**Tutorial Questions - Lecture 2**

1. Try Q2-5 in a Python Console or a Notebook.
2. Assign the following variables in python (use variable names ‘a’,’b’,’c’,’d’). Which data types are produced? (check with 'type')”
   1. 1
   2. 2.0
   3. ‘Exeter’
   4. ‘3.0’
3. Assign a variable: my\_string = 'Today we are learning about variables and data-types'. Extract (using slicing syntax) the following strings from my\_string:
   1. 'Today'
   2. 'data-types'
   3. 'learning about variables'
   4. 'sepyt-atad dna selbairav tuoba gninrael era ew yadoT' (backwards!)
   5. 'yadoT'
   6. 'Tdyw r erigaotvralsaddt-ye' (?!?)
4. Create a list with the following values: mylist=[1,2.5,'3','Exeter','Python']. How do you:
   1. Delete the value 1 from the list (assuming you know which position it is in)
   2. Delete the value 1 from the list (if you DON'T know which position it is in)
   3. Reverse the list (think slicing!)
   4. Add ['Exeter',5,1.2] to the list?
   5. How many times does 'Exeter' appear in the extended list?
5. Create a dictionary called ‘cities’ with the following index/values: - 1 - 'Exeter', 2 -'Bath', 3 - 'Bristol’, 4 - 'London'.
   1. Look up the value for the key 3.
   2. Make a list from the keys of ‘cities’
   3. Create an integer variable ‘i’ with a value 3
   4. Look up the value of ‘cities’ in position i.
6. Create a new python script in VSCode (or another editor of your choice). The function (strictly this is called a ‘method’ as it applies to a string object) ‘split’ takes a string and splits it down into elements (returned as a list) given another string ‘x’ used as a separator. The syntax is my\_string.split(x). Assign a variable my\_string as in question 2. Use the split command with a space character to generate a list of the individual words in the sentence.
   1. Assign a new variable containing a list with these words in reverse order.
   2. Output this variable (using print ())
   3. Concatenate sentence ‘I’m good at working with ’ with the 3rd word in this list, and output.
7. ‘find’ locates the lowest index of a substring in a string and returns -1 if not found. Optional arguments limit the range of the search, so that s.find('i',10,20) locates the first occurrence of ‘i’ in s between positions 10 and 20 (and is identical to s[10:20].find('i')). rfind works identically, only returning the highest index of the substring.
   1. Add to your script from above, find the location of first occurrence of the character ‘w’ in the string.
   2. Think of 2 different ways of finding the last occurrence of the character ‘w’ in the string.
   3. Write code to check that your two methods in (2) return the same value.
   4. Use (1) and one of your methods in (2) to work out if there are one or more occurences of ‘w’ in the string.
   5. Check this by making a list of the characters in the string, and using the list .count() function/method.
8. Create a dictionary called ‘number\_things’ with the following keys / values ‘text’ - ‘2’ and -'numeric' - 3.
   1. Write a code expression to output the arithmetic sum of the values corresponding to the keys ‘text’ and ‘numeric’
   2. Write a code expression to concatenate text values of the values corresponding to the keys ‘text’ and ‘numeric’