NPTEL Course: Programming, Data Structures and Algorithms in Python (*by* Prof. Madhvan Mukund)

Tutorial (Week 2)

Presented by: Jivesh Dixit

Problem 1: Calculate the weighted average of elements in a list

Weighted average: It is an arithmetic average, calculated for a set of numbers considering their relative importance.

Assuming the weights corresponding to numbers meant for averaging are, $\omega_1, \omega_2, \omega_3, \omega_4, \ldots\ldots, \omega_n$. If the weights are not normalised i.e.

$$\boxed{\sum \omega_i = \omega_1 + \omega_2 + \omega_3 + \omega_4 + \ldots\ldots + \omega_n \neq 1} \qquad (1)$$

We must normalise the weights using formula:

$$\boxed{\omega_{norm,\, i} = \omega_i / \sum \omega_i} \qquad (2)$$

Now weighted average for all the numbers in a list of length 'n' can be calculated as:

$$\boxed{\text{Weighted average} = (\omega_{norm,\, 1} * a_1 + \omega_{norm,\, 2} * a_2 + \ldots\ldots + \omega_{norm,\, n} * a_n)/n} \qquad (3)$$

Here $a_1, a_2, \ldots, a_n$ are the elements of the given list, and 'n' is the length of the list.

Approach:

- ➢ Initially want the list, and list of weights from user as an input. The lengths of both lists has to be same.
- ➢ Further we need to normalise the weights (if not already; as in (1)) using (2) for realistic averaging.
- ➢ Then we calculate the weighted average by applying relation (3).

*** notebook *link*

Problem 2(a): Checking if reverse of a string is same as the string

Approach:

- First of all we check for the input. If input is a string, we move forward, otherwise we raise a warning and exit.
- For string input, we arrange the letters of string in a reverse order.
- We check for condition, string == reverse of string.

Problem 2(b): Reversing words in a string

There can be various expected outcomes for reversing words in a given strings:

For example, if a given string is: *“we have tutorial for our course every friday”*

Reversed string can be:

1. *“friday every course our for tutorial have we”,*

2. *“ew evah lairotut rof ruo esruoc yreve yadirf”*

3. “yadirf yreve esruoc ruo rof lairotut evah ew”

Problem 3: Remove any empty sequence in a given list

Approach:

- ➢ First of all we check for the type of input, if it's not a list, we can return a warning message and then terminate the program.
- ➢ If input is a list, we check for each and every element for its type.
- ➢ If element in the list is a sequence, we check its length, and if length is zero, we remove it from the list.
- ➢ After processing all the elements we return the list.