≡   🔍 (https://profile.intra.42.fr/searches)   **xizhou**

(https://profile.intra.42.fr)

SCALE FOR PROJECT PISCINE PHP (/PROJECTS/PISCINE-PHP) / DAY 03 (/PROJECTS/42-PISCINE-C-FORMATION-PISCINE-PHP-DAY-03)

You should evaluate 1 student in this team

★

Git repository

```
git@vogsphere-v2.hive.fi:vogsphere/intra-uuid-d347a045-2d5e-40cd-8de  📋
```

Guidelines

You already know how peer2peer evaluation work by now and during
this PHP Piscine, it works as any other Piscine projects you have evaluated
so far. Take the time to discuss the assignments and to make sure that
the evaluated has understood the notions of the day.

Attachments

☐ Subject (https://cdn.intra.42.fr/pdf/pdf/10078/d03.en.pdf)   🔗 Install bitnami MAMP (https://youtu.be/RGJ3HnjDJtA)

☐ Resources (/uploads/document/document/1178/resources.tar)

Preliminaries

There are a lot of exercises, you will have to manage your time accordingly. Execute the simplest, fastest and most efficient tests. Be careful, this evaluation form won't necessarily tell you what to test, you have to come up with your own! As you most likely have created a bunch of them for your own assignments, feel free to re-use them. Once an exercise is wrong, the evaluation must stop but you can still discuss the rest of the assignments.

The basics

- The whole team must be present.
- Only grade the work that is in the student or group's GiT
repository.
- You must use Chrome for this defense.
- If there is nothing in the repository, the evaluation stops here.
Discuss what went wrong and how to avoid facing the same
situation tomorrow.

☑ Yes   ✕ No

Ex00

Bitnami MAMP

Is bitnami MAMP installed? 👀
Launch it now if it's not already!

☑ Yes   ✕ No

Ex01

phpinfo

Open the browser or curl the following address (modify the URL if necessary):

```
http://localhost:8080/d03/ex01/phpinfo.php
```

You should see the `phpinfo()` page. Check in the
source code that this function was used.

☑ Yes   ✕ No

Ex02

print_get

Execute the examples provided in the subject and then make
your own tests.

Check with, at least, the following example:

```
$> curl 'http://localhost:8080/d03/ex02/print_get.php?dubo=doyouwantbread&zaz=
dubo: doyouwantbread
zaz: mylittlepony
6x7: 42
$>
```

☑ Yes   ✕ No

Ex03

cookie_crisp

Execute the examples provided in the subject and then make
your own tests.

Create a cookie and check the set-cookie response header to see if the cookie was properly set.

```
$> curl -c cook.txt -v 'http://localhost:8080/d03/ex03/cookie_crisp.php?action=
```

Check that we can read the newly created cookie.

```
ok.txt 'http://localhost:8080/d03/ex03/cookie_crisp.php?action=get&name=drink'
```

Destroy the cookie and check that you are not able to get it anymore.

```
$> curl -c cook.txt 'http://localhost:8080/d03/ex03/cookie_crisp.php?action=de
$> curl -b cook.txt 'http://localhost:8080/d03/ex03/cookie_crisp.php?action=ge
$>
```

If everything works as expected, tick "Yes", otherwise tick "No" and the evaluation stops here.

☑ Yes   ✕ No

Ex04

raw_text

Access the file in the browser (`http://localhost:8080/d03/ex04/raw_text.php`) and access it with `curl` :

```
$> curl -v 'http://localhost:8080/d03/ex04/raw_text.php'
[...]
>
< Content-Type: text/plain
<
<html><body>Hello</body></html>
[...]
$>
```

The content must be **exactly** the same.

⊘ Yes   ✕ No

Ex05

read_img

Check first in Chrome that the image is properly displayed.
Then check with `curl` that the Content-type is `image/png` .

```
$> curl --head 'http://localhost:8080/d03/ex05/read_img.php'
HTTP/1.1 200 OK
Date: XXX
Server: XXX
X-Powered-By: XXX
Content-Type: image/png
$>
```

⊘ Yes   ✕ No

Ex06

members_only

First, test without login or password. The page must return an error message and the header must contain `HTTP/1.1 401 Unauthorized` .

```
$> curl -v 'http://localhost:8080/d03/ex06/members_only.php'
[...]
< HTTP/1.0 401 Unauthorized
[...]
< WWW-Authenticate: Basic realm=''Member area''
<html><body>That area is accessible for members only</body></html>
$>
```

Check that you get the error message when trying with a wrong password:

```
$> curl --user zaz:root 'http://localhost:8080/d03/ex06/members_only.php'
&lt;html&gt;&lt;body&gt;That area is accessible for members only&lt;/body&gt;&
```

Finally, try with the correct login and password combination. Check that an HTI

```
<pre>$> curl --user zaz:Ilovemylittleponey http://localhost:8080/d03/ex06/memb
<html><body>Hello Zaz<br />
<img src='data:image/png;base64,iVBORw0KGgoAAAA...
...
...
...6MIHnr2t+eeO4Fr+v/H80AmcVvzqAfAAAAAElFTkSuQmCC'>
</body></html>
$>
```

☑ Yes   ✕ No

Ratings

Don't forget to check the flag corresponding to the defense

✔ Ok

📄 Empty work   🗐 Cheat   🎇 Crash   🚫 Forbidden function

Conclusion

Leave a comment on this evaluation

Nora did a great job
and completed all the

Finish evaluation

Terms of use for video surveillance (https://signin.intra.42.fr/legal/terms/1)
Rules of procedure (https://signin.intra.42.fr/legal/terms/4)
Declaration on the use of cookies (https://signin.intra.42.fr/legal/terms/2)
General term of use of the site (https://signin.intra.42.fr/legal/terms/6)
Legal notices (https://signin.intra.42.fr/legal/terms/3)
Privacy policy (https://signin.intra.42.fr/legal/terms/5)