JLUFE Fall 2021(Sep-Jan)

Homework Assignment Report

JILIN UNIVERSITY OF FINANCE AND ECONOMICS

College of Managment Science and Information Engineering

BSc in Data Science and Big Data Technology

(2021)

MODULE: Intelligent Technology

Homework Assignment: 02

Strings and Lists

23/09/2021

Submitted by:

Milan(米兰) 0318021907632 (2005)
QQ: 3086215265 | Github ID: milaan9

In [ ]:

```

```

Instructions:

1. I have added tips and required learning resources for each question, which helps you to solve the problems.
2. Finish the assignment on your **OWN**. **Any student find copying/sharing from classmates or internet will get '0' points!!!**
3. After Accepting this assignment from ➞ [GitHub Clasroom link (https://classroom.github.com/a/ZPJFystv)](https://classroom.github.com/a/ZPJFystv), Github will create private repository of the assignment in your GitHub Classroom account.
4. In your repository Clone ➞ Download ZIP in your computer.
5. Change your ➞ **College**, **Major**, **Name**, **Student number**, **Class number**, **QQ number** and **GitHub ID**
6. Once you finish the Assignment [convert your .ipynb file into PDF (https://github.com/milaan9/91_Python_Mini_Projects/tree/main/001_Convert_IPython_to_PDF)](https://github.com/milaan9/91_Python_Mini_Projects/tree/main/001_Convert_IPython_to_PDF) (both **.ipynb** and **.pdf** file will be required!)
7. To submit your assignment, go to GitHub Classroom repository and Add file ➞ Upload files ➞ Commit changes

   A. Replace the question (**.ipynb**) file with your solution (**.ipynb**) file.

B. Also, upload (**.pdf**) converted file of your solution (**.ipynb**) file.

Python Assignment 02

Part A ➞ String Level 1

> **Note:** Please create new cell for each question

1. Concatenate the string `Python`, `4`, `Data`, `Science` to a single string, `Python 4 Data Science`.
2. Declare a variable named `course` and assign it to an initial value `Python 4 Data Science`.
3. Print the length of the `course` string using **len() (https://github.com/milaan9/04_Python_Functions/blob/main/002_Python_Functions_Built_in/040_Pyt** method and **print() (https://github.com/milaan9/04_Python_Functions/blob/main/002_Python_Functions_Built_in/051_Pyt**
4. Change all the characters of variable company to uppercase and lowercase letters using **upper() (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/026_Pytho** and **lower() (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/025_Pytho** method.
5. Use **capitalize() (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/001_Pytho title() (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/042_Pytho swapcase() (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/027_Pytho** methods to format the value of the string `Python 4 Data Science`.
6. Cut(slice) out the first word of `Python 4 Data Science`.
7. Check if `Python 4 Data Science` string contains a word `Python` using the method: **index() (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/010_Pytho find() (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/008_Pytho** or other methods.
8. Change `Python 4 Data Science` to `Python 4 Everybody` using the **replace() (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/035_Pytho** method or other methods.
9. Split the string `Python 4 Data Science` using space as the separator (**split() (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/038_Pytho**
10. `Google, Facebook, Microsoft, Apple, IBM, Oracle, Amazon` split the string at the comma.
11. What is the character at index 9 in the string `Python 4 Data Science`.
12. What is the second last index of the string `Python 4 Data Science`.
13. Create an acronym or an abbreviation for the name `Python 4 Data Science`.
14. Use **index() (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/010_Pytho** to determine the position of the first occurrence of `D` in `Python 4 Data Science`.
15. Use **rfind (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/036_Pytho** to determine the position of the last occurrence of `e` in `Python 4 Data Science`.

16. Use index() (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/010_Pytho or find() (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/008_Pytho to find the position of the first occurrence of the word `because` in the following sentence:

    - `We cannot end the sentence with ‘because’ , because ‘because’ is a conjunction.` .

17. Use rindex (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/037_Pytho to find the position of the first and last occurrence of the word `because` in the following sentence:

    - `We cannot end the sentence with ‘because’ , because ‘because’ is a conjunction.` .

18. Slice out the phrase `‘because’ , because ‘because’` in the following sentence:

    - `We cannot end the sentence with ‘because’ , because ‘because’ is a conjunction.` .

19. Does `Python 4 Data Science` start with a substring `Python` ?
20. Does ' `Python 4 Data Science` contains with a substring `Python` ?
21. `Python 4 DataScience` remove the left and right trailing spaces in the given string.
22. The following list contains the names of some of python libraries: `['Django', 'Flask', 'Bottle', 'Pyramid', 'Falcon']` . Join the list with a hash with space string.
23. Which one of the following variables return True when we use the method isidentifier() (https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/015_Pytho

    -
      ```
      2021PythonDataypes
      Python_Dataypes_2021
      ```

24. Make the following using string formatting methods:

    -
      ```
      8 + 6 = 14
      8 - 6 = 2
      8 * 6 = 48
      8 / 6 = 1.33
      8 % 6 = 2
      8 // 6 = 1
      8 ** 6 = 262144
      ```

