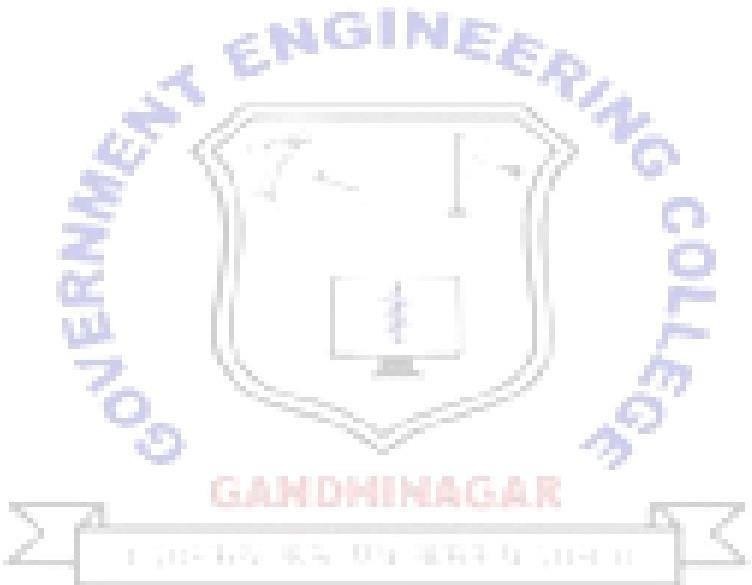
Government Engineering College

**Sec-28 Gandhinagar**

**Sem:-V (Computer Engineering Department) Subject: Python with Data Science [3150713]**

**Certificate**

**This is to certify that**



Mr./Ms

**NIRAJKUMAR ITALIYA**

Of

class

**CE\_Sem-5** Division **A2** Enrollment No **190130107041** Has

satisfactorily

completed

his/her

term

work

in

**PDS(*3150713)***

Subjec t

for

the term

ending in

November **2021-2022**.

Date:- **27/09/2021**

Practicals

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | **Basic Python Practicals** |  | **Date** | **Page No.** | **Sign** |
| 1 | **Develop a program to understand the control structures of python.** | CO1 | 31/07/2021 |  |  |
| 2 | **Develop a program to learn different types of structures (list, dictionary, tuples) in python.** | CO1 | 31/07/2021 |  |  |
| 3 | **Develop a program to learn concepts of functions scoping, recursion and list mutability.** | CO1 | 31/07/2021 |  |  |
| 4 | **Develop a program to understand working of exception handling and assertions.** | CO1 | 31/07/2021 |  |  |
| 5 | **Develop a program for data structure algorithms using python**  **– searching, sorting, and hash tables.** | CO1 | 31/07/2021 |  |  |
| 6 | **Develop a program to learn regular expressions using python.** | CO1 | 31/07/2021 |  |  |
| 7 | **Develop chat room applications using multithreading.(Assignment)** | CO1 | 31/07/2021 |  |  |
| 8 | **Learn to plot different types of graphs using PyPlot.** | CO1 | 31/07/2021 |  |  |
| 9 | **Implement classical ciphers using python. .(Assignment)** | CO1 | 31/07/2021 |  |  |
| 10 | **Draw graphics using Turtle. .(Assignment)** | CO1 | 31/07/2021 |  |  |
| 11 | **Develop a program to learn GUI programming using Tkinter. .(Assignment)** | CO1 | 31/07/2021 |  |  |
|  | **Pandas Library Practicals** |  |  |  |  |
| 12 | **Develop a program that reads a .csv dataset file using Pandas library and display the following content of the dataset.**   1. **First five rows of the dataset** 2. **Complete data of the dataset** 3. **Summary or metadata of the dataset** | CO2 | 30/08/2021 |  |  |
| 13 | **Develop a program that shows application of slicing and dicing over the raws and columns of the dataset.** | CO2 | 30/08/2021 |  |  |
| **14** | **Develop a program that shows usage of aggregate function over the input dataset.**   1. **describe** 2. **max** 3. **min** 4. **mean** 5. **median** 6. **count** 7. **std** 8. **Corr** | **CO2** | 30/08/2021 |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **15** | **Develop a program that applies split and merge operations on the datasets.** | **CO2** | 30/08/2021 |  |  |
| **16** | **Develop a program that shows the various data cleaning tasks over the dataset.**   1. **Identifying the null values** 2. **Identifying the empty values** 3. **Identifying the incorrect timestamp** | **CO3** | 30/08/2021 |  |  |
| **17** | **Develop a program that shows an application of a Lamda function over the dataset.** | **CO2** | 30/08/2021 |  |  |
|  |  |  |  |  |  |
|  | **NumPy Library Practicals** |  |  |  |  |
| **18** | **Develop a program that shows usage of following NumPy array operations:**   1. **any() h) isreal()** 2. **all() i) iscomplex()** 3. **isnan() j) isscalar()** 4. **isinf() k) less()** 5. **isfinite() l) greater()** 6. **isinf() m) less\_equal()** 7. **zeros() o) greater\_equal()** | **CO3** | 10/09/2021 |  |  |
| **19** | **Develop a program that shows usage of following NumPy library vector functions.**   1. **arrange()** 2. **reshape()** 3. **linspace()** 4. **randint()** 5. **dot()** | **CO3** | 10/09/2021 |  |  |
|  |  |  |  |  |  |
|  | **Matplotlib Library Practicals** |  |  |  |  |
| **20** | **Write a program to display below plot using matplotlib library** |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **For Values of X:[1,2,3,...,49], Values of Y (thrice of X):[3,6,9,12,...,144,147]** | CO4 | 10/09/2020 | 66-6 |  |
| 21 | **Write a program to display below plot using matplotlib library For the point values**  **x1 = [10,20,30], y1 = [20,40,10]**  **x2 = [10,20,30], y2 = [40,10,30]** | CO4 | 10/09/2020 | 67-67 |  |
| 22 | **Write a program to display below bar plot using matplotlib library**  **For value**  **Languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++'] popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]** | CO4 | 10/09/2020 | 68-69 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 23 | **Write a program to display below bar plot using matplotlib library**  **For below data display pie plot**  **languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++'] popuratity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]** | CO4 | 10/09/2020 | 70-70 |  |
|  | **colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728",** |
|  | **"#9467bd", "#8c564b"]** |
| 24 | **Write a program to display below bar plot using matplotlib library**  **For 200 random points for both X and Y display scatter plot:** | CO4 | 10/09/2020 | 71-71 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 25 | **Develop a program that reads .csv file from the url: (**[**https://github.com/chris1610/pbpython/blob/master/data/sam**](https://github.com/chris1610/pbpython/blob/master/data/sample-salesv3.xlsx?raw=true)[**ple-salesv3.xlsx?raw=true**](https://github.com/chris1610/pbpython/blob/master/data/sample-salesv3.xlsx?raw=true)**) and plot the data of the dataset stored in the .csv file.** | CO4 | 10/09/2020 | 72-72 |  |
|  |  |  |  |  |  |
|  | **Scikit Learn Practicals** |  |  |  |  |
| 26 |  | CO5 | 15/10/2020 | 73-73 |  |
| **Exercise 1: Language identification** |
| * **Write a text classification pipeline using a custom preprocessor and CharNGramAnalyzer using data from Wikipedia articles as a training set.** * **Evaluate the performance on some held out test sets.** |
| 27 |  | CO5 | 15/10/2020 | 74-75 |  |
| **Exercise 2: Sentiment Analysis on movie reviews** |
| * **Write a text classification pipeline to classify movie reviews as either positive or negative.** * **Find a good set of parameters using grid search.** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | * **Evaluate the performance on a held out test set.** |  |  |  |  |
| 28 |  | CO5 | 15/10/2020 | 76-77 |  |
| **CLI text classification utility**  **Using the results of the previous exercises and the cPickle module of the standard library, write a command line utility that detects the language of some text provided on stdin and**  **estimates the polarity (positive or negative) if the text is written in English.** |
| **Bonus point if the utility is able to give a confidence level for its predictions.** |

