

8. Implementing programs using written modules and Python Standard Libraries (pandas, numpy, Matplotlib, scipy)

Ex No : 8A

PANDAS

Pandas is a Python library used for working with data sets.
It has functions for analyzing, cleaning, exploring, and manipulating data.

Aim

To create a dataset in python using pandas.

Algorithm

1. Start
2. Import pandas using a variable 'pd'
3. Create a dataset named mydataset
4. Print the value by calling dataframe
5. stop

Coding

```
import pandas as pd
mydataset = {
    'cars': ["BMW", "Volvo", "Ford"],
    'passings': [3, 7, 2]
}
myvar = pd.DataFrame(mydataset)
print(myvar)
```

Output

```
cars passings
0  BMW      3
1  Volvo    7
2  Ford     2
```

Result

Thus the program for creating a dataset in python using pandas was executed successfully.

Ex No : 8B

NumPy

NumPy is a Python library used for working with arrays.

It also has functions for working in domain of linear algebra, fourier transform, and matrices.

Aim

To create an array in python using numpy.

Algorithm

1. Start
2. Import numpy using a variable 'np'
3. Create aarray with method np.array
4. Print the value of array and it's type
5. stop

Coding

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print(arr)
print(type(arr))
```

Output

```
[1 2 3 4 5]
<class 'numpy.ndarray'>
```

Result

Thus the program for creating anarray in python using numpy was executed successfully.

Ex No : 8C

Matplotlib

Matplotlib is a low level graph plotting library in python that serves as a visualization utility.

Matplotlib is open source and we can use it freely.

Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility.

Aim

To Draw a line in a diagram from position (0,0) to position (6,250) using Matplotlib in python.

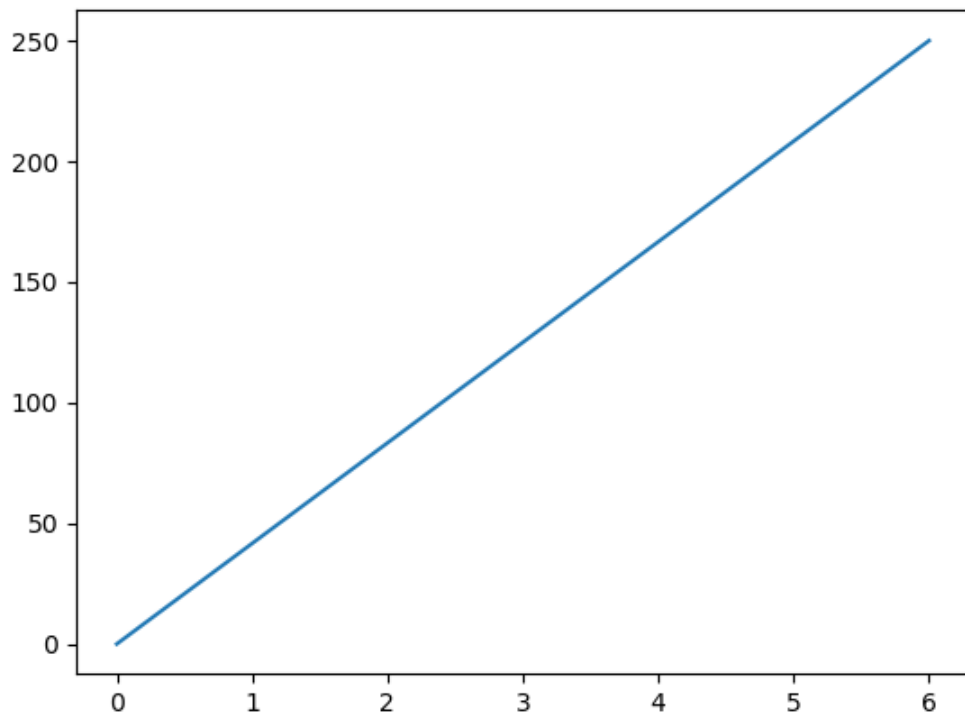
Algorithm

1. Start
2. Import matplotlib, numpy and sys packages in the program
3. Declare the points for x and y using numpy
4. Plot the points by using the method "plot"
5. stop

Coding

