

JLUFE

Fall

2021(Sep-Jan)

Homework Assignment Report

JILIN UNIVERSITY OF FINANCE AND ECONOMICS

School of International Exchange

BSc in Bachelor degree in e-commerce

(2021)

MODULE: Intelligent Technology

Homework Assignment: 02

Data Types (String, List, Tuple, Sets, Dictionary)

23/09/2021

Submitted by:

Wendy(陈芃佳) 0318031902145 (1921)

QQ: 1692677159 | Github ID: Wendy729

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 from → [GitHub Classroom link \(https://classroom.github.com/a/cXdhullE\)](https://classroom.github.com/a/cXdhullE), Github will create private repository of the assignment in your GitHub Classroom account.
4. In your repository → 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 → →
 - 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 → Operators Level 1

Note: Please create new cell for each question

1. Declare your age as integer variable
2. Declare your height as a float variable
3. Declare a variable that store a complex number
4. Write a code by taking user `input()`

(https://github.com/milaan9/04_Python_Functions/blob/main/002_Python_Functions_Built_in/032_Py) to enter base and height of the triangle and calculate an area of this triangle ($area = 0.5 * a * b$) and the perimeter of the triangle ($perimeter = a + b + c$).

- Enter base: 20
Enter height: 10
The area of the triangle is 100
The perimeter of the triangle is 12

5. Calculate the value of y : ($y = x^2 + 6x + 9$). Try to use different x values and figure out at what x value y is going to be 0.
6. Find the length of 'python' and 'datascience' and compare if the length are same using `==`.
7. Use `and` operator to check if `on` is found in both `python` and `cannon`
8. I hope this course is not full of jargon. Use `in` operator to check if `jargon` is in the sentence.
9. There is no `on` in both `python` and `cannon`
10. Find the length of the text `python` and convert the value to float and convert it to string
11. Even numbers are divisible by 2 and the remainder is zero. How do you check if a number is even or not using python?
12. Check if the floor division of 7 by 3 is equal to the int converted value of 2.7.
13. Check if type of "10" is equal to type of 10
14. Write a code that prompts the user to enter hours and rate per hour. Calculate pay of the person?

- Enter hours: 40
Enter rate per hour: 30
Your weekly earning is 1200

15. Write a script that prompts the user to enter number of years. Calculate the number of seconds a person can live. Assume a person can live hundred years

- Enter number of years you have lived: 100
You have lived for 3153600000 seconds.

16. Write a Python code that displays the following table using operators

- ```
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
5 10 15 20 25
```

In [2]:

```
Solution:
#1-3
age= 20
print(age)
height=1.63
print(height)
c=1+2j
print(c)
```

```
20
1.63
(1+2j)
```

In [ ]:

```
a=int(input("Enter base value:"))
b=int(input("Enter height value:"))
area=(0.5*a*b)
print(area)
a=int(input("Enter value:"))
b=int(input("Enter value:"))
c=int(input("Enter value:"))
perimeter=a+b+c
print(perimeter)
```

In [6]:

```
import math
a=int(input("a: "));
b=int(input("b: "));
c=int(input("c: "));
d=b*b-4*a*c;
if a==0:
 if b!=0:
 print("x=%0.2f"%(-c/b));
 else:
 print("none");
else:
 if d==0:
 print("x1=x2=%0.2f"%(-b/2/a));
 else:
 if d>0:
 x1=-b+math.sqrt(d)/2/a;
 x2=-b-math.sqrt(d)/2/a;
 print("x1=%0.2f"%x1);
 print("x2=%0.2f"%x2);
 else:
 print("virtual root");
```

```
a: 1
b: 6
c: 9
x1=x2=-3.00
```

In [2]:

```
a="python"
print(len(a))
b="datascience"
print(len(b))
a==b
```

6  
11

Out[2]:

False

In [20]:

```
a="on"
b="python"
c="cannon"
if a in b and a in c:
 print('yes')
else:
 print('no')
```

yes

In [17]:

```
a="I hope this course is not full of jargon"
if 'jargon' in a:
 print('yes')
```

yes

In [21]:

```
a="on"
b="python"
c="cannon"
if a in b and a in c:
 print('yes')
else:
 print('no')
```

yes

In [8]:

```
str1="python"
length=len(str1)
print(length)
print(float(length))
print(str(length))
```