Practicals

**enroll = 190130107041**

#### Develop a program to understand the control structures of python.

print("Enrollment no =",enroll)

print("Practical no= 1")

for i in range(5):

if i%2 == 0:

print(str(i) + ' Even')

elif i%3==0:

print(str(i) + ' divisible by 3')

else:

print(str(i) + ' odd')

i=0

while i < 3:

print("I'm loopy!")

i += 1

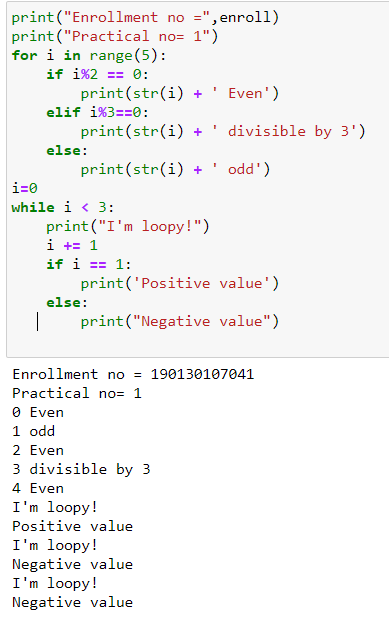
if i == 1:

print('Positive value')

else:

print("Negative value")

**Output :**



1. **Develop a program to learn different types of structures (list, dictionary, tuples) in python.**

print("Enrollment no =",enroll)

print("Practical no= 2")

print("Practical Title= understand different types of structure(list, dictionary, tuples)")

lst = list()

for i in range(1, 6):

lst.append(i)

print('Our list:', lst)

lst[2] = 'PYTHON'

print('after mutation:', lst)

dct = {}

fruit = ['apple', 'banana', 'grapes']

for i in range(1, 4):

dct[i] = fruit[i-1]

print('dict:', dct)

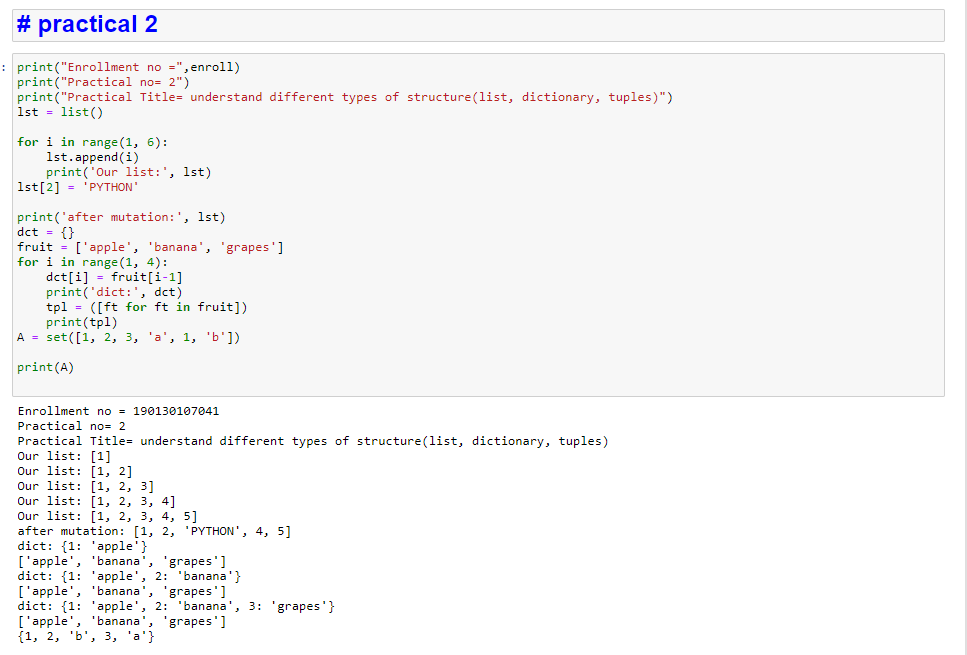
tpl = ([ft for ft in fruit])

print(tpl)

A = set([1, 2, 3, 'a', 1, 'b'])

print(A)

**Output :**



1. **Develop a program to learn concepts of functions scoping, recursion and list mutability.**

print("Enrollment no =",enroll)

print("Practical no= 3")

print("Practical Title= understand the functions scoping, recursion and list mutabilit")

import datetime as dt

def myfunc():

x=100

print(x)

myfunc()

def get\_date():

dt = datetime.datetime.now().date()

def printfnc():

print("Today's date: " + str(dt))

printfnc()

get\_date()

def my\_name():

name = 'Joey'

print(name)

name='Joey Tribbiani'

my\_name()

def get\_itr(num):

if num <= 0: return 1

elif num == 1: return 1

else:

return num\*get\_itr(num-1)

ans = get\_itr(5)

print(ans) #list

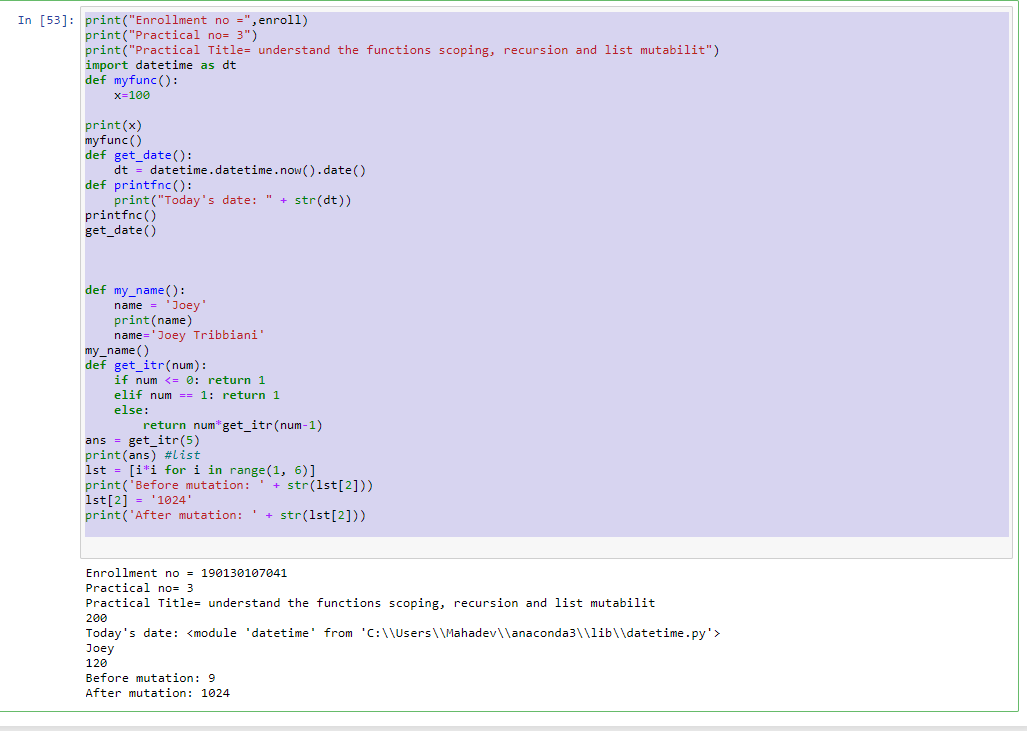
lst = [i\*i for i in range(1, 6)]

print('Before mutation: ' + str(lst[2]))

lst[2] = '1024'

print('After mutation: ' + str(lst[2]))

