

**MINOR PROJECT REPORT**  
**SUBMITTED IN PARTIAL FULFILLMENT OF THE DEGREE**  
**OF**  
**BTECH**

**BY**

**ADRIJA CHAKRABORTY (14200322020)**

**SECOND YEAR STUDENT OF**  
**MEGHANAD SAHA INSTITUTE OF TECHNOLOGY**



**Under the supervision of:**

**Mr. Mainak Deb**  
**Sikharthy Infotech Pvt. Ltd.**

## **Department of Electronics And Communication**

Date: 10.07.23

I hereby forward the documentation prepared by me **Adrija Chakraborty** under the supervision of Mr. Mainak Deb Sir entitled **Python SRS Documentation** accepted as fulfilment of the requirement for the Degree of Bachelor of Technology (BTech) in **Electronics And Communication** from **Meghnad Saha Institute of Technology** affiliated to **Maulana Abul Kalam Azad University of Technology (MAKAUT)**.

---

**Mr. Mainak Deb**

**(Senior Software Engineer &  
Project Manager)**

**Project Guide**

**Sikharthy Infotech Pvt. Ltd.**

---

**Adrija Chakraborty**

**Department of Electronics And  
Communication**

**Meghnad Saha Institute of  
Technology**

# **Python SRS Documentation**

By

**Adrija Chakraborty (14200322020)**

UNDER THE GUIDANCE OF

**Mr. Mainak Deb**

**Project Guide**

**Sikharthy Infotech Pvt. Ltd.**

THESIS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

**B.Tech**

IN

ELECTRONICS AND COMMUNICATIONS

**MEGHNAD SAHA INSTITUTE OF TECHNOLOGY**

**AFFILIATED TO**

**Maulana Abul Kalam Azad University of Technology**

## **Certificate of Approval**

The foregoing project is hereby approved as a creditable study for the BTech in Electronics And Communication and presented in a manner of satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approved any statement made, opinion express or conclusion therein but approve this project only for the purpose for which it is submitted.

Final Examination for  
Evaluation of the Project

-----

-----

Signatures of Examiners

## **ABSTRACT**

The purpose of the project entitled as “Python SRS Documentation” is to develop a Documentation on the 5 day course that was undertaken by the aforementioned professor and institute. It deals with the basics of Python , backend development using MYSQL Database.

Python Data Science Libraries and data analysis,sorting and visually plotting. This project has been uploaded to github and the link too the repository has been shared in this documentation.

## **ACKNOWLEDGEMENT**

It is a great pleasure for me to acknowledge the assistance and participation of a large number of individuals to this attempt. Our project report has been structured under the valued suggestion, support and guidance of **Mr. Mainak Deb**. Under his guidance we have accomplished the challenging task in a very short time.

Finally, we express our sincere thankfulness to our family members for inspiring me all throughout and always encouraging us.

**Adrija Chakraborty**

**Department of Electronics And Communication**

# **TABLE OF CONTENTS**

## **Chapter 1: Introduction**

1: Introduction

PHASE A:

BRIEF DESCRIPTION OF THE TOPICS AND MODULE LEARNT SO FAR

PHASE B:

