HW2 Python Review

January 29, 2019

1 Homework 2

In this homework you will complete a couple of simple exercises in order to show your understanding with Python. If these exercises are challenging or new to you, you may want to reconsider taking the class and/or brush up on your Python skills. For the following exercises you are not allowed to use any Python packages (i.e. Numpy, Pandas, etc.).

1.0.1 NAME : Isha Mangal

1.0.2 ID : 3031911156

1.0.3 Mandatory : Please print the output of each question below your code

1.1 Lists

1.1 Create an empty Python list called 'a' in the cell below.

```
In [25]: a = []
         a

Out[25]: []
```

1.2 Store all values between 1-100 (inclusive) with increments of 3 (i.e. 1, 4, 7...) in 'a'.

```
In [26]: # concatanate function to keep adding elements to the list
         a = list(range(1, 98, 3))
         # concatanate function to keep adding elements to the list
         a.append(100)
         print (a)

[1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49, 52, 55, 58, 61, 64, 67, 70,
```

1.3 Create another list called 'a2' with numbers from 2-46 (inclusive) with increments of 0.5 (i.e. 2, 2.5, 3...).

```
In [27]: # Because Python's range function can't take in a float value
         # write a custom function for this
```

```
def float_range(start, stop=None, step=None):
    # Use float number in range() function
    # if stop and step argument is null,
    # then set start=0.0 and step = 1.0
    if stop == None:
        stop = start + 0.0
        start = 0.0
    if step == None:
        step = 1.0

    while True:
        if step > 0 and start >= stop:
            break
        elif step < 0 and start <= stop:
            break
        yield ("%g" % start) # return float number
        start = start + step

a2 = list()
a2 = list(float_range(2, 46.5, 0.5))
print (a2)
```

```
['2', '2.5', '3', '3.5', '4', '4.5', '5', '5.5', '6', '6.5', '7', '7.5', '8', '8.5', '9', '9.5
```

1.4 Double every even integer element from list ’a’. Store the results back in ’a’.

```
In [28]: def double_even(list_provided):
             # if: odd number (identified by n % 2) return
             # else: for even number multiply by 2
             return [n if n % 2 else n * 2 for n in list_provided]
         a = double_even(a)
         print (a)
```

```
[1, 8, 7, 20, 13, 32, 19, 44, 25, 56, 31, 68, 37, 80, 43, 92, 49, 104, 55, 116, 61, 128, 67, 14
```

1.5 Add all numbers in ’a’ except for the 2nd and 21st elements (the 2nd element here means the element at list index 1 and similarly for the 21st element).

```
In [29]: # Add all elements of a, Except the index 1 and index 20th
         # Simpler to just add all and then subtract the 2 elements
         print (sum(a) - a[1] - a[20])
```

```
2532
```

1.6 Calculate the mean of ’a’.

```
In [30]: mean_val = float(sum(a)/len(a))
         print (mean_val)
```

```
76.5
```

1.7 Delete all elements greater than the mean value from list 'a'

```
In [31]: # Delete all numbers from list a which are greater than mean_val
         a = ([x for x in a if x <= mean_val])
         print (a)
```

```
[1, 8, 7, 20, 13, 32, 19, 44, 25, 56, 31, 68, 37, 43, 49, 55, 61, 67, 73]
```

1.2 Strings

2.1 Create an empty list called 'b'.

```
In [32]: b = []
         print (b)
```

```
[]
```

2.2 Store the words in the sentence below as elements into the list 'b'.
'I am so excited about Data-X. It is important to be able to work with data.'

```
In [33]: sentence = "I am so excited about Data-X. It is important to be able to work with dat
         b = sentence.split(" ")
         print (b)
```

```
['I', 'am', 'so', 'excited', 'about', 'Data-X.', 'It', 'is', 'important', 'to', 'be', 'able',
```

2.3 Return the count of the occurences of the lower-case letter 'e' in the list 'b'.

```
In [34]: sub_string = "e"
         counter = 0

         for each_word in b:
             counter = counter + each_word.count(sub_string)
         print (counter)
```

```
4
```

2.4 Replace every lower- or upper-case letter 'i' in the list b with a '1'.

```
In [35]: b = [words.replace('i', '1') for words in b]
         b = [words.replace('I', '1') for words in b]
         print (b)
```

```
['1', 'am', 'so', 'exc1ted', 'about', 'Data-X.', '1t', '1s', '1mportant', 'to', 'be', 'able',
```

2.5 Append the string "This is the end of the first HW." to the list ’b’.

```
In [36]: b.append("This is the end of the first HW.")
         print (b)
```

```
['1', 'am', 'so', 'exc1ted', 'about', 'Data-X.', '1t', '1s', '1mportant', 'to', 'be', 'able',
```

2.6 Print ’b’ as ONE string backwards (starting with "WH tsrif...").

```
In [37]: # function to reverse one string
         def reverse_word(string_val):
             return string_val[::-1]

         # Join elements of list in one large string
         joined_sentence = ' '.join(b)
         print (joined_sentence)

         # Now reverse the large sentence
         new_string = (reverse_word(joined_sentence))
         # Remove the first character of '.'
         new_string = new_string[1:]
         print (new_string)
```

```
1 am so exc1ted about Data-X. 1t 1s 1mportant to be able to work w1th data. This is the end of
WH tsrif eht fo dne eht si sihT .atad ht1w krow ot elba eb ot tnatropm1 s1 t1 .X-ataD tuoba de
```

1.3 Dictionaries

3.1 Put the following in a dictionary called ’codes’:
Keys: 1001, 1002, 1003, 1004, 1005
Values: ’Alpha’, ’Beta’, ’Gamma’, ’Delta’, ’Tau’
then traverse the dictionary by its keys and change every value to be all lower case.

```
In [38]: # Creating empty Dictionary using empty brackets
         codes = {}

         keys_list = [1001, 1002, 1003, 1004, 1005]
         values_list = ['Alpha', 'Beta', 'Gamma', 'Delta', 'Tau']

         # Merge the two lists to create a dictionary
         codes = dict( zip (keys_list, values_list) )
         codes = dict((key, value.lower()) for key, value in codes.items())
         codes
```

```
Out[38]: {1001: 'alpha', 1002: 'beta', 1003: 'gamma', 1004: 'delta', 1005: 'tau'}
```

3.2 Delete ’alpha’ from the dictionary.

```
In [39]: # If key is not present in dictionary, then del can throw KeyError
         try:
             del codes[1001]
         except KeyError:
             print("Key '1001' not found")
         codes
```

```
Out[39]: {1002: 'beta', 1003: 'gamma', 1004: 'delta', 1005: 'tau'}
```

1.4 Sets

4.1 Create a set called ’c’ with the all the odd numbers less than 10.

```
In [40]: # Create a list for range between 1 to 10,
         # with step value of 2
         c_list = list(range(1, 10, 2))
         # Convert the list into set
         c = set(c_list)
         print (c)
```

```
{1, 3, 5, 7, 9}
```

4.2 Create another set called ’d’ with elements 2, 5, 10, 30.

```
In [41]: d_list = [2,5,10,30]
         # Convert the list into set
         d = set(d_list)
         print (d)
```

```
{2, 10, 5, 30}
```

4.3 Find the union between sets ’c’ and ’d’ and store this in a new set called ’e’.

```
In [42]: e =  c.union(d)
         print (e)
```

```
{1, 2, 3, 5, 7, 9, 10, 30}
```

4.4 Find the intersection between sets ’c’ and ’d’.

```
In [43]: print (c.intersection(d))
```

```
{5}
```