**Output:**

‘

1. **Develop a program to understand working of exception handling and assertions. Practical 4.1:**

print("Enrollment no =",enroll) print("Practical no= 4.1")

print("Practical Title= understand the working of assertion") #P4\_1

#assertion

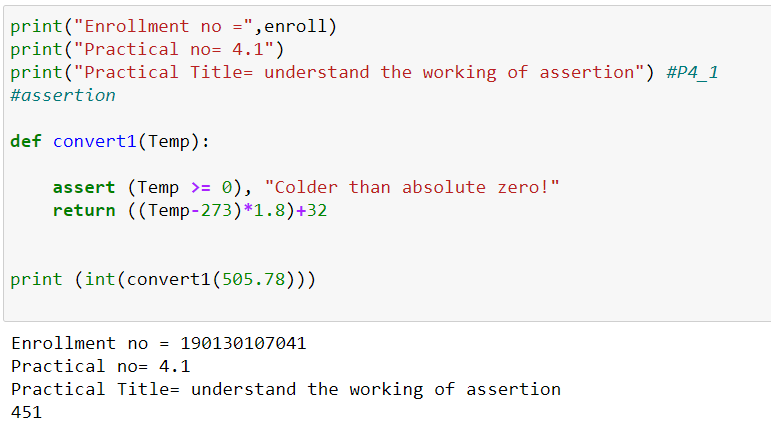
def convert1(Temp):

assert (Temp >= 0), "Colder than absolute zero!" return ((Temp-273)\*1.8)+32

#print (convert1(273))

print (int(convert1(505.78))) #print (convert1(-5))

**Output:**



**Practical 4.2:**

print("Enrollment no =",enroll) print("Practical no= 4.2")

print("Practical Title= understand the excaption handling") #P4\_2

a = 100

b = int(input('Enter b: '))

try:

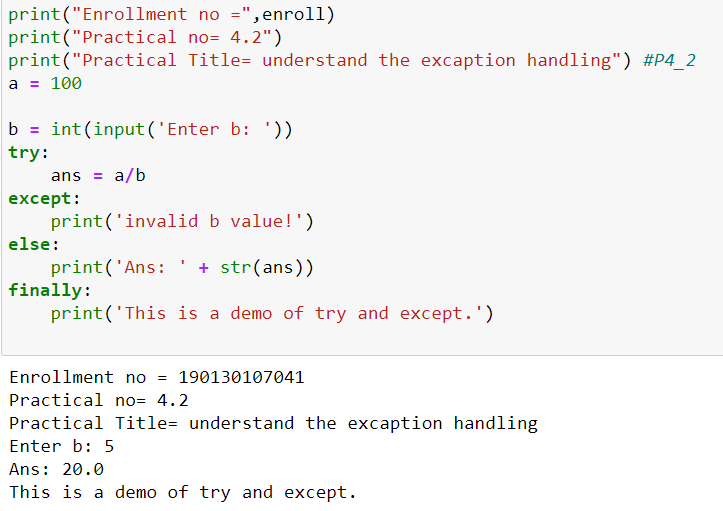
ans = a/b except:

print('invalid b value!') else:

print('Ans: ' + str(ans)) finally:

print('This is a demo of try and except.')

**Output :**



1. **Develop a program for data structure algorithms using python – searching, sorting, and hash tables.**

print("Enrollment no =",enroll) print("Practical no= 5.1")

print("Practical Title= insertion sort algorithm ") #P5\_1

def insertionSort(arr):

for i in range(1,len(arr)): key = arr[i]

j = i-1

while j >= 0 and key < arr[j] : arr[j+1] = arr[j]

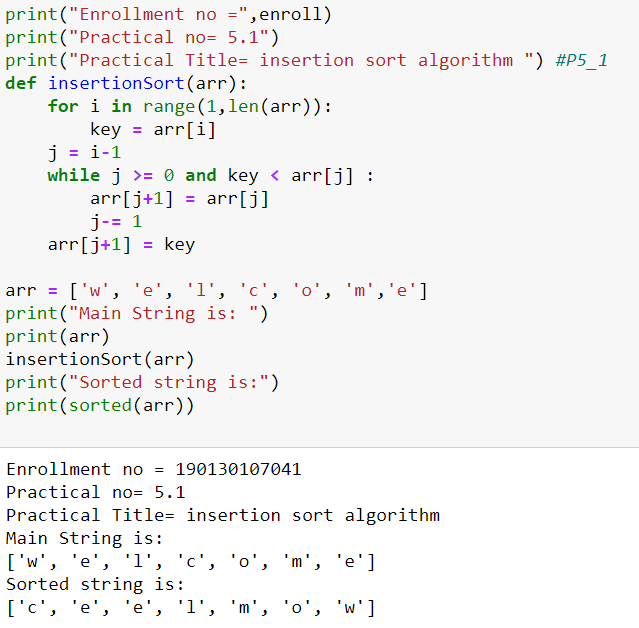
j-= 1

arr[j+1] = key

arr = ['w', 'e', 'l', 'c', 'o', 'm','e'] print("Main String is: ") print(arr)

insertionSort(arr) print("Sorted string is:") print(sorted(arr))

**Output:**



**Practical 5.2:**

print("Enrollment no =",enroll) print("Practical no= 5.2")

print("Practical Title= Mergesort algorithm") #P5\_2

def merge(arr, l, m, r): n1 = m - l + 1

n2 = r- m

L = [0] \* (n1)

R = [0] \* (n2)

for i in range(0 , n1): L[i] = arr[l + i]

for j in range(0 , n2): R[j] = arr[m + 1 + j]

i = 0

j = 0 k = l

while i < n1 and j < n2 : if L[i] <= R[j]:

arr[k] = L[i] i += 1

else:

arr[k] = R[j] j += 1

k += 1

while i < n1: arr[k] = L[i]

i += 1

k += 1

while j < n2: arr[k] = R[j] j += 1

k += 1

def mergeSort(arr,l,r):

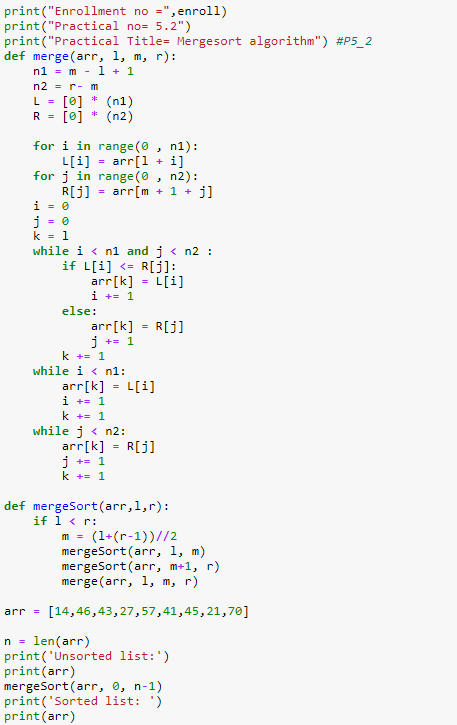
if l < r:

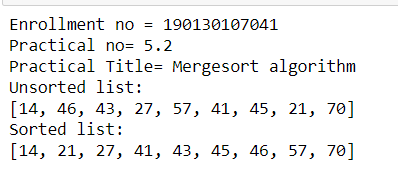
m = (l+(r-1))//2 mergeSort(arr, l, m) mergeSort(arr, m+1, r) merge(arr, l, m, r)

arr = [14,46,43,27,57,41,45,21,70]

n = len(arr) print('Unsorted list:') print(arr) mergeSort(arr, 0, n-1) print('Sorted list: ') print(arr)