PYTHON BACKEND USING MYSQL DATABASE

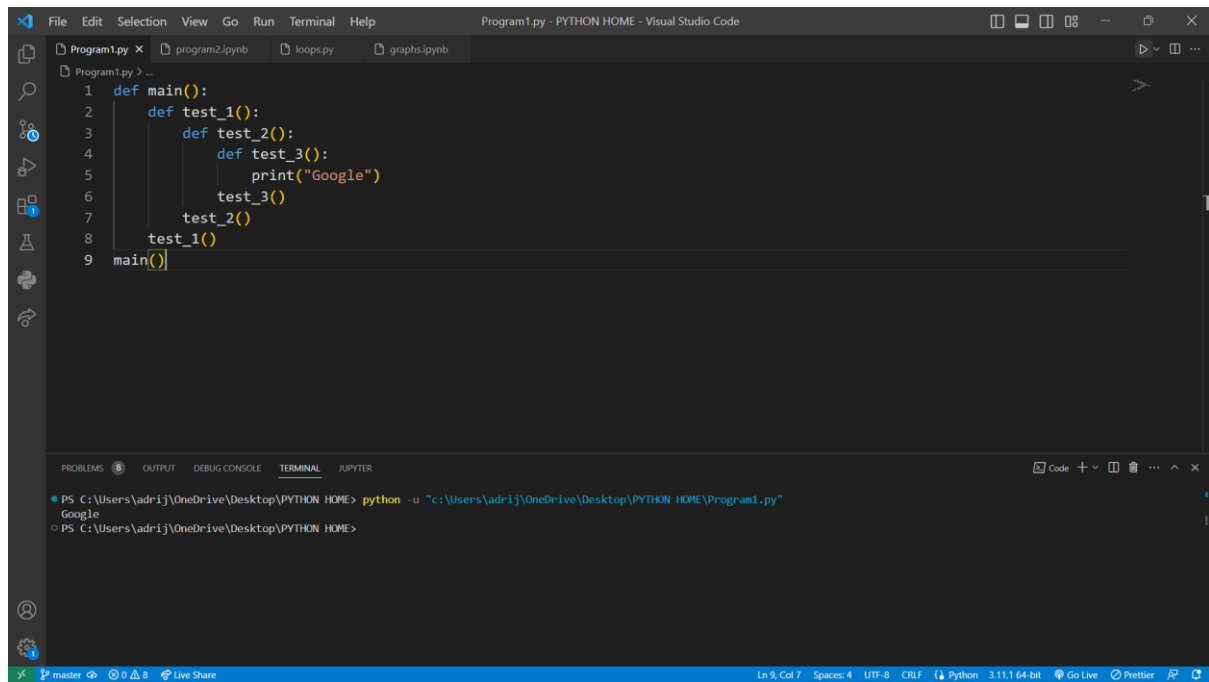
PYTHON DATASCIENCE LIBRARIES AND DATA ANALYSIS, SORTING AND  
VISUALLY PLOTTING

# INTRODUCTION

THIS IS A SRS DOCUMENTATION PROJECT ON THE PYTHON PROGRAMMING LANGUAGE. BRIEFLY ON THE MODULES AND TOPICS LEARNT SO FAR.KNOWLEDGE GAINED AND PRACTICAL IMPLEMENTATIONS FOR FUTURE PROSPECTUS.PYTHON AS WE ALL KNOW IS A HIGHLY DEVELOPED AND HIGH LEVEL PROGRAMMING LANGUAGE FOR DEVELOPERS. IT IS AN OBJECT ORIENTED LANGUAGE USEFUL IN MANY WAYS.

## INHERITANCE

WE USED PYTHON IN THIS AS THE DEVELOPING PROGRAMMING LANGUAGE. FOLLOWING IS THE EXAMPLE OF INHERITANCE AS TAUGHT BY OUR RESPECTED PROFESSOR.



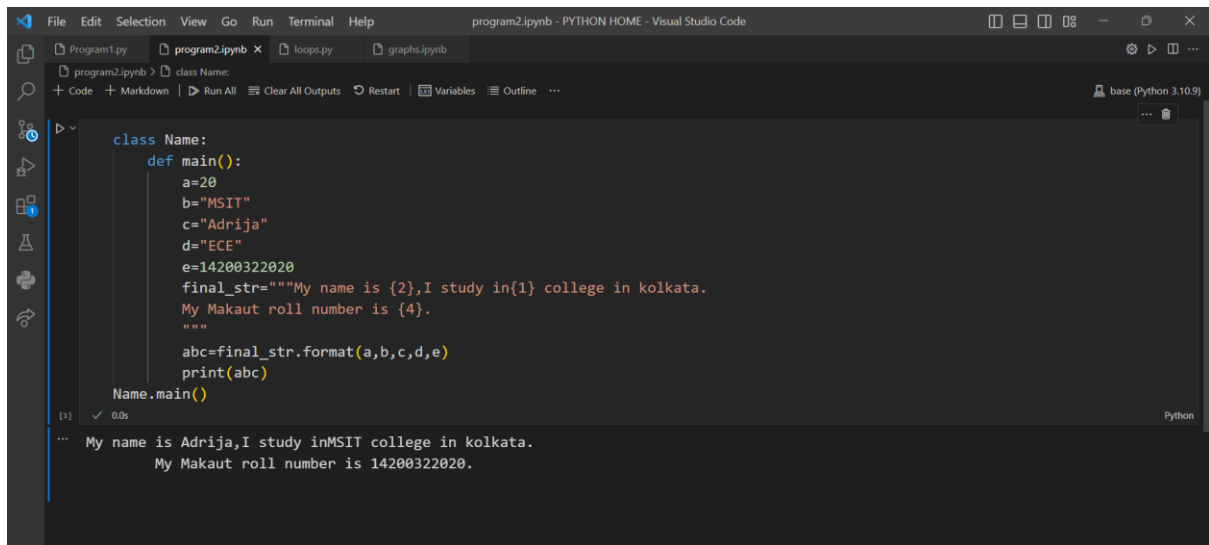
```
1 def main():
2     def test_1():
3         def test_2():
4             def test_3():
5                 print("Google")
6                 test_3()
7             test_2()
8         test_1()
9     main()
```

The screenshot shows a Visual Studio Code window titled 'Program1.py - PYTHON HOME - Visual Studio Code'. The editor displays a Python script with nested function definitions: `main()` calls `test_1()`, which calls `test_2()`, which calls `test_3()`, which prints 'Google'. Below the editor, the 'TERMINAL' panel shows the command `python -u "c:\Users\adrij\OneDrive\Desktop\PYTHON HOME\Program1.py"` being executed, resulting in the output 'Google'.

