

## Assignment 4.4: Lists

### Summary

In this graded assignment, you will write a Python program that stores data in and manipulates lists, and then use the Coursera automatic testing utility to submit and evaluate it.

### Learning Outcomes

In completing this assignment, you will:

- Write a Python program using lists
- Use built-in Python functionality to add and remove data from a list
- Use the Coursera automatic testing utility to get feedback on the correctness of your program

### Description

In this assignment, you will write a very simple program that manipulates a list of numbers.

The list that you will manipulate is called “values”, and the program has been started for you below:

1	def test() : # do not change this line!
2	values = [4, 7, 8, 9, -5] # do not change this line!
3	
4	# write your code here so that it modifies values
5	
6	# be sure to indent your code!
7	
8	print(values)
9	return values # do not change this line!
10	# do not write any code below here
11	
12	test() # do not change this line!
13	# do not remove this line!

The automatic grading utility that we will use for this activity requires that your code be put into a **function**, which is a named group of code; we will learn more about functions in a subsequent lesson. The function you are creating here is called “test” as you can see on line 1, and ends on line 9.

**BEFORE YOU BEGIN**, click the “Run” button to the right of the code block as you have done in previous activities. This will cause the “test” function to execute, which will create the list of values on line 2, print it on line 8, and then end. You can continue to use “print” statements and run the program in the code block as you develop your program.

Starting at line 4, modify the code so that it does the following:

- See if the last element of the list is negative. If so, remove it from the list.
- Then, calculate the average of the first two elements of the list and insert it in between those two elements.
- Last, swap the values of the first and last elements of the list.

At the end of the program, print out the list using “print(values)”. In this case, it should print [9, 5.5, 7, 8, 4]

### IMPORTANT NOTES!

- In order for your program to be automatically graded as part of this activity, all of your code must be within the “test” function that starts on line 1 and ends with the line “return values”. **Please do not change either of those lines**, otherwise your code will not be correctly executed.
- Also, **please be sure to indent all of the code starting with line 2** and ending with “return values” so that it is considered to be part of the function, and thus is part of the code that is evaluated by the automatic grading utility. Be sure to double-indent anything that was singly indented in your original code, and so on. The Coursera code editor will likely do the correct indentation for you, but everything you write in this activity should be indented.
- Last, note that the automatic grading utility will ignore any code that comes after the line “return values”, so **please do not write any code after that line**, and **please do not modify any of the code that comes after it**.

### Hints:

- *Review the previous lesson and make sure you understand how indices are used to access individual elements in a list. Keep in mind that indices are 0-based, meaning that the first element is at index 0, and you can use negative indices as well, e.g. index -1 is used to reference the last element.*
- *Review the use of the append, insert, remove, and pop functions for modifying the list contents.*
- *Swapping values can be difficult so be sure you pay close attention to how you do that. You may want to change the values directly, or do some combination of removing/adding elements.*
- *As in previous activities, don’t forget that you can use the “print” function to print out intermediate values as your code is performing operations so that you can see what is happening.*

Use the “Run” button to the right of the code block to run your code and see the values in the list before submitting your program for grading. Here are some common Python errors that you may run into:

- **IndentationError:** this means that your code is not properly indented. Keep in mind that all the code you write for this activity should be within the body of the “test” function and indented by at least one tab.
- **SyntaxError:** this is the general error message for invalid Python code, but if you see one that reads “return outside function,” that means the final line of the function (“return values”) is not properly indented; it should be indented by exactly one tab.
- **IndexError:** it is likely that you are trying to access an illegal index in the list. Keep in mind that for a list of  $N$  elements, the valid indices range from 0 to  $N-1$  going forward, and from  $-1$  to  $-1*N$  going backwards.
- **ValueError:** indicates that you are trying to remove a value in the list that does not exist.
- **TypeError:** this typically relates to using characters, numbers, etc. in invalid ways, but if you see a message involving “positional arguments,” it may mean that you changed the “test” function that is used by the grading utility. Please do not change the line “def test():” on line 1 of the starter code, or the line “test()” at the end of the program.

After you have run your program using the “Run” button and believe that it is producing the correct output, accept the terms of the Coursera Honor Code and then click the “Submit Quiz” button below to submit this assignment and have it graded.

A few seconds after you submit the assignment, you will see the result at the top of the screen. If it reads “Congratulations! You passed!” then you are ready to proceed to the next lesson.

However, if it reads “Try again once you are ready.” then this means that the automatic grading program indicated that your program is not correct. You can find the error message from the grading program beneath your code. In that case, click the “Retake” button to go back to the quiz and try again.

If you believe that your code was correct, be sure to click the “Run” button before submitting and inspect the result of the “print” statement that is right before the “return” statement. This allows you to see the output that your code is producing before it is evaluated. If it looks right, but the automatic grading utility is still indicating that it is incorrect, then post a message on the discussion board to ask for assistance, but please do not post your code so that you do not reveal your solution to other learners.