```
import sys
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np
xpoints = np.array([0, 6])
ypoints = np.array([0, 250])
plt.plot(xpoints, ypoints)
plt.show()
plt.savefig(sys.stdout.buffer)
sys.stdout.flush()
```

Output



Result

Thus the program for drawing a line in a diagram from position (0,0) to position (6,250) using Matplotlib in python was executed successfully.

Ex No : 8D

SCIPY

SciPy is a scientific computation library that uses NumPy underneath.

SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing.

Aim

To Create a Triangulation from the given points using Scipy in python.

Algorithm

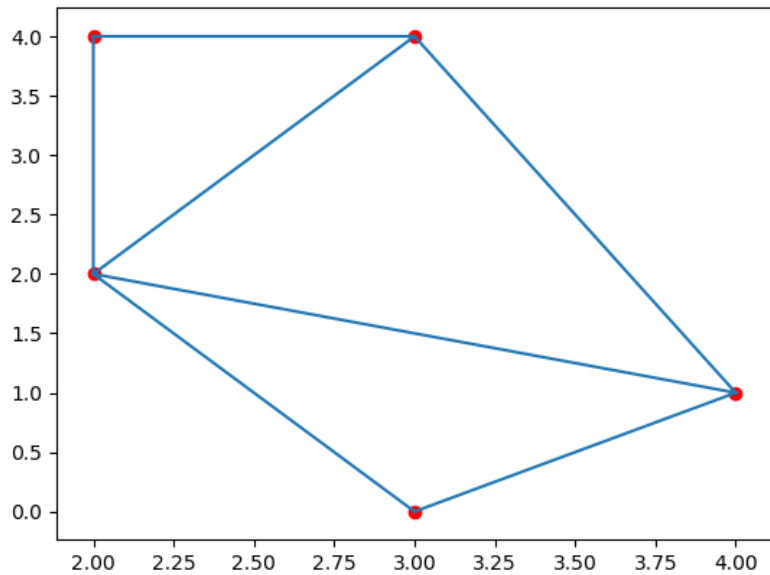
1. Start
2. Import Scipy, matplotlib, numpy and sys packages in the program
3. Declare the points for triangle by array initialization using matplotlib.pyplot
4. Plot the points by using the method triplot and scatter
5. stop

Coding

```
import sys
import matplotlib
matplotlib.use('Agg')
import numpy as np
from scipy.spatial import Delaunay
import matplotlib.pyplot as plt
points = np.array([
    [2, 4],
    [3, 4],
    [3, 0],
    [2, 2],
    [4, 1]
])
simplices = Delaunay(points).simplices
plt.triplot(points[:, 0], points[:, 1], simplices)
```

```
plt.scatter(points[:, 0], points[:, 1], color='r')  
plt.show()  
plt.savefig(sys.stdout.buffer)  
sys.stdout.flush()
```

Output



Result

Thus the program for Creating a Triangulation from the given points using Scipy in python. was executed successfully.

Ex.No:9 Implementing real-time/technical applications using File handling. (copy from one file to another, word count, longest word)

Ex.no.9.A: Python program to copy from one file to another

Aim

To write a program for copying the contents from one file to another using Python.

Algorithm

1. Start
2. Open the file in write mode
3. Write the contents to the file name file1.txt using f as file object
4. Read each line from the file object f
5. Copy and Write the text in text files file2 in object file f1 from file1
6. Read each line from the file name file2
7. Open the text file file2 in read mode
8. Print the file contents file2.txt copied from file1.txt
9. Close the file
10. Stop

Coding

