

Subject : Python

**Shree M.M.Ghodasara Mahila College**  
**Junagadh**

---

**Computer Science Department**  
**B.C.A.**  
**Lab Manual**

**PYTHON**



**Shree M.M.Ghodasara Mahila College**

Opp. Motibaug, Junagadh - 362 001

Ph. (0285) - 2670523, 2671523

## INDEX

No.	DEFINITION	Page No.
<b>CH – 1 Introduction to Python</b>		
<b>1</b>	Write a program in Python to input principle amount, rate and no. of year and find Simple Interest.	<b>9</b>
<b>2</b>	Write a program in Python to input Three Subjects Name and Marks and Calculate Total and Percentage, if Percentage is more than 70 then print appropriate message.	<b>10</b>
<b>3</b>	Write a program in Python to input Two Number from user and check which number is Maximum	<b>11</b>
<b>4</b>	Calculate Python strings.	<b>12</b>
<b>5</b>	Calculate Python list.	<b>13</b>
<b>6</b>	Calculate Python tuple.	<b>14</b>
<b>7</b>	Calculate Python dictionary.	<b>15</b>
<b>8</b>	Write a program in Python to check enter number is positive, negative, zero or not a number.	<b>16</b>
<b>9</b>	Write a program in Python to check input number is prime or not.	<b>17</b>
<b>10</b>	Write a program to input any character from keyboard and check input character is vowel or not.	<b>18</b>
<b>11</b>	Write a program in Python to print square of all number present in list.	<b>19</b>
<b>12</b>	Write a program in Python to print sum of first five natural number.	<b>20</b>
<b>13</b>	Write a program to print following output. 0,10 0,11 0,12 0,13 1,10 1,11 1,12 1,13 2,10 2,11 2,12 2,13	<b>21</b>
<b>14</b>	Write a program in Python to print following series. 1 4 7	<b>22</b>
<b>15</b>	Write a program in Python to display following output. 1,5 2,6 3,7	<b>23</b>
<b>16</b>	Write a program in Python to display following output. 10	<b>24</b>

## Subject : Python

	9 8 7 Loop is finished.	
<b>17</b>	Write a program in Python to display all elements before number 88.	<b>25</b>
<b>18</b>	Write a program in Python to input character and check input character is <b>Alphabet or not.</b>	<b>26</b>
<b>19</b>	Write a program to find factorial number.	<b>27</b>
<b>20</b>	Create an iterator that returns numbers starting with 1 and each sequence will increase by one.( returning 1, 2, 3, 4, 5 etc.)	<b>28</b>
<b>21</b>	Write a program to stop after 10 iteration.	<b>29</b>
<b>22</b>	Write a program in Python to open text file ( .txt ) and read it.	<b>30</b>
<b>23</b>	Write a program in Python to write text file ( .txt ) and read only 20 character from it.	<b>31</b>
<b>24</b>	Write a program in Python to write position into a file.	<b>32</b>

### CH – 2 OOP using Python

<b>1</b>	Write a program in Python for Exception Handling.	<b>33</b>
<b>2</b>	Write a program in Python for Nested Exception with finally block.	<b>34</b>
<b>3</b>	Write a program in Python Exception with argument.	<b>35</b>
<b>4</b>	Write a program in Python for Assertion.	<b>36</b>
<b>5</b>	Write a program in Python to destroy class object.	<b>37</b>
<b>6</b>	Write a program in Python to create employee class and count it.	<b>38</b>
<b>7</b>	Write a program in Python to inheritance.	<b>39</b>
<b>8</b>	Write a program in Python to encapsulation and information hiding.	<b>40</b>
<b>9</b>	Write a program in Python to linear search.	<b>41</b>
<b>10</b>	Write a program in Python to interpolation.	<b>42</b>
<b>11</b>	Write a program in Python to insertion sort.	<b>43</b>
<b>12</b>	Write a program in Python to selection sort.	<b>44</b>
<b>13</b>	Write a program in Python to bubble sort.	<b>45</b>
<b>14</b>	Write a program to in Python to merge sort.	<b>46</b>
<b>15</b>	Write a program in Python for shell short.	<b>47</b>
<b>16</b>	Write a program in Python for operator overloading.	<b>48</b>
<b>17</b>	Write a program in Python for unary operator overloading.	<b>49</b>
<b>18</b>	Write a program in Python for Hash table example.	<b>50</b>

### CH – 3 Plotting using PyLab

<b>1</b>	Write a program in Python for plotting.	<b>51</b>
<b>2</b>	Write a program in Python for multiple figure.	<b>52</b>
<b>3</b>	Write a program in Python to calculate rate of interest and display figure	<b>53</b>

## Subject : Python

	plotting compound growth of investment.	
<b>4</b>	Write a program in to calculate principle amount and display growth of principle amount plotting with title and label.	<b>54</b>
<b>5</b>	Write a program in Python to change the type size and line width.	<b>55</b>
<b>6</b>	Write a program in Python to display following dashed color output.	<b>56</b>
<b>7</b>	Write a program in Python to create Bar Chart.	<b>57</b>
<b>8</b>	Write a program in Python to create Histogram chart.	<b>58</b>
<b>9</b>	Write a program in Python to create Scatter plot.	<b>59</b>
<b>10</b>	Write a program in Python to create Pie Chart.	<b>60</b>
<b>11</b>	Write a program in Python for plotting curves of given equation for <code>x=np.arange(0.2*(np.pi),0.1)</code> <code>y=np.sin(x)</code>	<b>61</b>
<b>12</b>	Write a program in Python for Knapsack	<b>62</b>

### CH – 4 Regular Expressions

<b>1</b>	Return the first word of given string.	<b>63</b>
<b>2</b>	Return the first two character of each word.	<b>64</b>
<b>3</b>	Return the domain types of given email-id.	<b>65</b>
<b>4</b>	Result date from given string.	<b>66</b>
<b>5</b>	Return all words of a string those starts with vowel.	<b>67</b>
<b>6</b>	Validate a phone no.(Phone no. Must be of 10 digits and starts 8 or 9).	<b>68</b>
<b>7</b>	Split a string with multiple delimiters.	<b>69</b>
<b>8</b>	Write a program in python to search and match given strings.	<b>70</b>
<b>9</b>	Write a program to check phone number pattern.	<b>71</b>
<b>JSON</b>		
<b>1</b>	Write a program in JSON to create JSON to Dictionary.	<b>72</b>
<b>2</b>	Write a program in to read JSON file.	<b>73</b>
<b>3</b>	Write a program to convert dictionary to JSON	<b>74</b>
<b>4</b>	Write a program for writing JSON to file.	<b>75</b>
<b>5</b>	Write a program for pretty print JSON.	<b>76</b>
<b>CSV</b>		
<b>1</b>	Write a program to create CSV file.	<b>77</b>
<b>2</b>	Write a program to read CSV file .	<b>78</b>
<b>3</b>	Write a program to modifying existing rows of person.CSV file.	<b>79</b>
<b>4</b>	Write a program to append new record to person.CSVfile.	<b>80</b>
<b>5</b>	Write a program to writing with custom delimiter as pipe( )	<b>81</b>
<b>6</b>	Write a program to writing CSV file using line terminator.	<b>82</b>
<b>7</b>	Write a program to writing CSV file with quochars.	<b>83</b>
<b>8</b>	Write a program to writing dictionary into peak.CSV file.	<b>84</b>
<b>9</b>	Write a program to writing dictionary into grade.CSV file with custom dialects.	<b>85</b>

## Subject : Python

<b>10</b>	Write a program to read people.CSV file, where after delimiter we have spaces.	<b>86</b>
<b>11</b>	Write a program to read quotes.CSV file, where delimiter is comma(,) but with quotes.	<b>87</b>
<b>12</b>	Write a program to read dialect.CSV file	<b>88</b>
<b>13</b>	Write a program in Python to Read people_data.csv into a dictionary by registering a new dialect.	<b>89</b>
<b>XML</b>		
<b>1</b>	Write a program in python to write xml file name bca.xml.	<b>90</b>
<b>2</b>	Write a program in python to read xml file.	<b>91</b>
<b>3</b>	Write a program in python to modify xml file.	<b>92</b>
<b>4</b>	Write a program in python to read xml file using element tree.	<b>93</b>

<b>CH – 5 Python and Data Analytics</b>		
<b>1</b>	Comparison of MSE, MAE and RMSE—regressionErrorMeasures.py	<b>94</b>

## How to Run Python?

From start menu click on python -> IDLE (3.7 64-bit)



Now, you will get the python home screen like this.

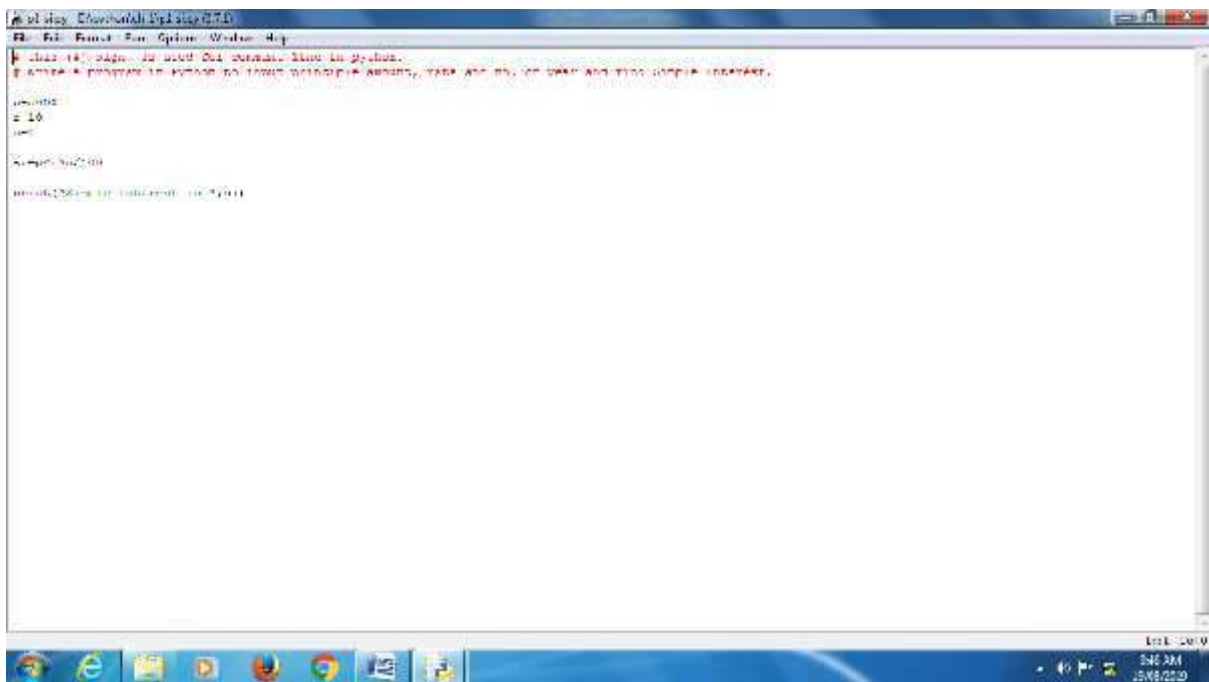


## Subject : Python

Now, click on file menu and then click on new file ( ctrl + n)



Then write code in it and save it with .py extension.



## Subject : Python

To run a program press F5 key from keyboard or from Run menu click on Run Module.

Now, you can see the output on python screen.



The screenshot shows a Python IDE window titled 'Python17194'. The window contains the following text:

```
Python 3.7.1 [AMD64] on win32
File "C:\Python37\python.exe", line 1, in <module>:
  <code>
>>>
```

The window also shows a Windows taskbar at the bottom with various application icons and a system tray on the right displaying the time as 9:24 AM on 2/18/2019.



## CH – 1 Introduction to Python

- 1) Write a program in Python to input principle amount, rate and no. of year and find Simple Interest.

