C4

# Working with Lists

CHAPTER ASSIGNMENT

Name _____ Class _____ No. _____ Date _____

Answer every question in the space given. Show your working. You may use the chapter page to help you.

Questions from *Python Crash Course*, 2nd edition, Eric Matthes (No Starch Press), for classroom use.

## Try It Yourself

4-1 **Pizzas:** Think of at least three kinds of your favorite pizza. Store these pizza names in a list, and then use a `for` loop to print the name of each pizza.

• Modify your `for` loop to print a sentence using the name of the pizza instead of printing just the name of the pizza. For each pizza you should have one line of output containing a simple statement like *I like pepperoni pizza*.

• Add a line at the end of your program, outside the `for` loop, that states how much you like pizza. The output should consist of three or more lines about the kinds of pizza you like and then an additional sentence, such as *I really love pizza!*

4-2 **Animals:** Think of at least three different animals that have a common characteristic. Store the names of these animals in a list, and then use a `for` loop to print out the name of each animal.

• Modify your program to print a statement about each animal, such as *A dog would make a great pet.*

• Add a line at the end of your program stating what these animals have in common. You could print a sentence such as *Any of these animals would make a great pet!*

4-3 **Counting to Twenty:** Use a `for` loop to print the numbers from 1 to 20, inclusive.

4-4 **One Million:** Make a list of the numbers from one to one million, and then use a `for` loop to print the numbers. (If the output is taking too long, stop it by pressing ctrl-C or by closing the output window.)

4-5 **Summing a Million:** Make a list of the numbers from one to one million, and then use `min()` and `max()` to make sure your list actually starts at one and ends at one million. Also, use the `sum()` function to see how quickly Python can add a million numbers.

4-6 **Odd Numbers:** Use the third argument of the `range()` function to make a list of the odd numbers from 1 to 20. Use a `for` loop to print each number.

4-7 **Threes:** Make a list of the multiples of 3 from 3 to 30. Use a `for` loop to print the numbers in your list.

4-8 **Cubes:** A number raised to the third power is called a *cube*. For example, the cube of 2 is written as `2**3` in Python. Make a list of the first 10 cubes (that is, the cube of each integer from 1 through 10), and use a `for` loop to print out the value of each cube.

4-9 **Cube Comprehension:** Use a list comprehension to generate a list of the first 10 cubes.

4-10 **Slices:** Using one of the programs you wrote in this chapter, add several lines to the end of the program that do the following:

• Print the message *The first three items in the list are:*. Then use a slice to print the first three items from that program's list.

• Print the message *Three items from the middle of the list are:*. Use a slice to print three items from the middle of the list.

• Print the message *The last three items in the list are:*. Use a slice to print the last three items in the list.

4-11 **My Pizzas, Your Pizzas:** Start with your program from Exercise 4-1 (page 56). Make a copy of the list of pizzas, and call it `friend_pizzas`. Then, do the following:

• Add a new pizza to the original list.

• Add a different pizza to the list `friend_pizzas`.

• Prove that you have two separate lists. Print the message *My favorite pizzas are:*, and then use a `for` loop to print the first list. Print the message *My friend's favorite pizzas are:*, and then use a `for` loop to print the second list. Make sure each new pizza is stored in the appropriate list.

4-12 **More Loops:** All versions of *foods.py* in this section have avoided using `for` loops when printing to save space. Choose a version of *foods.py*, and write two `for` loops to print each list of foods.

4-13 **Buffet:** A buffet-style restaurant offers only five basic foods. Think of five simple foods, and store them in a tuple.

- Use a `for` loop to print each food the restaurant offers.
- Try to modify one of the items, and make sure that Python rejects the change.
- The restaurant changes its menu, replacing two of the items with different foods. Add a line that rewrites the tuple, and then use a `for` loop to print each of the items on the revised menu.

4-14 **PEP 8:** Look through the original PEP 8 style guide at *https://python.org/dev/peps/pep-0008/*. You won't use much of it now, but it might be interesting to skim through it.

4-15 **Code Review:** Choose three of the programs you've written in this chapter and modify each one to comply with PEP 8:

- Use four spaces for each indentation level. Set your text editor to insert four spaces every time you press tab, if you haven't already done so (see Appendix B for instructions on how to do this).
- Use less than 80 characters on each line, and set your editor to show a vertical guideline at the 80th character position.
- Don't use blank lines excessively in your program files.

End of assignment · Check your answers against the chapter at gurujazzy.com/subjects/python-crash-course/chapter-4-working-with-lists/