

SCALE FOR PROJECT PISCINE PHP (/PROJECTS/PISCINE-PHP) / DAY 01 (/PROJECTS/42-PISCINE-C-**FORMATION-PISCINE-PHP-DAY-01)**

Guidelines

You know a little about how peer2peer correction by now it works as usual.

Preliminaries

Let's do it. This is the scale of the first day of PHP in the PHP Piscine. There is a lot of exercises, don't stay to long on each exercises else you'll never finish. Execute the simplest, fastest and most efficient tests. You should have executed a lot of tests for your own exercises just redo those. BE CAREFUL: this scale will not tell you what tests to do. Yes, some will maybe execute more(+) strange tests, that might end end break you programs (and stop the evaluation). Solution: multiply defense to avoid this problem. Edit: If it's not clear for everybody. As soon as an exercise is wrong, the evaluation stops the next exercises won't be graded. You can still discuss exercises but can't grade them. It's also explained in the 7th point of the subject.

The basics

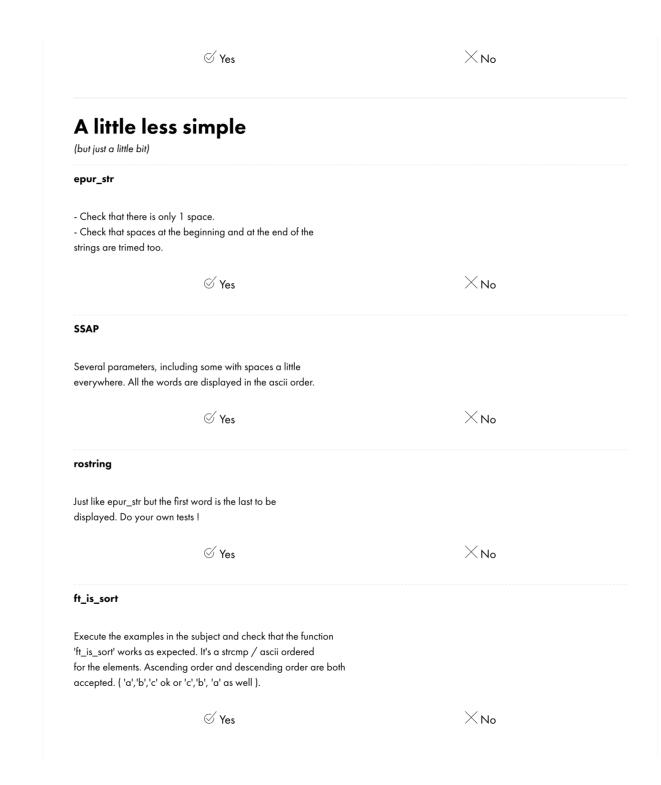
- Is everybody present?
- Only grade the work that is in the student or group's GiT repository.
- Chrome must be used for this defense. If there is nothing in the repository put 0 and stop evaluation. Discuss what went wrong (not necessarily on the technical side) and how to avoid having the same situation tomorrow.



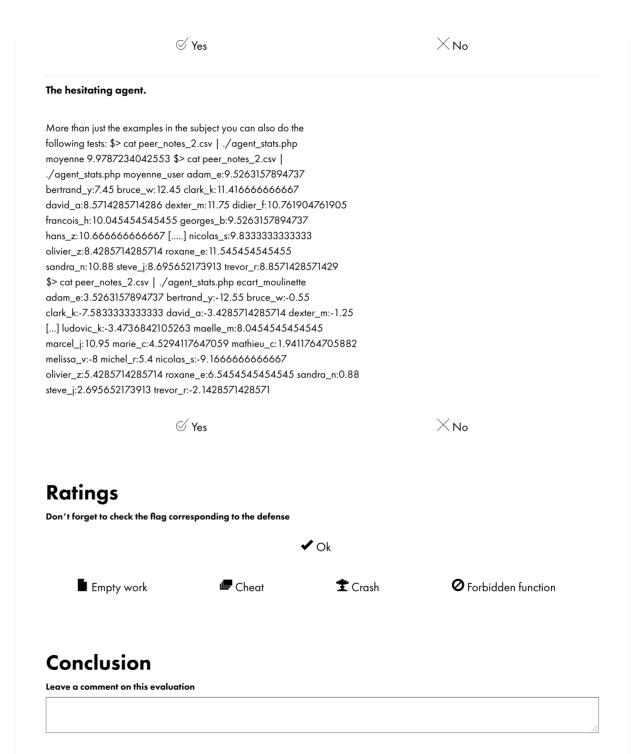


The easy ones

The first exercises are pretty simple		
HW		
If the program displays Hello World, it's good. Else the evaluation stops.		
⊘ Yes	XNo	
mIX		
Execute 'wc -c' to check that the php script is under 100 characters (the whole file, everything included). Also check he number of X, especially the fact that there isn't 999 or 1001. It's a make-or-break. (reminder: if it breaks the evaluation stops).		
⊘ Yes	×No	
To the divine		
The program must ask a number in the command line and say if it's odd or even The result must be ok The message display must be accurate The characters other that numbers must be detected Test at least the examples given in the subject, more are better No test with float numbers.		
⊘ Yes	XNo	
ft_split		
Execute the examples in the subject and check that it works as expected. Execute extra tests.		
⊘ Yes	×No	
aff_param		



ssap2	
Reminder: the different parameters are splitted into words (separated by spaces). The words must be sorted in the specific order explained in the subject. The comparisons are made character by character. => az < a4 < a# < Cb < 42 < 5v < _too < _TOTO	
⊗ Yes	$ imes_{No}$
do_op	
So let's do it. Test properly do_op. No error management required except for the number of arguments. So you have to test with 3 arguments, put 3 correct arguments, eventually with spaces. No division by 0.	
⊗ Yes	imesNo
do_op_2 - la version ++	
this time if there is more or less than one argument, it must have the same error message from the last exercise. Check that something that doesn't look like a calculation displays a syntax error. If you test the last example (the one with rm -rf between backquotes), do it on the other student account. Not yours But it's totally up to you. No division by 0.	
⊗ Yes	imesNo



Preview!!!

General term of use of the site (https://signin.intra.42.fr/legal/terms/6)

Privacy policy
(https://signin.intra.42.fr/legal/terms/5)

Legal notices
(https://signin.intra.42.fr/legal/terms/3)

Declaration on the use of cookies (https://signin.intra.42.fr/legal/terms/2)

Terms of use for video surveillance
(https://signin.intra.42.fr/legal/terms/1) (ht

(https://sigr