# this (#) sign is used for comment line in python.

```
p=5000
```

```
r=10
```

```
n=1
```

```
si=p*r*n/100
```

```
print("Simple Interest is=",si)
```

**Output :**

Simple Interest is = 500.0

## Subject : Python

- 2) Write a program in Python to input Three Subjects Name and Marks and Calculate Total and Percentage, if Percentage is more than 70 then print appropriate message.

```
s1=input("Enter first subject name=")
s2=input("Enter second subject name=")
s3=input("Enter third subject name=")
m1=int(input("Enter first subject mark="))
m2=int(input("Enter second subject mark="))
m3=int(input("Enter third subject mark="))

tot=m1+m2+m3
per=tot/3

if(per>70):
    print("Congratulation, you achieve distinction")

print(s1,"=",m1)
print(s2,"=",m2)
print(s3,"=",m3)

print("total=",tot)
print("Percentage=",per)
```

### Output :

```
Enter first subject name=ANDROID
Enter second subject name=PYTHON
Enter third subject name=DW & DM
Enter first subject mark=87
Enter second subject mark=75
Enter third subject mark=83
Congratulation, you achieve distinction
ANDROID = 87
PYTHON = 75
DW & DM = 83
total= 245
Percentage= 81.66666666666667
```

## Subject : Python

- 3) Write a program in Python to input Two Number from user and check which number is Maximum

```
num1=int(input("Enter number1="))
num2=int(input("Enter number2="))

print("num1=",num1)
print("num2=",num2)

if(num1>num2):
    print(num1,"is Maximum")

else:
    print(num2,"is Maximum")
```

### Output :

```
Enter number1=100
Enter number2=300
num1= 100
num2= 300
300 is Maximum
```

4) Calculate Python strings.

```
str="Hello World!"  
print (str)  
print (str[0])  
print (str[2:5])  
print (str[2:])  
print (str*2)  
print (str+"Test")
```

**Output :**

```
Hello World!  
H  
llo  
llo World!  
Hello World!Hello World!  
Hello World!Test
```

5) Calculate Python list.

```
list=['Python',786,2.23,'Jagruti',70.2]
tinylis=[123,'Hello']
print (list)
print (tinylis)
print (list[0])
print(list[1:3])
print (list[2:])
print (list*2)
print(tinylis*2)
print (list+tinylis)
print(list[-1])
print(list[-3])
print(list[-4])
```

**Output :**

```
['Python', 786, 2.23, 'Jagruti', 70.2]
[123, 'Hello']
Python
[786, 2.23]
[2.23, 'Jagruti', 70.2]
['Python', 786, 2.23, 'Jagruti', 70.2, 'Python', 786, 2.23, 'Jagruti', 70.2]
[123, 'Hello', 123, 'Hello']
['Python', 786, 2.23, 'Jagruti', 70.2, 123, 'Hello']
70.2
2.23
786
```

6) Calculate Python tuple.

```
tuple=('Python',786,2.23,'Jagruti',70.2);  
tinytuple=(123,'Sneha')  
print (tuple);  
print (tuple[0]);  
print (tuple[1:3]);  
print(tuple[2:]);  
print (tinytuple*2);  
print(tuple+tinytuple);
```

**Output :**

```
('Python', 786, 2.23, 'Jagruti', 70.2)  
Python  
(786, 2.23)  
(2.23, 'Jagruti', 70.2)  
(123, 'Sneha', 123, 'Sneha')  
('Python', 786, 2.23, 'Jagruti', 70.2, 123, 'Sneha')
```

7) Calculate Python dictionary.

```
dict={ }
dict['one']="This is One"
dict[2]="This is Two"
tinydict={'name':'Jagruti','code':6852,'dept':'computer'}
print (tinydict)
print(tinydict.keys())
print(tinydict.values())
print(dict['one'])
print(dict[2])
print (tinydict)
print(tinydict.keys())
print(tinydict.values())
```

**Output :**

```
{'name': 'Jagruti', 'code': 6852, 'dept': 'computer'}
dict_keys(['name', 'code', 'dept'])
dict_values(['Jagruti', 6852, 'computer'])
This is One
This is Two
{'name': 'Jagruti', 'code': 6852, 'dept': 'computer'}
dict_keys(['name', 'code', 'dept'])
dict_values(['Jagruti', 6852, 'computer'])
```

## Subject : Python

- 8) Write a program in Python to check enter number is positive, negative, zero or not a number.

```
number=int(input("Enter any number:"))  
  
if number<0:  
    print("The entered number is Negative")  
  
elif number>0:  
    print("The Entered number is Positive")  
  
elif number==0:  
    print("Number is Zero")  
  
else:  
    print("The input is not a number")
```

### Output :

```
Enter any number:10  
The Entered number is Positive
```



**9) Write a program in Python to check input number is prime or not.**

```
# taking input from user
number = int(input("Enter any number: "))

# prime number is always greater than 1
if number > 1:
    for i in range(2, number):
        if (number % i) == 0:
            print(number, "is not a prime number")
            break
    else:
        print(number, "is a prime number")

# if the entered number is less than or equal to 1
# then it is not prime number
else:
    print(number, "is not a prime number")
```

**Output :**

```
Enter any number: 151
151 is a prime number
```

## Subject : Python

**10) Write a program to input any character from keyboard and check input character is vowel or not.**

```
ch=input("Enter a character:")
```

```
if(ch=='A' or ch=='a' or ch=='E' or ch=='e' or ch=='I' or ch=='i' or ch=='O' or ch=='o' or  
ch=='U' or ch=='u'):  
    print(ch,"is a Vowel")
```

```
else:  
    print(ch,"is a consonant")
```

### **Output :**

```
Enter a character:a  
a is a Vowel
```

## Subject : Python

11) Write a program in Python to print square of all number present in list.

```
number=[1,2,4,6,11,20]  
sq=0
```

```
for val in number:  
    sq=val*val  
    print(sq)
```

**Output :**

```
1  
4  
16  
36  
121  
400
```

**12) Write a program in Python to print sum of first five natural number.**

```
sum=0  
  
for val in range(1,6):  
    sum=sum+val  
  
print(sum)
```

**Output :**

15

13) Write a program to print following output.

**Output :**

```
0 , 10
0 , 11
0 , 12
0 , 13
1 , 10
1 , 11
1 , 12
1 , 13
2 , 10
2 , 11
2 , 12
2 , 13
```

**Program**

```
for num1 in range(3):
    for num2 in range(10,14):
        print(num1,",",num2)
```

**14) Write a program in Python to print following series.**

**1  
4  
7**

```
num=1
```

```
while num<10:  
    print(num)  
    num=num+3
```

**Output :**

**1  
4  
7**

15) Write a program in Python to display following output.

1,5  
2,6  
3,7

```
i=1  
j=5  
  
while i<4:  
    while j<8:  
        print(i,"",j)  
        j=j+1  
    i=i+1
```

**Output :**

1 , 5  
2 , 6  
3 , 7

16) Write a program in Python to display following output.

```
10
9
8
7
Loop is finished
```

```
num=10
```

```
while num>6:
    print(num)
    num=num-1
```

```
else:
    print("Loop is finished")
```

**Output :**

```
10
9
8
7
Loop is finished
```



**17) Write a program in Python to display all elements before number 88.**

```
for num in[11,9,5,88,20,40,3,18]:  
    print(num)  
    if(num==88):  
        print("The number 88 is found")  
        print("Terminating the loop")  
        break
```

**Output :**

```
11  
9  
5  
88  
The number 88 is found  
Terminating the loop
```

## Subject : Python

18) Write a program in Python to input character and check input character is Alphabet or not.

```
ch=input("Enter a character:")  
  
if((ch=="a" and ch<="z") or (ch>="A" and ch<="z")):  
    print(ch,"is an Alphabet")  
  
else:  
    print(ch,"is not an Alphabet")
```

### Output :

```
Enter a character:8  
8 is not an Alphabet
```

**19) Write a program to find factorial number.**

```
def factorial(num):
    """This is a recursive function that calls it self to find the factorial of given number"""
    if(num==1):
        return num
    else:
        return num*factorial(num-1)

num=int(input("Enter a Number:"))

if(num<0):
    print("Factorial can not be found for negative numbers")

elif(num==0):
    print("Factorial of 0 is 1")

else:
    print("Factorial of",num,"is:",factorial(num))
```

**Output :**

```
Enter a Number:5
Factorial of 5 is: 120
```

**20) Create an iterator that returns numbers starting with 1 and each sequence will increase by one. ( returning 1, 2, 3, 4, 5 etc.)**

```
class MyNumbers:
    def __iter__(self):
        self.a = 1
        return self

    def __next__(self):
        x = self.a
        self.a += 1
        return x

myclass = MyNumbers()
myiter = iter(myclass)

print(next(myiter))
print(next(myiter))
print(next(myiter))
print(next(myiter))
print(next(myiter))
```

**Output :**

```
1
2
3
4
5
```

21) Write a program to stop after 10 iteration.

```
class MyNumbers:
    def __iter__(self):
        self.a = 1
        return self

    def __next__(self):
        if self.a <= 10:
            x = self.a
            self.a += 1
            return x
        else:
            raise StopIteration

myclass = MyNumbers()
myiter = iter(myclass)

for x in myiter:
    print(x)
```

**Output :**

```
1
2
3
4
5
6
7
8
9
10
```

## Subject : Python

22) Write a program in Python to open text file ( .txt ) and read it.

```
f=open("d://python/ch_1_saru/demo.txt","r")
str=f.read()
print(str)
```

**Output :**

```
Helloo
Welcome to m.m.ghodasara college
```

## Subject : Python

23) Write a program in Python to write text file ( .txt ) and read only 20 character from it.

```
obj=open("test.txt","w")
obj.write("Shree M.M.Ghudasara College")
print("Writing Completed:")
```

```
obj=open("test.txt","r")
str=obj.read(20)
print("The Reading String is:",str)
```

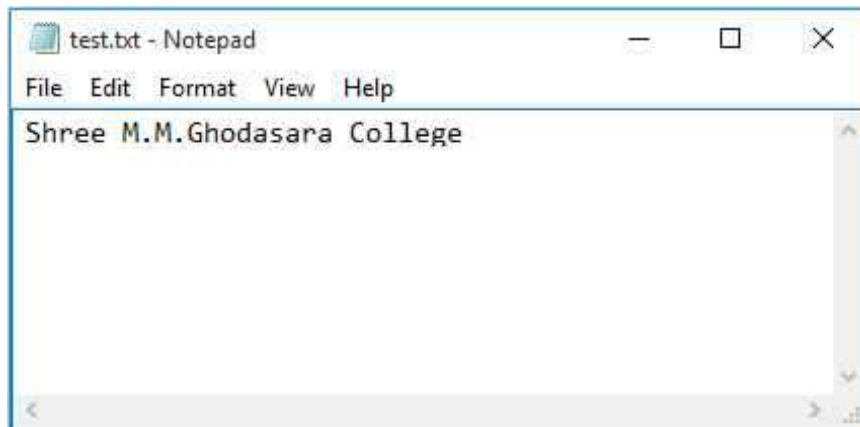
```
obj.close()
```

### Output :

```
Writing Completed:
The Reading String is: Shree M.M.Ghudasara
```

24) Write a program in Python to write position into a file.

Note: First create test.txt file



```
obj=open("test.txt","r+")
str=obj.read(10)
print("The Reading String is:",str)

position=obj.tell()
print("The Current Position is:",position)

position=obj.seek(0,0)
str=obj.read(20)
print("The Current Position is:",str)
obj.close()
```

**Output :**

The Reading String is: Shree M.M.  
The Current Position is: 10  
The Current Position is: Shree M.M.Ghodasara



## CH – 2 OOP using Python

### 1) Write a program in Python for Exception Handling.

```
#Standard Exception Handling
try:
    fh=open("test.txt","r")
    fh.write("Shree M.M.Ghodasara College")
    a=10
    b=0
    c=a/b

except (IOError,ZeroDivisionError):
    print("Error in open File")
    print("Value Can't Division By Zero")

else:
    print("Data Written Successfully")
```

#### Output :

```
Error in open File
Value Can't Division By Zero
```

**2) Write a program in Python for Nested Exception with finally block.**

**Note: Don't need to create demo.txt**

```
#nested exception with finally block
try:
    fh=open("demo.txt","w")
    try:
        fh.write("Content Here")
    finally:
        print("File is going to be closed")
        fh.close()

except IOError:
    print("Error in Open File")
```

**Output :**

File is going to be closed

**3) Write a program in Python Exception with argument.**

```
#Exception With Argument

def temp_convert(var):
    try:
        return int(var)
    except ValueError as Argument:
        print("Error")

#Call The Function

temp_convert("Hello")
```

**Output :**

Error

4) Write a program in Python for Assertion.

```
#Assertion

def kelvintofren(tempt):
    assert(tempt>=0),"Colder Than Actual"
    return ((tempt-273)*1.8)+32

print(kelvintofren(273))

print(kelvintofren(505.23))

print(kelvintofren(-5))
```

**Output :**

```
32.0
450.014000000000007
Traceback (most recent call last):
```

5) Write a program in Python to destroy class object.

```
#Destroy class object
class circle:
    def __init__(self,r=0): #double underscore
        self.r=r
    def __del__(self):
        class_name=self.__class__.__name__
        print(class_name,"Destroyed")

c1=circle()
c2=c1
c3=c1
print(id(c1),id(c2),id(c3))

del c1
del c2
del c3
```

**Output :**

```
277699804464 277699804464 277699804464
circle Destroyed
```

**6) Write a program in Python to create employee class and count it.**

```
class employee:
    empcount=0

    def __init__(self,name,salary):
        self.name=name
        self.salary=salary

    def empdisp(self):
        employee.empcount+=1
        print("Employee Name is:",self.name)
        print("Employee Salary is:",self.salary)

emp1=employee("Ram",50000)
emp1.empdisp()

emp2=employee("Shyam",40000)
emp2.empdisp()

print("Number of Employee:",employee.empcount)
```

**Output :**

```
Employee Name is: Ram
Employee Salary is: 50000
Employee Name is: Shyam
Employee Salary is: 40000
Number of Employee: 2
```

7) Write a program in Python to inheritance.

```
#inheritance
class parent:
    parentattr=100
    def __init__(self):
        print("Parent Class Constructor Called")
    def parentmethod(self):
        print("Parent Class Method Called")
    def setAttr(self,attr):
        parent.parentattr=attr
    def getAttr(self):
        print("Parent Attribute is:",parent.parentattr)

class child(parent):
    def __init__(self):
        print("Child Class Constructor")
    def childmethod(self):
        print("Child Class Method")

c=child()
c.childmethod()
c.parentmethod()
c.setAttr(500)
c.getAttr()
```