6  
6.0

---

-----

Traceback (most recent call last)

<ipython-input-8-1254025cf8e0> in <module>

3 print(length)

4 print(float(length))

----> 5 print(str(length))

TypeError: 'str' object is not callable

In [ ]:

```
n=int(input('Enter the number: '))
if n%2==0:
 print('{} is even number'.format(n))
```

In [2]:

```
a=7
b=3
c = a / b
print (c)
c=="2.3"
```

2.3333333333333335

Out[2]:

False

In [6]:

```
type(10)
```

Out[6]:

int

In [8]:

```
type('10')
```

Out[8]:

str



- [https://github.com/milaan9/02\\_Python\\_Datatypes/blob/main/002\\_Python\\_String\\_Methods/042\\_Python\\_swapcase\(\)](https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/042_Python_swapcase())
- [https://github.com/milaan9/02\\_Python\\_Datatypes/blob/main/002\\_Python\\_String\\_Methods/027\\_Python\\_methods\\_to\\_format\\_the\\_value\\_of\\_the\\_string\\_Python\\_4\\_Data\\_Science](https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/027_Python_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_Python_find())
- [https://github.com/milaan9/02\\_Python\\_Datatypes/blob/main/002\\_Python\\_String\\_Methods/010\\_Python\\_find\(\)](https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/010_Python_find())
- [https://github.com/milaan9/02\\_Python\\_Datatypes/blob/main/002\\_Python\\_String\\_Methods/008\\_Python](https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/008_Python_or_other_methods) 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_Python_replace())
- [https://github.com/milaan9/02\\_Python\\_Datatypes/blob/main/002\\_Python\\_String\\_Methods/035\\_Python](https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/035_Python_replace()) 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_Python_split()))
- [https://github.com/milaan9/02\\_Python\\_Datatypes/blob/main/002\\_Python\\_String\\_Methods/038\\_Python](https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/038_Python_split())
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_Python_index())
- [https://github.com/milaan9/02\\_Python\\_Datatypes/blob/main/002\\_Python\\_String\\_Methods/010\\_Python](https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/010_Python_index()) 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_Python_rfind())
- [https://github.com/milaan9/02\\_Python\\_Datatypes/blob/main/002\\_Python\\_String\\_Methods/036\\_Python](https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/036_Python_rfind()) 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_Python_index())
- [https://github.com/milaan9/02\\_Python\\_Datatypes/blob/main/002\\_Python\\_String\\_Methods/010\\_Python](https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/010_Python_index()) or [find\(\)](https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/008_Python_find())
- [https://github.com/milaan9/02\\_Python\\_Datatypes/blob/main/002\\_Python\\_String\\_Methods/008\\_Python](https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/008_Python_find()) 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_Python_rindex())
- [https://github.com/milaan9/02\\_Python\\_Datatypes/blob/main/002\\_Python\\_String\\_Methods/037\\_Python](https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/037_Python_rindex()) 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_Python_isidentifier())
- [https://github.com/milaan9/02\\_Python\\_Datatypes/blob/main/002\\_Python\\_String\\_Methods/015\\_Python](https://github.com/milaan9/02_Python_Datatypes/blob/main/002_Python_String_Methods/015_Python_isidentifier())
- 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 [45]:

```
Solution:

str1="Python"+" "
str2="4"+" "
str3="Data"+" "
str4="Science"+" "
str5=str1+str2+str3+str4
print(str5)
```

Python 4 Data Science

In [48]:

```
str01="course"
str5=str1+str2+str3+str4+str01
print(str5)
```

Python 4 Data Science course

In [50]:

```
str01="course"
print(len(str01))
```

6

In [61]:

```
str = "Tencent"
print (str.upper())
str = "Tencent"
print (str.lower())
```

TENCENT  
tencent



In [66]:

```
s = 'Python 4 Data Science'
s.capitalize()
```

Out[66]:

'Python 4 data science'

In [67]:

```
s = 'Python 4 Data Science'
s.title()
```

Out[67]:

'Python 4 Data Science'

In [69]:

```
s = 'Python 4 Data Science'
s.swapcase()
```

Out[69]:

'pYTHON 4 dATA sCIENCE'

In [72]:

```
s = 'Python 4 Data Science'
s2=s.replace("Python", " ")
print(s2)
```

4 Data Science

In [75]:

```
str1 = "Python 4 Data Science"
str2=str1.index("Python")
print(str2)
```

0

In [77]:

```
str1 = "Python 4 Data Science"
str2=str1.find("Python")
print(str2)
```

0

In [88]:

```
s = 'Python 4 Data Science'
s2=s.replace('Data', 'Everybody').replace('Science', '')
print(s2)
```

Python 4 Everybody

In [93]:

```
s = 'Python 4 Data Science'
s2=s.split()
print(s2)
```

['Python', '4', 'Data', 'Science']

In [96]:

```
a="Google Facebook Microsoft Apple IBM Oracle Amazon"
b=a.split()
print(b)
```

['Google', 'Facebook', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon']

In [2]:

```
string="Python 4 Data Science"
print("the character at index 9 of",string,"is",string[9])
```

the character at index 9 of Python 4 Data Science is D

In [3]:

```
string="Python 4 Data Science"
print("the second last index of",string,"is",string[-2])
```

the second last index of Python 4 Data Science is c

In [4]:

```
string="Python 4 Data Science"
[i[0] for i in string.split(" ")]
```

Out[4]:

['P', '4', 'D', 'S']

In [5]:

```
str1 = "Python 4 Data Science"
str2=str1.index("D")
print(str2)
```

9

In [7]:

```
str1 = "Python 4 Data Science"
str2=str1.rfind("e")
print(str2)
```

20

In [12]:

```
str1 = "We cannot end the sentence with 'because', because 'because' is a conjunctio
str2=str1.index("because")
print(str2)
```

33

In [15]:

```
str1 = "We cannot end the sentence with 'because', because 'because' is a conjunctio
str2=str1.rindex("because")
print(str2)
```

52

In [24]:

```
str1 = "We cannot end the sentence with 'because', because 'because' is a conjunctio
print(str1[32:60])
```

'because', because 'because'

In [28]:

```
s = 'Python 4 Data Science'
s2= s.startswith('Python')
print(s2)
```

True

In [30]:

```
'Python' in 'Python 4 Data Science'
```

Out[30]:

True

In [32]:

```
s = 'Python 4 Data Science'
s.strip()
```

Out[32]:

'Python 4 Data Science'

In [35]:

```
a=['Django', 'Flask', 'Bottle', 'Pyramid', 'Falcon']
'#'.join(a)
```

Out[35]:

```
'Django#Flask#Bottle#Pyramid#Falcon'
```

In [37]:

```
a= "2021PythonDataypes"
b= a.isidentifier()
print(b)
c= "Python_Dataypes_2021"
d= c.isidentifier()
print(d)
```

False

True

In [64]:

```
m = 8
n = 6
sum1 = m + n
sum2 = m - n
sum3=m * n
sum4=m / n
sum5=m % n
sum6=m // n
sum7=m ** n
print("m+n:", sum1, "m-n:" , sum2, "m*n:", sum3, "m/n:", sum4, "m % n :", sum5, "m // n :", s
```

```
m+n: 14 m-n: 2 m*n: 48 m/n: 1.3333333333333333 m % n : 2 m // n : 1 m ** n : 262144
```

In [66]:

```
print("Name\tAge\tCountry\tCity")
print("Milaan\t96\tFinland\tTampere")
```

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

## Part C → List Level 1 & 2

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

## Part C → 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](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](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](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 [87]:

```
Solution:
a= ["Chen",20,1.63,"China","ChangChun"]
print(a)
print(type(a))
```

```
['Chen', 20, 1.63, 'China', 'ChangChun']
<class 'list'>
```

