from scipy import linalg

from scipy import special

import numpy as np

help(linalg)

#To get information about any function

# We are trying to solve a linear algebra system which can be given as

#         x + 3y +10z = 10

#         2x + 12y + 7z = 18

#         5x + 8y + 8z = 30

'''a = np.array([[1, 3, 10], [2, 12, 7], [5, 8, 8]])# Creating input array

b = np.array([[10], [18], [30]])# Solution Array

x = linalg.solve(a, b)# Solve the linear algebra

print(x)# Print results'''

# To find Determinent of matrix

'''A = np.array([[5,9],[8,4]])

x = linalg.det(A)  #Passing the values to the det function

print(x)  #printing the result'''

'''l, v = linalg.eig(A)# To calculate eigen values and eigenvectors

print(l)#printing the result for eigenvalues

print(v)#printing the result for eigenvectors'''

# Make use of special functions

a = special.exp10(3)

print(a)

b = special.exp2(3)

print(b)

c = special.sindg(90)

print(c)

d = special.cosdg(45)

print(d)

from matplotlib import pyplot as plt

from matplotlib import style

import numpy as np

# To plot line

plt.plot([1,2,3],[4,5,1])  #Plotting to our canvas

plt.show() #Showing what we plotted

x = [5,2,7]

y = [2,16,4]

plt.plot(x,y)

plt.title('Info')

plt.ylabel('Y axis')

plt.xlabel('X axis')

plt.show()

#to plot line using style

style.use('ggplot')

x = [5,8,10]

y = [12,16,6]

x2 = [6,9,11]

y2 = [6,15,7]

plt.plot(x,y,'g',label='line one', linewidth=5)

plt.plot(x2,y2,'c',label='line two',linewidth=2)

plt.title('Epic Info')

plt.ylabel('Y axis')

plt.xlabel('X axis')

plt.legend()

plt.grid(True,color='k')

plt.show()

#to plot bar

plt.bar([0.25,1.25,2.25,3.25,4.25],[50,40,70,80,20],label="BMW",width=0.5)

plt.bar([.75,1.75,2.75,3.75,4.75],[80,20,20,50,60],label="Audi", color='r',width=0.5)

plt.legend()

plt.xlabel('Days')

plt.ylabel('Distance (kms)')

plt.title('Information')

plt.show()

#To plot histogram

population\_age = [22,55,62,45,21,22,34,42,42,4,2,102,95,85,55,110,120,70,65,55,111,115,80,75,65,54,44,43,42,48]

bins = [0,10,20,30,40,50,60,70,80,90,100]

plt.hist(population\_age, bins, histtype='stepfilled', rwidth=0.8)#histtype value can be step,barstacked,stepfilled

plt.xlabel('age groups')

plt.ylabel('Number of people')

plt.title('Histogram')

plt.show()

#To plot Scaater grap

x = [1,1.5,2,2.5,3,3.5,3.6]

y = [7.5,8,8.5,9,9.5,10,10.5]

x1=[8,8.5,9,9.5,10,10.5,11]

y1=[3,3.5,3.7,4,4.5,5,5.2]

plt.scatter(x,y, label='high income low saving',color='r')

plt.scatter(x1,y1,label='low income high savings',color='b')

plt.xlabel('saving\*100')

plt.ylabel('income\*1000')

plt.title('Scatter Plot')

plt.legend()

plt.show()

#To plot piechart

y = np.array([35, 25, 25, 25])

mylabels = ["Apples", "Bananas", "Cherries", "Dates"]

myexplode = [0.2, 0, 0.4, 0]

plt.pie(y,labels=mylabels,explode=myexplode,autopct='%1.1f%%')

plt.show()

import pandas as pd

import numpy as np

data = np.array(['a','b','c','d'])

s=pd.Series(data)

print(s)

print(s[1])

data1= {

"calories": [420, 380, 390],

"duration": [50, 40, 45]

}

df = pd.DataFrame(data1)#load data into dataframe object

print(df)

print(df.loc[1])#to locate sepecified row0

# Reading CSV File

df = pd.read\_csv('cars.csv')

#print(df)#print reduced sample

#print(df.to\_string())#to print entire dataframe

print(df.dtypes)#give type of each column

#print(df.head(10))#give first five row

#print(df.tail())#give last five row

#print(df.to\_numpy())#Convert each row into array

print(df.describe())#shows you quick statistic summary of your data

#print(df.sort\_index(axis=1,ascending=True))#This method with parameter axis=1 means sorts column names and ascending=True means sorts the rows in ascending order

print(df.info())

left = pd.DataFrame({

'id':[1,2,3,4,5],

'Name': ['Alex', 'Amy', 'Allen', 'Alice', 'Ayoung'],

'subject\_id':['sub1','sub2','sub4','sub6','sub5']})

