07 Team Activity: Lists

Instructions

Arrange a one hour synchronous meeting with your team for this activity. Online students should coordinate a video-sharing meeting. Campus students will use class time for this meeting. You should prepare for this meeting by completing the preparation material and the individual checkpoint assignment beforehand.

Problem Statement

There are a few details about writing and calling functions in Python that, if you understand, will help you be a more effective programmer. These details include default parameter values and pass by reference. As a team during this activity, you will write and call a function that demonstrates both default parameter values and pass by reference.

Helpful Documentation

- The Python programming language allows a programmer to specify a default value for a function parameter. When a parameter has a default value, the corresponding argument is optional. The prepare content for lesson 4 includes a section about default parameter values.

- Within a Python program, when a number is passed as an argument to a function, the computer copies the number from the argument into the parameter. In other words, the parameter gets a copy of the value that is in the argument. This idea of passing a copy of an argument into a parameter is known as pass by value.

  In Python, when a list is passed as an argument to a function, the computer does *not* copy the list. Instead, the computer passes a reference to the list into the function. This means the argument and parameter refer to the same list. This idea of passing a reference into a parameter is known as pass by reference. With pass by reference, if a called function changes a list that was passed into the function, this will of course change the list in the calling function because both the argument and parameter refer to the same list.

  The prepare content for this lesson includes a section about passing arguments by value and by reference.

- The Python `random` module contains functions to generate pseudo random numbers. The `random.uniform` function generates random numbers that are not integers.

- The built-in Python `round` function rounds a number to a specified number of digits after the decimal place.

Assignment

Write a Python program named `random_numbers.py` that creates a list of numbers, appends more numbers onto the list, and prints the list. The program must have two functions named `append_random_numbers` and `main` as follows:

1. `append_random_numbers`
   a. Has two parameters: a list named *numbers_list* and an integer named *quantity*. The parameter *quantity* has a default value of 1
   b. Computes *quantity* pseudo random numbers by calling the `random.uniform` function.
   c. Rounds the *quantity* pseudo random numbers to one digit after the decimal.
   d. Appends the *quantity* pseudo random numbers onto the end of the *numbers_list*.

2. `main`

   a. Has no parameters
   b. Creates a list named *randnums* like this:

```
randnums = [16.2, 75.1, 52.3]
```

   c. Prints *randnums*
   d. Calls the `append_random_numbers` function with only one argument to add one number to *randnums*
   e. Prints *randnums*
   f. Calls the `append_random_numbers` function again with two arguments to add three numbers to *randnums*
   g. Prints *randnums*

3. At the bottom of your program, write a line of code to call the `main` function. Then run your program and verify that your program works correctly.

Core Requirements

1. Your program includes two functions named `main` and `append_random_numbers`. The `append_random_numbers` function has two parameters named *numbers_list* and *quantity*, and *quantity* has a default value of 1.

2. The *append_random_numbers* function includes a loop that appends *quantity* random numbers at the end of *numbers_list*.

3. The `main` function calls `append_random_numbers` twice, first with one argument and second with two arguments.

Stretch Challenges

If your team finishes the core requirements in less than an hour, complete one or more of these stretch challenges. Note that the stretch challenges are optional.

1. Add a function named `append_random_words` that meets the following criteria:
   a. Has two parameters: a list named *words_list* and an integer named *quantity*. The parameter *quantity* has a default value of 1
   b. Randomly selects *quantity* words from a list of words and appends the selected words at the end of *words_list*.

2. Add something or change something in your program that you think would make your program better, easier for the user, more elegant, or more fun. Be creative.

Testing Procedure

Verify that your program works correctly by following each step in this testing procedure:

1. Download the `test_random_numbers.py` Python file and save it in the same folder where you saved your `random_numbers.py` program. Run the `test_random_numbers.py` file and ensure that the `test_random_numbers` function passes. If it doesn't pass, there is a mistake in your `random_numbers` function. Read the output from `pytest`, fix the mistake, and run the `test_random_numbers.py` file again until the test function passes.

```
> python test_random_numbers.py
====================== test session starts ======================
platform win32--Python 3.8.6, pytest-6.1.2, py-1.9.0, pluggy-0.13.
rootdir: C:\Users\cse111\lesson07
collected 1 item

test_random_numbers.py::test_random_numbers PASSED          [100%]

======================= 1 passed in 0.11s =======================
```

2. Run your program and ensure that your program's output is similar* to the output below.

```
> python random_numbers.py
randnums [16.2, 75.1, 52.3]
randnums [16.2, 75.1, 52.3, 84.2]
randnums [16.2, 75.1, 52.3, 84.2, 99.5, 20.4, 25.3]
randwords ['join', 'love', 'smile', 'love', 'cloud', 'head']
```

* Because your program is appending random numbers to the *randnums* list, your program's output will be slightly different than the output shown above. Specifically, the last four random numbers and the random words will be different.

Sample Solution

Please work diligently with your team for the one hour meeting. After the meeting is over, please compare your approach to the sample solution [↓]. Please ***do not look at the sample solution*** until you have either finished the program or diligently worked for at least one hour. At the end of the hour, if you are still struggling to complete the assignment, you may use the sample solution to help you finish.

Ponder

Look again at the `append_random_numbers` function (and the `append_random_words` function if you wrote it for the stretch challenge). The first parameter, *numbers_list*, is a list. Recall that in Python, lists are passed by reference which means that any changes made to the list in `append_random_numbers` (and `append_random_words`) will change the list in `main`. You can verify this is true by looking at the output of your program. The numbers that were appended onto the list in the `append_random_numbers` function are part of the same list in `main`.

Submission

When you have finished the activity, please report your progress via the associated I-Learn quiz. When asked about which of the requirements you completed, feel free to include any work done during the team meeting or after the meeting, including work done with the help of the sample solution, if necessary. In short, report on what you were able to accomplish, regardless of when you completed it or if you needed help from the sample solution.