25. Use a **new line** and **tab** escape sequence to print the following lines.

    -
      ```
      Name    Age   Country   City
      Milaan  96    Finland   Tampere
      ```

In [114]:

```
# Solution:
str_word1= "Python"
str_word2= "4"
str_word3= "Data"
str_word4= "Science"
print(str_word1+" "+str_word2+" "+str_word3+" "+str_word4)

Python_4_Data_Science="course"
print(Python_4_Data_Science)
print(len(Python_4_Data_Science))

word="company"
word_a=word.upper()
word_b=word_a.lower()
print(word_a)
print(word_b)

word="Python 4 Data Science"
print(word.capitalize())
print(word.title())
print(word.swapcase())

word=["Python",4,"data","science"]
print(word.pop(0))

str_word="Python 4 Data Science"
print("Python "in word)

word=["Python",4,"data","science"]
del word[2]
del word[2]
print (word)
word.insert(2,"Everybody")
print(word)

word="Python 4 Data Science"
print(word.split())

word="Google,Facebook,Microsoft,Apple,IBM,Oracle,Amazom"
print(word.split(","))

word="Python 4 Data Science"
print(word[9])

word="P y t h o n 4 D a t a  S  c  i  e  n  c  e"
#     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
#   -18-17-16-15-14-13-12-11-10-9-8-7-6-5-4-3-2-1

word="Python 4 Data Science"
word1=word[0]
word2=word[7]
word3=word[9]
word4=word[14]
print(word1+" "+word2+" "+word3+" "+word4)

str_word="Python 4 Data Science"
print(word.index("D"))

str_word="Python 4 Data Science"
```

```
print(word.rfind("e"))

word=["We"," cannot"," end","the","sentence","with","'because'", "because","'because'","is","a","cor
print(word.index("because"))

word=["We"," cannot"," end","the","sentence","with","'because'", "because","'because'","is","a","cor
print("Python" in "a")

word="Python 4 Data Science"
print(word.lstrip("4"))

word=['Django', 'Flask', 'Bottle', 'Pyramid', 'Falcon']
print(" ".join(word))

word1="2021PythonDataypes"
word2="Python_Dataypes_2021"
print(word1.isidentifier())
print(word2.isidentifier())

print("8 {} 6 = {}".format('+','14'))
print("8 {} 6 = {}".format('-','2'))
print("8 {} 6 = {}".format('*','48'))
print("8 {} 6 = {}".format('/','1.33'))
print("8 {} 6 = {}".format('%','2'))
print("8 {} 6 = {}".format('//','1'))
print("8 {} 6 = {}".format('**','262144'))
```

```
Python 4 Data Science
course
6
COMPANY
company
Python 4 data science
Python 4 Data Science
pYTHON 4 dATA sCIENCE
Python
False
['Python', 4]
['Python', 4, 'Everybody']
['Python', '4', 'Data', 'Science']
['Google', 'Facebook', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazom']
D
P 4 D S
9
20
7
False
Python 4 Data Science
Django Flask Bottle Pyramid Falcon
False
True
8 + 6 = 14
8 - 6 = 2
8 * 6 = 48
8 / 6 = 1.33
8 % 6 = 2
8 // 6 = 1
8 ** 6 = 262144
```

Part B ➞ List Level 1

> **Note:** Please create new cell for each question

1. Declare a list with more than 5 items with different data types
2. Find the length of your list
3. Get the first item, the middle item and the last item of the list
4. Declare a list `called my_info` , put your (name, age, height, marital status, country)
5. Declare a list variable named `mix_fruits` and assign initial values Guava, Mango, Apple, Pear, Fig, Orange and Banana and print the list.
6. Print the list using print() (https://github.com/milaan9/04_Python_Functions/blob/main/002_Python_Functions_Built_in/051_Py
7. Print the number of `mix_fruits` in the list
8. Print the first, middle and last fruit
9. Print the list after modifying one of the fruit
10. Add an fruit to variable `mix_fruits`
11. Insert an fruit in the middle of the `mix_fruits` list
12. Change one of the fruit names to uppercase
13. Join the elements in `mix_fruits` with a string `-#-`
14. Check if a certain fruit exists in the `mix_fruits` list.
15. Sort the list using sort() (https://github.com/milaan9/02_Python_Datatypes/blob/main/003_Python_List_Methods/009_Python_ method
16. Reverse the list in descending order using reverse() (https://github.com/milaan9/02_Python_Datatypes/blob/main/003_Python_List_Methods/008_Python_ method
17. Slice out the first 3 fruits from the list
18. Slice out the last 3 fruits from the list
19. Slice out the middle fruit or fruits from the list
20. Remove the first fruit from the list
21. Remove the middle fruit or companies from the list
22. Remove the last fruit from the list
23. Remove all fruits from the list
24. Delete the fruits list
25. Join the following lists:

    -
      ```
      front_end = ['HTML', 'CSS', 'JS', 'React', 'Redux']
      back_end = ['Node','Express', 'MongoDB']
      ```

26. After joining the lists in question 25. Copy the joined list and assign it to a variable full_stack. Then insert `'Python'` and `'SQL'` after `'Redux'` .

