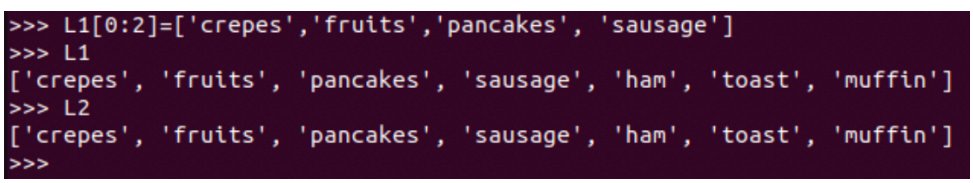
Part 1: Lists
Please conduct this part using the Python Interpreter Shell
1. Create a simple list L1 containing 5 strings. Example: L1 = [‘spam’, ‘eggs’, ‘ham’, ‘toast’, ‘muffin’]
2. Create a second list L2 as ‘L2 = L1’. For ...
3. Replace the list items at offset 1 and 2 with 2 different values in-place using a single statement. You may use any other numbers as the replacements.

i. Paste a screenshot of the statement you used and the output.

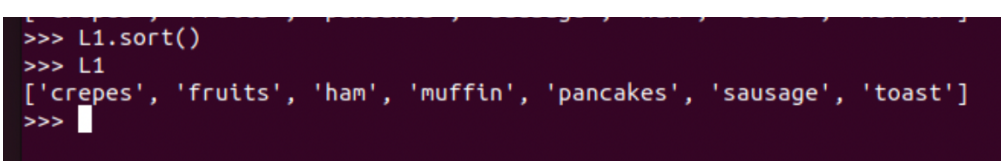
ii. Print L2 to see if the mutability of lists in maintained when you make this change.
4. Now, replace the list items at offset 1 and 2 with 4 different numbers in-place using a single statement. You may use any other numbers as the replacements. The final length of L1 after you execute this statement should be 7.

i. Paste a screenshot of the statement you used and the output.



ii. Print L2 to see if the mutability of lists in maintained when you make this change.
5. The ‘sort’ method applies to lists and is used to sort a list in ascending order by default. Other orders are also possible but we will use the default order. To use the sort function on L1, use ‘L1.sort()’. This will sort L1 but give you no output. To see the output, print L1.

i. Paste a screenshot of the statement you used and the output.



ii. Print L2 to see if the mutability of lists in maintained when you make this change.
iii. Re-assign L1 with some of the strings starting with an upper-case. Example: L1 = [‘Spam’, ‘eggs’, ‘Ham’, ‘toast’, ‘muffin’].
i. What happens if you sort this list using the sort() function?
After sorting, the list arranges words beginning with an uppercase letter first. Then, all the words that begin with a lowercase are sorted afterwards.
ii. What observations can you make about the sorting method that the sort() function uses?
The sort method seems to use ASCII values to determine sorting order. That's why words

beginning with an uppercase are the first in the list.

iii. Would a sort method work if a list comprised multiple data types?

I believe so. Every data type could contain a value that can be found in the ASCII table. Whether it's an int, char, or string, those values can be assigned to an ASCII value and be sorted from there.

Part 2: Dictionary

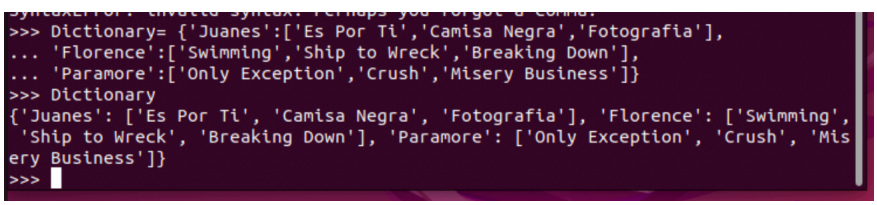
Please conduct this part using the Python Interpreter Shell

1. We are going to use a dictionary type to store 3 records in this example. You may use any records that adhere to the following record-keeping style:

Dictionary = {Key1:[Value[0], Value[1], Value[2]], Key2:[Value[0], Value[1], Value[2]], Key3: [Value[0], Value[1], Value[2]]}

An example of this is movieDB that we discussed in class today.

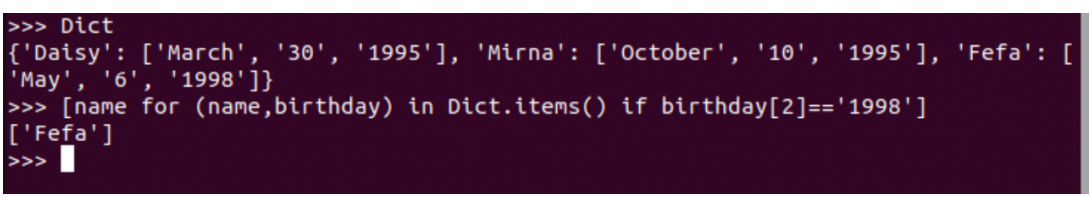
i. Paste a screenshot of the dictionary you created.



ii. Store all keys from the dictionary in a list using the list() and keys() methods.

iii. Write a comprehension statement that helps you find the correct key from your dictionary by using a known value. An example of this is shown in slide 18 of today's class material.

i. Paste a screenshot of your comprehension statement and its output



Part 3: Variable Scope in Python

Please conduct this part using the python script 'function.py' in the class git repository. This script is located in cs3035/python/week7 in the cs3035 folder of your Ubuntu machine.

We studied variable scope in C through local, function, and global variables. Modify the function.py code to test the scope of local, function, and global variables in Python. Write your observations for each of the following parts in 1-2 sentences.

i. In Python, does declaring global, local, and function variables follow the same rules as C?

Yes, it follows the same rules. Variables declared in locals or functions have their own scopes and cannot be accessed or modified outside that scope.

ii. What happens when we declare the same variables both globally and locally?

The local variable is able to modify the global variable. No issue occurs.

iii. What happens when we declare the same functions both locally and within the function?

The variable in the function is only within the scope of the function. It does not effect the same variable name in local.

iv. Does python use static variables?

No, python does not seem to use static variables. There isn't a static key word and each variable declared only pertains to its respective scope.