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.

✓ Yes ✗ No

The easy ones

The first exercises are pretty simple...

HW

If the program displays Hello World, it's good. Else the evaluation stops.

✓ Yes ✕ No

mlX

Execute 'wc -c' to check that the php script is under 100 characters (the whole file, everything included). Also check the 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 ✕ No

ft_split

Execute the examples in the subject and check that it works as expected. Execute extra tests.

✓ Yes ✕ No

aff_param

Does it work? Tests are up to you.

✓ Yes ✕ No

A little less simple

(but just a little bit)

epur_str

- Check that there is only 1 space.
- Check that spaces at the beginning and at the end of the strings are trimed too.

✓ Yes ✕ No

SSAP

Several parameters, including some with spaces a little everywhere. All the words are displayed in the ascii order.

✓ Yes ✕ No

rostring

Just like epur_str but the first word is the last to be displayed. Do your own tests !

✓ Yes ✕ No

ft_is_sort

Execute the examples in the subject and check that the function 'ft_is_sort' works as expected. It's a strcmp / ascii ordered for the elements. Ascending order and descending order are both accepted. ('a','b','c' ok or 'c','b', 'a' as well).

✓ Yes ✕ No

more difficult

it's getting complicated!

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 < _toto < _TOT0

✓ Yes ✕ 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 ✕ No

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 ✕ No

search_it!

Who seeks finds ! At least the subject example must be tested. Don't hesitate to do more test.

✓ Yes ✗ No

The hesitating agent.

More than just the examples in the subject you can also do the following tests: $> cat peer_notes_2.csv | ./agent_stats.php moyenne 9.9787234042553 $> cat peer_notes_2.csv | ./agent_stats.php moyenne_user adam_e:9.5263157894737 bertrand_y:7.45 bruce_w:12.45 clark_k:11.416666666667 david_a:8.5714285714286 dexter_m:11.75 didier_f:10.761904761905 francois_h:10.045454545455 georges_b:9.5263157894737 hans_z:10.666666666667 [.....] nicolas_s:9.8333333333333 olivier_z:8.4285714285714 roxane_e:11.545454545455 sandra_n:10.88 steve_j:8.695652173913 trevor_r:8.8571428571429 $> cat peer_notes_2.csv | ./agent_stats.php ecart_moulinette adam_e:3.5263157894737 bertrand_y:-12.55 bruce_w:-0.55 clark_k:-7.5833333333333 david_a:-3.4285714285714 dexter_m:-1.25 [...] ludovic_k:-3.4736842105263 maelle_m:8.0454545454545 marcel_j:10.95 marie_c:4.5294117647059 mathieu_c:1.9411764705882 melissa_v:-8 michel_r:5.4 nicolas_s:-9.1666666666667 olivier_z:5.4285714285714 roxane_e:6.5454545454545 sandra_n:0.88 steve_j:2.695652173913 trevor_r:-2.1428571428571

✓ 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

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)

(https://sig