CS Homework #17

Deadline: February 20th, 9:00 pm.

- *Save your code as lastname_homework17.py and submit on Google Classroom.*
- *Please, run your code before submitting.*
- *If you get an error, try to fix it before submitting your homework.*
- *If you get help from anyone, please, make sure that you actually understand the solution.*

Task 1

Create a **lambda function** that returns a square root of a variable (reminder: square root is the same as power of 0.5). Test your function using your own example. Using f-strings print the input and output.

Task 2

Improve the lambda function from Task 1 by rounding your result to two decimal points. Test your function using your own example. Using f-strings print the input and output.

Task 3

Go back to your function from Task 1 (the one without rounding). Test it again using f-strings. This time format your output using the following code {val:.2f}, where *val* is the value that you print. What is the difference between the rounding approaches in Task 2 and Task 3. Print your answer to this question, using print().

Task 4

Create a lambda function that return the power of 2 of a number if the number is even and the power of 3 if the number is odd. For example, the argument 4 will return 16, while the argument 5 will return 125. Hint. you can use the following format in your lambda function: (do something) if (a condition) else (do something else).

Task 5

Create a function (let's call it *create_dictionary*) that accepts two lists and creates a corresponding dictionary, similar to how we did it before: dict(zip(list A, list B)). This time, create a LOCAL function (let's call it *verify_lists*) that verifies that both list A and list B are actually lists and have equal size. If verification fails, *create_dictionary* must return an empty dictionary. Verify that you are not able to run it outside of your main function.

Task 6

This time both create_dictionary and verify_lists are global. You still need to use verify_lists, when using create_dictionary. Same as in Task 5, return an empty dictionary if the arguments are not valid.