In [91]:

```
a=["Chen",20,1.63,"China","ChangChun"]
len(a)
```

Out[91]:

5

In [93]:

```
a=["Chen",20,1.63,"China","ChangChun"]
print(a[0])
print(a[2])
print(a[4])
```

Chen  
1.63  
ChangChun

In [31]:

```
called_my_info=["name","age","height","marital_status","country"]
print(called_my_info)
```

['name', 'age', 'height', 'marital\_status', 'country']

In [106]:

```
value_list=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
mix_fruits=[value_list]
print(mix_fruits)
```

[['Guava', 'Mango', 'Apple', 'Pear', 'Fig', 'Orange', 'Banana']]

In [109]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
print(len(mix_fruits))
```

7

In [112]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
print(mix_fruits[0])
print(mix_fruits[3])
print(mix_fruits[6])
```

Guava  
Pear  
Banana

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
mix_fruits[0]="Peach"
print(mix_fruits)
```

In [3]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
mix_fruits.append("Peach")
print(mix_fruits)
```

In [4]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
mix_fruits.insert(1,"Peach")
print(mix_fruits)
```

In [8]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
[a.upper() for a in ["Guava"]]
```

[ ' GUAVA ' ]

In [18]:

```
str="#-#";
mix_fruits=("Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana");
print (str.join(mix_fruits))
```

In [30]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
if "Mango" in mix_fruits:
 print ("True")
```

In [33]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
mix_fruits.sort()
print(mix_fruits)
```

15/27

In [38]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
mix_fruits.reverse()
print(mix_fruits)
```

```
['Banana', 'Orange', 'Fig', 'Pear', 'Apple', 'Mango', 'Guava']
```

In [44]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
print(mix_fruits[:3])
```

```
['Guava', 'Mango', 'Apple']
```

In [48]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
print(mix_fruits[4:])
```

```
['Fig', 'Orange', 'Banana']
```

In [52]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
print(mix_fruits[3:4])
```

```
['Pear']
```

In [56]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
mix_fruits.remove("Guava")
print(mix_fruits)
```

```
['Mango', 'Apple', 'Pear', 'Fig', 'Orange', 'Banana']
```

In [57]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
mix_fruits.remove("Pear")
print(mix_fruits)
```

```
['Guava', 'Mango', 'Apple', 'Fig', 'Orange', 'Banana']
```

In [58]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
mix_fruits.pop()
print(mix_fruits)
```

```
['Guava', 'Mango', 'Apple', 'Pear', 'Fig', 'Orange']
```



In [60]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
mix_fruits.clear()
print(mix_fruits)
```

[]

In [65]:

```
mix_fruits=["Guava", "Mango", "Apple", "Pear", "Fig", "Orange","Banana"]
del mix_fruits[0:7]
print(mix_fruits)
```

[]

In [67]:

```
front_end = ['HTML', 'CSS', 'JS', 'React', 'Redux']
back_end = ['Node', 'Express', 'MongoDB']
A=front_end +back_end
print(A)
```

['HTML', 'CSS', 'JS', 'React', 'Redux', 'Node', 'Express', 'MongoDB']

In [71]:

```
A=['HTML', 'CSS', 'JS', 'React', 'Redux','Node','Express', 'MongoDB']
A.insert(5, 'Python')
A.insert(6, 'SQL')
print(A)
```

['HTML', 'CSS', 'JS', 'React', 'Redux', 'Python', 'SQL', 'Node', 'Express', 'MongoDB']

## Part C → 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. `['India', 'Russia', 'China', 'Finland', 'Sweden', 'Norway', 'Denmark']` . Unpack the first three countries and the rest as scandic countries.