**Output :**

```
Child Class Constructor
Child Class Method
Parent Class Method Called
Parent Attribute is: 500
```

**8) Write a program in Python to encapsulation and information hiding.**

```
class counter:
    __setcount=0

    def __count(self):
        self.__setcount+=1
        print(self.__setcount)

cunt=counter()
cunt._counter__count()
cunt._counter__count()

print(cunt._counter__setcount)
```

**Output :**

```
1
2
2
```



9) Write a program in Python to linear search.

```
#Linear Search
def linear_search(value,search_value):
    pos=0
    search_res=False

#Logic For Search Data into list
    while (pos<len(value)and search_res is False):
        if(value[pos]==search_value):
            search_res=True
        else:
            pos=pos+1
    return search_res

l=[int(x) for x in input("Enter Number For List:").split(",")]
print(l)
res=int(input("Enter Number For Search:"))
print(linear_search(l,res))
```

**Output :**

```
Enter Number For List:1,2,3,4,5
[1, 2, 3, 4, 5]
Enter Number For Search:6
False
```

**10) Write a program in Python to interpolation.**

```
#Interpolation Search
def inter_search(values,x):
    startval=0;
    searchind=(len(values)-1)
    while(startval<=searchind and x>=values[startval] and x<=values[searchind]):

#Find The Mid Point
        mid=startval+int(((float(searchind-startval)/(values[searchind]-values[startval]))*(x-
values[startval])));

#Compare Values
        if(values[mid]==x):
            return "Found "+str(x)+" at Index "+str(mid)
        if(values[mid]<x):
            startval=mid+1
        return "Search Element Not In The List"

l=[2,6,11,19,27,31,45,127]
print(inter_search(l,11))
```

**Output :**

Found 11 at Index 2

**11) Write a program in Python to insertion sort.**

```
#Insertion Sort
def insert_sort(inputlist):
    for i in range(1,len(inputlist)):
        j=i-1
        k=inputlist[i]

    #Compare Values
    while(inputlist[j]>k and j>=0):
        inputlist[j+1]=inputlist[j]
        j=j-1
    inputlist[j+1]=k

list=[19,2,31,45,30,11,121,27]
insert_sort(list)
print(list)
```

**Output :**

[2, 11, 19, 27, 30, 31, 45, 121]

12) Write a program in Python to selection sort.

```
#Selection Sort
def selection_sort(input_list):
    for i in range(len(input_list)):
        minval=i
        for j in range(i+1,len(input_list)):
            if(input_list[minval]>input_list[j]):
                minval=j

#Swap the minimum value with comapred value
    input_list[i],input_list[minval]=input_list[minval],input_list[i]

l=[19,2,34,45,30,11,121,27]
selection_sort(l)
print(l)
```

**Output :**

[2, 11, 19, 27, 30, 34, 45, 121]

**13) Write a program in Python to bubble sort.**

```
#bubble sort
def bubble_sort(list):

#Swap and Sort the list
    for i in range(len(list)-1,0,-1):
        for j in range(i):
            if(list[j]>list[j+1]):
                temp=list[j]
                list[j]=list[j+1]
                list[j+1]=temp

list=[19,2,31,45,6,11,121,27]
bubble_sort(list)
print(list)
```

**Output :**

```
[2, 6, 11, 19, 27, 31, 45, 121]
```

**14) Write a program to in Python to merge sort.**

```
#Merge Sort
def merge_sort(unsorted_list):
    if(len(unsorted_list)<=1):
        return unsorted_list

#Find The Middle Point And Devide It
    middle=len(unsorted_list)//2
    left_list=unsorted_list[:middle]
    right_list=unsorted_list[middle:]
    left_list=merge_sort(left_list)
    right_list=merge_sort(right_list)
    return list(merge(left_list,right_list))

#Merge The Sorted List
def merge(left_half,right_half):
    res=[]
    while(len(left_half)!=0 and len(right_half)!=0):
        if(left_half[0]<right_half[0]):
            res.append(left_half[0])
            left_half.remove(left_half[0])
        else:
            res.append(right_half[0])
            right_half.remove(right_half[0])
    if(len(left_half)==0):
        res=res+right_half
    else:
        res=res+left_half
    return res

unsorted_list=[64,34,25,12,22,11,90,23]
print(merge_sort(unsorted_list))
```

**Output :**

```
[11, 12, 22, 23, 25, 34, 64, 90]
```

15) Write a program in Python for shell sort.

```
#Shell Sort
def shell_sort(input_list):
    gap=len(input_list)//2
    while(gap>0):
        for i in range(gap,len(input_list)):
            temp=input_list[i]
            j=i

        #Sort the sub list for this gap
        while(j>=gap and input_list[j-gap]>temp):
            input_list[j]=input_list[j-gap]
            j=j-gap
        input_list[j]=temp

    #Reduce the gap for next element
    gap=gap//2

list=[19,2,31,45,30,11,121,27]
shell_sort(list)
print(list)
```

**Output :**

[2, 11, 19, 27, 30, 31, 45, 121]

**16) Write a program in Python for operator overloading.**

```
#Operator OVerloading
class vector:
    def __init__(self,a,b):
        self.a=a
        self.b=b

    def __str__(self):
        return "vector (%d, %d)"%(self.a,self.b)

    def __add__(self,other):
        return vector(self.a+other.a,self.b+other.b)

v1=vector(2,10)
v2=vector(4,5)
print(v1+v2)
```

**Output :**

vector (6, 15)



**17) Write a program in Python for unary operator overloading.**

```
#Unary - Operator Overloading
class vector:
    def __init__(self,a):
        self.a=a

    def __str__(self):
        return "vector (%d)"%(self.a)

    def __neg__(self):
        return vector(-self.a)

v1=vector(10)
v2=-v1
print(v2)
```

**Output :**

vector (-10)

**18) Write a program in Python for Hash table example.**

```
#Hash Table Example
dict={'Name':'ABC','Age':20,'Class':'BCA'}
print("dict[Name]:",dict['Name'])
print("dict[Age]:",dict['Age'])

#Update The Dictionary
dict['Age']=21
print("dict[Age]:",dict['Age'])

#Delete The Dictionary Value
del dict['Name']
dict.clear()
del dict
print(dict['Name'])
```

**Output :**

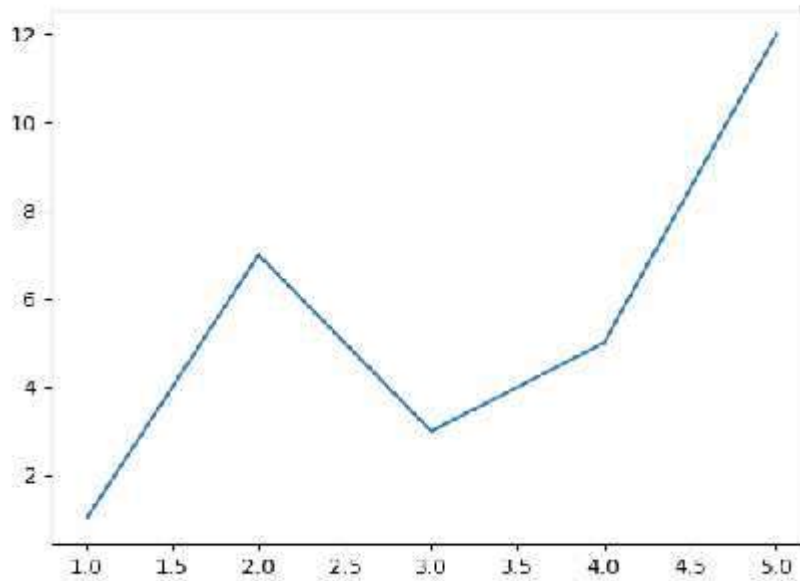
```
dict[Name]: ABC
dict[Age]: 20
dict[Age]: 21
Traceback (most recent call last):
  File "F:\PYTHON\python\ch_2\hashtable.py", line 14, in <module>
    print(dict['Name'])
TypeError: 'type' object is not subscriptable
```

## CH – 3 Plotting using PyLab

1) Write a program in Python for plotting.

```
import pylab
pylab.figure(1)
pylab.plot([1,2,3,4,5],[1,7,3,5,12])
pylab.show()
```

**Output :**



2) Write a program in Python for multiple figure.

```
import pylab  
pylab.figure(1)  
pylab.plot([1,2,3,4],[1,2,3,4])
```

```
pylab.figure(2)  
pylab.plot([1,4,2,3],[5,6,7,8])
```

```
pylab.savefig('figure-Addie')  
pylab.figure(1)  
pylab.plot([5,6,10,3])  
pylab.savefig('figure-Jane')
```

Output :

figure-Addie.png

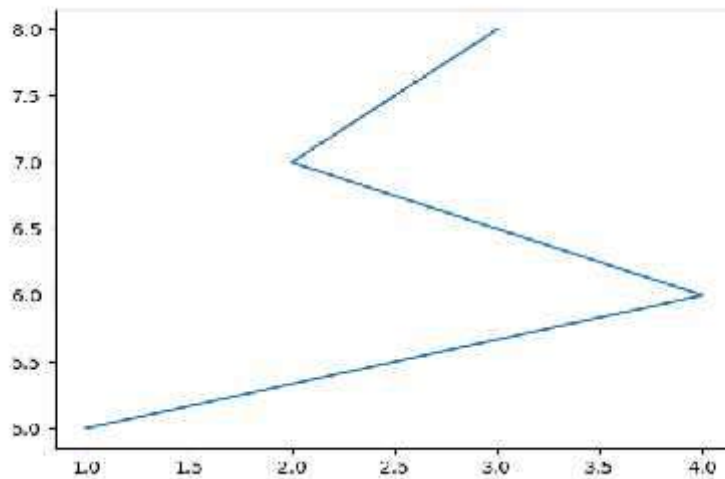
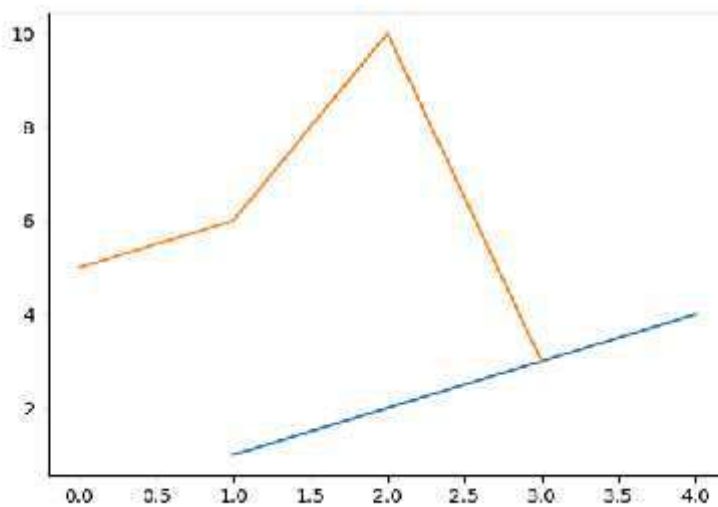


figure-Jane.png



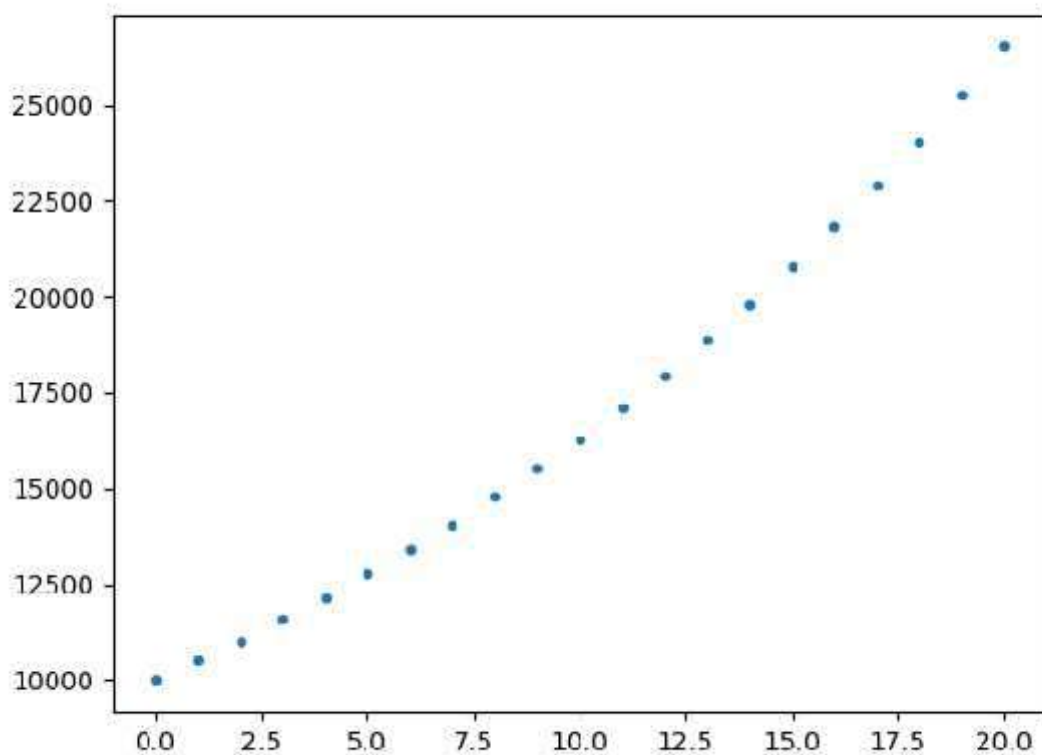
- 3) Write a program in Python to calculate rate of interest and display figure plotting compound growth of investment.

```
import pylab
p=10000
r=0.05
y=20
values=[]

for i in range(y+1):
    values.append(p)
    p+=p*r

pylab.plot(values, '.')
pylab.savefig('FigurePlottingCompoundGrowth')
```

