

Neural Network Deep Learning

Assignment – 1

Name: Vineela Gurram

Student ID: 700757960

GitHub Link

https://github.com/vineelagurram2002/Neural_networks_coursework/blob/main/Vineela_Gurram_700757960.ipynb

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"`

Note: You need to create a function named “string_alternative” for this program and call it from main function.

In the below code snippet, we are accepting first name and last name from the console and returning every other character in the fullname string.



```
one.py x two.py three.py
Assignment 2 > one.py > main
1 # Write a program that takes two strings from the user: first_name, last_name. Pass these variables to
2 # fullname function that should return the (full name).
3
4 # o For example:
5 # ▪ First_name = "your first name", last_name = "your last name"
6 # ▪ Full_name = "your full name"
7
8 first_name = input("Enter the First String: ")
9 last_name = input("Enter the Second String: ")
10
11 def fullname(first_name,last_name):
12     # concatenating the first_name and last_name and storing in fullname
13     fullname = first_name + ' ' + last_name
14     return fullname
15 print("Full Name: ",fullname(first_name,last_name))
16
```

o Write function named “string_alternative” that returns every other char in the full_name string.

Str = “Good evening”

Output: Go vnn

```
one.py  X  two.py  three.py
Assignment 2 > one.py > ...
18  # o Write function named “string_alternative” that returns every other char in the full_name string.
19  # Str = “Good evening”
20  # Output: Go vnn
21
22  # Note: You need to create a function named “string_alternative” for this program and call it from
23  # main function.
24
25  def string_alternative(fullname):
26      fullNameValue = fullname(first_name,last_name)
27
28      # First Approach through for loop
29      finalFullName = ''
30      for i in range(len(fullNameValue)):
31          if(i % 2 == 0):
32              finalFullName += fullNameValue[i]
33      print("Full Name with every other Character: ",finalFullName)
34
35      # Second Approach through step
36      # finalFullName = fullNameValue[::2]
37      # print("Full Name Alternate Characters: ",finalFullName)
38
39  def main():
40      string_alternative(fullname)
41
42  if __name__ == "__main__":
43      main()
```

Output:

```
"Enter first name: Vineela\n",
"Enter last name: Gurram\n",
"Full Name: Vineela Gurram\n",
"Output: VneaGra\n"
```