In [5]:

```
Solution:
ages = [19, 23, 19, 25, 21, 20, 25, 26, 25, 24]
ages.sort()
print(ages)
print (min(ages))
print (max(ages))
```

```
[19, 19, 20, 21, 23, 24, 25, 25, 25, 26]
19
26
```

In [18]:

```
ages = [19, 23, 19, 25, 21, 20, 25, 26, 25, 24]
ages.insert(0, "19")
ages.append("26")
print(ages)
```

```
['19', 19, 23, 19, 25, 21, 20, 25, 26, 25, 24, '26']
```

In [32]:

```
#Find the median age
```

In [64]:

```
ages = [19, 23, 19, 25, 21, 20, 25, 26, 25, 24]
mean = sum(ages)/len(ages)
print(mean)
```

```
22.7
```

In [19]:

```
max=ages[-1]
min=ages[0]
print(min,max)
```

```
19 26
```

In [76]:

```
mina=abs(-3.7)
maxa=abs(3.3)
print(mina)
print(maxa)
```

```
3.7
3.3
```

In [81]:

```
scandic_countries=['India', 'Russia', 'China', 'Finland', 'Sweden', 'Norway', 'Denma
print(scandic_countries[0:3])
print("scandic_countries:",scandic_countries[3:7])
```

```
['India', 'Russia', 'China']
```

```
scandic_countries: ['Finland', 'Sweden', 'Norway', 'Denmark']
```

## Part D → Tuple Level 1 & 2

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

### Part D → Level 1

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

1. Create a tuple containing names of fruits and vegetables
2. Join fruits and vegetables tuples and assign it to fruits\_vegetables
3. How many fruits\_vegetables do you have?
4. Modify the fruits\_vegetables tuple and add the name of your favorite mushroom and beverage and assign it to food\_tuple

In [5]:

```
Solution:
fruits=("peach")
vegetables=("tomato")
```

In [14]:

```
fruits=("peach")
vegetables=("tomato")
fruits_vegetables=(fruits,vegetables)
print("fruits_vegetables",fruits_vegetables)
```

```
fruits_vegetables ('peach', 'tomato')
```

In [15]:

```
fruits=("peach")
vegetables=("tomato")
fruits_vegetables=(fruits,vegetables)
print(len(fruits_vegetables))
```

2

In [24]:

```
fruits_vegetables=('peach', 'tomato', 'needle mushroom', 'orange juice')
favorite_mushroom="needle mushroom"
beverage="orange juice"
food_tuple=(fruits_vegetables, favorite_mushroom, beverage)
print("food_tuple", food_tuple)
```

```
File "<ipython-input-24-80977a13619b>", line 2
 favorite_mushroom="needle mushroom"
 ^
```

SyntaxError: invalid syntax

## Part D → Level 2

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

1. Unpack `fruits_vegetables` and `mushroom` and `beverage` from `food_tuple`
2. Change the about `food_tuple` tuple to a `food_list` list
3. Slice out the middle item or items from the `food_tuple` tuple or `food_list` list.
4. Slice out the first three items and the last three items from `food_list` list
5. Delete the `food_tuple` tuple completely
6. Check if an item exists in tuple:
  - Check if 'Finland' is a asian country
  - Check if 'India' is a asian country

```
asian_countries = ('India', 'China', 'Singapore', 'Thailand', 'Indonesia')
```

In [23]:

```
Solution:
fruits_vegetables=('peach', 'tomato', 'needle mushroom', 'orange juice')
print(fruits_vegetables[0:3])
```

```
('peach', 'tomato', 'needle mushroom')
```

In [25]:

```
food_tuple=('peach', 'tomato', 'needle mushroom', 'orange juice')
food_list=list(food_tuple)
print("food_list:", food_list)
print(type(food_list))
```

```
food_list: ['peach', 'tomato', 'needle mushroom', 'orange juice']
<class 'list'>
```

In [26]:

```
food_list: ['peach', 'tomato', 'needle mushroom', 'orange juice']
food_list[1:3]
```

Out[26]:

```
['tomato', 'needle mushroom']
```

In [27]:

```
food_list: ['peach', 'tomato', 'needle mushroom', 'orange juice']
food_list[0:3]
```

Out[27]:

```
['peach', 'tomato', 'needle mushroom']
```

In [28]:

```
food_list: ['peach', 'tomato', 'needle mushroom', 'orange juice']
food_list[-3:]
```

Out[28]:

```
['tomato', 'needle mushroom', 'orange juice']
```

In [22]:

```
food_list=['peach', 'tomato', 'needle mushroom', 'orange juice']
food_list.clear()
print(food_list)
```

```
[]
```

In [3]:

```
asian_countries = ('India', 'China', 'Singapore', 'Thailand', 'Indonesia')
if 'Finland' not in asian_countries:
 print("No, 'Finland' is not a asian country")
```

No, 'Finland' is not a asian country

In [1]:

```
asian_countries = ('India', 'China', 'Singapore', 'Thailand', 'Indonesia')
if 'India' in asian_countries:
 print("Yes, 'India' is a asian country")
```

Yes, 'India' is a asian country

## Part E → Sets Level 1, 2 and 3

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

```
mix_fruits = {'Guava', 'Pear', 'Mango', 'Apple', 'Fig', 'Orange', 'Banana'}
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
num = [22, 19, 24, 25, 26, 24, 25, 24]
```

**Part E** → **Level 1**

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

1. Find the length of the set `mix_fruits`
2. Add 'Kiwi' to `mix_fruits`
3. Insert multiple fruits at once to the set `mix_fruits`
4. Remove one of the fruit from the set `mix_fruits`
5. What is the difference between **remove** and **discard**

In [58]:

```
Solution:
mix_fruits = {'Guava', 'Pear', 'Mango', 'Apple', 'Fig', 'Orange', 'Banana'}
print(len(mix_fruits))
```

7

In [60]:

```
mix_fruits = {'Guava', 'Pear', 'Mango', 'Apple', 'Fig', 'Orange', 'Banana'}
mix_fruits.add('Kiwi')
print(mix_fruits)
```

{'Orange', 'Banana', 'Kiwi', 'Pear', 'Mango', 'Fig', 'Apple', 'Guava'}

In [62]:

```
mix_fruits = {'Guava', 'Pear', 'Mango', 'Apple', 'Fig', 'Orange', 'Banana'}
mix_fruits.update(["watermelon", "lemon", "strawberry"])
print(mix_fruits)
```

{'Orange', 'strawberry', 'Banana', 'watermelon', 'Pear', 'Mango', 'Fig', 'Apple', 'Guava', 'lemon'}

In [64]:

```
mix_fruits = {'Guava', 'Pear', 'Mango', 'Apple', 'Fig', 'Orange', 'Banana'}
mix_fruits.remove("Pear")
print(mix_fruits)
```

{'Orange', 'Banana', 'Mango', 'Fig', 'Apple', 'Guava'}

In [ ]:

```
#If the item to be deleted does not exist,remove () raises an error.
#If the item to be deleted does not exist,discard () does not raise an error.
```

**Part E** → **Level 2**

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

Use Imaginary values for Set A and B

1. Join A and B
2. Find A intersection B
3. Is A subset of B
4. Are A and B disjoint sets
5. Join A with B and B with A
6. What is the symmetric difference between A and B
7. Delete the sets completely

In [67]:

```
Solution:
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
print(A|B)
```

```
{19, 20, 22, 24, 25, 26, 27, 28}
```

In [69]:

```
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
print(A&B)
```

```
{19, 20, 22, 24, 25, 26}
```

In [71]:

```
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
A.issubset(B)
```

Out[71]:

```
True
```

In [73]:

```
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
A.isdisjoint(B)
```

Out[73]:

```
False
```

In [83]:

```
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
C = A.union(B)
print(C)

A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
D = B.union(A)
print(D)
```

```
{19, 20, 22, 24, 25, 26, 27, 28}
{19, 20, 22, 24, 25, 26, 27, 28}
```

In [85]:

```
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
E = A.symmetric_difference(B)
print(E)
```

```
{27, 28}
```

In [87]:

```
A.clear()
print(A)
B.clear()
print(B)
```

```
set()
set()
```

## Part E → Level 3

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

1. Convert the `num` to a set and compare the length of the list and the set, which one is bigger?
2. Explain the difference between the following data types: **string**, **list**, **tuple** and **set**
3. I am a researcher cum teacher and I love to inspire and teach people. . How many unique words have been used in the sentence? Use the `split` methods and set to get the unique words.