**Output :**





1. **Develop a program to learn regular expressions using python.**

print("Enrollment no =",enroll) print("Practical no= 6")

print("Practical Title= understand the regular expression") #P6

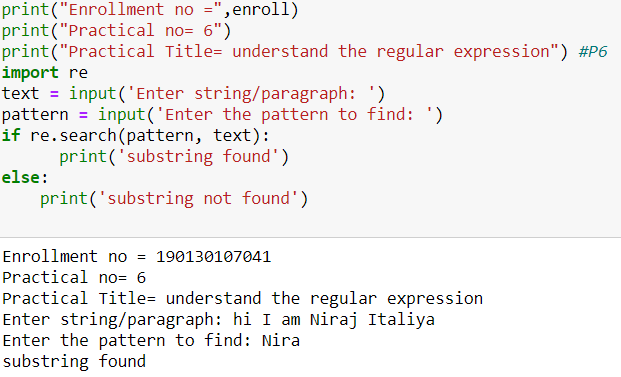
import re

text = input('Enter string/paragraph: ') pattern = input('Enter the pattern to find: ') if re.search(pattern, text):

print('substring found') else:

print('substring not found')

Output :



1. **Learn to plot different types of graphs using PyPlot.**

print("Enrollment no =",enroll)

print("Practical no= 8")

print("Practical Title= learn different types of graphs ") #P8\_1

import matplotlib.pyplot as plt

x = [1,2,3,4,5,6,7,8,9]

y = [1,8,27,64,125,256,343,512,729]

plt.plot(x, y)

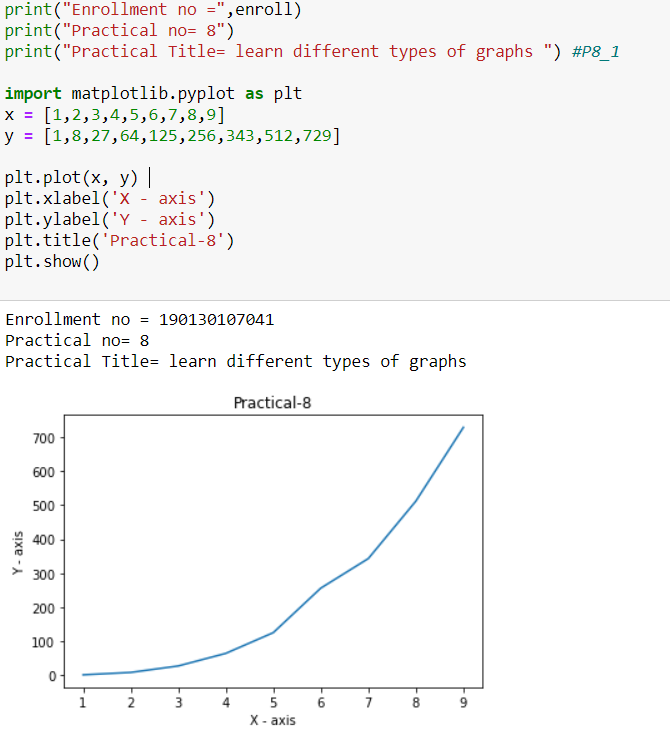
plt.xlabel('X - axis')

plt.ylabel('Y - axis')

plt.title('Practical-8')

plt.show()

**Output:**



**Practical 8.2:**

print("Enrollment no =",enroll)

print("Practical no = 8.2")

print("Practical Title= learn different types of graphs ") #P8\_2

#import matplotlib.pyplot as plt

def compound\_interest(principle, rate, time):

result = principle \* (pow((1 + rate / 100), time))

return result

p = float(input("Enter the principal amount: "))

r = float(input("Enter the interest rate: "))

endyear = float(input("Enter the Year : "))

yearlist = []

pamountlist = []

interestlist = []

for i in range(int(endyear)):

yearlist.append(i)

amount = compound\_interest(p, r, i)

intamount = int(amount)

pamountlist.append(intamount)

interest = amount - p

intins = int(interest)

interestlist.append(intins)

print(yearlist)

print(pamountlist)

print(interestlist)

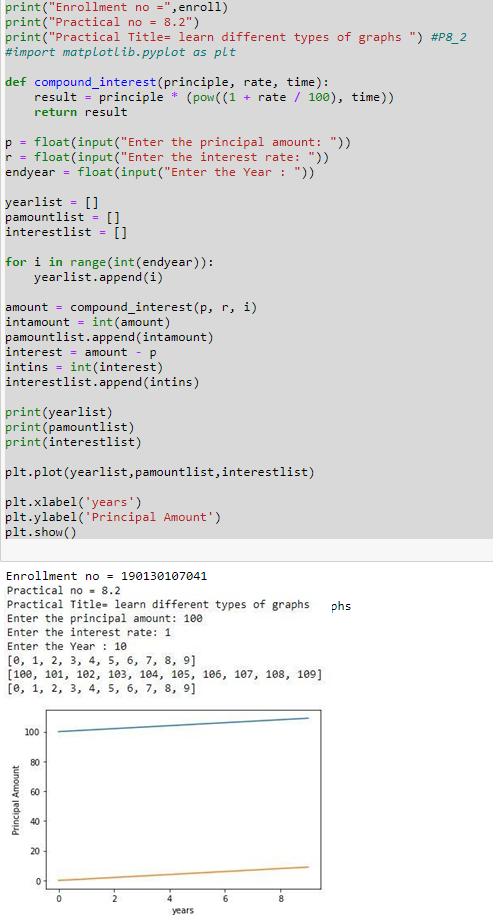
plt.plot(yearlist,pamountlist,interestlist)

plt.xlabel('years')

plt.ylabel('Principal Amount')

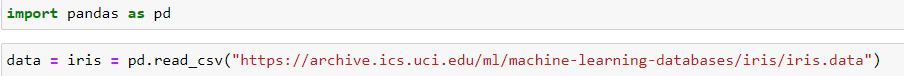
plt.show()

**Output:**

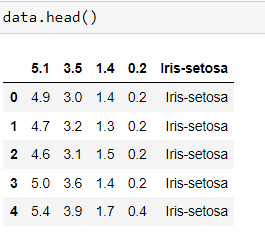


**Pandas library practicals:**

Develop a program that reads a .csv dataset file using Pandas library and display the following content of the dataset.



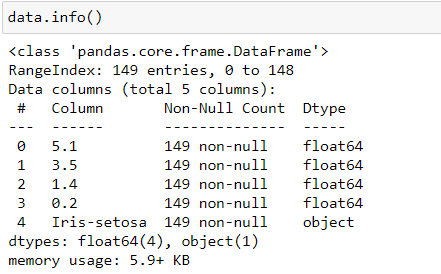
1. **Develop a program that reads a .csv dataset file using Pandas library and display the following content of the dataset.**
2. First five rows of the dataset



