

Student ID: 700752115

Student Name: Rumana Shaik

Neural Networks & Deep Learning - ICP-2

Github link: <https://github.com/ShaikRumana301/Neural-Network-DL-ICP-2>

1. Write a program that takes two strings from the user: first_name, last_name. Pass these variables to fullname function that should return the (full name).
 - o For example:
 - First_name = "your first name", last_name = "your last name"
 - Full_name = "your full name"
 - o Write function named "string_alternative" that returns every other char in the full_name string.

Str = "Good evening"

Output: Go vnn

```
In [16]: #defined the function for returning the alterbative characters of the string
def string_alternative(result):
    return result[::2];

#enter first name as input
first_name=input("please enter first name: ");
#enter last name as input
last_name=input("please enter last name: ");

#defined the function for returning the full name
def full_name(first_name,last_name):
    #concatenating the string
    return first_name+" "+last_name

print("Resultant String Alternative: "+string_alternative(full_name(first_name,last_name)));

please enter first name: Rumana
please enter last name: Shaik
Resultant String Alternative: Rmn hi
```

2. Write a python program to find the wordcount in a file (input.txt) for each line and then print the output.
 - o Finally store the output in output.txt file.

Example:

Input: a file includes two lines:

Python Course

Deep Learning Course

Output:

Python Course

Deep Learning Course

Word_Count:

Python: 1

Course: 2

Deep: 1

Learning: 1

```

In [7]: #read the Input file
input_file= open("Input.txt", "r")

#Write the number of words count and split the data from input file
output_file = open("output.txt", "w")

context = {}

for line in input_file:
    output_file.write(line)
    #splits the number of lines fom the input file context
    new_line = line.split()
    for x in new_line:
        if context.get(x)==None:
            context[x]=1
        else:
            context[x] = context[x] + 1

#writes the word count of the context to the output file
output_file.write("\n\nWord_Count : \n\n")
for key, value in context.items():
    output_file.write('%s:%s\n' % (key, value))
input_file.close()
output_file.close()

```

Input.txt:

```

1 Python Course
2 Deep Learning Course
3 Neural Network Learning Course

```

Output.txt:

jupyter output.txt 11 minutes ago

Logout

File Edit View Language

Plain Text

```

1 Python Course
2 Deep Learning Course
3 Neural Network Learning Course
4
5 Word_Count :
6
7 Python:1
8 Course:3
9 Deep:1
10 Learning:2
11 Neural:1
12 Network:1
13

```

- Write a program, which reads heights (inches.) of customers into a list and convert these heights to centimeters in a separate list using:

1) Nested Interactive loop.

2) List comprehensions

Example: L1: [150,155, 145, 148]

Output: [68.03, 70.3, 65.77, 67.13]

```

In [24]: #read the input of customer heights in inches
data = input("enter customer heights : ")

#convert the height of customer in inches to centimeters
def inchesToCentimeters(value):
    return value*2.54

heights=data.split()
new_list=[]
heights_list = list(map(float, heights))

def convert_height_nested_loop(heights):
    for x in heights:
        #appending the list using Nested interactive Loop
        value=int(x)
        new_list.append(inchesToCentimeters(value))
    return new_list

def convert_height_list_comp(heights):
    return [height * 2.54 for height in heights]

print("Resultant heights in centimeters using Nested Interactive : ",convert_height_nested_loop(heights))

print("Resultant heights in centimeters using List comprehension : ",convert_height_list_comp(heights_list))

enter customer heights : 145 130 165 150
Resultant heights in centimeters using Nested Interactive : [368.3, 330.2, 419.1, 381.0]
Resultant heights in centimeters using List comprehension : [368.3, 330.2, 419.1, 381.0]

```