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

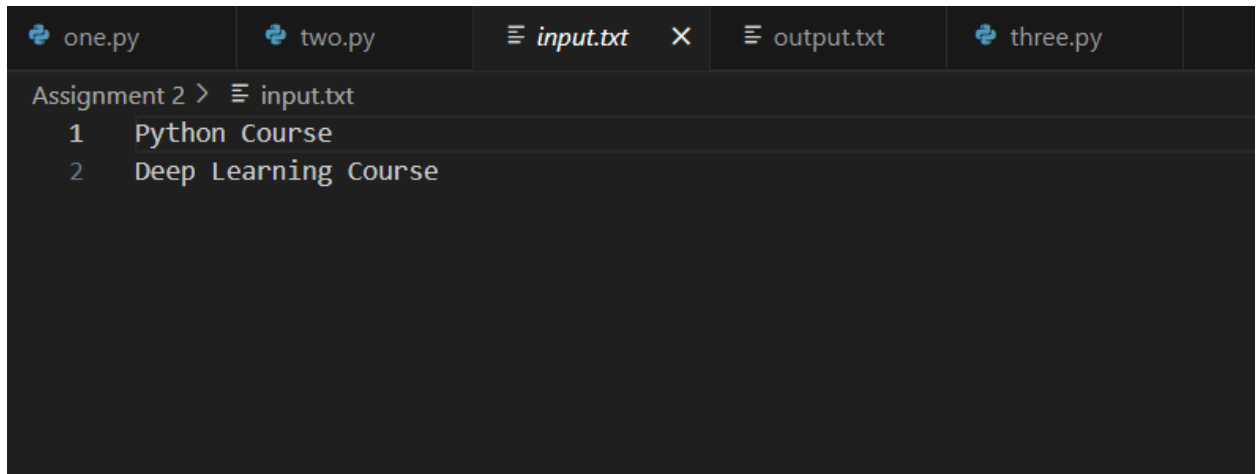
Deep: 1

Learning: 1

In the below Code snippet, we are replacing python with pythons

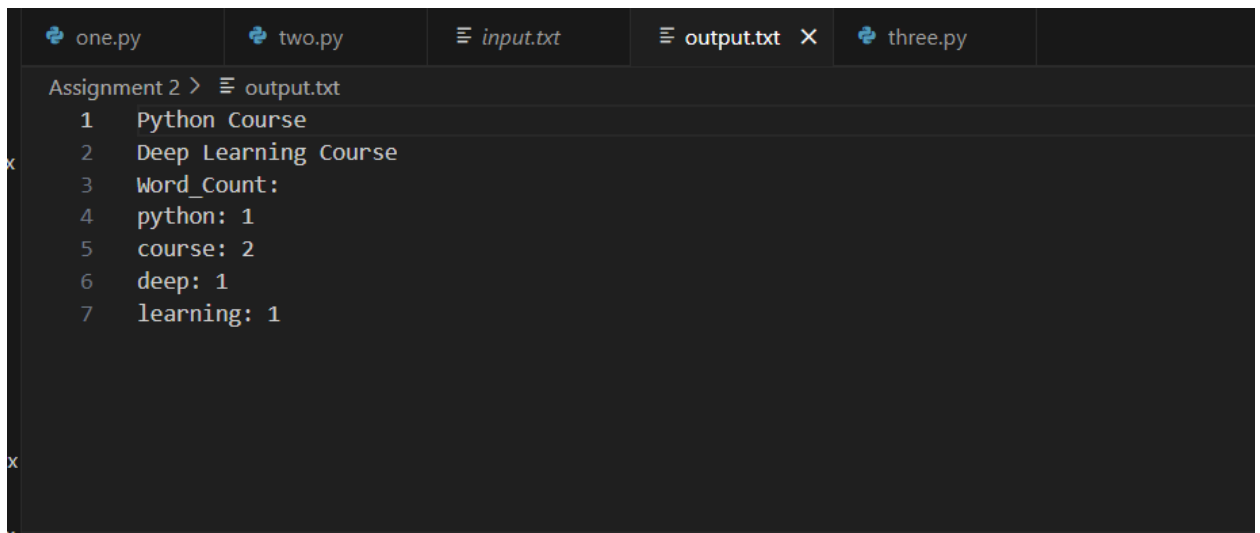
```
one.py two.py x three.py
Assignment 2 > two.py > ...
1  # Write a python program to find the wordcount in a file (input.txt) for each line and then print the output.
2  # o Finally store the output in output.txt file
3
4  inputfile = open("input.txt", 'r')
5  outputfile = open("output.txt", 'w')
6  # We are creating empty dictionary, it stores values in key value pair and it has unique key
7  d = dict()
8
9  # Writing input file content into output file
10 outputfile.write(inputfile.read() + '\n')
11 outputfile.write("Word_Count:")
12 # Reset the file pointer to the beginning of the file
13 inputfile.seek(0)
14
15 for line in inputfile:
16     # Remove the leading spaces and newline character
17     line = line.strip()
18     # we are splitting the line into list of words
19     words = line.split()
20     for word in words:
21         words = word.lower()
22         if words in d:
23             d[words] = d[words] + 1
24         else:
25             d[words] = 1
26 # Write word counts to the output file
27 for key in list(d.keys()):
28     outputfile.write(f"\n{key}: {d[key]}")
29
30 # we are closing the input and output files
31 inputfile.close()
32 outputfile.close()
```

Output:



The screenshot shows a code editor with a dark theme. The top bar contains five tabs: 'one.py', 'two.py', 'input.txt', 'output.txt', and 'three.py'. The 'input.txt' tab is active and highlighted. Below the tabs, the text 'Assignment 2 > input.txt' is displayed. The main content area shows a numbered list with two items: '1 Python Course' and '2 Deep Learning Course'.

```
Assignment 2 > input.txt
1 Python Course
2 Deep Learning Course
```



The screenshot shows the same code editor with the 'output.txt' tab active. The text 'Assignment 2 > output.txt' is displayed. The main content area shows a numbered list with seven items: '1 Python Course', '2 Deep Learning Course', '3 Word Count:', '4 python: 1', '5 course: 2', '6 deep: 1', and '7 learning: 1'. There are also some faint 'x' marks on the left side of the editor.

```
Assignment 2 > output.txt
1 Python Course
2 Deep Learning Course
3 Word Count:
4 python: 1
5 course: 2
6 deep: 1
7 learning: 1
```

3. 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

In the below Code snippet, we are calculating the height of the customer into centimeters using two approaches.

Nested Interactive Loop:

```
one.py two.py input.txt output.txt three.py X
Assignment 2 > three.py > ...
1  # Write a program, which reads heights (inches.) of customers into a list and convert these
2  # heights to centimeters in a separate list using:
3  # 1) Nested Interactive loop.
4  # 2) List comprehensions
5
6  # Nested Interactive loop.
7  num_customers = int(input("Enter the Total Number of Customers: "))
8  # creating empty list for height in inches
9  heights_inches = []
10 # creating empty list for height in centimeters
11 heights_cm = []
12
13 # This function converts inches to centimeters of a customers
14 def inches_to_cm(height_in_inches):
15     return height_in_inches * 2.54
16
17 for i in range(num_customers):
18     height = float(input(f"Enter Height (in inches) for Customer {i + 1}: "))
19     heights_inches.append(height)
20     heights_cm.append(inches_to_cm(height))
21
22 # Printing the height of the customers in centimeters
23 print("Heights in centimeters:", heights_cm)
24
```

List Comprehensions :

```
one.py two.py input.txt output.txt three.py X
Assignment 2 > three.py > ...
25 # List comprehensions
26 def inches_to_cm(height_in_inches):
27     return height_in_inches * 2.54
28
29 num_customers = int(input("Enter the Total Number of Customers: "))
30 heights_inches = [float(input(f"Enter height (in inches) for customer {i + 1}: ")) for i in range(num_customers)]
31 heights_cm = [inches_to_cm(height) for height in heights_inches]
32
33 # Printing the height of the customers in centimeters
34 print("Heights in centimeters:", heights_cm)
```

Output:

```
"Enter heights in inches (type 'done' to finish):\n",
"Enter height (or 'done'): 150\n",
"Enter height (or 'done'): 155\n",
"Enter height (or 'done'): 145\n",
"Enter height (or 'done'): 148\n",
"Enter height (or 'done'): done\n",
"Heights in inches: [150.0, 155.0, 145.0, 148.0]\n",
"Heights in centimeters: [381.0, 393.7, 368.3, 375.92]\n"
```