**Output :**



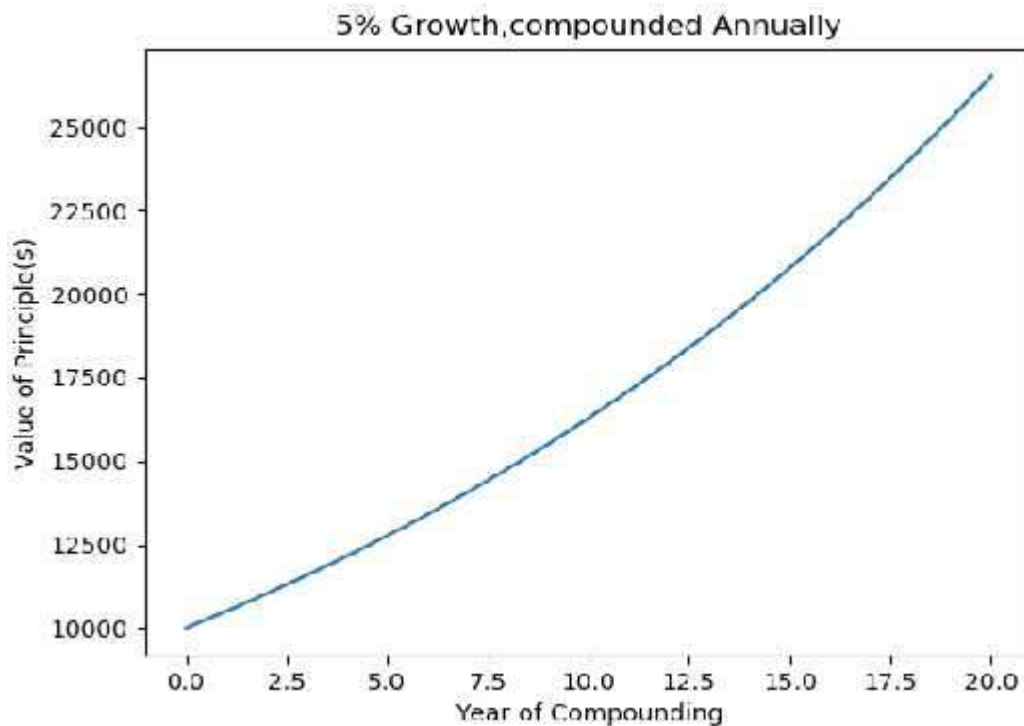
- 4) Write a program in to calculate principle amount and display growth of principle amount plotting with title and label.

```
import pylab
p=10000
r=0.05
y=20
values=[]

for i in range(y+1):
    values.append(p)
    p+=p*r

pylab.plot(values)
pylab.title('5% Growth,compounded Annually')
pylab.xlabel('Year of Compounding')
pylab.ylabel('Value of Principle($)')
pylab.show()
```

**Output :**



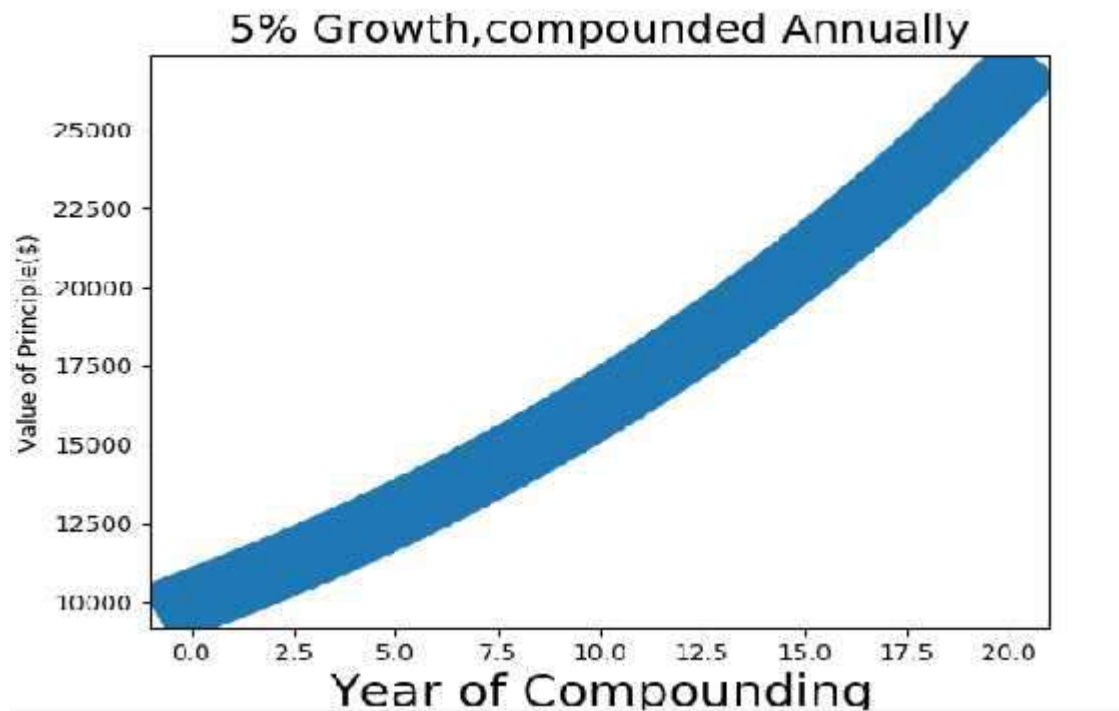
5) Write a program in Python to change the type size and line width.

```
import pylab
p=10000
r=0.05
y=20
values=[]

for i in range(y+1):
    values.append(p)
    p+=p*r

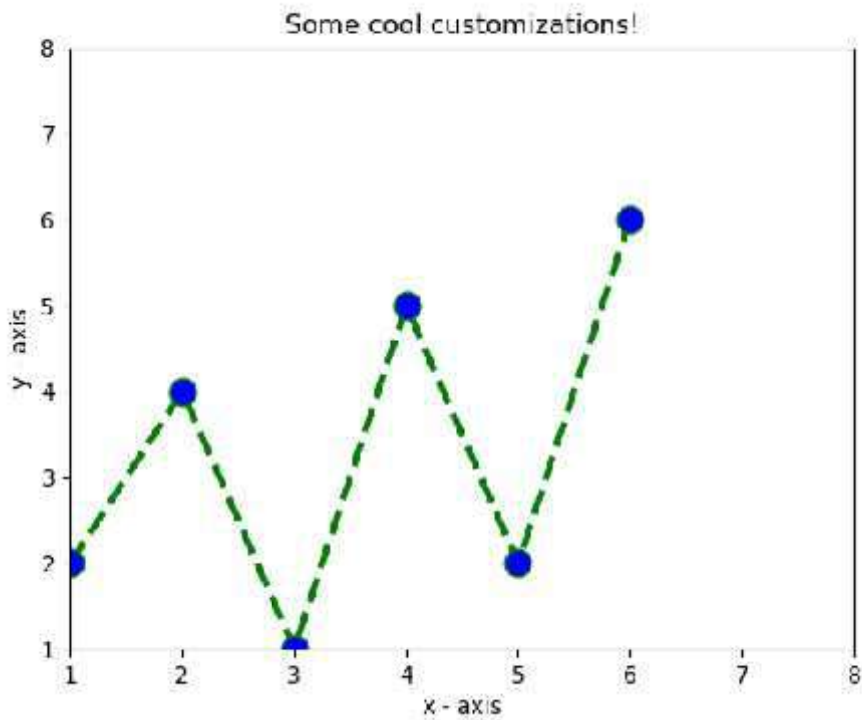
pylab.plot(values,linewidth=30)
pylab.title('5% Growth,compounded Annually',fontsize='xx-large')
pylab.xlabel('Year of Compounding',fontsize='20')
pylab.ylabel('Value of Principle($)' )
pylab.show()
```

**Output :**



6) Write a program in Python to display following dashed colored output.

Output :



```
import pylab as plt

# x axis values
x = [1,2,3,4,5,6]
# corresponding y axis values
y = [2,4,1,5,2,6]

# plotting the points
plt.plot(x, y, color='green', linestyle='dashed', linewidth = 3,
         marker='o', markerfacecolor='blue', markersize=12)

# setting x and y axis range
plt.ylim(1,8)
plt.xlim(1,8)

# naming the x axis
plt.xlabel('x - axis')
# naming the y axis
plt.ylabel('y - axis')

# giving a title to my graph
plt.title('Some cool customizations!')

# function to show the plot
plt.show()
```



7) Write a program in Python to create Bar Chart.

```
import pylab as plt

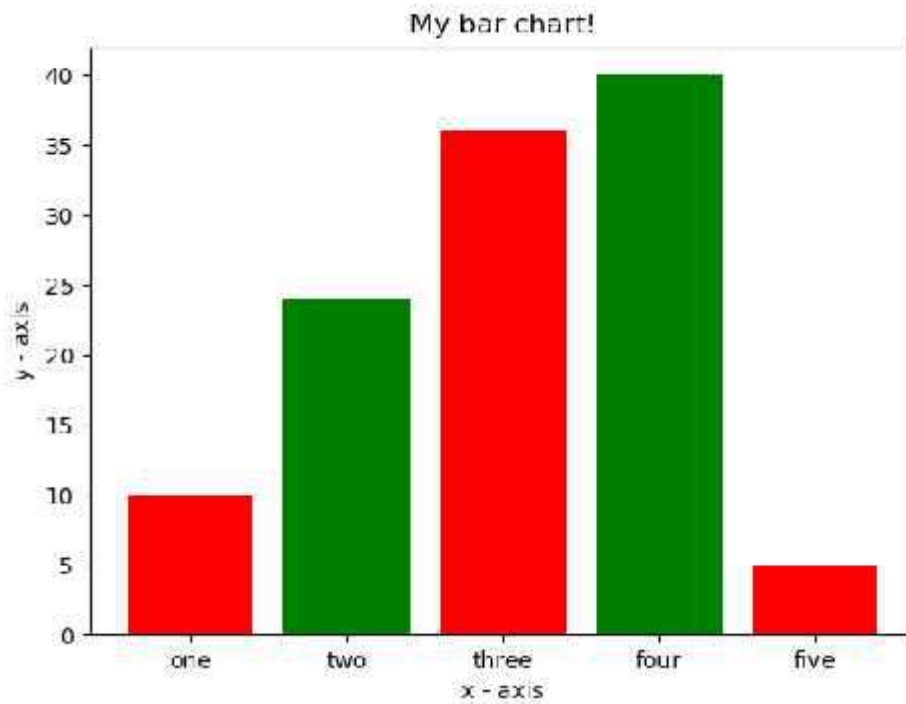
left=[1,2,3,4,5]
height=[10,24,36,40,5]

tick_label=['one','two','three','four','five']

plt.bar(left,height,tick_label=tick_label,width=0.8,color=['red','green'])

plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.title('My bar chart!')
plt.show()
```

**Output :**



## Subject : Python

### 8) Write a program in Python to create Histogram chart.

```
import pylab as plt

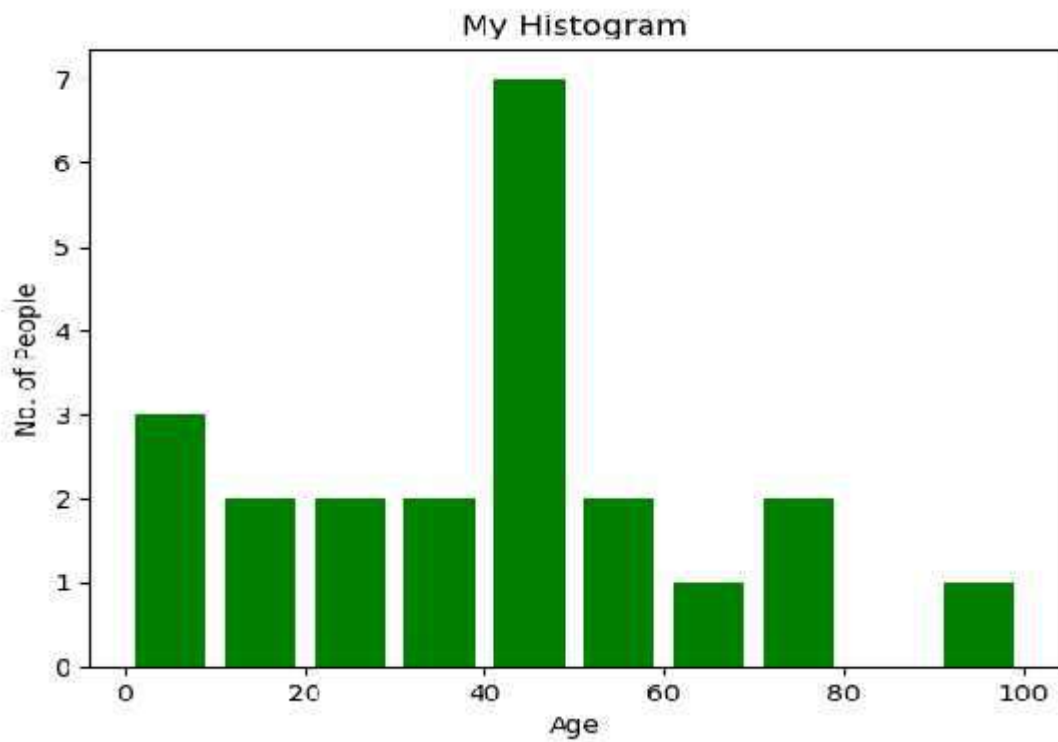
ages=[2,5,70,40,30,45,50,45,43,40,44,60,7,13,57,18,90,77,32,21,20,40]
range=(0,100)
bins=10

plt.hist(ages,bins,range,color='blue',histtype='bar',rwidth=0.8)

plt.xlabel('Age')
plt.ylabel('No of People')
plt.title('My Histogram')

plt.show()
```

