

CSC110 Fall 2022 Assignment 1: Written Questions

Jaeyong Lee

September 27, 2022

Part 1: Data and Comprehensions

1. Imagine this scenario...

- (a) (The total amount of money you're planning to spend on your trip.)

A float would best represent this data since our budget would be expressed in dollars and there may be a decimal point.

- (b) (The restaurant names in your sister's "top ten restaurants" message, in the order of her preferences.)

A list where the elements are strings of the names of the restaurants would be ideal in this scenario. A list is chosen over a set because lists maintain order, which is important since the restaurants are ordered in terms of preference.

- (c) (The number of places you are staying that have laundry service.)

An integer could be used to represent this data as the number of places will be a whole number. A float is not necessary.

- (d) (The names of the cities you will be visiting and the corresponding number of days you are staying in each city.)

A dictionary would be ideal in this case because we need to associate each city with a number of days we plan to stay. So, the keys would be name of the city (a string) and the values would be the number of days we plan to stay (an integer).

- (e) (Whether or not you have a valid passport.)

The boolean data type will be ideal here. This is because the question "is my passport valid" can be answered with either true or false.

2. Exploring comprehensions.

- (a) i. ['B', 'l', 'u', 'e', 'b', 'e', 'r', 'r', 'y']

ii. This is a list and each of its elements are strings.

- (b) i. {'B', 'l', 'b', 'y', 'e', 'r', 'u'}

ii. This is a set and each of its elements are strings.

iii. In terms of size, the set is smaller since it does not contain any duplicate elements. The elements themselves are also in different order than the list because in a set, order does not matter.

- (c) Expression 1 evaluates to ['David!', 'Tom!', 'Mario!'] and expression 2 evaluates to ['David', 'Tom', 'Mario', '!', '!', '!']. Their values are different because while the former contains a string concatenation in its expression (each name is concatenated with '!'), the latter is concatenating two separate comprehensions to return one list. The two lists are the identity comprehension (which returns the list as is) and a comprehension that is returning a list that contains a '!' for every name in the list names. Expressions 1 and 2 evaluate very differently.

Part 2: Programming Exercises

Complete this part in the provided `a1_part2.py` starter file. Do **not** include your solution in this LaTeX file.

Part 3: Pytest Debugging Exercise

1. `test_two_customers` failed, `test_just_food` failed, and `test_single_bill` passed
2. The test `test_two_customers` raises a `KeyError` because of an issue in the test itself. In the assignment statement for the list `'bills,'` the keys `'Food'` and `'Songs'` have capitalized first letters. This causes an error because the function `get_largest_bill` calls `calculate_total_cost` and tries to pass in parameters by performing a key lookup for `'food'` and `'songs'` where the first letters are lowercase. When the Python interpreter is unable to find those keys, a `KeyError` is raised, meaning the program tried accessing keys in the dictionary that do not exist. Thus, the test fails.

The test `test_just_food` raises an `AssertionError`, meaning the actual test result did not match the expected result. This occurs because the function `calculate_total_cost` is written so that it takes `menu.amount` as the first parameter and then the number of songs. However, when we call this function from the body of `get_largest_bill`, we pass in the number of songs first and then the menu amount, which is the incorrect order. The reason we get a result different from the expected result is because the program treated the food bills as the numbers of songs sang.

3. The test `test_single_bill` passed because in that test, the bill has 10 as both the food bill and the number of songs sang. So although there are errors in the code, this test still passes because in this case, it does not matter which number gets passed in first into the function since the numbers are the same.

Part 4: Colour Rows

Complete this part in the provided `a1_part4.py` starter file. Do **not** include your solution in this LaTeX file.

Part 5: Working with Image Data

Complete this part in the provided `a1_part5.py` starter file. Do **not** include your solution in this LaTeX file.