## PROGRAM 2

THE FOLLOWING IS THE SECOND PROGRAM WRITTEN IN PYTHON. THE KERNEL SELECTED IS PYTHON.BASE AND THE CODE SNIPPET IS TYPED IN A JUPYTER->. ipynb FILE. IN JUPYTER WE CAN TYPE CODE SNIPPETS AS SHOWN BELOW.





```
class Name:
    def main():
        a=20
        b="MSIT"
        c="Adrija"
        d="ECE"
        e=14200322020
        final_str="My name is {2},I study in{1} college in kolkata.
        My Makaut roll number is {4}."
        """
        abc=final_str.format(a,b,c,d,e)
        print(abc)
Name.main()
```

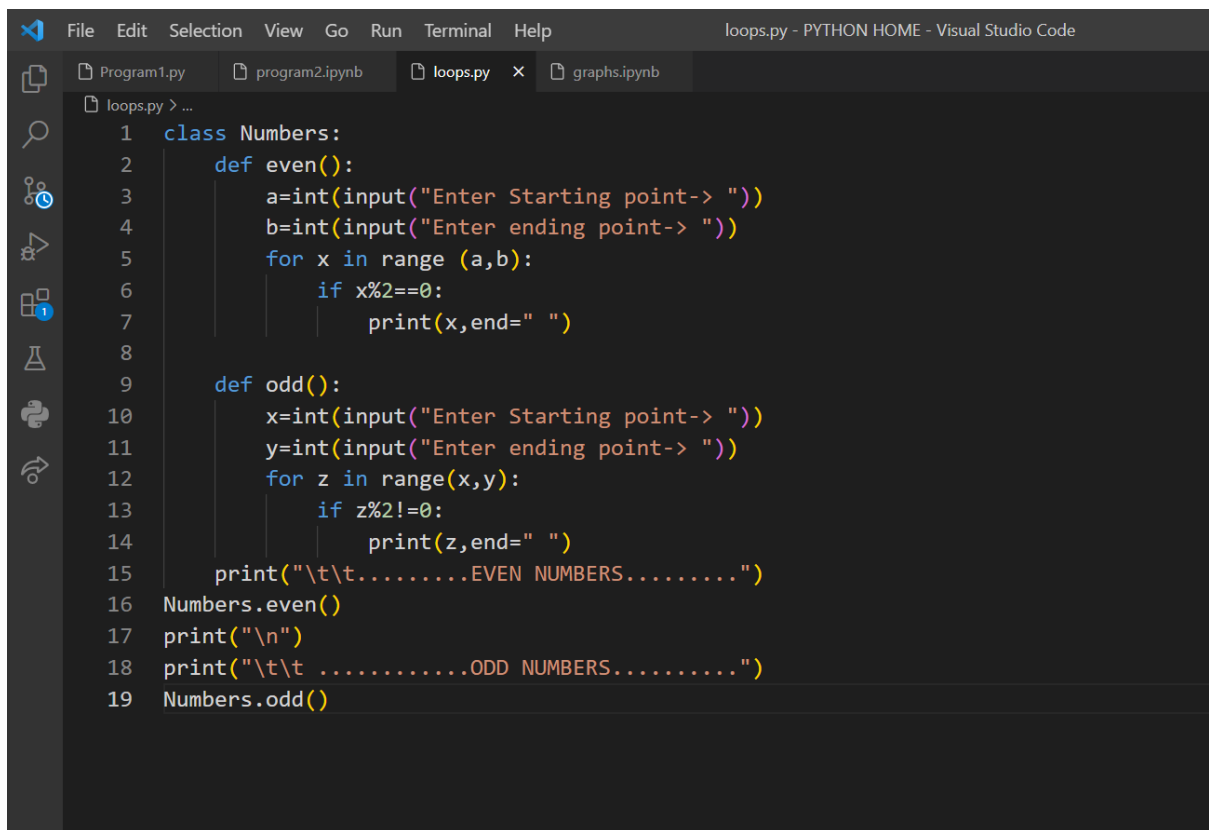
My name is Adrija,I study inMSIT college in kolkata.  
My Makaut roll number is 14200322020.