#### Output :



9) Write a program in Python to create Scatter plot.

```
import pylab as plt

x=[1,2,3,4,5,6,7,8,9,10]
y=[2,4,5,7,6,8,9,11,12,12]

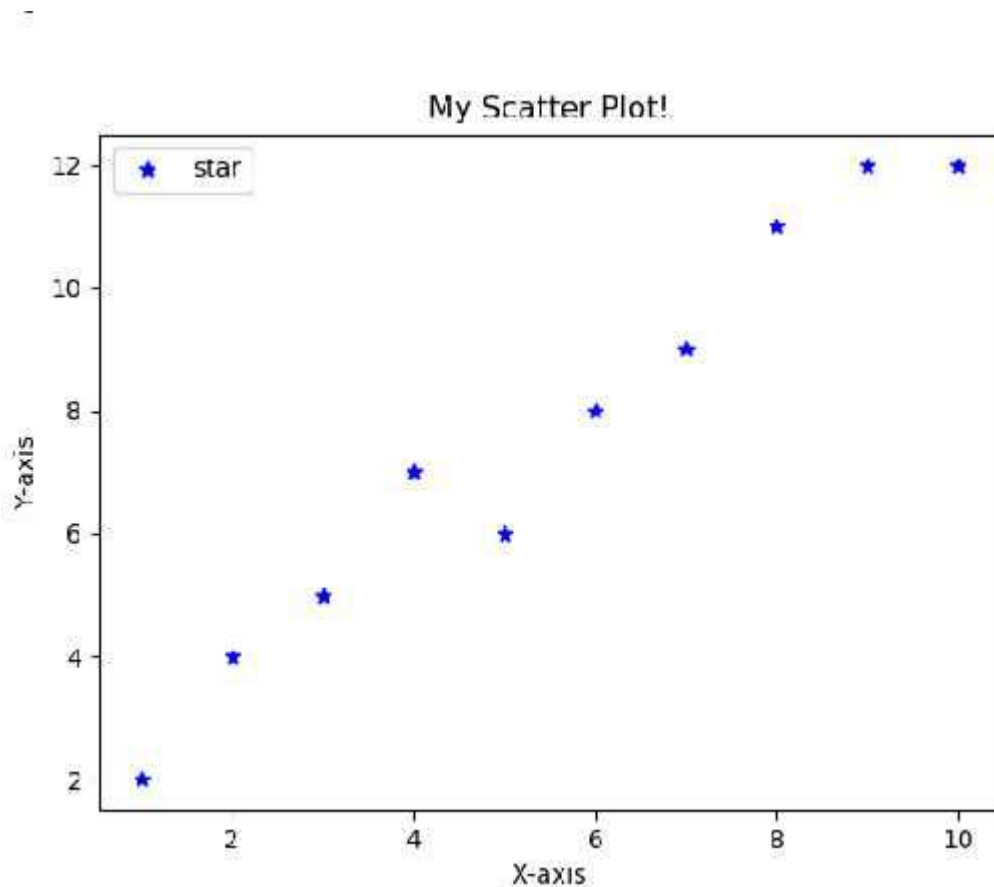
plt.scatter(x,y,label='star',color='blue',marker='*',s=35)

plt.xlabel('X-axis')
plt.ylabel('Y-axis')

plt.title('My Scatter Plot!')

plt.legend()
plt.show()
```

**Output :**



10) Write a program in Python to create Pie Chart.

```
import pylab as plt
# defining labels
activities = ['eat', 'sleep', 'work', 'play']

# portion covered by each label
slices = [3, 7, 8, 6]

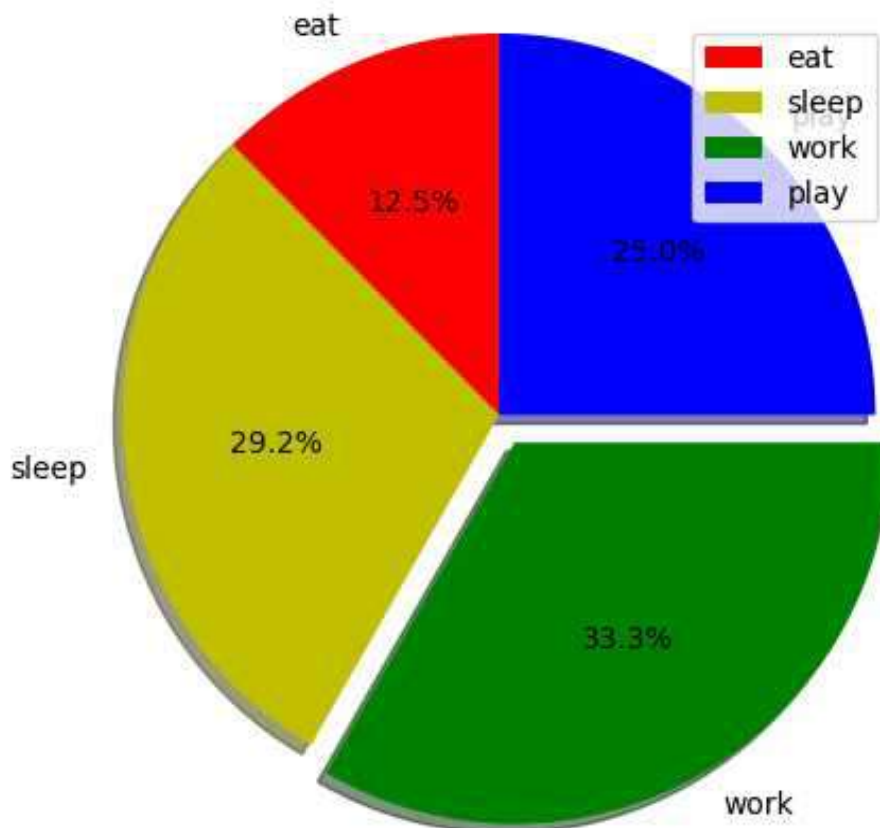
# color for each label
colors = ['r', 'y', 'g', 'b']

# plotting the pie chart
plt.pie(slices, labels = activities, colors=colors,
        startangle=90, shadow = True, explode = (0, 0, 0.1, 0),
        radius = 1.2, autopct = '%1.1f%%')

# plotting legend
plt.legend()

# showing the plot
plt.show()
```

**Output :**



## Subject : Python

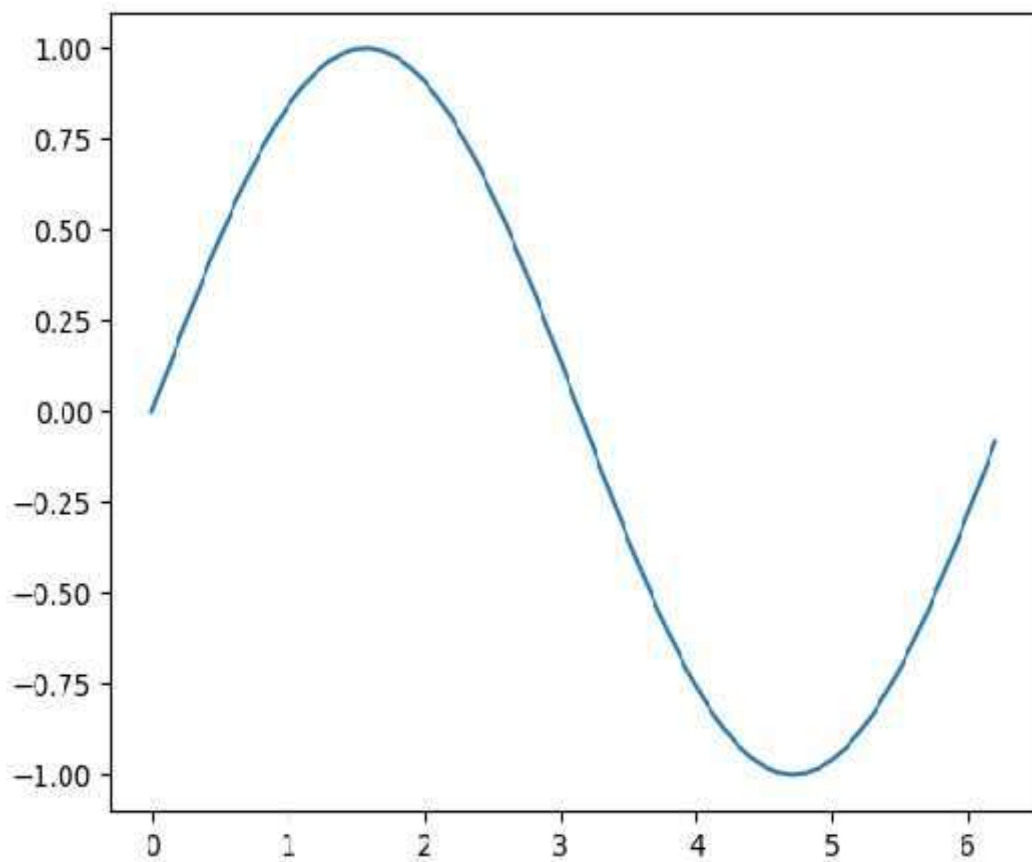
11) Write a program in Python for plotting curves of given equation for  
`x=np.arange(0,2*(np.pi),0.1)`  
`y=np.sin(x)`

```
import pylab as plt  
import numpy as ny
```

```
x=ny.arange(0,2*(ny.pi),0.1)  
y=ny.sin(x)
```

```
plt.plot(x,y)  
plt.show()
```

**Output :**



12) Write a program in Python for Knapsack.

```
def knapsack (w,wt,val,n):  
  
    k=[[0 for x in range (w+1)]for x in range (n+1)]  
    for i in range(n+1):  
        for j in range(w+1):  
            if(i==0 or j==0):  
                k[i][j]=0  
            elif(wt[i-1]<=j):  
                k[i][j]=max(val[i-1]+k[i-1][j-wt[i-1]],k[i-1][j])  
            else:  
                k[i][j]=k[i-1][j]  
    return k[n][w]  
  
val=[60,100,120]  
wt=[10,20,30]  
w=50  
n=len(val)  
print(knapsack(w,wt,val,n))
```

**Output :**

220

## CH – 4 Regular Expressions

### 1) Return the first word of given string.

```
import re
#Extract each character
result=re.findall(r'.','Av is largerst Analytics community of India')
print(result)

#Above space is also extraxted,now to avoid it use"\w' instead of","
result=re.findall(r'\w','Av is largerst Analytics community of India')
print("\n",result)

#Eatract each word(using "*" or "+")
result=re.findall(r'\w*','Av is largerst Analytics community of India')
print("\n",result)

#Above space is also extraxted,now to avoid it use"\w+' instead of"\w*"
result=re.findall(r'\w+', 'Av is largerst Analytics community of India')
print("\n",result)

#Extract each word(using "^")
result=re.findall(r'^\w+', 'Av is largerst Analytics community of India')
print("\n",result)

#If we will use "$" instead of "^",it will return the word from the end of the string.let's at
it
result=re.findall(r'\w+$', 'Av is largerst Analytics community of India')
print("\n",result)
```

### Output :

```
['A', 'v', ' ', 'i', 's', ' ', 'l', 'a', 'r', 'g', 'e', 'r', 's', 't', ' ', 'A', 'n', 'a', 'l', 'y', 't', 'i', 'c', 's', ' ', 'c', 'o', 'm', 'm', 'u', 'n', 'i', 't', 'y', ' ', 'o', 'f', ' ', 'I', 'n', 'd', 'i', 'a']

['A', 'v', 'i', 's', 'l', 'a', 'r', 'g', 'e', 'r', 's', 't', 'A', 'n', 'a', 'l', 'y', 't', 'i', 'c', 's', 'c', 'o', 'm', 'm', 'u', 'n', 'i', 't', 'y', 'o', 'f', 'I', 'n', 'd', 'i', 'a']

['Av', ' ', 'is', ' ', 'largerst', ' ', 'Analytics', ' ', 'community', ' ', 'of', ' ', 'India', '']

['Av', 'is', 'largerst', 'Analytics', 'community', 'of', 'India']

['Av']

['India']
```

