#Roll No: 19H61A3528

#Date: 25/08/2021

#Variables and Arithmetic

print()

a=3

a

b=10

a+2

mike=17

print(mike\*b+1)

print(mike\*\*b)

b=0

print(mike\*b+1)

print(mike\*\*b)

#STRING VARIABLES

greeting = 'Hello. My name is '

print(greeting)

name='mike'

print(greeting + name)

print(greeting + name + '!')

age = 10

print('I am' + str(age) + 'Years old.')

#ASSIGNMENT

print('My name is ' + name + ', I am now ' + str(age) + ' years old, and in 6 years I\'ll be ' + str(age+2) + ' years old.')

#The NUMPY MODULE

numlist = [1,2,3,4,4]

print(numlist)

# importing a module

import numpy as np

print(np.mean(numlist))

print(np.sqrt(4))

print(np.linspace(1,10,7))

print(np.linspace(1,10,12))

funout = np.linspace(1,10,12)

print(funout + 2)

#ASSIGNMENT

print(np.linspace(4,100,15))

arr = np.round(np.linspace(4,100,15))

print(arr)

arr1 = np.round(np.linspace(4,100,15),2)

print(arr1)

arr=np.sqrt(arr)

print('square root of list is ' + str(arr))

# MATPLOTLIB MODULE

import matplotlib.pyplot as plt #importing a matplotlib module

import numpy as np

plt.plot(1,3)

plt.plot(1,3,'ko')

plt.show()

x=np.arange(-9,10)

print(x)

y=x\*\*2

plt.plot(x,y,'r')

plt.show()

plt.plot(x,y/2,'gs')

plt.show()

#creating a line

plt.plot([0,3],[-1,1])

plt.show()

#creating more than on line

plt.plot([0,3],[-1,1],label='first line')

plt.plot([-2,0],[-4,1],label='second line')

plt.legend()

plt.show()

#matrix

M=np.random.randint(0,10,size=(4,5))

print(M)

plt.imshow(M)#plt.impshow() used to represent the matrix with colors

plt.show()

#SCALAR AND VECTOR MULTIPLICATIONS

vec=[3,4,5,2] #creating a vector

s = 2

print(vec\*s)

s=3

print(vec\*s)

vec=np.array([3,4,5,2])

s=3

print(vec\*s)

vec2d = np.array([1,2])

s1=2

s2=0.5

s3=-1

print(vec2d[0])

plt.plot([0,vec2d[0]],[0,vec2d[1]],'bs-')

plt.show

plt.plot([0,vec2d[0]],[0,vec2d[1]],'bs-',label='v')

plt.plot([0,s1\*vec2d[0]],[0,s1\*vec2d[1]],'ro-',label='v\*s1')

plt.plot([0,s2\*vec2d[0]],[0,s2\*vec2d[1]],'kp-',label='v\*s2')

plt.plot([0,s3\*vec2d[0]],[0,s3\*vec2d[1]],'g\*-',label='v\*s3')

plt.axis('square')

plt.xlim([-4,4])

plt.ylim([-4,4])

plt.grid()

plt.legend()

plt.show()

#VECTOR DOT PRODUCT

v1=np.array([4,5,6,2])

v2=np.array([3,0,5])

print(np.dot(v1,v2))

v1=np.array([5,6,2])

v2=np.array([3,0,5])

print(np.dot(v1,v2))

v3=np.array([-4,3,1])

print(np.dot(v2,v3))

v4=np.array([0,0,0])

print(np.dot(v2,v4))

v3=np.array([-4,3,1])

v5=np.array([2,3,-1])

print(np.dot(v3,v5))

#ASSIGNMENT

#1)

v1=np.array([5,9])

v2=np.array([-18,10])

v3=np.array([1,2])

print(np.dot(v1,v2))

print(np.dot(v2,v3))

print(np.dot(v1,v3))

#2)

plt.plot([0,v1[0]],[0,v1[1]],'bs-',label='v1')

plt.plot([0,v2[0]],[0,v2[1]],'ro-',label='v2')

plt.plot([0,v3[0]],[0,v3[1]],'ko-',label='v3')

plt.axis('square')

plt.xlim([-10,10])

plt.ylim([-10,10])

plt.grid()

plt.legend()

plt.show()

#MATRICES

print(np.eye(3))

print(np.zeros((3,4)))

print(np.full((5,2),7))

M= np.array([[1,2,3],[4,5,6],[7,8,9]])

print(M)

#ASSIGNMENT

M1=np.random.randn(2,2)

M2=np.random.randn(2,2)

M3=np.random.randn(3,2)

print(M1)

v1= np.array([2,3,-1])

print(v1)

v1= np.array([2,3,-1],ndmin=2)#ndmin is used to transpose a matrix

print(v1)

print(' ')

print(v1.T)

M=np.random.randn(3,3)#random 3\*3 matrix

print(M)

M=np.round(np.random.randn(3,3))

print(M)

print(' ')

print(M.T)

#ASSIGNMENT

#1)

print(M.T)

print(M.T.T)#twice transposing

#2)

M=np.round(np.random.randn(3,4))

print(M)

print(' ')

print(M.T)

#MATRIX MULTIPLICATION

M1=np.random.randn(4,5)

M2=np.random.randn(4,5)

#these 2 matrices cannote be multipied since the condition is not satisfied

print(np.matmul(M1,M2.T))

print(' ')

print(M1@M2.T)

print(np.matmul(M1,M2.T) - M1@M2.T)

#ASSIGNMENT

M= np.array([[1,2,3],[4,5,6],[7,8,9]])

print(M)

print(np.matmul(M,np.eye(3)))

print(np.matmul(M,np.zeros(3)))

print(np.matmul(M,M.T))

#MATRIX INVERSION

A=np.random.randn(4,4)#4\*4 random matrix

print(A)

A=np.random.randn(4,4)

Ainv= np.linalg.inv(A)

AinvA = A@Ainv

print(A)

print(' ')

print(Ainv)

print(' ')

print(A)

print(AinvA)

fig,ax = plt.subplots(1,3,figsize=(6,5))

ax[0].imshow(A)

ax[1].imshow(Ainv)

ax[1].set\_title('A$^{-1}$')

ax[2].imshow(AinvA)

ax[2].set\_title('A$^{-1}$A')

plt.show()