```
f = open("file1.txt", "wt")
f.write("Python is an interpreted high-level general-purpose programming language.")
with open("file1.txt") as f:
    with open("file2.txt", "w") as f1:
        for line in f:
            f1.write(line)
f2=open("file2.txt","rt")
print("File copied successfully.The contents are:\n",f2.read())
f2.close()
```

Output

File copied successfully.The contents are:

Python is an interpreted high-level general-purpose programming language.

Result

Thus the program for copying the contents from one file to another using Python was successfully executed.

Ex.no.9.b: Python program to count words in a file

Aim

To write a program for counting the words in a file using Python.

Algorithm

1. Start
2. Open the text file in read mode
3. Read each line from the file object f
4. Split the word using split() function and assign to the variable ,words
5. Count the number of words using len() function
6. Print the total number of words read from the file
7. Close the file
8. Stop

Coding

```
f = open("demofile.txt", "rt")  
  
data = f.read()  
  
words = data.split()  
  
print("The total number of words are:",len(words))  
  
f.close()  
  
print('Number of words in text file :', len(words))
```

Output

('The total number of words are:', 2)

('Number of words in text file :', 2)

Result

Thus the program for counting the words in a file using Python was successfully executed.

Ex.no.9.c: Python program to find longest word in a file

Aim

To write a program to find the longest word in a file using Python.

Algorithm

1. Start
2. Open the text file in read mode
3. Read each line from the file object fin
4. Split the word using split() function and assign to the variable ,words
5. Count the number of words using len() function
6. Using max(),find the maximum length of word by finding the length of each words
7. Print the longest word read from the file
8. Close the file
9. Stop

Coding

```
fin = open("demofile.txt","r")
str = fin.read()
words = str.split()
max_len = len(max(words, key=len))
for word in words:
    if len(word)==max_len:
longest_word =word
print(longest_word)
```

Output

hello

Result

Thus the program for finding the longest word in a file using Python was successfully executed.

Ex.No:10.Implementing real-time/technical applications using Exception handling. (divide by zero error, voter's age validity, student mark range validation)

Ex.No:10.a. Python program to handle divide by zero error using Exceptional handling

Aim

To write a program to handle divide by zero error using Exceptional handling in Python.

Algorithm

1. Start
2. Use Try block
3. Get the input values for a and b
4. Calculate $c=a/b$
5. Use the exceptvalueerror and print wrong data
6. Use the Zerodivisionerror and print zero error division
7. Print the result C.

Coding

try :

```
a=int(input("Enter value of a: "))
```

```
b=int(input("Enter value of b: "))
```

```
c=a/b
```

except ValueError:

```
print("You have entered wrong data")
```

except ZeroDivisionError:

```
print("Divide by Zero Error!!!")
```

else:

```
print("The result: ",c)
```

Output

Enter value of a: 45

Enter value of b: 32

('The result: ', 1)

Result

Thus the program to handle divide by zero error using Exceptional handling in Python was successfully executed.

Ex.No:10.b. Python program to voter's age validity using Exceptional handling

Aim

To write a program to find voter's age validity using Exceptional handling in Python.

Algorithm

1. Start
2. Get the input values for age
3. Use if condition and check the age is above 18 or not
4. Use the except keyword and print the statement
5. Stop

Coding

try:

```
age=int(input("Enter your age"))
```

```
if age>18:
```

```
    print("Eligible to vote")
```

```
else:
```

```
    print("Not eligible to vote")
```

except:

```
    print("age must be a valid number")
```

Output

```
Enter your age 12
```

```
Not eligible to vote
```

Result

Thus the program to find voter's age validity using Exceptional handling in Python was successfully executed.

Ex.No:10.c. Python program to validate student mark range using Exceptional handling

Aim

To write a program to validate student mark range using Exceptional handling in Python.

Algorithm

1. Start
2. Get the input value for mark
3. Use if condition and check the mark is above 0 or less than 100
4. Use the except keyword and print the statement
5. Stop

Coding

try:

```
mark=int(input("Enter your mark"))
```

```
if mark>0 and mark<=100:
```

```
    print("Valid Mark")
```

```
else:
```

```
    print("Invalid mark")
```

except:

```
    print("Mark must be a valid number")
```

Output

Enter your mark 34

Valid Mark

Result

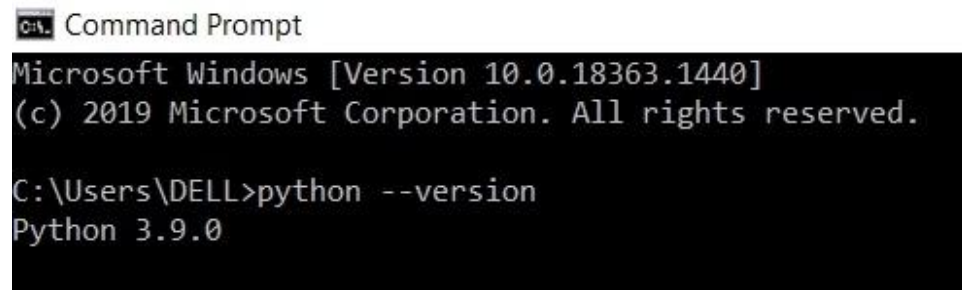
Thus, the program to validate student mark range using Exceptional handling in Python was successfully executed.

Ex.No:11 Exploring Pygame

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language.

Step 1: Check for Python Installation

In order to install Pygame, Python must be installed already in your system. To check whether Python is installed or not in your system, open the command prompt and give the command as shown below.



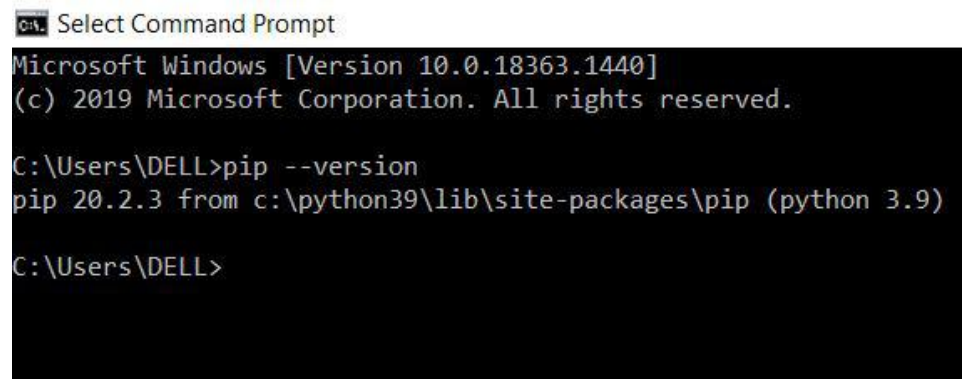
```
C:\> Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\DELL>python --version
Python 3.9.0
```

If this command runs successfully, and we are able to get a Python version then we are good to go.

Step 2: Check for PIP installation

PIP is a tool that is used to install python packages. PIP is automatically installed with Python 2.7.9+ and Python 3.4+. Open the command prompt and enter the command shown below to check whether pip is installed or not.



```
C:\> Select Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\DELL>pip --version
pip 20.2.3 from c:\python39\lib\site-packages\pip (python 3.9)

C:\Users\DELL>
```

Step 3: Install Pygame

To install Pygame, open the command prompt and give the command as shown below:

```
pip install pygame
```

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\DELL>python --version
Python 3.9.0

C:\Users\DELL>pip --version
pip 20.2.3 from c:\python39\lib\site-packages\pip (python 3.9)

C:\Users\DELL>pip install pygame
Collecting pygame
  Downloading pygame-2.0.1-cp39-cp39-win_amd64.whl (5.2 MB)
    |#####| 5.2 MB 148 kB/s
Installing collected packages: pygame
Successfully installed pygame-2.0.1
WARNING: You are using pip version 20.2.3; however, version 21.0.1 is available.
You should consider upgrading via the 'c:\python39\python.exe -m pip install --upgrade pip' command.

C:\Users\DELL>
```

Pygame is successfully installed as shown in the image above.

Step 4: Check Whether PyGame is Working or not

Now open a new terminal and import the Pygame library to see whether it is working fine or not in our system. The library is imported successfully means we got success.

```
Command Prompt - python
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\DELL>python
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import pygame
pygame 2.0.1 (SDL 2.0.14, Python 3.9.0)
Hello from the pygame community. https://www.pygame.org/contribute.html
>>> █
```

In this way, we can install the pygame module in Python.