## Subject : Python

### 2) Return the first two character of each word.

```
import re
#Extract Consecutive two characters of each word, excluding spaces(using "\w")
result=re.findall(r"\w\w",'Av is largest Analytics community of India')
print(result)
```

```
#Extract Consecutive two characters those available at start of word boundary(using "\b")
result=re.findall(r"\b\w\w.",'Av is largest Analytics community of India')
print("\n",result)
```

#### Output :

```
['Av', 'is', 'la', 'rg', 'es', 'An', 'al', 'yt', 'ic', 'co', 'mm', 'un', 'it', 'of', 'In', 'di']
```

```
['Av', 'is', 'la', 'An', 'co', 'of', 'In']
```



**3) Return the domain types of given Email-Id.**

```
import re
#To explain it in simple manner, i will again go with a step wise apporach:
#Extract all character after "@"
result=re.findall(r'@\w+', 'abc.test@gmail.com,xyz@test.in,test.first@analyticsvidhya.com, first.test@rest.biz')
print (result)

#Above you can see that ".com",".in" part is not extracted.To add it ,we will go with below code
result=re.findall(r'@\w+\.\w+', 'abc.test@gmail.com,xyz@test.in,test.first@analyticsvidhya.com, first.test@rest.biz')
print (result)

#Extract only domain name using “( )”
result=re.findall(r'@\w+(\.\w+)', 'abc.test@gmail.com,xyz@test.in,test.first@analyticsvidhya.com, first.test@rest.biz')
print (result)
```

**Output :**

```
['@gmail', '@test', '@analyticsvidhya', '@rest']
['@gmail.com', '@test.in', '@analyticsvidhya.com', '@rest.biz']
['com', 'in', 'com', 'biz']
```

**4) Return date from given string.**

```
import re
#here we will use "\d" to extract digit
result=re.findall(r'\d{2}-\d{2}-\d{4}','Amit 34-3456 12-05-2007, XYZ 56-4532 11-11-2011, ABC 67-8945 12-01-2009')
print (result)
```

```
#If you want to extract only year again parenthesis "()" will help you
result=re.findall(r'\d{2}-\d{2}-(\d{4})','Amit 34-3456 12-05-2007, XYZ 56-4532 11-11-2011, ABC 67-8945 12-01-2009')
print (result)
```

**Output :**

```
['12-05-2007', '11-11-2011', '12-01-2009']
['2007', '2011', '2009']
```

## Subject : Python

### 5) Return all words of a string those starts with vowel.

```
import re
result=re.findall(r'\w+', 'AV is largest Analytics community of India')
print (result)

#return words starts with alphabets(using[])
result=re.findall(r'[aeiouAEIOU]\w+', 'AV is largest Analytics community of India')
print (result)

#solution
result=re.findall(r'\b[aeiouAEIOU]\w+', 'AV is largest Analytics community of India')
print (result)

#Above example using ^
result=re.findall(r'\b[^aeiouAEIOU]\w+', 'AV is largest Analytics community of India')
print (result)

#Above you can see that it has returned words starting with space.to drop it from outputs
include space in square bracket[]
result=re.findall(r'\b[^aeiouAEIOU ]\w+', 'AV is largest Analytics community of India')
print (result)
```

#### Output :

```
['AV', 'is', 'largest', 'Analytics', 'community', 'of', 'India']
['AV', 'is', 'argest', 'Analytics', 'ommunity', 'of', 'India']
['AV', 'is', 'Analytics', 'of', 'India']
['is', 'largest', ' Analytics', ' community', ' of', ' India']
['largest', 'community']
```

## Subject : Python

### 6) Validate a phone number (phone number must be of 10 digits and starts with 8 or 9).

```
import re
li=['9999999999','999999-999','99999x9999']
for val in li:
    if re.match(r'[8-9]{1}[0-9]{9}',val) and len(val) == 10:
        print ('yes')
    else:
        print ('no')
```

#### Output :

```
yes
no
no
```

**7) Split a string with multiple delimiters.**

```
import re

#re.split
line = 'asdf fjdk;afed,fjek,asdf,foo' # String has multiple delimiters (";",","," ").
result= re.split(r'[;,\s]', line)
print (result)

#re.sub()
line = 'asdf fjdk;afed,fjek,asdf,foo'
result= re.sub(r'[;,\s]', ' ', line)
print (result)
```

**Output :**

```
['asdf', 'fjdk', 'afed', 'fjek', 'asdf', 'foo']
asdf fjdk afed fjek asdf foo
```

**8) Write a program in python to search and match given strings.**

```
import re

line = "Cats are smarter than dogs";
m = re.match( r'dogs', line)
if m:
    print ("match --> : ", m)
else:
    print( "No match!!")
m= re.search( r'dogs', line)
if m:
    print( "search --> ", m)
else:
    print ("No match!!")
```

**Output :**

```
No match!!
search --> <re.Match object; span=(22, 26), match='dogs'>
```

**9) Write a program to check phone number pattern.**

```
import re
phone="2004-959-559 #This is phone number"

#delete python-style comment
num=re.sub(r'#.*$',"",phone)
print("Phone number:",num)

#remove
num=re.sub(r'\D','',phone)
print("Phone number:",num)
num=re.sub(r'\d','',phone)
print("Phone number:",num)
```

**Output :**

```
Phone number: 2004-959-559
Phone number: 2004959559
Phone number: -- #This is phone number
```

## **CH – 4 JSON**

**1) Write a program to JSON to create JSON to dictionary.**

```
import json
```

```
person='{"Name": "Jagruti", "Languages": ["English", "Gujarati"]}'  
person_dict = json.loads(person)
```

```
print(person_dict)  
print(person_dict['Languages'])
```

**Output :**

```
{'Name': 'Jagruti', 'Languages': ['English', 'Gujarati']}  
['English', 'Gujarati']
```



**2) Write a program in Python to read JSON file.**

```
person.json  
{ "name": "pragna",  
  "languages": ["English", "Gujarati"]  
}
```

```
import json  
  
with open('d:\python37\program\json\person.json') as f:  
    data = json.load(f)  
  
print(data)
```

**Output :**

```
{'name': 'pragna', 'languages': ['English', 'Gujarati']}
```

**3) Write a program in Python to convert dictionary to JSON.**

```
import json

person_dict = {'name': 'Jagruti',
               'age': 25,
               'children': None
               }
person_json = json.dumps(person_dict)

print(person_json)
```

**Output :**

```
{"name": "Jagruti", "age": 25, "children": null}
```

**4) Write a program in Python for writing JSON to file.**

```
import json

person_dict = {"name": "Pragna",
               "languages": ["English", "Gujarati"],
               "married": True,
               "age": 32
               }

with open('person.json', 'w') as json_file:
    json.dump(person_dict, json_file)
```

**Output :**

```
{"name": "Pragna", "languages": ["English", "Gujarati"], "married": true, "age": 32}
```

**5) Write a program in Python pretty print JSON**

```
import json

person_string = '{"name": "Pragna", "languages": "English", "numbers": [2, 1.6, null]}'

# Getting dictionary
person_dict = json.loads(person_string)

# Pretty Printing JSON string back
print(json.dumps(person_dict, indent = 4, sort_keys=True))
```

**Output :**

```
{
    "languages": "English",
    "name": "Pragna",
    "numbers": [
        2,
        1.6,
        null
    ]
}
```

## Subject : Python

### CH – 4 CSV

#### 1) Write a program in Python to create CSV file.

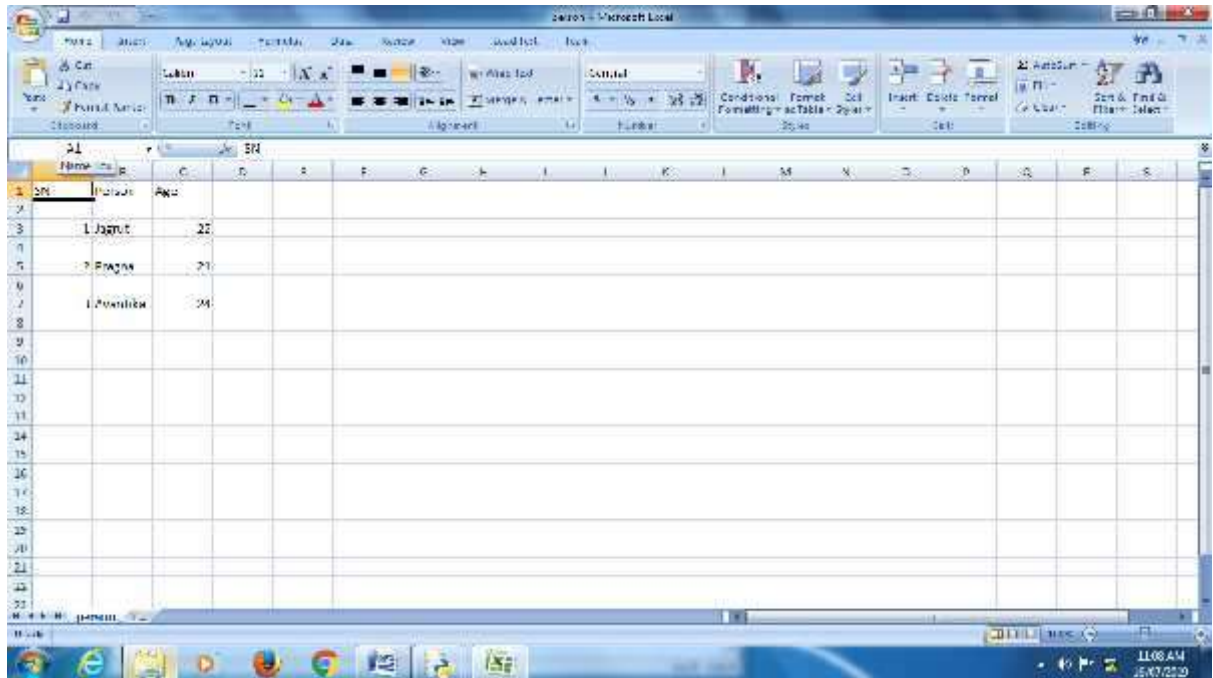
```
import csv

csvData = [['SN','Person', 'Age'], ['1','Jagruiti', '22'], ['2','Pragna', '21'], ['3','Avantika', '24']]

with open('person.csv', 'w') as csvFile:
    writer = csv.writer(csvFile)
    writer.writerows(csvData)
```

#### Output :

**Note:** open person.csv file from your folder



SN	Person	Age
1	Jagruiti	22
2	Pragna	21
3	Avantika	24

**2) Write a program in Python to read CSV file .**

```
import csv

with open(' D:\python\ch_4\csv\person.csv', 'r') as csvfile:
    reader = csv.reader(csvfile)
    for row in reader:
        print(row)
```

**Output :**

['SN', 'Person', 'Age']

[]

['1', 'Jagruti', '22']

[]

['2', 'Pragna', '21']

[]

['3', 'Avantika', '24']

[]

## Subject : Python

### 3) Write a program in Python to modifying existing rows of person.csv.

```
import csv

row = ['1', 'Mishti', '27']

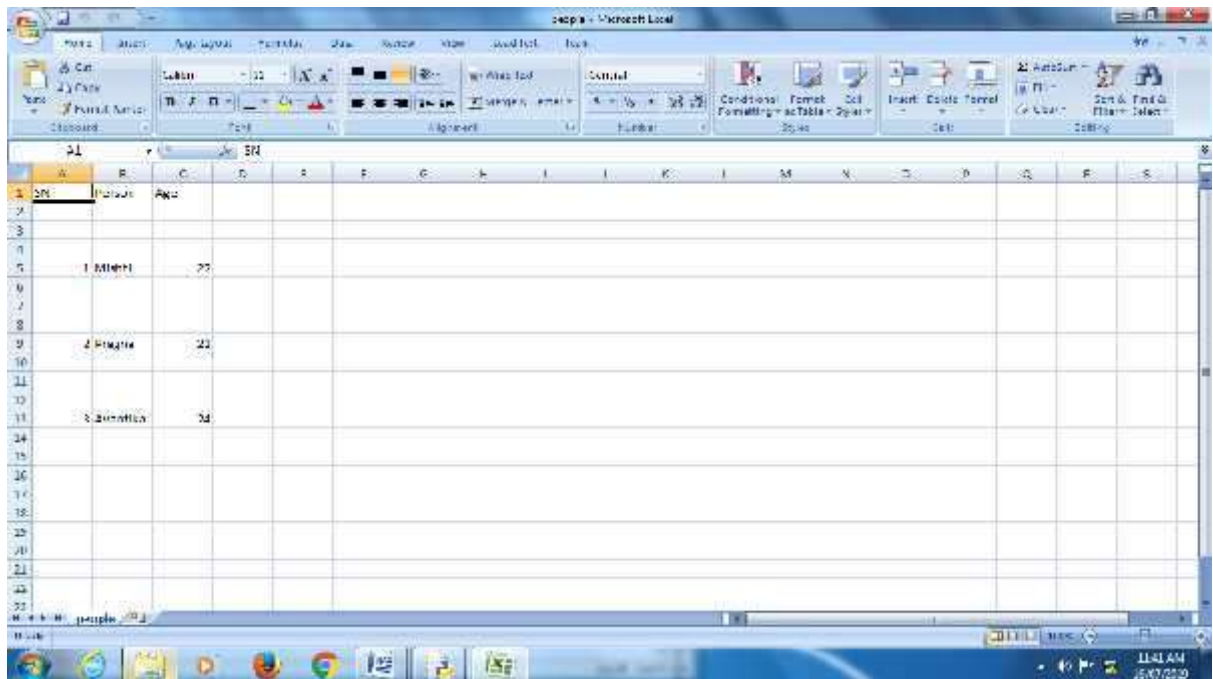
with open('person.csv', 'r') as readfile:
    reader = csv.reader(readfile)
    lines = list(reader)
    lines[2] = row

with open('people.csv', 'w') as writefile:
    writer = csv.writer(writefile)
    writer.writerows(lines)

readfile.close()
writefile.close()
```