1. Complete data of the dataset

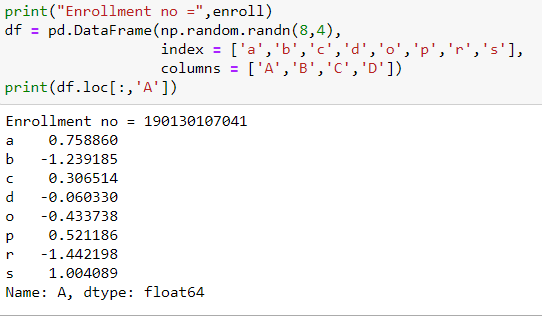


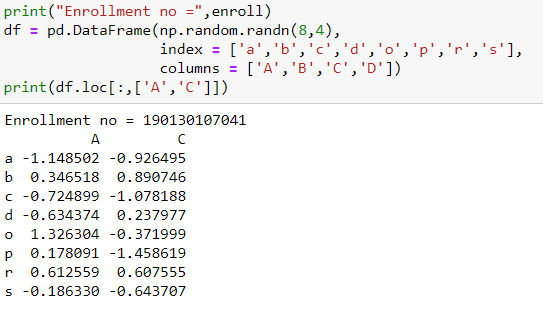
1. Summary or metadata of the dataset

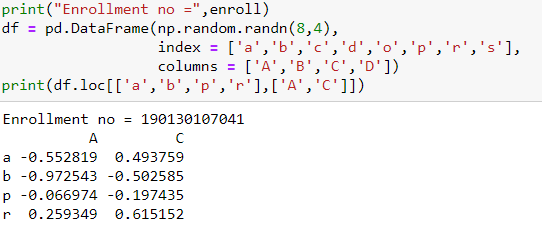


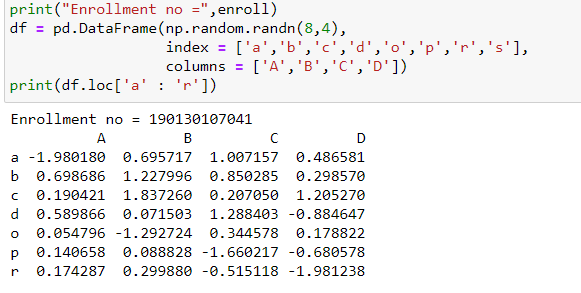
1. **Develop a program that shows application of slicing and dicing over the raws and columns of the dataset.**

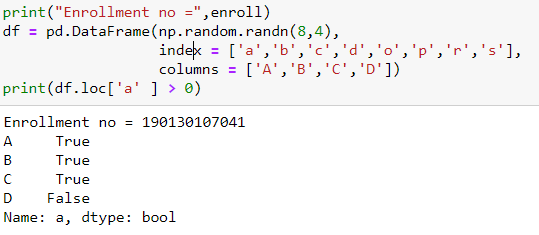
C:\Users\EMPEROR\Desktop\pds_ss\ss_1.JPG

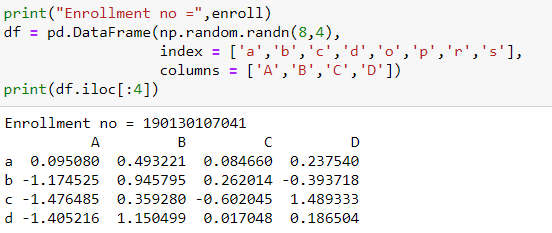


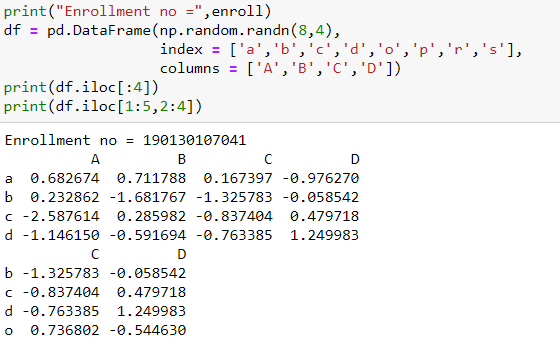




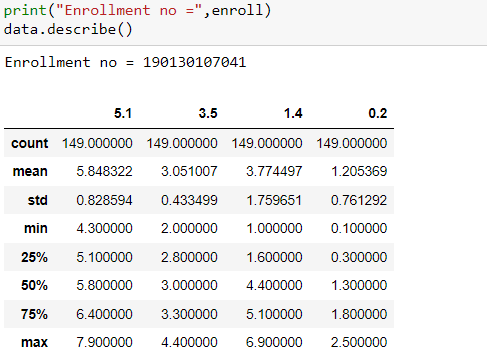




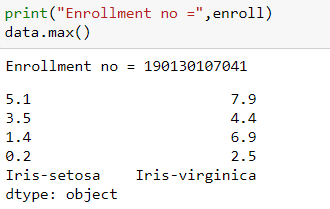




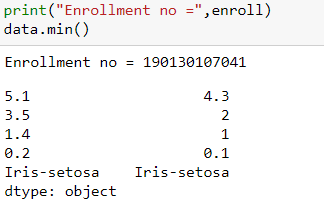
1. **Develop a program that shows usage of aggregate function over the input dataset.**
2. **describe**



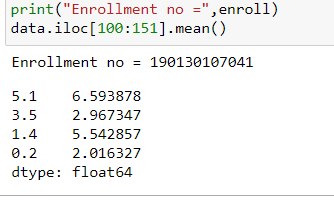
1. **max**



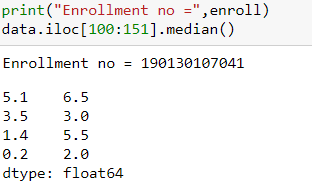
1. **min**



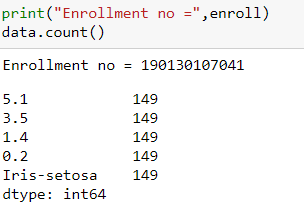
1. **mean**



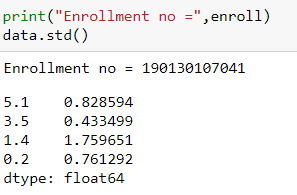
1. **median**



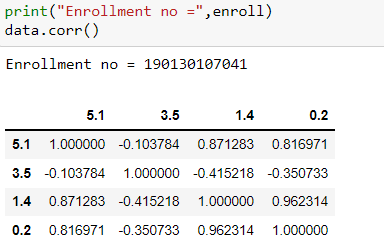
1. **count**



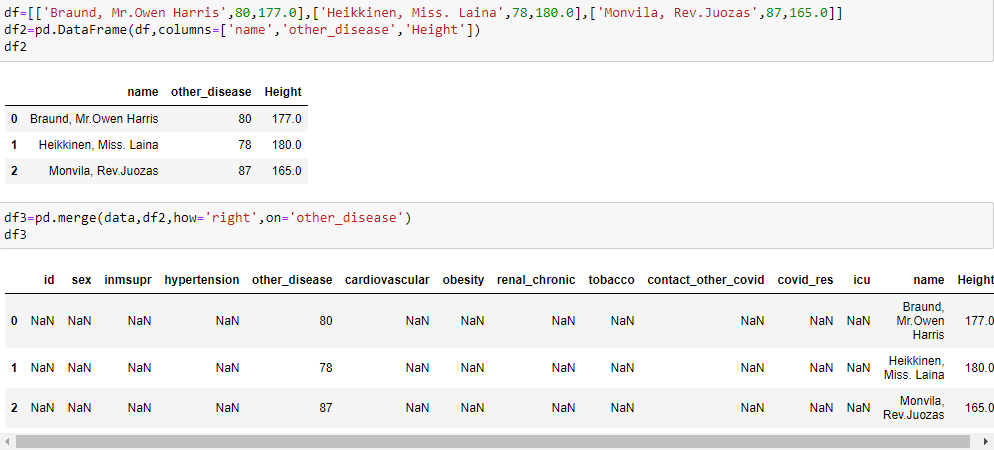
1. **std**



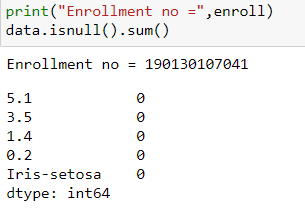
1. **corr**



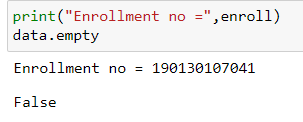
1. **Develop a program that applies split and merge operations on the datasets.**