right = pd.DataFrame(

{'id':[1,2,3,4,5],

'Name': ['Billy', 'Brian', 'Bran', 'Bryce', 'Betty'],

'subject\_id':['sub2','sub4','sub3','sub6','sub5']})

print (left)

print (right)

print (pd.merge(left,right,on='id'))#merge data on id

print (pd.merge(left,right,on=['id','subject\_id']))#merge on multiple keys

print (pd.merge(left, right, on='subject\_id', how='left'))#perform left outer join on data

print (pd.merge(left, right, on='subject\_id', how='right'))#perform right outer join

print (pd.merge(left, right, on='subject\_id', how='outer'))#perform outer join

print (pd.merge(left, right, on='subject\_id', how='inner'))#perform inner join

from tkinter import \*

import mysql.connector

conn = mysql.connector.connect(user='root', password='root', host='localhost', database='mydb')

cursor=conn.cursor()

class First:

def \_\_init\_\_(self,root):

self.f=Frame(root,height=500,width=500)

self.f.pack()

self.l1=Label(text='First Name :')

self.l2=Label(text='Last Name :')

self.l3=Label(text='Age')

self.l4=Label(text='Tel No:')

self.e1=Entry(self.f,width=18)

self.e2=Entry(self.f,width=18)

self.e3=Entry(self.f,width=18)

self.e4=Entry(self.f,width=18)

self.b1=Button(self.f,text="Submit",command=self.display)

self.l1.place(x=50,y=30)

self.e1.place(x=200,y=30)

self.l2.place(x=50,y=60)

self.e2.place(x=200,y=60)

self.l3.place(x=50,y=90)

self.e3.place(x=200,y=90)

self.l4.place(x=50,y=130)

self.e4.place(x=200,y=130)

self.b1.place(x=200,y=150)

def display(self):

str1=self.e1.get()

str2=self.e2.get()

str3=int(self.e3.get())

str4=int(self.e4.get())

sql="Insert into EMPLOYEE1(FIRST\_NAME,LAST\_NAME,AGE,TEL\_NO)values('%s','%s','%d','%d')"%(str1,str2,str3,str4)

try:

cursor.execute(sql)

conn.commit()

except:

conn.rollback()

print("Data Inserted")

conn.close()

self.next\_win()

def next\_win(self):

root.destroy()

import next\_win

root=Tk()

root.title("Welcome To Tkinter GUI Programme")

obj=First(root)

root.mainloop()

from tkinter import \*

import mysql.connector

conn = mysql.connector.connect(user='root', password='root', host='localhost', database='mydb')

cursor=conn.cursor()

class next:

def \_\_init\_\_(self,root):

self.f=Frame(root,height=500,width=500)

self.f.pack()

self.b1=Button(self.f,text="Show",command=self.display)

self.b1.place(x=200,y=150)

def display(self):

print ("----------View records-----------")

#sql = "SELECT \* FROM EMPLOYEE1 limit 0,10"

cursor.execute("SELECT \* FROM EMPLOYEE1 limit 0,10")

i=0

try:

for d in cursor:

for j in range(len(d)):

e=Entry(self.f,width=10,fg='blue')

e.grid(row=i,column=j)

e.insert(END,d[j])

i=i+1

except:

print ("Error: unable to fecth data")

root=Tk()

root.title("Welcome To Next Window")

obj=next(root)root.mainloop()

from tkinter import \*

class First:

def \_\_init\_\_(self,root):

self.f=Frame(root,height=500,width=500)

self.f.pack()

self.l1=Label(text='First Name :')

self.l2=Label(text='Last Name :')

self.l3=Label(text='Age')

self.l4=Label(text='Tel No:')

self.e1=Entry(self.f,width=18)

self.e2=Entry(self.f,width=18)

self.e3=Entry(self.f,width=18)

self.e4=Entry(self.f,width=18)

self.b1=Button(self.f,text="Submit",command=self.display)

self.l1.place(x=50,y=30)

self.e1.place(x=200,y=30)

self.l2.place(x=50,y=60)

self.e2.place(x=200,y=60)

self.l3.place(x=50,y=90)

self.e3.place(x=200,y=90)

self.l4.place(x=50,y=130)

self.e4.place(x=200,y=130)

self.b1.place(x=200,y=150)

def display(self):

str1=self.e1.get()

str2=self.e2.get()

str3=(self.e3.get())

str4=(self.e4.get())

l5=Label(text='Name :'+str1+" "+str2).place(x=50,y=220)

l6=Label(text='Age :'+str3).place(x=50,y=250)

l7=Label(text='Tel No :'+str4).place(x=50,y=280)

root=Tk()

root.title("Welcome To Tkinter GUI Programme")

obj=First(root)

root.mainloop()

from tkinter import \*

from PIL import ImageTk,Image

root=Tk()

root.title("My First Tkinter Application") #Giving a title to bar

c=Canvas(root,bg="orange",height=700, width=800)

id=c.create\_line(80,80,200,80,200,200,width=6,fill="cyan3")