#### Output :

**Note:** open people.csv file from your folder



The screenshot shows a Microsoft Excel spreadsheet with the following data:

SN	Name	Age
1	Mishti	27
2	Mishti	22
3	Mishti	24





## Subject : Python

### 5) Write a program in Python writing with custom delimiter as pipe(|).

```
import csv

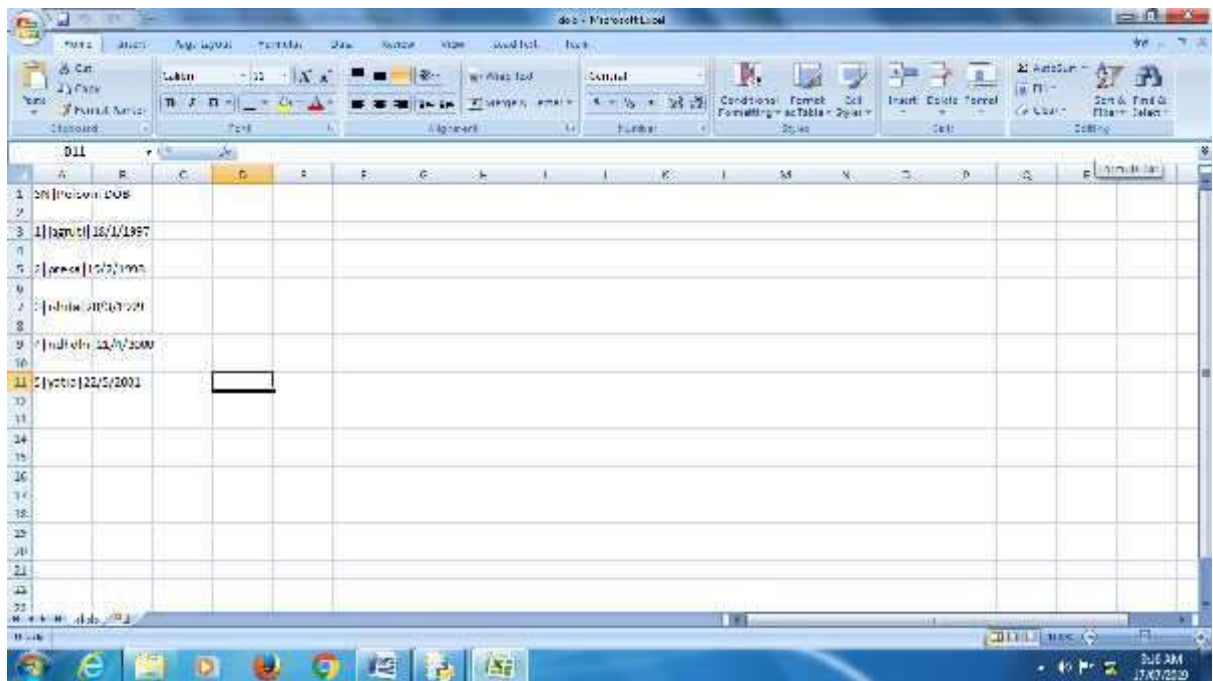
person = [['SN', 'Person', 'DOB'],
          ['1', 'jagruti', '18/1/1997'],
          ['2', 'prexa', '19/2/1998'],
          ['3', 'ishita', '20/3/1999'],
          ['4', 'ridhdhi', '21/4/2000'],
          ['5', 'yatra', '22/5/2001']]

csv.register_dialect('myDialect',
                    delimiter = '|',
                    quoting=csv.QUOTE_NONE,
                    skipinitialspace=True)

with open('dob.csv', 'w') as f:
    writer = csv.writer(f, dialect='myDialect')
    for row in person:
        writer.writerow(row)
f.close()
```

#### Output :

**Note:** open dob.csv file from your folder



SN	Person	DOB
1	jagruti	18/1/1997
2	prexa	19/2/1998
3	ishita	20/3/1999
4	ridhdhi	21/4/2000
5	yatra	22/5/2001

## Subject : Python

### 6) Write a program in python to writing csv file using line terminator.

```
import csv
```

```
csvData = [['Fruit', 'Quantity'], ['Apple', '5'], ['Orange', '7'], ['Mango', '8']]
```

```
csv.register_dialect('myDialect', delimiter = '|', lineterminator = '\r\n\r\n')
```

```
with open('lineterminator.csv', 'w') as f:
```

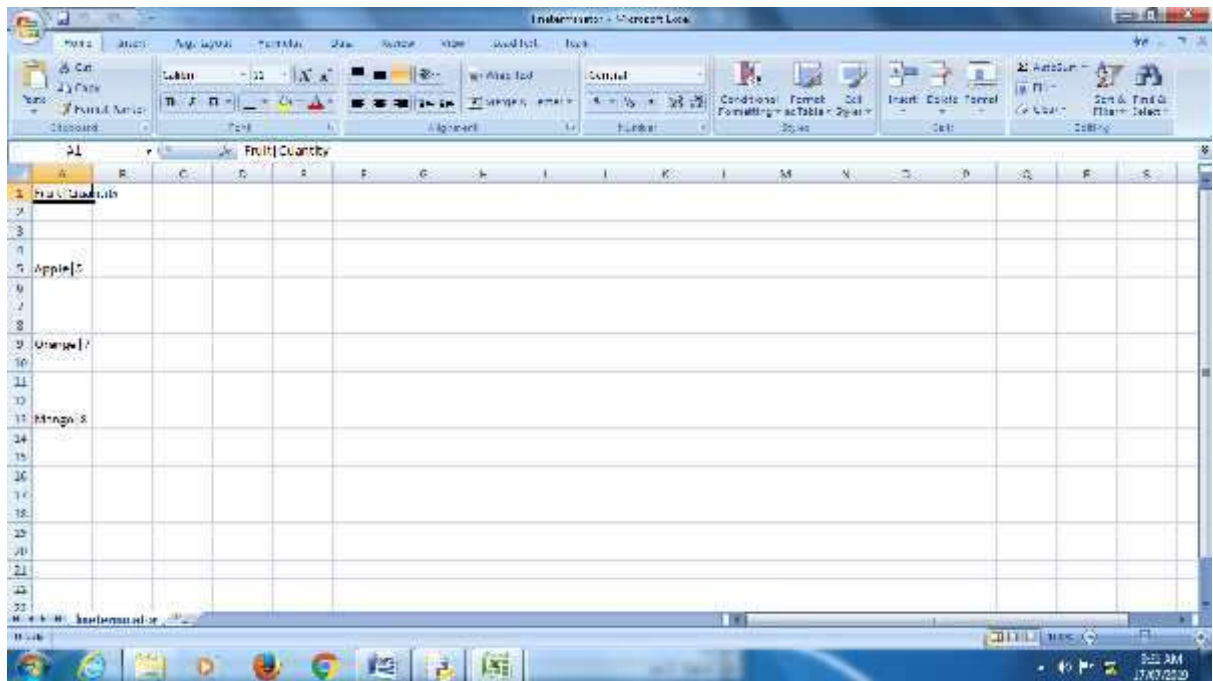
```
    writer = csv.writer(f, dialect='myDialect')
```

```
    writer.writerows(csvData)
```

```
f.close()
```

#### Output :

**Note:** open lineterminator.csv file from your folder



## Subject : Python

### 7) Write a program in python to writing csv file with quochars.

```
import csv

csvData = [['SN', 'Items'], ['1', 'Pen'], ['2', 'Book'], ['3', 'Copy']]

csv.register_dialect('myDialect',
delimiter = '|',
quotechar = '"',
quoting=csv.QUOTE_ALL,
skipinitialspace=True)

with open('quotechar.csv', 'w') as csvFile:
    writer = csv.writer(csvFile, dialect='myDialect')
    writer.writerows(csvData)

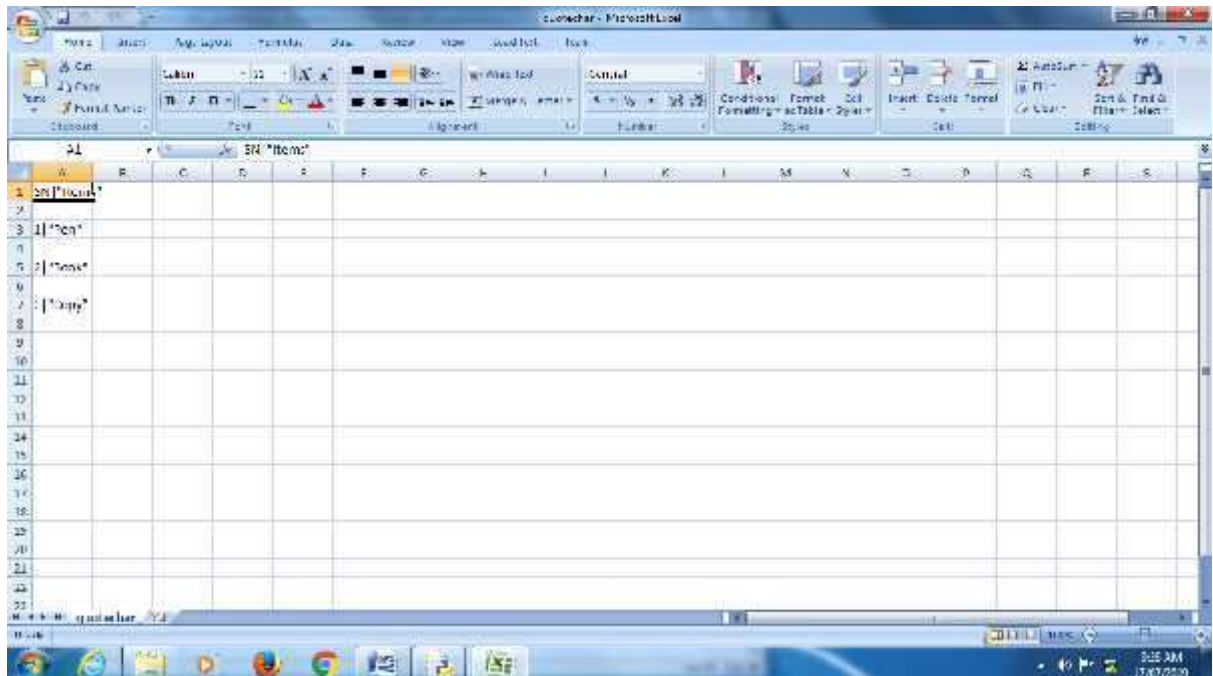
print("writing completed")

csvFile.close()
```

#### Output :

writing completed

**Note:** open quotechar.csv file from your folder .



## Subject : Python

### 8) Write a program in python to writing dictionary into peak.csv file.

```
import csv

data = [{'mountain': 'Everest', 'height': '8848'},
        {'mountain': 'K2', 'height': '8611'},
        {'mountain': 'Kanchenjunga', 'height': '8586'}]

with open('peak.csv', 'w') as csvFile:
    fields = ['mountain', 'height']
    writer = csv.DictWriter(csvFile, fieldnames=fields)
    writer.writeheader()
    writer.writerows(data)

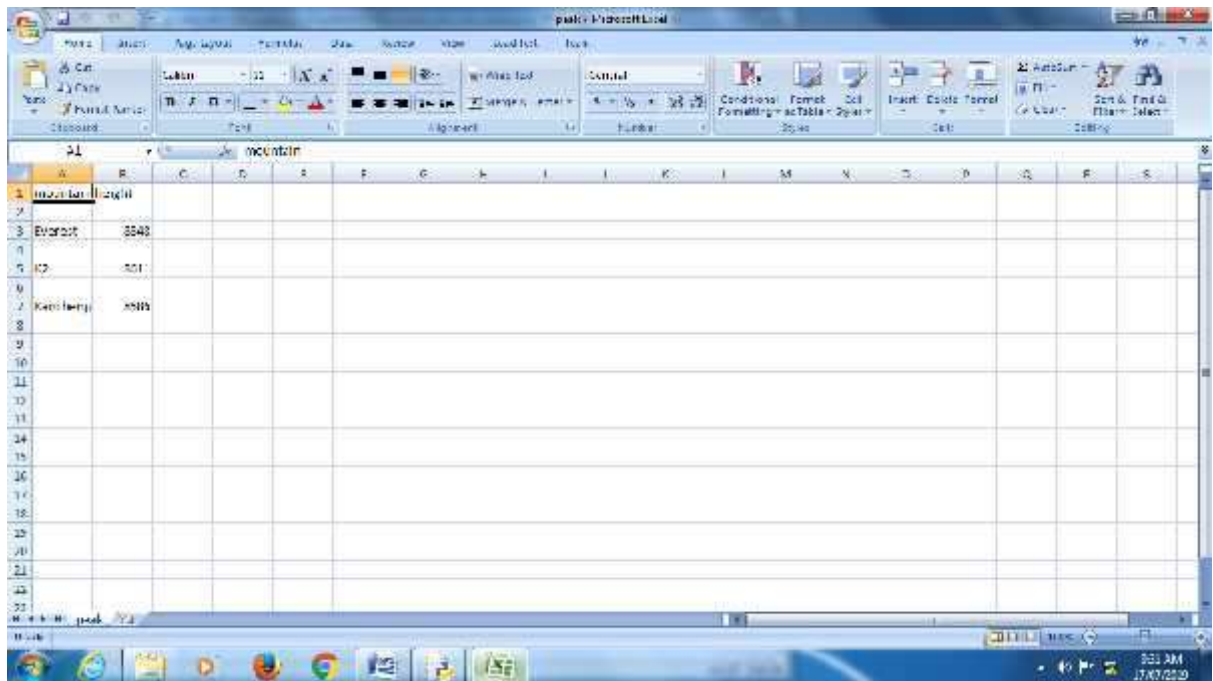
print("writing completed")

csvFile.close()
```