**THE OUTPUT IS THE FINAL STRING. BUT HEREIN, WE HAVEN'T DIRECTLY PRINTED THE STRING; WHEREAS WE HAVE USED REFERENCE TO DO THE SAME.**

## **LOOPS**

**THE FOLLOWING SHOWS THE LOOPS IN PYTHON AS IN HOW TO IMPLEMENT ITERATION OR LOOPS IN PYTHON. A VERY BASIC PROGRAM ON ODD AND EVEN NUMBERS HAS BEEN WRITTEN IN THE FOLLOWING EXAMPLE.**

**WE HAVE INITIALLY DEFINED THE CLASS AND OBJECT AND HAVE TAKEN INPT FROM THE USER ABOUT WHAT SHALL BE THE RANGE AND NUMBERS . AND THEN CORRESPONDING TO THAT THE NUMBER'S WHETHER ODD OR EVEN HAS BEEN MENTIONED IN THE TERMINAL AS OUTPUT..**



```
File Edit Selection View Go Run Terminal Help loops.py - PYTHON HOME - Visual Studio Code
Program1.py program2.ipynb loops.py x graphs.ipynb
loops.py > ...
1 class Numbers:
2     def even():
3         a=int(input("Enter Starting point-> "))
4         b=int(input("Enter ending point-> "))
5         for x in range (a,b):
6             if x%2==0:
7                 print(x,end=" ")
8
9     def odd():
10        x=int(input("Enter Starting point-> "))
11        y=int(input("Enter ending point-> "))
12        for z in range(x,y):
13            if z%2!=0:
14                print(z,end=" ")
15        print("\t\t.....EVEN NUMBERS.....")
16    Numbers.even()
17    print("\n")
18    print("\t\t .....ODD NUMBERS.....")
19    Numbers.odd()
```

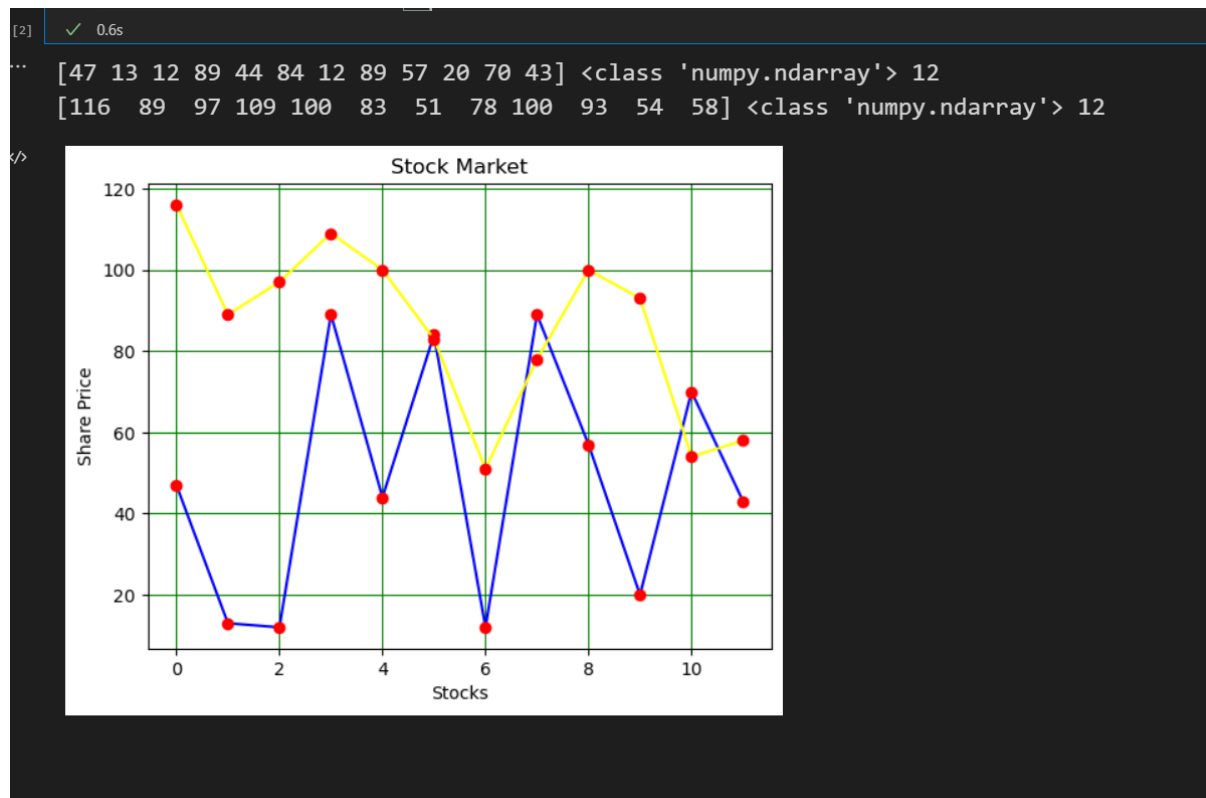
## GRAPH

```
import numpy as np
import matplotlib.pyplot as plt
class Graph:
    def RandomArrayGraph():
        x_axis=np.random.randint(10,90,12)
        y_axis=np.random.randint(50,120,12)
        # x_axis=np.array_split(x_axis,2)
        # y_axis=np.array_split(y_axis,2)
        print(x_axis,type(x_axis),len(x_axis))
        print(y_axis,type(y_axis),len(y_axis))
        #
plt.plot(x_axis,y_axis,marker="o",mfc="red",mec="red",
color="blue")
```

```

plt.plot(x_axis,marker="o",mfc="red",mec="red"
,color="blue")
plt.plot(y_axis,marker="o",mfc="red",mec="red"
,color="yellow")
plt.grid(color="green")
plt.xlabel("Stocks")
plt.ylabel("Share Price")
plt.title("Stock Market")
plt.show()
Graph.RandomArrayGraph()

```



ABOVE IS THE CODE AND OUT PUT OF HOW WE DEPLOY A GRAPH IN PYTHON IN JUPYTER NOTEBOOK. INITIALLY WE HAVE TAKEN A RANDOM INTEGER USING THE RANDINT FEATURE OF PYTHON. THE GRAPH HAS BEEN DRAWN WITH RESPECT TO THE RANDOM INTEGER AS SELECTED BY THE SYSTEM SINCE IT HAS BEEN DONE IN A VERY BASIC FORMAT FOR OUR BASIC UNDERSTANDNG.

## SORTING OF AN ARRAY

```
class Arrays:
    def main():
        arr_1=[]
        n=int(input("Enter the number of array elements -> "))
        for x in range(0,n):
            y=int(input("Enter the Array Element -> "))
            arr_1.append(y)
        print("\nFinal Array ->",arr_1," ",type(arr_1))
        print("Length of the Array is -> ",len(arr_1))
        arr_1.sort()
        print("\nThe Array is Increasing Order -> ",arr_1)
        arr_1.sort(reverse=True)
        print("\nThe Array is Decreasing Order -> ",arr_1)
Arrays.main()
```

1] ✓ 27.8s Python

ABOVE IS THE CODE FOR SORTING AN ARRAY.THE CLASS AND OBJECT HAS BEEN SPECIFIED OR DEFINED. WE HAVE TAKEN INPUT FROM THE USER AND LATER APPENDED THE SAME INTO THE ARRAY.

```
...
Final Array -> [20, 30] <class 'list'>
Length of the Array is -> 2

The Array is Increasing Order -> [20, 30]

The Array is Decreasing Order -> [30, 20]
```

+ Code + Markdown

ABOVE IS THE OUTPUT OF SORTING AN ARRAY IN JUPYTER NOTEBOOK AFTER CODING IN PYTHON.

## FILE HANDLING

```
import os
class FileHandling:
    def main():
        x=str(input("Enter a file name -> "))
        y=str(input("Enter a text in file -> "))
        b=open(x,"w")
        b.write(y)
        b.close()
        print(x,"\nFile created successfully")
        print("\n\t\t-----Reading File Contents-----")
        z=open(x,"r")
        print("\n")
        print(z.read())
    def FileDelete():
        xyz=input("Enter the file name to delete -> ")
        a=os.remove(xyz)
        print("File deleted successfully")
FileHandling.main()
FileHandling.FileDelete()
```

THE ABOVE IS THE CODE OF FILE HANDLING . FILE HANDLING IS AS WE ALL KNOW A VERY IMPORTANT PART OF ANY PROGRAMMING LANGUAGE.  
USING THE ABOVE CODE WE CAN OPEN, READ ,,WRITE AND DELETE A FILE(IN THIS CASE A SAMPLE FILE HAS BEEN CREATED.

## LIBRARIES IN PYTHON

**PYTHON AS WE ALL KNOW IS A HIGHLY DEVELOPED ANND CODER FRIENDLY LANGUAGE WITH ALMOST ALL FUNCTIONS/METHODS BEING PREDEFINED AND AVAILABLE TO US.**

**THERE ARE SEVERAL LIBRARIES IN PYTHON. SOME OF THEM AR NUMPY,SCIPY,PANDAS, AND ALL.WE CAN INSTALL ANACONDA AT ONE GO TO AVOID ANY MESSY INSTALLATIONS OF ONE LIBRARY AT A TIME.**

## DATABASE HANDLING

```
#Create Database  
import mysql.connector
```

```
mydb = mysql.connector.connect(  
    host="localhost",  
    user="root",  
    password=""  
)  
  
abc = mydb.cursor()  
  
abc.execute("CREATE DATABASE msit")  
print("Database Created Successfully")
```

In [2]:

```
#Create Table  
import mysql.connector
```

```
db = mysql.connector.connect(  
    host="localhost",  
    user="root",  
    password="",  
    database="msit"  
)  
  
mysql_query = db.cursor()  
  
mysql_query.execute("CREATE TABLE customers_students(id INT AUTO_INCREMENT  
PRIMARY KEY,name VARCHAR(255), address VARCHAR(255), email VARCHAR(100),  
gender TEXT(50))")  
print("Table Created Successfully")
```

```
#Data Insert in Table  
import mysql.connector
```

```
mydb = mysql.connector.connect(  
    host="localhost",  
    user="root",  
    password="",  
    database="msit"  
)  
  
mycursor = mydb.cursor()  
  
sql = "INSERT INTO customers (name, address) VALUES (%s, %s)"
```

```

val = ("Mainak", "Kolkata")
mycursor.execute(sql, val)

sql_2= "INSERT INTO customers_students (name, address,email,gender) VALUES
(%s, %s,%s, %s)"
val_2 = ("Test1", "addressTest","test@gmail.com","Male")
mycursor.execute(sql_2, val_2)

mydb.commit()

print(mycursor.rowcount, "record inserted.")

#Data Insert in Table
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="msit"
)

mycursor = mydb.cursor()

sql = "INSERT INTO customers (name, address) VALUES (%s, %s)"
a=str(input("Enter your name -> "))
b=str(input("Enter your address -> "))
val = (a, b)
mycursor.execute(sql, val)

sql_2= "INSERT INTO customers_students (name, address,email,gender) VALUES
(%s, %s,%s, %s)"
a=str(input("Enter your name ->"))
b=str(input("Enter your address -> "))
c=str(input("Enter your email"))
d=str(input("Enter your gender -> "))
val_2=(a,b,c,d)
mycursor.execute(sql_2, val_2)
mydb.commit()
print(mycursor.rowcount, "record inserted.")

#fetch data
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="msit"
)

mycursor = mydb.cursor()

mycursor.execute("SELECT * FROM customers_students")

myresult = mycursor.fetchall()

```

```

for x in myresult:
    print(x)

#Search Data
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="msit"
)

mycursor = mydb.cursor()

sql = "SELECT * FROM customers_students WHERE email = 'test@gmail.com'"

mycursor.execute(sql)

myresult = mycursor.fetchall()

for x in myresult:
    print(x)

```

## PANDAS

```

In [4]: import pandas as pd
import numpy as np
class Test:
    def main():
        abc={
            "Name":["Soham","Indrasish","tanmoy","kaustav","aqib"],
            "Roll-Number":[1,2,3,4,5]
        }
        xyz=pd.DataFrame(abc)
        print(xyz)
Test.main()

```

```

In [2]: import pandas as pd
print(pd.__version__)

```

1.5.3

**PANDAS IS A LIBRARY USED IN PYTHON FOR DEVELOPERS. NUMPY IS USED FOR ARRAYS.**

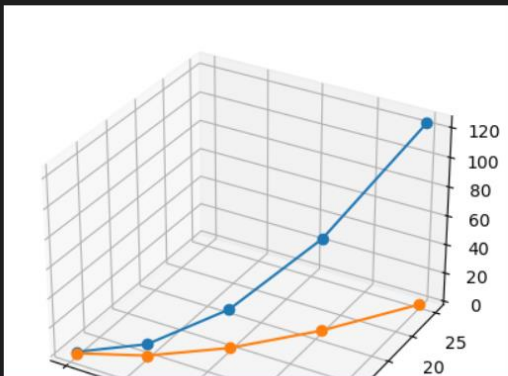


## MATPLOTLIB

**MATPLOTLIB IS A LIBRARY USED FOR PLOTTING VISUALLY EFFECTIVE GRAPHS REPRESENTING DATA.FOLLOWING IS SUCH AN EXAMPLE.**

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]
z = [1, 8, 27, 64, 125]
# Creating the figure object
fig = plt.figure()
# keeping the projection = 3d
# creates the 3d plot
ax = plt.axes(projection = '3d')
ax.plot3D(x,y,z,marker="o")
ax.plot3D(x,y,marker="o")
plt.rcParams['grid.color'] = "grey"
# plt.legend()
```

✓ 1.4s



Ln 13, Col 15 CRLF Cell 1 of 1 Go Live

**WE IMPORT MATPLOTLIB AS PLT.AND THUS THE CODE DEPLOYS A 3D GRAPH.**

## SUBPLOT