In [131]:

```
# Solution:
L=["string",1,1.5,(1,2),["x","y"]]
print(L)
print(len(L))
print(L[0])
print(L[2])
print(L[4])

my_info=["Wangbo","19","175","unmarried","China"]
mix_fruits=["Guava","Mango","Apple","Pear","Fig","Orange","Banana"]
print(my_info)
print(mix_fruits)
print(len(mix_fruits))
print(mix_fruits[0])
print(mix_fruits[3])
print(mix_fruits[6])
mix_fruits[0]="Grape"
print(mix_fruits)
mix_fruits. insert(3, "Pear")
print(mix_fruits)
fruit="Grape"
print(fruit. upper ())
print ("pear" in mix_fruits)
print(mix_fruits.sort())
print(mix_fruits.reverse())
mix_fruits=["Guava","Mango","Apple","Pear","Fig","Orange","Banana"]
print(mix_fruits[:3])
print(mix_fruits[4:])
del mix_fruits[3]
print(mix_fruits)
del mix_fruits[0]
print(mix_fruits)
mix_fruits.remove("Apple")
print(mix_fruits)
del mix_fruits

front_end = [ 'HTML','CSS','JS','React','Redux']
back_end = ['Node','Express', 'MongoDB']
full_stack=front_end+back_end
print(full_stack)
full_stack.insert (5,"Python")
print(full_stack)
full_stack.insert (6,"SQL")
print(full_stack)
```

```
['string', 1, 1.5, (1, 2), ['x', 'y']]
5
string
1.5
['x', 'y']
['Wangbo', '19', '175', 'unmarried', 'China']
['Guava', 'Mango', 'Apple', 'Pear', 'Fig', 'Orange', 'Banana']
7
Guava
Pear
Banana
['Grape', 'Mango', 'Apple', 'Pear', 'Fig', 'Orange', 'Banana']
```

```
['Grape', 'Mango', 'Apple', 'Pear', 'Pear', 'Fig', 'Orange', 'Banana']
GRAPE
False
None
None
['Guava', 'Mango', 'Apple']
['Fig', 'Orange', 'Banana']
['Guava', 'Mango', 'Apple', 'Fig', 'Orange', 'Banana']
['Mango', 'Apple', 'Fig', 'Orange', 'Banana']
['Mango', 'Fig', 'Orange', 'Banana']
['HTML', 'CSS', 'JS', 'React', 'Redux', 'Node', 'Express', 'MongoDB']
['HTML', 'CSS', 'JS', 'React', 'Redux', 'Python', 'Node', 'Express', 'MongoDB']
['HTML', 'CSS', 'JS', 'React', 'Redux', 'Python', 'SQL', 'Node', 'Express', 'MongoD
B']
```

Part B ➞ List Level 2

> **Note:** Please create new cell for each question

1. The following is a list of 10 students ages:

   -
     ```
     ages = [19, 23, 19, 25, 21, 20, 25, 26, 25, 24]
     ```

     - Sort the list and find the min and max age
     - Add the min age and the max age again to the list
     - Find the median age (one middle item or two middle items divided by two)
     - Find the average age (sum of all items divided by their number)
     - Find the range of the ages (max - min)
     - Compare the value of (min - average) and (max - average), use abs() (https://github.com/milaan9/04_Python_Functions/blob/main/002_Python_Functions_Built_in method
2. Find the middle country(ies) in the countries list (https://github.com/milaan9/02_Python_Datatypes/blob/main/countries_data.py)
3. Divide the countries list into two equal lists if it is even if not one more country for the first half.
4. ['India', 'Russia', 'China', 'Finland', 'Sweden', 'Norway', 'Denmark']. Unpack the first three countries and the rest as scandic countries.

In [140]:

```
# Solution:
ages = [19, 23, 19, 25, 21, 20, 25, 26, 25, 24]
print(min(ages))
print(max (ages))
ages.insert(1,17)
ages.insert(4,30)
print(ages)
ages.sort ()
print(ages)
print(len (ages))
age1=ages[5]
age2=ages[6]
age3= (age1+age2)/2
print(age3)
a=ages[0]
b=ages[1]
c=ages[2]
d=ages[3]
e=ages[4]
f=ages[5]
g=ages[6]
h=ages[7]
i=ages[8]
j=ages[9]
k=ages[10]
l=ages[11]
print((a+b+c+d+e+g+h+i+j+k+1)/12)
A=min(ages)
B=max (ages)
print(A)
print(B)
C=['India','Russis','China','Finland','Sweden','Norway','Denmark']
print(len(C))
print(C[3])
print("scandic countries=")
print(C[:3])
```

```
19
26
[19, 17, 23, 19, 30, 25, 21, 20, 25, 26, 25, 24]
[17, 19, 19, 20, 21, 23, 24, 25, 25, 25, 26, 30]
12
23.5
18.5
17
30
7
Finland
scandic countries=
['India', 'Russis', 'China']
```