Ex.NO:12 Developing a game activity using Pygame like bouncing ball.

Aim

To write a program to develop a game activity using Pygame like bouncing ball in python.

Algorithm

6. Start
7. Get the input value for mark
8. Use if condition and check the mark is above 0 or less than 100
9. Use the except keyword and print the statement
10. Stop

Coding

```
import sys, pygame

pygame.init()

size = width, height = 800, 400

speed = [1, 1]

background = 255, 255, 255

screen = pygame.display.set_mode(size)

pygame.display.set_caption("Bouncing ball")

ball = pygame.image.load("D://bb.png")

ballrect = ball.get_rect()

while 1:

    for event in pygame.event.get():

        if event.type == pygame.QUIT:

            sys.exit()

            ballrect = ballrect.move(speed)

    if ballrect.left < 0 or ballrect.right > width:

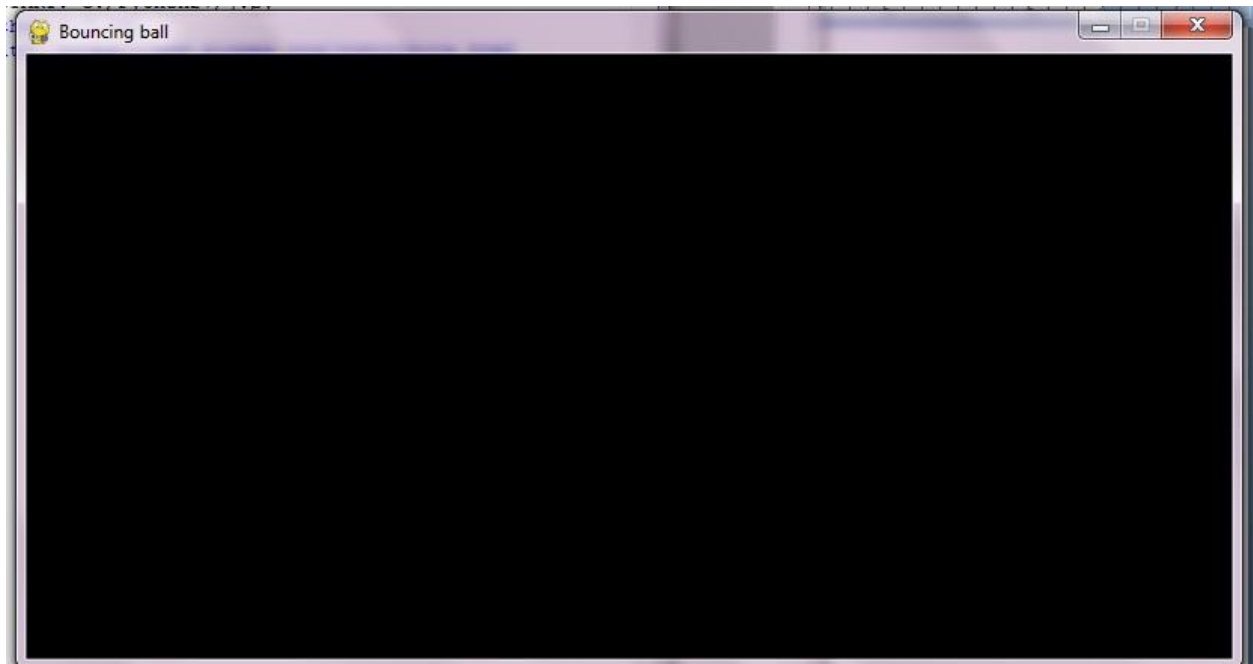
        speed[0] = -speed[0]

    if ballrect.top < 0 or ballrect.bottom > height:

        speed[1] = -speed[1]
```

```
screen.fill(background)
screen.blit(ball, ballrect)
pygame.display.flip()
```

Output



Result

Thus, the program to develop a game activity using Pygame like bouncing ball in python was successfully executed.