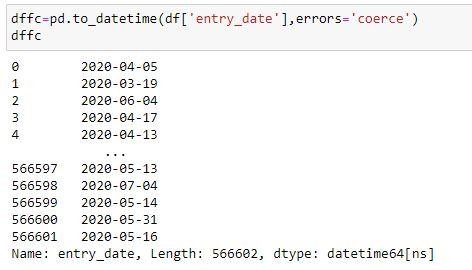
1. **Develop a program that shows the various data cleaning tasks over the dataset.**
2. **Identifying the null values**



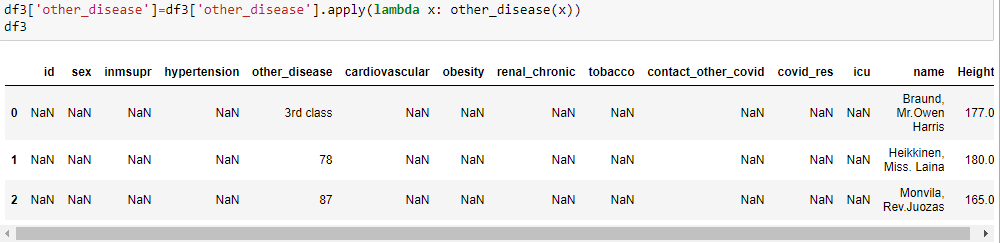
1. **Identifying the empty values**



1. **Identifying the incorrect timestamp**



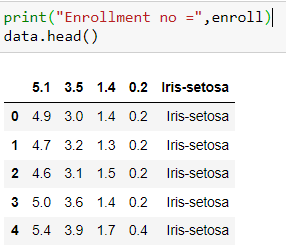
1. **Develop a program that shows an application of a Lamda function over the dataset.**



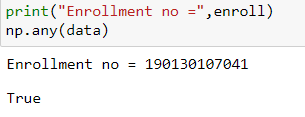
**Numpy practicals:**

1. **Develop a programme that shows usage of following NumPy array operations:**

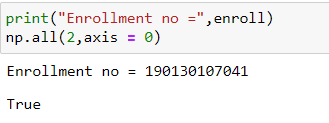
E:\Vishwa\Sem 5\Submission pds\s1.PNG



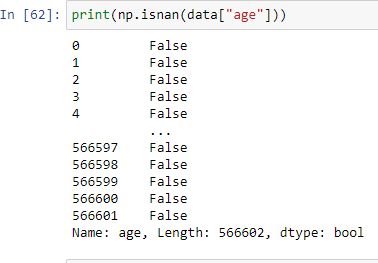
* 1. **any()**



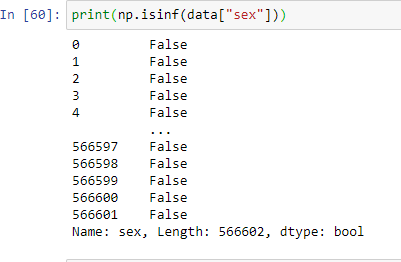
* 1. **all()**



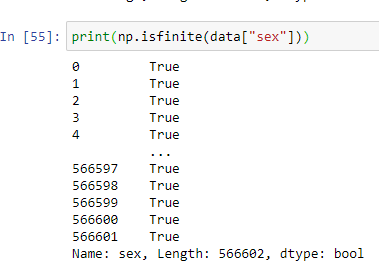
* 1. **isnan()**



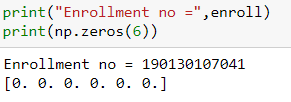
* 1. **isinf()**



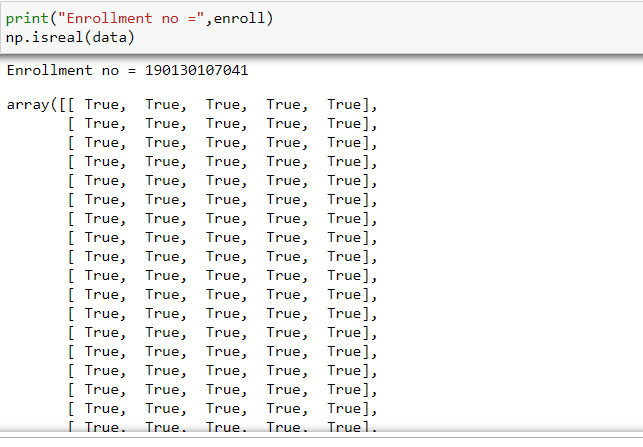
* 1. **isfinite()**

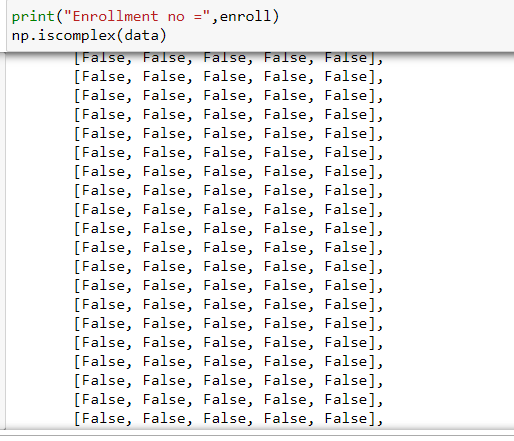


* 1. **zeros()**



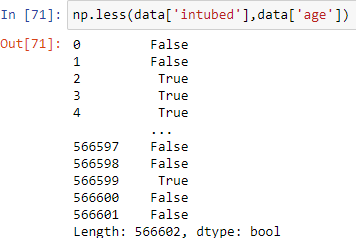
* 1. **isreal()**



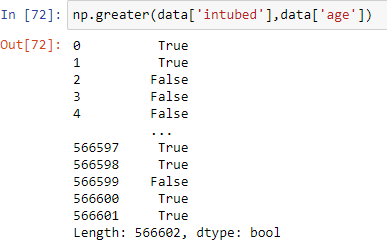
* 1. **iscomplex()**
  2. 
  3. **isscaler()**



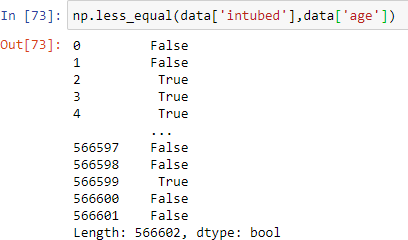
* 1. **less()**



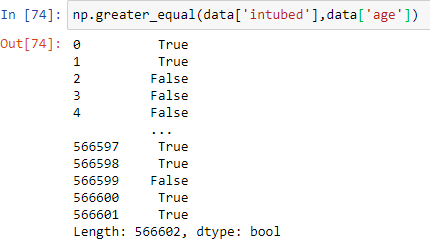
* 1. **greater()**



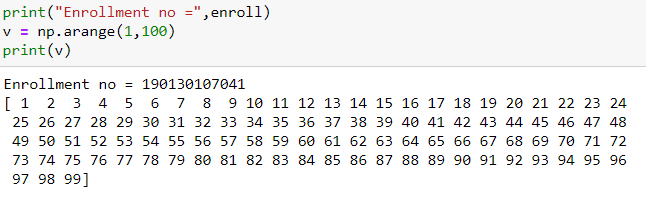
* 1. **less\_equal()**



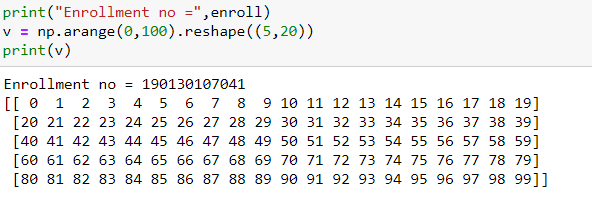
* 1. **greater\_equal()**



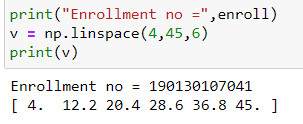
1. **Develop a program that shows usage of following NumPy library vector functions.**
2. **arrange()**