```
import matplotlib.pyplot as plt
import numpy as np
class Subplot:
    def Plot():
        x_axis_1=np.random.randint(10,50,12)
        y_axis_1=np.random.randint(50,100,12)
        plt.subplot(2,2,1)
        plt.plot(x_axis_1,y_axis_1,marker="o",mfc="red")

        x_axis_2=np.random.randint(10,50,12)
        y_axis_2=np.random.randint(50,100,12)
        plt.subplot(2,2,2)
        plt.scatter(x_axis_2,y_axis_2)

        x_axis_3=np.random.randint(10,50,12)
        y_axis_3=np.random.randint(50,100,12)
        plt.subplot(2,2,3)
        plt.plot(x_axis_3,y_axis_3,marker="o",mfc="red")

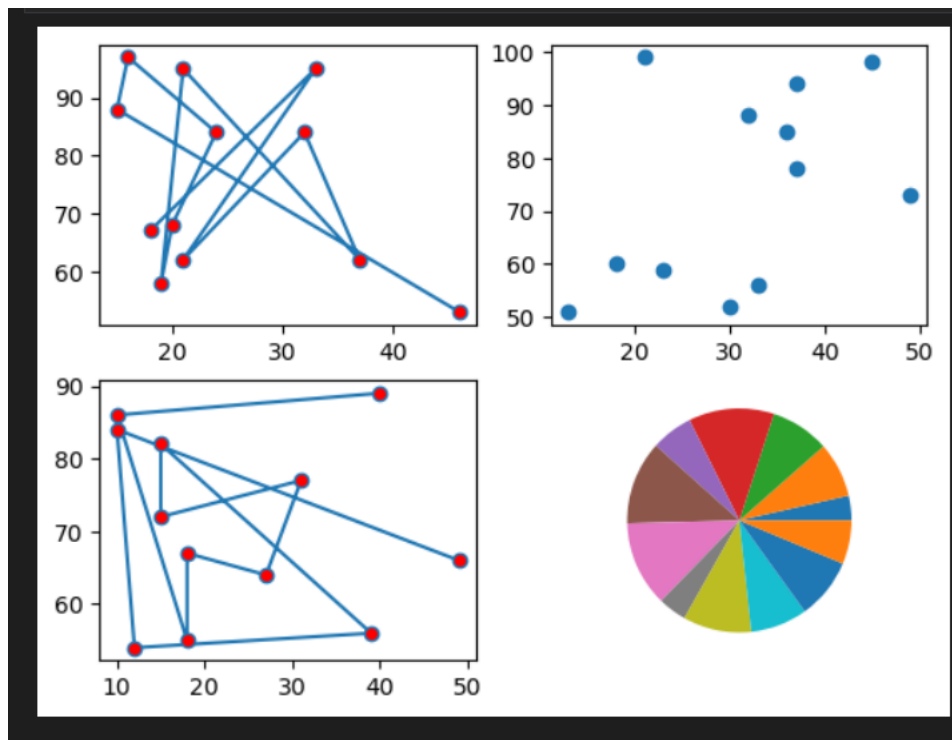
        x_axis_4=np.random.randint(10,50,12)
        plt.subplot(2,2,4)
        plt.pie(x_axis_4)

