**Assignment - 1**

**Question 1:**

#The following is a list of 10 students ages:

#ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

#import numpy data

import numpy as np

ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

#• Sort the list and find the min and max age

ages.sort()

Ages

min(ages), max(ages)

#• Add the min age and the max age again to the list

ages.append(min(ages))

ages.append(max(ages))

ages

#• Find the median age (one middle item or two middle items divided by two)

ages.sort()

np.median(ages)

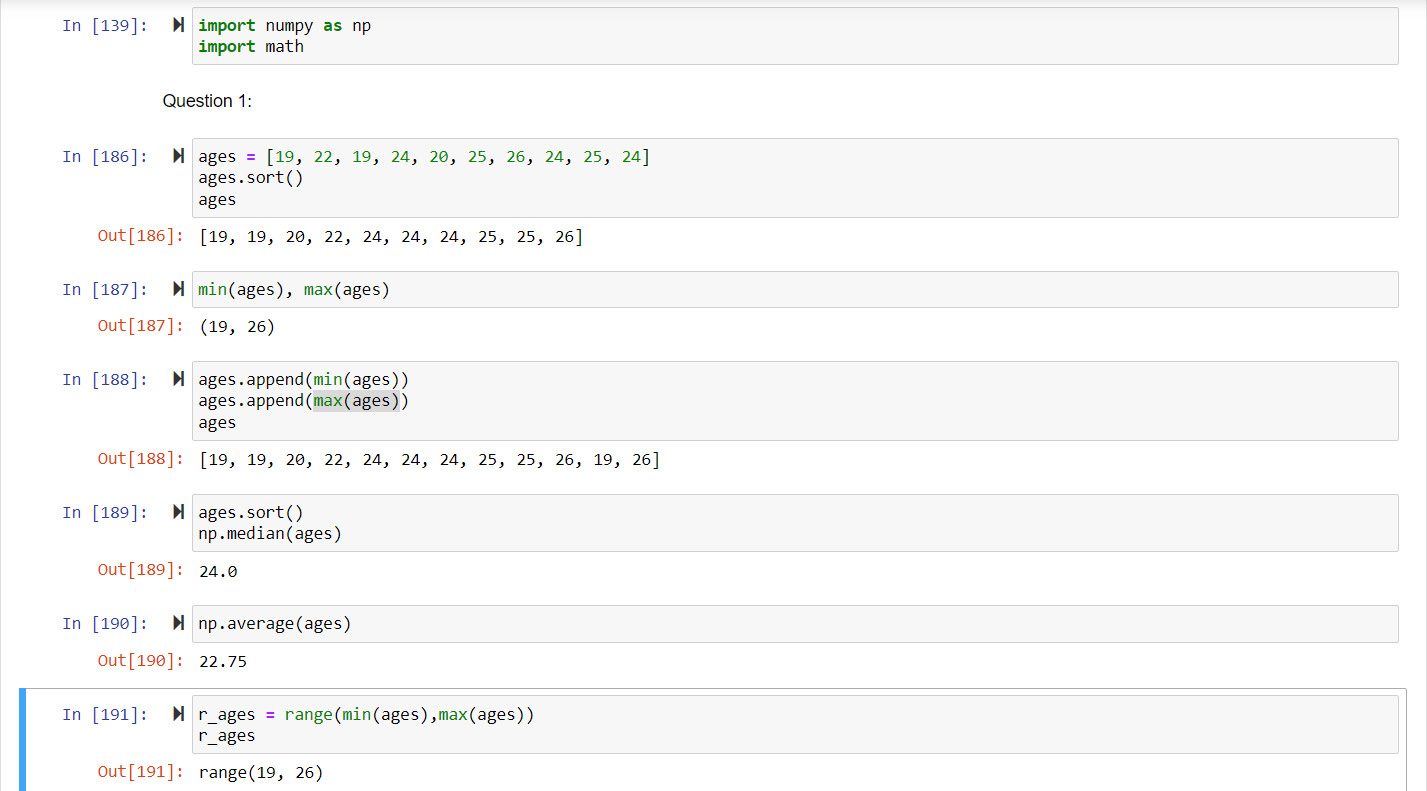
#• Find the average age (sum of all items divided by their number)

np.average(ages)

#• Find the range of the ages (max minus min)

r\_ages = range(min(ages),max(ages))

r\_ages

****

**Question 2:**

#• Create an empty dictionary called dog

#• Add name, color, breed, legs, age to the dog dictionary

dog = { }

dog = {

'name' : 'Joe',

'color' : 'white',

'breed' : 'Bichon Frise',

'legs' : 'four',

'age' : '13 months'

}

dog

#• Create a student dictionary and add first\_name, last\_name, gender, age, marital status, skills, #country, city and address as keys for the dictionary

student = {

'first\_name' : 'Navasahitha',

'last\_name' : 'Inuganti',

'gender' : 'Female',

'age' : 25,

'marital\_status' : 'married',

'skills' : ['never give up attitude'],

'city' : 'St.Louis',

'address' : 'Basston Dr'

}

student

#• Get the length of the student dictionary

len(student)

#• Get the value of skills and check the data type, it should be a list

print(student['skills'])

print(type(student['skills']))

#• Modify the skills values by adding one or two skills

student['skills'].append('multi-tasking')

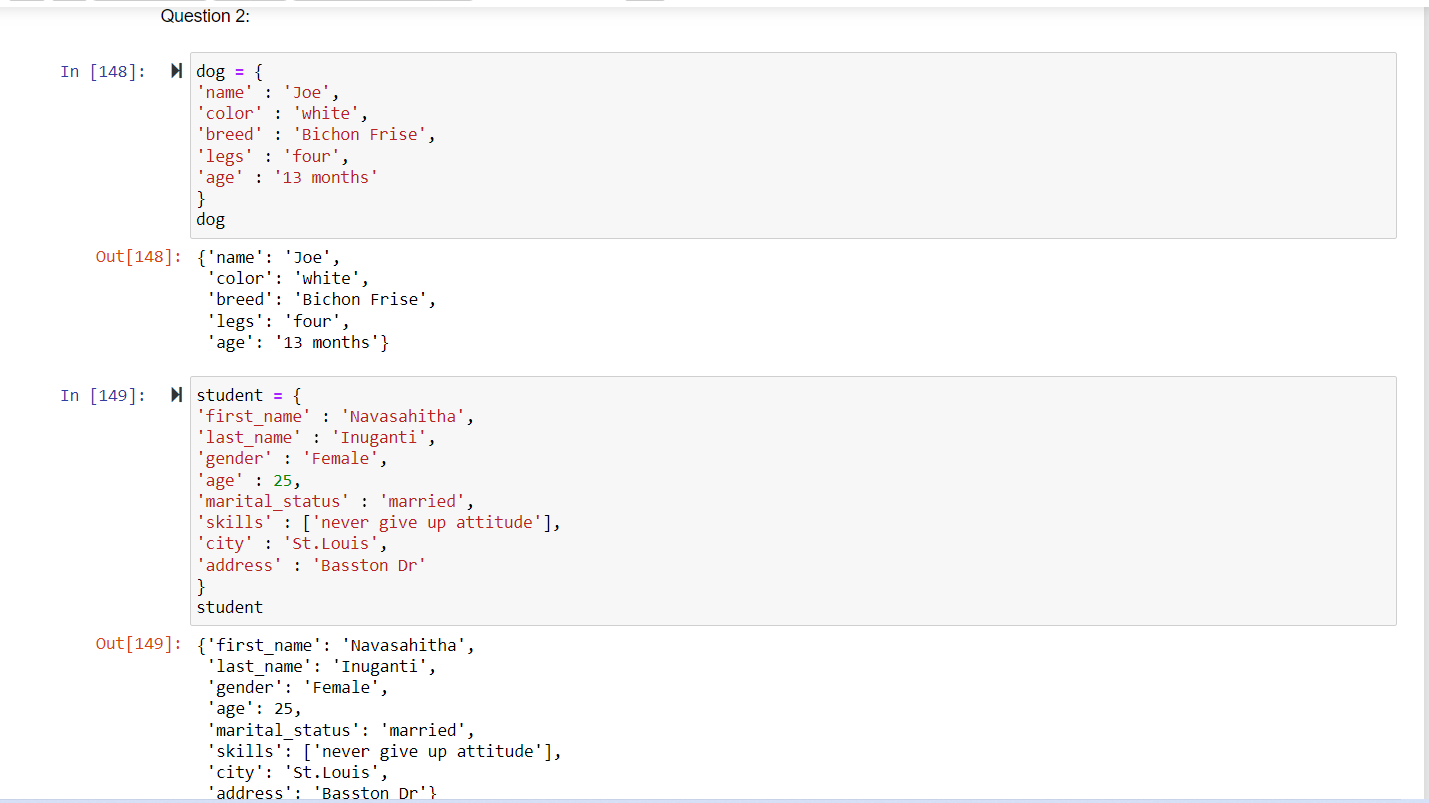
student

#• Get the dictionary keys as a list

student.keys()

#• Get the dictionary values as a list

student.values()





**Question 3:**

#• Create a tuple containing names of your sisters and your brothers (imaginary siblings are #fine)

siblings = ('sahitha','yaswanth','santhi','bavina','priyansh')

siblings

#• Join brothers and sisters tuples and assign it to siblings

sisters = ('sahitha','santhi','bavina')

brothers = ('yaswanth','priyansh')

new\_siblings = sisters + brothers

new\_siblings

#• How many siblings do you have?

len(new\_siblings)

#• Modify the siblings tuple and add the name of your father and mother and assign it to #family\_members

father = 'prasad'

mother = 'kiran'