#### Output :

writing completed

**Note:** open peak.csv file from your folder .



## Subject : Python

- 9) Write a program in python to writing dictionary into grade.csv file with custom dialects.

```
import csv
```

```
csv.register_dialect('myDialect', delimiter = '|', quoting=csv.QUOTE_ALL)
```

```
with open('grade.csv', 'w') as csvfile:
```

```
    fieldnames = ['Name', 'Grade']
```

```
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames, dialect="myDialect")
```

```
    writer.writeheader()
```

```
    writer.writerows([{'Grade': 'B', 'Name': 'Jagruti'},
```

```
                      {'Grade': 'A', 'Name': 'Pragna'},
```

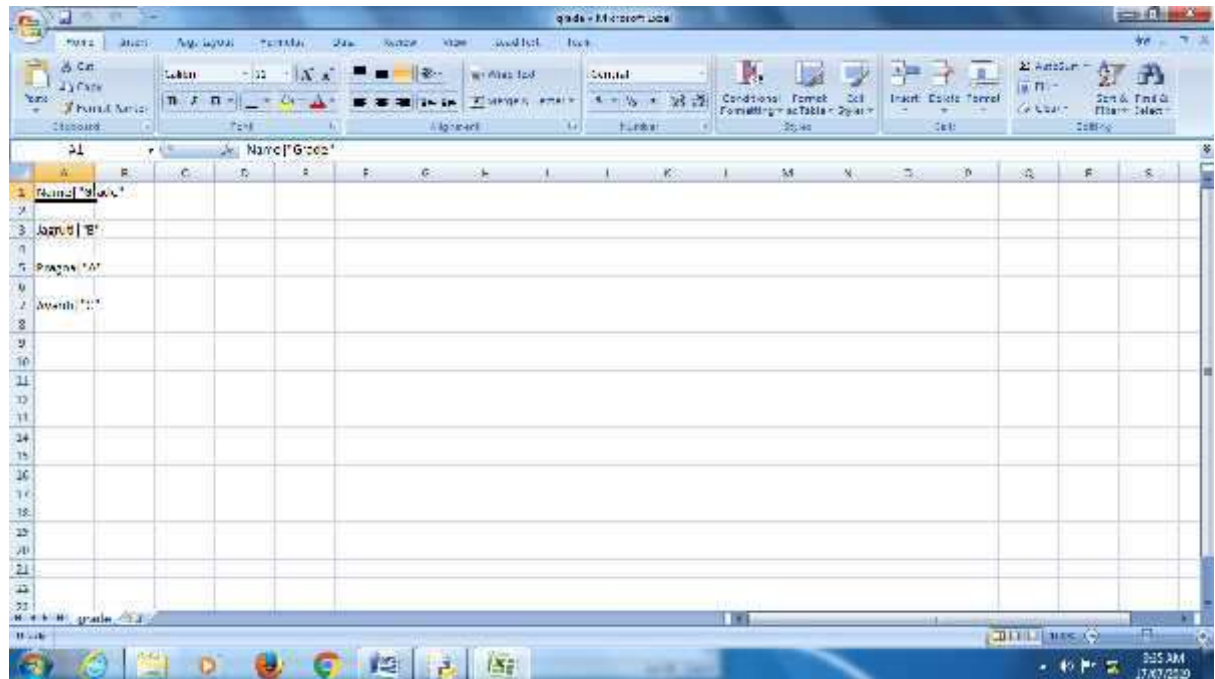
```
                      {'Grade': 'C', 'Name': 'Avanti'}])
```

```
print("writing completed")
```

### Output :

writing completed

**Note:** open grade.csv file from your folder .



## Subject : Python

10) Write a program in python to read people.csv file, where after delimiter we have spaces.

```
import csv

csv.register_dialect('myDialect',
delimiter = ',',
skipinitialspace=True)

with open('people.csv', 'r') as csvFile:
    reader = csv.reader(csvFile, dialect='myDialect')
    for row in reader:
        print(row)

csvFile.close()
```

### Output :

```
['Person', 'Age']
[]
['Jagruti', '22']
[]
['Pragna', '21']
[]
['Avantika', '24']
[]
```

## Subject : Python

11) Write a program in python to read quotes.csv file, where delimiter is comma ( , )but with quotes.

**Note:** first of all create quotes.csv file.

```
"SN", "Name", "Quotes"  
1, Buddha, "What we think we become"  
2, Mark Twain, "Never regret anything that made  
you smile"  
3, Oscar Wilde, "Be yourself everyone else is already  
taken"
```

```
import csv  
  
csv.register_dialect('myDialect',  
delimiter = ',',  
quoting=csv.QUOTE_ALL,  
skipinitialspace=True)  
  
with open('quotes.csv', 'r') as f:  
    reader = csv.reader(f, dialect='myDialect')  
    for row in reader:  
        print(row)
```

### Output :

```
["SN", "Name", "Quotes"]  
[1, Buddha, "What we think we become"]  
[2, Mark Twain, "Never regret anything that made you smile"]  
[3, Oscar Wilde, "Be yourself everyone else is already taken"]
```

12) Write a program in python to read dialect.csv file.

**Note:** first of all create dialects.csv file.

```
"pencil"|"eraser"|"sharpner"  
"book"|"chair"|"table"  
"apple"|"mango"|"grapes"
```

```
import csv  
  
csv.register_dialect('myDialect',  
delimiter = ',',  
quoting=csv.QUOTE_ALL,  
skipinitialspace=True)  
  
with open('quotes.csv', 'r') as f:  
    reader = csv.reader(f, dialect='myDialect')  
    for row in reader:  
        print(row)
```

**Output :**

```
["pencil"|"eraser"|"sharpner"]  
["book"|"chair"|"table"]  
["apple"|"mango"|"grapes"]
```



## Subject : Python

- 13) Write a program in python to read people\_data.csv into a dictionary by registering a new dialect.

```
import csv

csv.register_dialect('myDialect',
delimiter = '|',
skipinitialspace=True)

with open("people.csv", 'r') as csvfile:
    reader = csv.DictReader(csvfile, dialect='myDialect')
    for row in reader:
        print(row)

csvfile.close()
```

### Output :

```
OrderedDict([('Person,Age', 'Jagruti,22')])
OrderedDict([('Person,Age', 'Pragna,21')])
OrderedDict([('Person,Age', 'Avantika,24')])
```

## CH – 4 XML

1) Write a program in python to write xml file name bca.xml.

```
import xml.etree.ElementTree as ET
data=ET.Element('data')
items=ET.SubElement(data,'items')
item1=ET.SubElement(items,'item')
item2=ET.SubElement(items,'item')
item1.set('name','item1')
item2.set('name','item2')
item1.text='BCA'
item2.text='Bsc-IT'

mydata=ET.tostring(data)
print(mydata.decode("utf-8"))
myfile=open("bca.xml","w")
myfile.write(mydata.decode("utf-8"))
myfile.close()
```

**Output :**

```
<data><items><item name="item1">BCA</item><item name="item2">Bsc-
IT</item></items></data>
```

**2) Write a program in python to read xml file.**

```
from xml.dom import minidom

mydoc=minidom.parse('items.xml')
print(mydoc)

items=mydoc.getElementsByTagName('item')
print(items)

#one specific item attribute
print('item tag has two attribute')
print(items[0].attributes['name'].value)
print(items[1].attributes['name'].value)
#print all item attribute

for elem in items:
    print(elem.attributes['name'].value)
#second item data
print('\n Item2 Data')
print(items[1].firstChild.data)
print(items[1].childNodes[0].data)

#all item data

print('\n All Item Data')
for elem in items:
    print(elem.firstChild.data)
```

**Output :**

```
<xml.dom.minidom.Document object at 0x000000000328BFA8>
[<DOM Element: item at 0x2f90f20>, <DOM Element: item at 0x3283048>]
item tag has two attribute
item1
item2
item1
item2

Item2 Data
item2abc
item2abc

All Item Data
item1abc
item2abc
```

3) Write a program in python to modify xml file.

```
#modify Element

import xml.etree.ElementTree as ET

tree=ET.parse('items.xml')
root=root.getroot()

#change field text

for elem in root.iter('item'):
    elem.text='new Text'

#modify an attribute

for elem in root.iter('item'):
    elem.set('name','newitem')

#adding an attribute

for elem in root.iter('item'):
    elem.set('type','value')

#add second Element

attrib={ }
element=root.makeelement('bca',attrib)
root.append(element)

attrib={'name2':'secondelement2'}
subelement=root[0][1].makeelement('student',attrib)
ET.SubElement(root[1],'student',attrib)
root[1][0].text='bca class'
tree.write('modifyitem.xml')
```

**Output:**

```
<data>
  <items>
    <item name="newitem" type="value">new Text</item>
    <item name="newitem" type="value">new Text</item>
  </items>
<bca><student name2="secondelement2">bca class</student></bca></data>
```

**4) Write a program in python to read xml file using element tree.**

```
import xml.etree.ElementTree as ET
tree=ET.parse('items.xml')
root=tree.getroot()

#one specific item attribute

print(root[0][1].attrib)

#print all attribute
for elem in root:
    for subitem in elem:
        print(subitem.attrib)

#print value for specific attribute
print(root[0][1].text)
```

**Output :**

```
{'name': 'item2'}
{'name': 'item1'}
{'name': 'item2'}
item2abc
```

## CH – 5 Python and Data Analytics

### 1) Comparison of MSE, MAE and RMSE .

```
#here are some made-up numbers to start with
target = [1.5, 2.1, 3.3, -4.7, -2.3, 0.75]
prediction = [0.5, 1.5, 2.1, -2.2, 0.1, -0.5]

error = [ ]
for i in range(len(target)):
    error.append(target[i] - prediction[i])
#print the errors print("Errors ",) print(error)
#ans: [1.0, 0.60000000000000009, 1.1999999999999997, -2.5, #-2.3999999999999999,
1.25]

#calculate the squared errors and absolute value of errors
squaredError = [ ]
absError = [ ]
for val in error:
    squaredError.append(val*val)
    absError.append(abs(val))
#print squared errors and absolute value of errors print("Squared Error")
print(squaredError)
#ans: [1.0, 0.36000000000000001, 1.4399999999999993, 6.25, #5.7599999999999998,
1.5625] print("Absolute Value of Error")

print(absError)
#ans: [1.0, 0.60000000000000009, 1.1999999999999997, 2.5, #2.3999999999999999,
1.25]

#calculate and print mean squared error MSE
print("MSE = ", sum(squaredError)/len(squaredError))
#ans: 2.72875

from math import sqrt
#calculate and print square root of MSE (RMSE)
print("RMSE = ", sqrt(sum(squaredError)/len(squaredError)))
#ans: 1.65189285367

#calculate and print mean absolute error MAE
print("MAE = ", sum(absError)/len(absError))
#ans: 1.49166666667

#compare MSE to target variance
targetDeviation = [ ]
targetMean = sum(target)/len(target)
for val in target:
```

## Subject : Python

```
targetDeviation.append((val - targetMean)*(val - targetMean))

#print the target variance
print("Target Variance = ", sum(targetDeviation)/len(targetDeviation))
#ans: 7.5703472222222219

#print the the target standard deviation (square root of variance)
print("Target Standard Deviation = ", sqrt(sum(targetDeviation)
/len(targetDeviation)))
#ans: 2.7514263977475797
```

### Output :

```
[1.0, 0.36000000000000001, 1.4399999999999993, 6.25, 5.76, 1.5625]
[1.0, 0.60000000000000001, 1.1999999999999997, 2.5, 2.4, 1.25]
MSE = 2.72875
RMSE = 1.651892853668179
MAE = 1.4916666666666665
Target Variance = 7.570347222222222
Target Standard Deviation = 2.7514263977475797
```

**Lab Manual  
Developed at  
Computer Science Laboratory  
Of  
Shree M.M.Ghodasara Mahila College  
Under guidance of**

<b>Departmental Head</b>	<b>Dr. Raksha Bathani</b>
<b>Lab Incharge</b>	<b>Mr. Rajesh Makwana</b>
<b>Faculty</b>	<b>Dr. Jagruti Jadav</b>
	<b>Miss. Sarika Odiya</b>

*~ There is no alteration of Hardwork*