        plt.show()
Subplot.Plot()
```

The **subplot()** function takes three arguments that describes the layout of the figure.

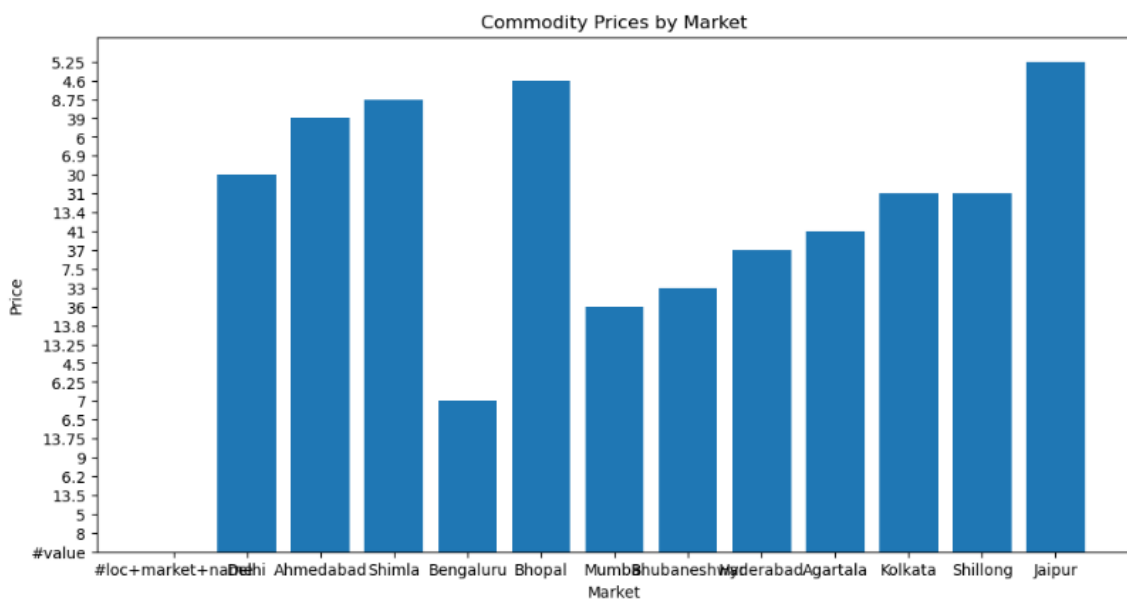
The layout is organized in rows and columns, which are represented by the *first* and *second* argument.

The third argument represents the index of the current plot.



**FOLLOWING IS THE VISUAL PLOTING OF THE BELOW MENNTIONED DATASHEET**

[wfp food\\_prices.xlsx](#)



## SORTING OF THE DATASHEET

```
In [14]: import pandas as pd
import matplotlib.pyplot as plt

In [8]: data = pd.read_csv('C:\\Users\\adrij\\OneDrive\\Desktop\\wfp_food_prices.csv')

In [10]: print(data.head())
```

	date	admin1	market	commodity	unit	priceflag
0	#date	#admin1+name	#loc+market+name	#item+name	#item+unit	
1	1994-01-15	Delhi	Delhi	Rice	KG	
2	1994-01-15	Delhi	Delhi	Wheat	KG	
3	1994-01-15	Delhi	Delhi	Sugar	KG	
4	1994-01-15	Gujarat	Ahmedabad	Wheat	KG	

```
priceflag  currency  price
0 #item+price+flag #currency #value
1          actual    INR      8
2          actual    INR      5
3          actual    INR    13.5
4          actual    INR      6.2

In [11]: print(data.tail())
```

	date	admin1	market	commodity	unit	priceflag
39	1994-02-15	Meghalaya	Shillong	Rice	KG	actual
40	1994-02-15	Meghalaya	Shillong	Oil (mustard)	KG	actual
41	1994-02-15	Orissa	Bhubaneshwar	Rice	KG	actual
42	1994-02-15	Orissa	Bhubaneshwar	Wheat	KG	actual
43	1994-02-15	Orissa	Bhubaneshwar	Wheat	KG	actual

jupyter Untitled1 Last Checkpoint: 32 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (pykernel)

```
date admin1 market commodity unit priceflag \
39 1994-02-15 Meghalaya Shillong Rice KG actual
40 1994-02-15 Meghalaya Shillong Oil (mustard) KG actual
41 1994-02-15 Orissa Bhubaneshwar Rice KG actual
42 1994-02-15 Orissa Bhubaneshwar Wheat KG actual
43 1994-02-15 Rajasthan Jaipur Wheat KG actual

currency price
39 INR 8
40 INR 31
41 INR 5
42 INR 5
43 INR 5.25

In [18]: df = pd.DataFrame(data)

# Create a bar plot
plt.figure(figsize=(12, 6))
plt.bar(df['market'], df['price'])
plt.xlabel('Market')
plt.ylabel('Price')
plt.title('Commodity Prices by Market')
plt.show()
```

Market	Price
Rajasthan	5.25
Orissa	5
Meghalaya	31
Shillong	8

THE TAIL AND HEAD OF THE EXCEL FILE HAS BEEN DISPLAYED IN THE JUPYTER NOTEBOOK

## **CONCLUSION**

**THIS IS A SRS DOCUMENTATION PROJECT ON THE PYTHON PROGRAMMING LANGUAGE. BRIEFLY ON THE MODULES AND TOPICS LEARNT SO FAR.KNOWLEDGE GAINED AND PRACTICAL IMPLEMENTATIONS FOR FUTURE PROSPECTUS.PYTHON AS WE ALL KNOW IS A HIGHLY DEVELOPED AND HIGH LEVEL PROGRAMMING LANGUAGE FOR DEVELOPERS. IT IS AN OBJECT ORIENTED LANGUAGE USEFUL IN MANY WAYS. THIS PROJECT WAS A WONDERFUL LEARNING EXPERIENCE.**

**GITHUB LINK : <https://github.com/adrija-12>**