family\_members = new\_siblings + (father,mother)

family\_members



**Question 4:**

it\_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}

A = {19, 22, 24, 20, 25, 26}

B = {19, 22, 20, 25, 26, 24, 28, 27}

age = [22, 19, 24, 25, 26, 24, 25, 24]

#• Find the length of the set it\_companies

len(it\_companies)

#• Add 'Twitter' to it\_companies

it\_companies.add('Twitter')

it\_companies

#• Insert multiple IT companies at once to the set it\_companies

it\_companies.update(['Equifax','Cognizant'])

it\_companies

#• Remove one of the companies from the set it\_companies

it\_companies.remove('Cognizant')

it\_companies

#• What is the difference between remove and discard

it\_companies.remove('Cognizant')

it\_companies

it\_companies.discard('Cognizant')

it\_companies

#If we use remove for the same value after removing it throws an exception whereas for discard #it does not throw any exceptions.

#• Join A and B

A.union(B)

#• Find A intersection B

A.intersection(B)

#• Is A subset of B

A.issubset(B)

#• Are A and B disjoint sets

A.isdisjoint(B)

#• Join A with B and B with A

A.union(B) and B.union(A)

#• What is the symmetric difference between A and B

A.symmetric\_difference(B)

#• Delete the sets completely

del A

del B

#• Convert the ages to a set and compare the length of the list and the set.

set\_ages = set(age)

set\_ages

len(age),len(set\_ages)

len(age) == len(set\_ages)





**Question 5:**

#The radius of a circle is 30 meters.

#• Calculate the area of a circle and assign the value to a variable name of \_area\_of\_circle\_

import math

radius = 30

\_area\_of\_circle\_ = math.pi\*radius\*\*2

\_area\_of\_circle\_

#• Calculate the circumference of a circle and assign the value to a variable name of #\_circum\_of\_circle\_