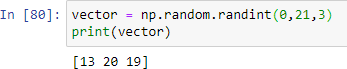
1. **reshape()**

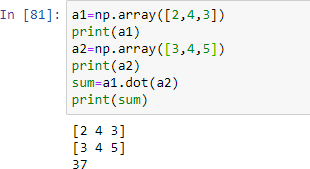
a

1. **linespace()**



1. **randint()**

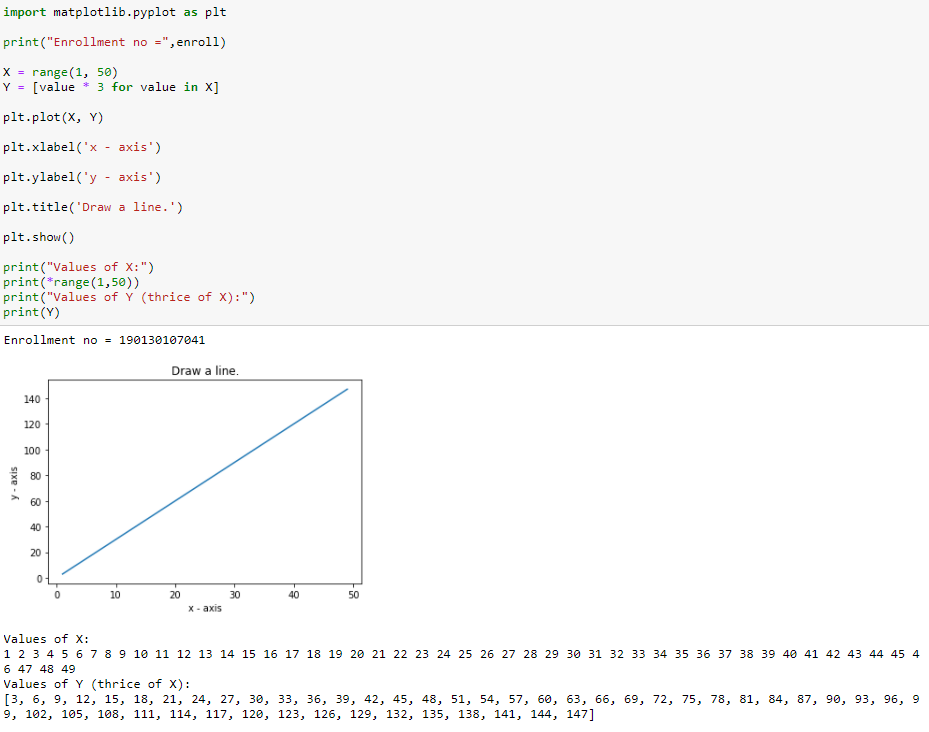


1. **dot()**

**Matplotlib library practicals:**

1. **Write a program to display below plot using matplotlib library**

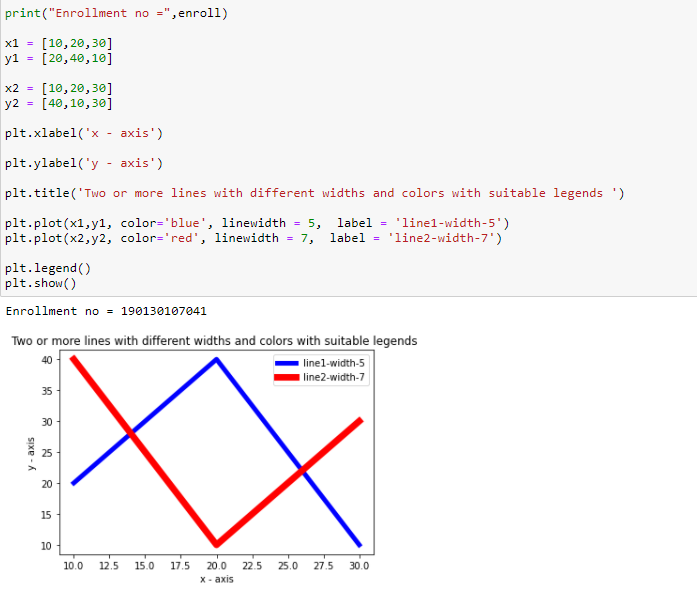
**For Values of X:[1,2,3,...,49], Values of Y (thrice of X):[3,6,9,12,...,144,147]**



1. **Write a program to display below plot using matplotlib library for the point values**

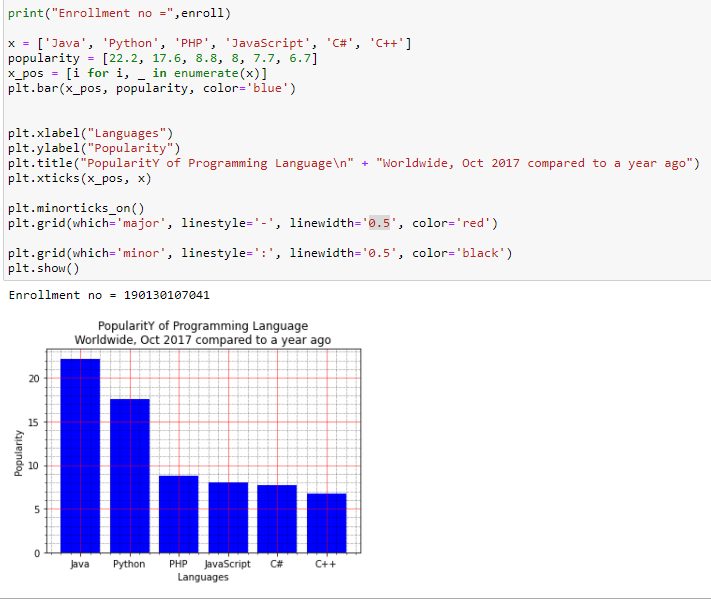
**x1 = [10,20,30], y1 = [20,40,10]**

**x2 = [10,20,30], y2 = [40,10,30]**



1. **Write a program to display below plot using matplotlib library for the point values**

**Languages =[‘Java’, ’Python’, ’PHP’, ‘JavaScript’, ‘C#’, ‘C++’] popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]**



1. **Write a program to display below plot using matplotlib library for the point values**

languages = [‘Java’, ’Python’, ’PHP’, ’JavaScript’, ’C#’, ’C++’] popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

colours = [#1f77b4, #ff7f0e, #2ca02c, #d62728, #9467bd, #8c564b]

**input : -**

print("Enrollment no =",enroll)

languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']

popuratity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd", "#8c564b"]

explode = (0.1, 0, 0, 0,0,0)