In [1]:

```
Solution:
num = [22, 19, 24, 25, 26, 24, 25, 24]
s = set(num)
print(s)
c = list(num)
print(c)
print(type(s))
print(len(num))
print(len(s))
print(len(c))
```

```
{19, 22, 24, 25, 26}
[22, 19, 24, 25, 26, 24, 25, 24]
<class 'set'>
8
5
8
```

In [97]:

```
#string: String cannot be changed
#list:ordered collection, can add and subtract
#tuple:ordered list, cannot be modified
#set: unordered sets, keys do not repeat
```

In [2]:

```
a = "I am a researcher cum teacher and I love to inspire and teach people."
b = a.split()
print(b)
print(len(b))
```

```
['I', 'am', 'a', 'researcher', 'cum', 'teacher', 'and', 'I', 'love', 'to', 'inspir
e', 'and', 'teach', 'people.']
14
```

## Part F → Dictionary Level 1

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

1. Create an empty dictionary called `bird`
2. Add `name`, `color`, `breed`, `legs`, `age` to the `bird` dictionary
3. Create a `student` dictionary and add `first_name`, `last_name`, `gender`, `age`, `marital_status`, `skills`, `country`, `city` and `address` as keys for the dictionary
4. Get the length of the `student` dictionary
5. Get the value of `skills` and check the data type, it should be a list
6. Modify the `skills` values by adding one or two skills
7. Get the dictionary keys as a list
8. Get the dictionary values as a list
9. Change the dictionary to a list of tuples using [items\(\)](#)

([https://github.com/milaan9/02\\_Python\\_Datatypes/blob/main/005\\_Python\\_Dictionary\\_Methods/005\\_P](https://github.com/milaan9/02_Python_Datatypes/blob/main/005_Python_Dictionary_Methods/005_P)

method

10. Delete one of the items in the dictionary

11. Delete one of the dictionaries

In [22]:

```
Solution:
```

```
bird={}
print(bird)
bird={"name":"bird","color":"yellow","breed":"cuckoo","legs":2,"age":2}
print(bird)
```

```
{
{'name': 'bird', 'color': 'yellow', 'breed': 'cuckoo', 'legs': 2, 'age': 2}
```

In [24]:

```
student={}
print(student)
student={"first_name":"Pengjia","last_name":"Chen","gender":"female","age":"20","marital_status":"unmarried","skills":"study","country":"China","city":"Changchun","address":"JiLin"}
print(student)
```

```
{
{'first_name': 'Pengjia', 'last_name': 'Chen', 'gender': 'female', 'age': '20', 'marital_status': 'unmarried', 'skills': 'study', 'country': 'China', 'city': 'Changchun', 'address': 'JiLin'}
```

In [27]:

```
print(len(student))
```

9

In [30]:

```
skills=[]
print(type(skills))
```

<class 'list'>

In [32]:

```
skill1,skill2=["study"],["sleep"]
skill=skill1+skill2
print(skill)
```

['study', 'sleep']

In [34]:

```
key_list=["first_name","last_name","gender","age","marital_status","skills","country"]
```

In [35]:

```
value_list=["Pengjia", "Chen", "female", "20", "unmarried", "study", "China", "Changchun",
```

In [56]:

```
student={"first_name":"Pengjia","last_name":"Chen","gender":"female","age":"20","ma:
s=student.items()
print(s)
print(type(s))
```

```
dict_items([('first_name', 'Pengjia'), ('last_name', 'Chen'), ('gender', 'female'),
('age', '20'), ('marital_status', 'unmarried'), ('skills', 'study'), ('country', 'Ch
ina'), ('city', 'Changchun'), ('address', 'JiLin')])
<class 'dict_items'>
```

In [52]:

```
student={"first_name":"Pengjia","last_name":"Chen","gender":"female","age":"20","ma:
pop_obj=student.popitem()
print(pop_obj)
```

```
('address', 'JiLin')
```

In [54]:

```
student={"first_name":"Pengjia","last_name":"Chen","gender":"female","age":"20","ma:
student.clear();
```