Eric Zorn

1/13/2018

ICT 4370

Professor Luke Kaim

**Dictionary Versus List Argument for HW Assignment 3 in Module 3**

There are many reasons why dictionaries and lists are used respectively while programming in Python. Although most any program can be written using completely one or the other, there are reasons that one of these data structures may be uses instead of the other. In Python 3 and other older versions of Python, the definition of a list is a mutable or changeable ordered sequence of elements/values. A standard variable cannot contain multiple value assignments without being formatted into a list, dictionary, or even tuple (less frequently used). Lists in Python are similar to arrays in other programming languages that you may be familiar with. The list begins and ends with an opening and closing square bracket, []. Each of the values in a Python list starts at the position zero because all lists and tuples are zero based. That means that the first element would be accessed like this **elements[0]** instead of **elements[1]**. The latter of the two examples would return the second value in the list or tuple. The only difference between lists and tuples in the Python programming language is that tuples are immutable, simply meaning that the values cannot be updated or changed.

The other most common data structure in the Python programming language is known as a dictionary. Similar to a list or tuple, dictionaries can contain multiple values. The main difference is that the values are assigned in key value pairs. The key is essentially the name that is given to the value, instead of using an index number like **[0]** or **[1]** in a list. Dictionaries are begun and closed with curly brackets, instead of the square brackets like **{}.** Tuples on the other hand begin and close with standard parentheses like **()**. Digital Ocean give the example of a dictionary with a username and information about the user sammy = {'username': 'sammy-shark', 'online': **True**, 'followers': 987}. This can be thought of similar to a relational database in MySQL with an ID and different values following the ID in the table within the database.

Tuples, lists, and dictionaries can be iterated over with different loops, such as a **for loop** and a **while loop**. A loop for a list would like **for j in grades: print(j)**. This would simply assign the variable j as the iterator and then print ever index of the list. The difference with a dictionary is that there are different methods that would be able to be called on a dictionary. These may be **keys()** or **values().** You would be able to loop over a dictionary with

**for key, value in grades: print(key, grades[value])**. This would iterate over the grades dictionary and print the key and the value for each student’s grade. The last method of a dictionary is the **items()** method which will return tuple pairs of the dictionary keys and values.

After a little background behind the three main Python data structures, I would definitely recommend lists as the proper data structure in this stock table assignment. As we have done, the assignment can definitely be completed completely using either a dictionary or list for all data values. When it comes to looping and printing data, I personally find it easier to use lists in Python and arrays in other programming languages like JavaScript. When we had to loop over the data and print the different values and titles for the table, it is a bit less code to deal with. In a program that may consist of larger datasets, it may be easier to use dictionaries. This is because of the different keys that would make selecting a specific value much easier. If you are planning on looping over the entire dataset and selecting/printing each value, a list may be a smarter choice. The interesting thing is that all data structures can be nested within another. Due to the fact that this dataset only held a small amount of values for each variable and each value was important to the outcome, I found that lists were the better alternative for the outcome.