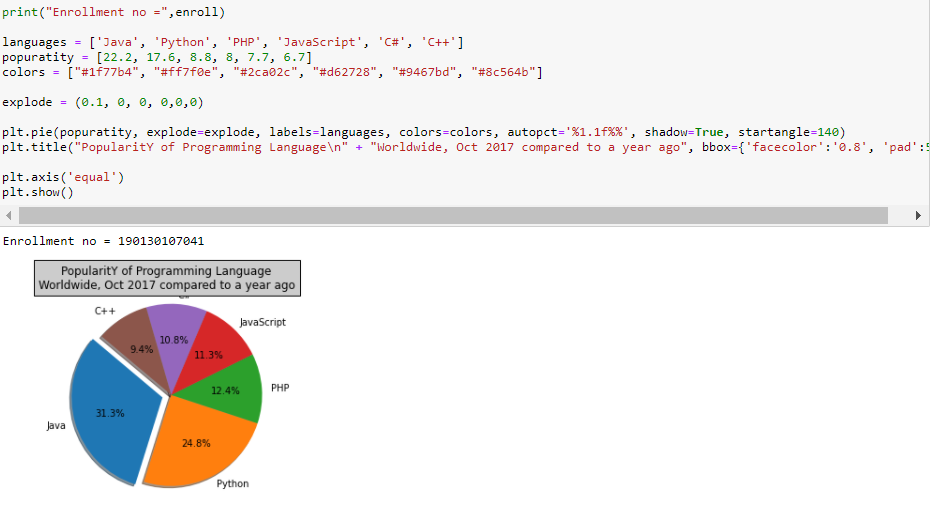
plt.pie(popuratity, explode=explode, labels=languages, colors=colors, autopct='%1.1f%%', shadow=True, startangle=140)

plt.title("PopularitY of Programming Language\n" + "Worldwide, Oct 2017 compared to a year ago", bbox={'facecolor':'0.8', 'pad':5})

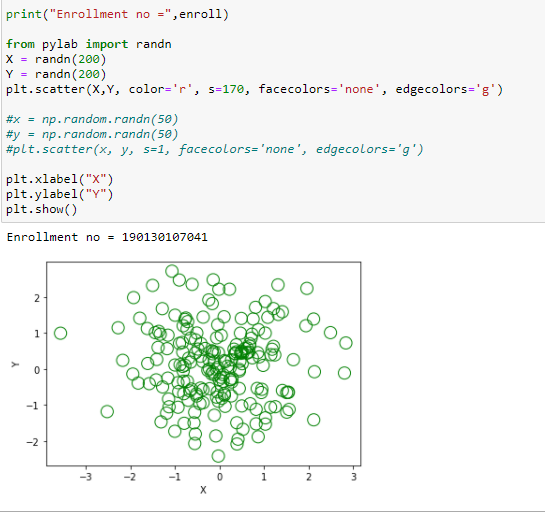
plt.axis('equal')

plt.show()

**output : -**



1. **Write a program to display below bar plot using matplotlib library For 200 random points for both X and Y display scatter plot:**



1. **Develop a program that reads .csv file from the url:**

**(https://github.com/chris1610/pbpython/blob/master/data/sample- salesv3.xlsx?raw=true) and plot the data of the dataset stored in the .csv file.**

**Input :-"**

print("Enrollment no =",enroll)

languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']

popuratity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd", "#8c564b"]

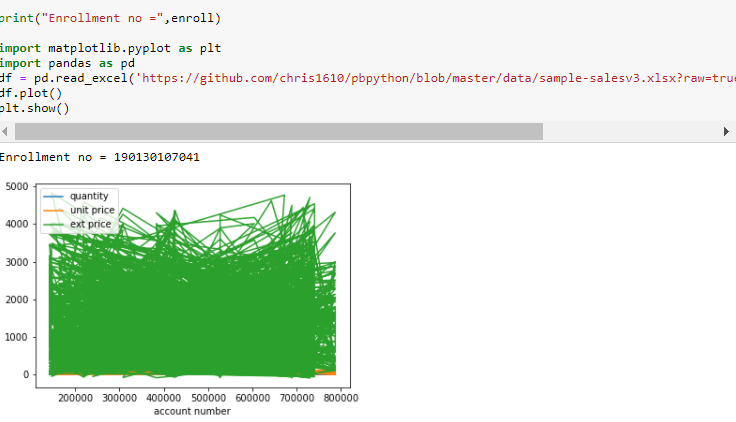
explode = (0.1, 0, 0, 0,0,0)

plt.pie(popuratity, explode=explode, labels=languages, colors=colors, autopct='%1.1f%%', shadow=True, startangle=140)

plt.title("PopularitY of Programming Language\n" + "Worldwide, Oct 2017 compared to a year ago", bbox={'facecolor':'0.8', 'pad':5})

plt.axis('equal')

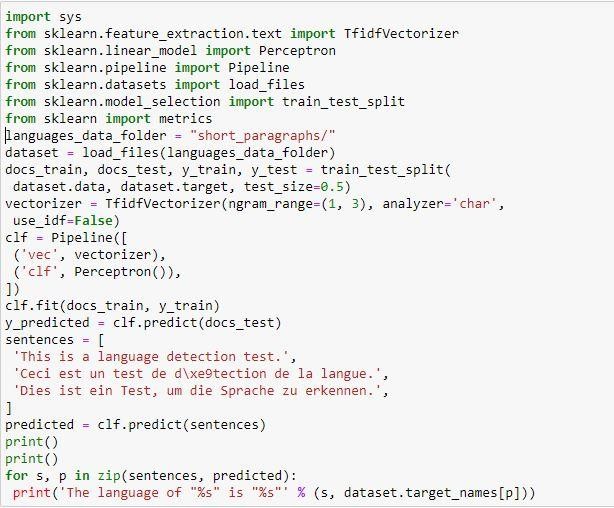
plt.show()

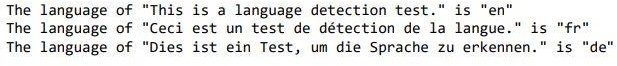
**roll**

**Scikit Learn practicals:**

1. **Exercise 1: Language identification**

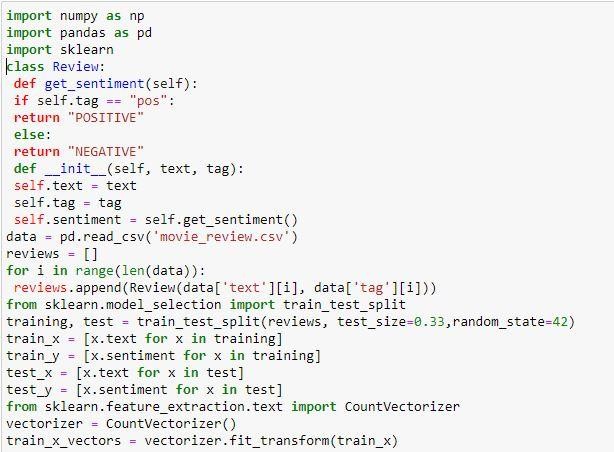
* **Write a text classification pipeline using a custom preprocessor and CharNGramAnalyzer using data from Wikipedia articles as a training set.**
* **Evaluate the performance on some held out test sets.**

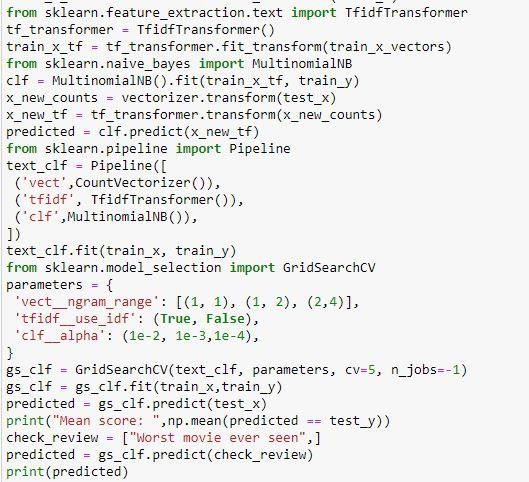




1. **Exercise 2: Sentiment Analysis on movie reviews**

* **Write a text classification pipeline to classify movie reviews as either positive or negative.**
* **Find a good set of parameters using grid search.**
* **Evaluate the performance on a held out test set.**







1. **CLI text classification utility**

**Using the results of the previous exercises and the cPickle module of the standard library, write a command line utility that detects the language of some text provided on stdin and estimates the polarity (positive or negative) if the text is**

**written in English.**

**Bonus point if the utility is able to give a confidence level for its predictions.**

