**04: SPECIAL TOPICS IN STANDARD PYTHON**

**1: How are strings the same as lists? How are they different?**

Strings and lists share many important properties. They are both data types which can be assigned to a variable or passed as an argument for a function. Both store alphanumerical data (although any numbers contained in strings are read simply as characters; they have to be converted back to int/float data types in order to be used as numbers). Both strings and lists can be iterated through using a for loop and the range function. Elements of lists and strings can be accessed by their index; the order of the elements in both data structures is important. Lists can be concatenated, or added together, in the same way as lists can.

The main difference between strings and lists are that lists are mutable, while strings are immutable. For example, you can add elements to lists, change the value of an element at a particular index or delete elements from the list. However, once you create a string, its elements cannot be changed. In order to remove/add/change elements in a string, a second string must be made including those changes. Furthermore, as mentioned above, each element of a string is itself a string; no other data types are stored within a string. Lists, meanwhile can contain elements of a variety of different types; one list could contain floats, integers, strings and other lists/tuples as elements.

**2: Why are list comprehensions useful?**

List comprehensions are useful because they reduce the number of lines of code needed to carry out a given function, i.e. to iterate through the elements of a loop. Using a for loop takes at least 3 lines to programme the same logic as can be contained in a one-line list comprehension. This makes the code more elegant and easier to read. List comprehensions are much faster than for loops, where you often need to ‘append’ new data each time you iterate through the list.

**3: Write a try/except block that will tell you if matplotlib is installed and working on your system. Bonus: if they are installed, print out the name and version number for each.**

import matplotlib as plt

try:

name = plt.\_name\_

version = plt.\_\_version\_\_

print(name, version)

except:

print("matplotlib is not installed on this computer")

**4: Write a function that performs a dot product using enumerate. Write a second function using zip.**

Each vector is represented here as a list. 3-D vectors will be used.

v1 = [1, 2, 1]

v2 = [6, 7, 8]

#Enumerate method.

dot\_1 = [(i, x) for i,x in enumerate(v1)]

dot\_2 = [(i, x) for i,x in enumerate(v2)]

dot\_enumerate = dot\_1 + dot\_2

print(dot\_enumerate)

dot\_product\_enumerate = 0

for pair in dot\_enumerate:

index\_ = pair[0]

number\_ = pair[1]

for element in dot\_enumerate:

if dot\_enumerate.index(pair) != dot\_enumerate.index(element) and index\_ == element[0]:

dot\_product\_enumerate += number\_ \* element[1]

dot\_product\_enumerate //= 2

print(dot\_product\_enumerate)

#Zip method.

dot\_product\_zip = 0

for a,b in zip(v1, v2):

dot\_product\_zip += a \* b

print(dot\_product\_zip)