\_circum\_of\_circle\_ = 2\*math.pi\*radius

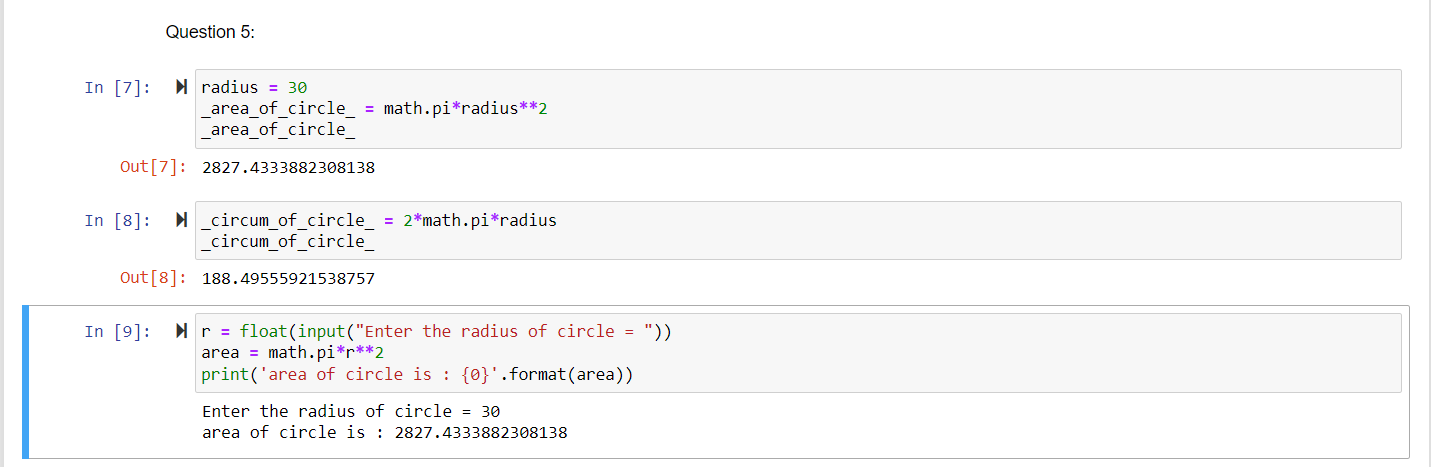
\_circum\_of\_circle\_

#• Take the radius as user input and calculate the area.

r = float(input("Enter the radius of circle = "))

area = math.pi\*r\*\*2

print('area of circle is : {0}'.format(area))



**Question 6:**

#“I am a 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.

sentence = "I am a teacher and I love to inspire and teach people"

s1 = sentence.split(' ')

set\_s1=set(s1)

set\_s1

len(set\_s1)

**Question 7:**

#Use a tab escape sequence to get the following lines.

print('Name\tAge\tCountry\tCity\nAsabeneh\t250\tFinlan\tHelsinki')

**Question 8:**

#Use the string formatting method to display the following:

#radius = 10

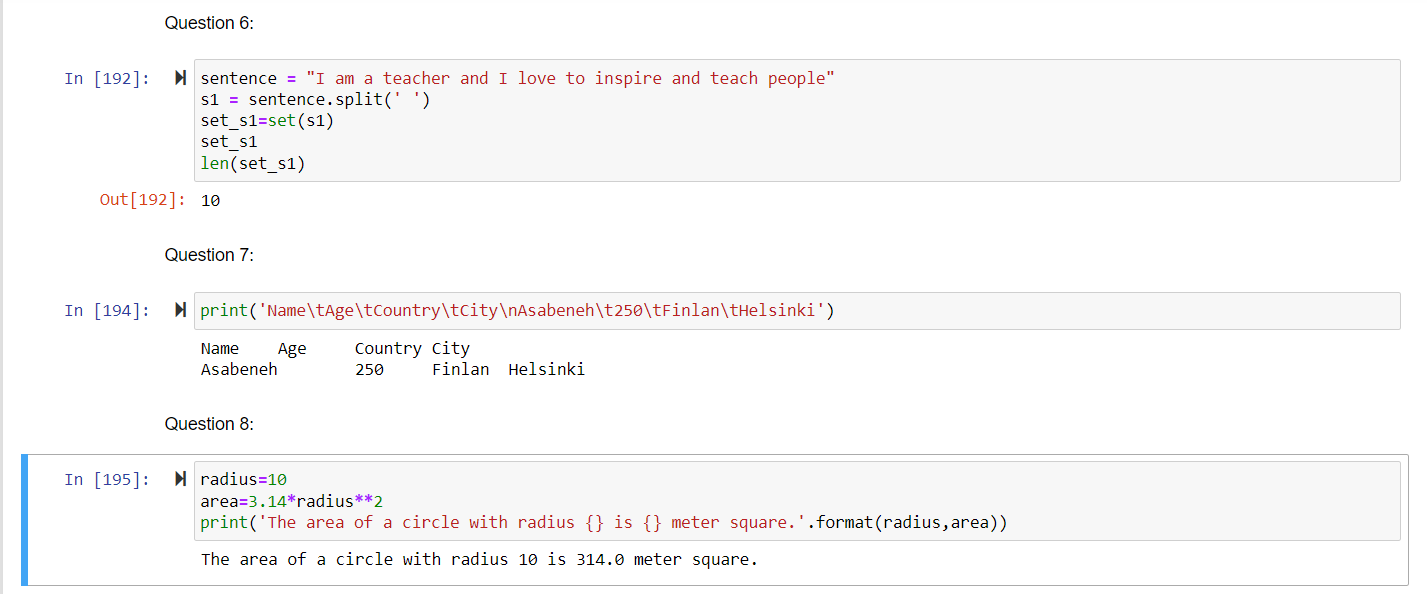
#area = 3.14 \* radius \*\* 2

#“The area of a circle with radius 10 is 314 meters square.”

radius = 10

area = 3.14\*radius\*\*2

print('The area of a circle with radius {} is {} meter square.'.format(radius,area))



**Question 9:**

#Write a program, which reads weights (lbs.) of N students into a list and convert these weights to kilograms in a separate list using Loop. N: No of students (Read input from user)

#Ex: L1: [150, 155, 145, 148]

#Output: [68.03, 70.3, 65.77, 67.13]

weights = []

weight\_in\_kgs = []

no\_of\_students = int(input("Enter no of students: "))

print('Enter student weights in lbs:')

for i in range(0, no\_of\_students):

weights.append(int(input()))

for i in range(0, no\_of\_students):

weight\_in\_kgs.append(round(0.453592 \* weights[i], 2))

print(weight\_in\_kgs)



**Question 10:**

Given the data set consists of 2 classes and 8 data points

Let the classes be C1 and C2

Where as data points be dots = 1 and X = 0

Therefore dataset = [[1,0],[2,1],[3,0],[6,0],[6,0],[7,1],[10,1],[11,1]]

C1 = [[1],[2],[3],[6],[6],[7],[10],[11]]

C2 = [1,1,0,0,0,1,1,1]

1. Now divide the data equally into 2 parts, first part is training and second part is testing

Training = [[1,1],[2,1],[3,0],[6,0]]

Testing = [[6,0],[7,1],[10,1],[11,1]]

Given k = 3

We are going to find the euclidean distance between data points.

d = √[ (x22 – x11)2

Let’s find out the predicted output for the first data point in the testing

For the testing 6 and training 6 i.e (6,6) d = 0

(6,3) d = 3

(6,2) d = 4

(6,1) d = 5

As per the data the 3 nearest neighbor labels for this datapoint is (0,0,1)

so , the predicted label for the testing 6 is 0. Similarly , for 7,10,11 are 0,0,0.

The testing set with predicted label is

Ptesting = [[6,0,0],[7,1,0],[10,1,0],[11,1,0]]

1. True positive(TP) : When the actual label is positive(1) and the machine learning model also predicts the label as positive(1).

True negative(TN) : When the actual label is negative(0) and the machine learning model also predicts the label as negative(0).

False positive(FP) : When the actual label is negative(0) and the machine learning model also predicts the label as positive(1).

False negative(FN) : When the actual label is positive(1) and the machine learning model also predicts the label as negative(0).

Confusion matrix :

[[6,0,0,TN],

[7,1,0,FN],

[10,1,0,FN],

[11,1,0,FN]]

Therefore total TN = 1, FN = 3, TP = 0, FP = 0

Accuracy = (TP+TN)/(TP+TN+FP+FN) = 0.25

Sensitivity = TP/(TP+FN) = 0

Specificity = TN/(FP+TN) = 1

**Video Explanation : https://github.com/NXI57230/Assignment1\_700725723.git**