $\sim$ /mamp/docs/demo/)
- $\bullet$  mettez le fichier httpd-app.conf, httpd-vhosts.conf et bitnami-apps-vhosts.conf dans votre rendu dans ex00/

#### Chapitre IV

## Exercice ex01: phpinfo

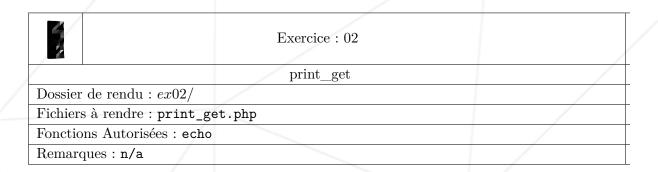
	Exercice: 01	
	phpinfo	
Dossier de rendu : $ex01/$		
Fichiers à rendre : phpinfo.php		
Fonctions Autorisées : phpinfo()		
Remarques : n/a		

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

```
$> curl 'http://j03.local.42.fr:8080/ex01/phpinfo.php'
<!D0CTYPE 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  $ex02 : print\_get$ 

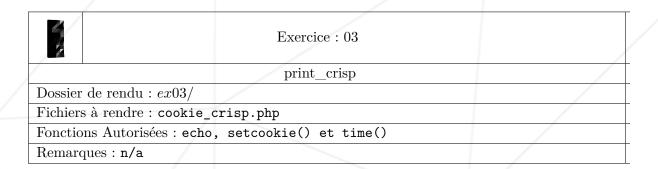


Créez une page print\_get.php qui affiche toutes les variables passées dans l'url Exemple :

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

## Chapitre VI

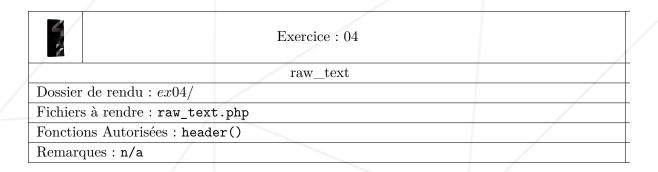
Exercice ex03 : cookie\_crisp



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

#### Chapitre VII

Exercice  $ex04 : raw\_text$ 



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

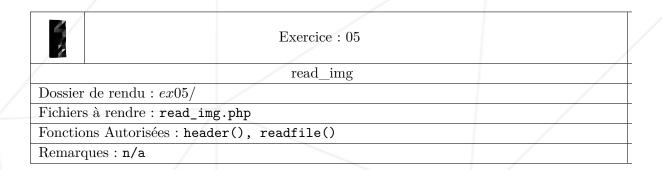
Si vous aviez lynx vous pourriez tester de cette façon (au saut de ligne prêt):

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

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

## Chapitre VIII

Exercice ex05 : read\_img



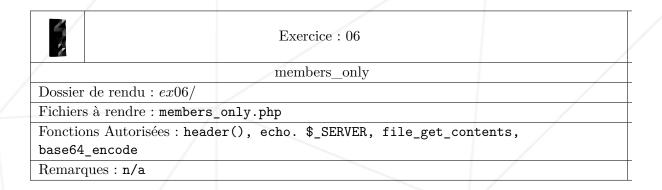
Créez une page read\_img.php qui renvoie au navigateur le fichier 42.png avec le bon Content-Type. Vous trouverez ce fichier dans le sujet sur l'intranet. Vous devez le mettre dans votre dépôt dans "/img/42.png" pour que nous puissions le réutiliser dans plusieurs exercices.



```
$> curl --head http://j03.local.42.fr:8080/ex05/read_img.php
HTTP/1.1 200 0K
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



Créez une page members\_only.php qui réclame un login/pass au niveau du protocol 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://j03.local.42.fr:8080/ex06/members_only.php
<html><body>
Bonjour Zaz<br />
<img src='data:image/png;base64,iVBORwOKGgoAAAA...
...
...
...
...
...
6MIHnr2t+ee04Fr+v/H80AmcVvzqAfAAAAAE1FTkSuQmCC'>
</body></html>
$>
```

Piscine PHP Jour 03

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://j03.local.42.fr:8080/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 /ex06/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: */*
 \ensuremath{\mathsf{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
```