id=c.create\_line(80,80,300,80,300,300,width=6,fill="cyan3")

id=c.create\_oval(500,300,200,200,width=4,fill="yellow",outline="pink")

id=c.create\_oval(120,120,400,300,width=6,fill="red",outline="cornsilk1",activefill="green")#Drawing a Oval

fnt=('Times',22,'bold italic underline') #Setting Font

id=c.create\_text(250,40,text="My first Tkinter Application on canvas",font=fnt,fill="red",activefill="green") #Drawing a Text

id=c.create\_polygon(320,320,320,420,420,320,width=6,fill="yellow",outline="cornsilk2",activefill="lightblue")#Drawing Polygon

id=c.create\_rectangle(360,460,550,550,width=6,fill="brown",outline="cornsilk3",activefill="chocolate")#Drawing Rectangle

id=c.create\_arc(500,100,600,300,width=6,start=0,extent=180,outline="white",style="arc")#Drawing arc

file1=ImageTk.PhotoImage(Image.open("Cat1.jpg"),height=200,width=200)

file2=ImageTk.PhotoImage(Image.open("dog1.jpg"),height=700,width=800)

#id=c.create\_image(500,500,anchor=NE,image=file2)

id=c.create\_image(700,800,anchor=CENTER,image=file2,activeimage=file1)#Drawing Image

c.pack()

root.mainloop()

import pickle,studentpickle

print ("\*\*\*\*\*\*WRITE file\*\*\*\*\*\*\*\*\*\*\*\*")

f=open('myfile.txt','w') #Create file if not exist and open it for writting

s1=input('Enter Your Text to be written in to file: ')

f.write(s1) #write text into file

f.close() #Close file

print("")

print ("\*\*\*\*\*\*Read file \*\*\*\*\*\*\*\*\*\*\*\*")

f1=open('myfile.txt','r') #open file for reading

print ("Contents of file are")

s=f1.read()

print (s)

f1.close()#Close file

print("")

print ("\*\*\*\*\*\*Append file\*\*\*\*\*\*\*\*\*\*\*\*")

f=open('myfile.txt','a') #open file for Appending

s2=input('Enter text to be appended in original file..')

f.write(s2)

f.close() #Close file

print("")

print ("")

print ("\*\*\*\*\*\*Read Appended file\*\*\*\*\*\*\*\*\*\*\*\*")

f1=open('myfile.txt','r') #Create file if not exist and open it for reading

s=f1.read()

print (s)

f.close()

print ("\*\*\*\*\*\*WITH OPEN\*\*\*\*\*\*\*\*\*\*\*\*")

with open('sample.txt','w') as f:

s3=input('Enter text for WITHOPEN write..')

f.write(s3)

with open('sample.txt','r') as f:

for line in f:

print (line)

print ("\*\*\*\*\*\*PICKLE\*\*\*\*\*\*\*\*\*\*\*\*")

f3=open('studentpickle.dat','wb')

n=int(input('How many students?? '))

for i in range (n):

roll=int(input("Enter your roll no : "))

name=input("Enter your name : ")

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

s=studentpickle.student(roll,name,age)

pickle.dump(s,f3)

f3.close()

########## StudentPickle File################

class student:

def \_\_init\_\_(self,roll,name,age):

self.roll=roll

self.name=name

self.age=age

def display(self):

print("{:5d}{:20s}{:10.2f}".format(self.roll,self.name,self.age))

i=1

while i<=5:

    n=int(input("Please enter numbers between 1 to 5 to see diffrent Exceptions : "))

    if n==1:

        try:

            a=int(input("Please enter number a : "))

            b=int(input("Please enter number b (put b=0): "))

            c=a/b

        excep        print ("Existing The Program")

        exit()

i+=1t ZeroDivisionError:

            print("Oops! Number Divisible by Zero Exception Occurs.")

        else:

            print ("Division is",c)

    elif n==2:

        try:

            a=int(input("Please enter number a : "))

            b=int(input("Please enter number b (put b='a'): "))

            c=a/b

        except ValueError:

            print("Oops! Value Error Exception Occurs.Please enter a valid number.")

        else:

            print ("Division is",c)

    elif n==3:

        try:

            a=int(input("Please enter number a : "))

            b=int(input("Please enter number b : "))

            c=k/b

        except NameError:

            print("Oops! Name Error Exception Occurs due to c=k/b (k is not defined ).Please enter a valid variable number.")

    elif n==4:

        try:

            r='2'+2

        except TypeError:

            print("Oops! Type Error Exception Occurs (due to '2'+2).Please Provide Valid data type. ")

    elif n==5:

        try:

            n=int(input("Please enter Numbers between 2 to 3: (Check for other nos) "))

            assert n>=2 and n<=3

            print("The Number Entered is",n)

        except AssertionError:

            print("Oops! Assertion Error Occurs..Please enter number between 2 to 5.")

    else: