**Programas da aula 09 –**

import numpy as np

vetor = np.array(((1., 0., 0.),(0.,1.,2.)), dtype=complex)

print(vetor)

import numpy as np

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

print(np.ones((2,3,4), dtype = np.int16))

print(np.empty((2,3)))

print(np.arange(10,30,5))

print(np.arange(0,2,0.3),'\n')

print(np.random.random(15))

print(np.random.random(15).reshape(3,5))

import numpy as np

from numpy import pi

#x = np.linspace(0,2,9)

x = np.linspace(0,2\*pi,100)

f = np.sin(x)

print(x)

print(f)

import numpy as np

a = np.array([20, 30, 40, 50])

b = np.arange(4)

print(a)

print(b)

c = a-b

print(c)

print(b\*\*2)

print(10\*np.sin(a))

print(a<35)

print(a-35)

print(35-a)

import numpy as np

A = np.array([[1,1],

             [0,1]])

B = np.array([[2,0],

              [3,4]])

print(A)

print(B)

print(A\*B)

print(A-B)

print(A+B)

print(A@B)

print(A.dot(B))

import numpy as np

a = np.ones((2,3), dtype = int)

b = np.random.random((2,3))

print(a);print(b)

a \*= 3

print(a)

#Slide

import numpy as np

array = np.array([[1., 0., 0], [0., 1., 2.]])

print(array)

print(type(array))

import numpy as np

array\_1 = np.arange(15).reshape(3, 5)

print(array\_1)

print(array\_1.shape)

print(array\_1.ndim)

print(array\_1.dtype.name)

print(array\_1.itemsize)

print(array\_1.size)

print(type(array\_1))

array\_2 = np.array([6, 7, 8])

print(array\_2)

print(type(array\_2))

import numpy as np

array\_1 = np.array([2,3,4])

print(array\_1)

print(array\_1.dtype, "\n")

array\_2 = np.array([1.2, 3.5, 5.1])

print(array\_2)

print(array\_2.dtype)

import numpy as np

array\_1 = np.array([1,2,3,4])

array\_2 = np.array([(1.5,2,3), (4,5,6)])

print(array\_2, "\n")

array\_3 = np.array( [ [1,2], [3,4] ], dtype=complex )

print(array\_3, "\n")

import numpy as np

print(np.zeros( (3,4) ))

print(np.ones( (2,3,4), dtype=np.int16 ))

print(np.empty( (2,3) ))

print(np.arange( 10, 30, 5))

print(np.arange(0, 2, 0.3), "\n")

import numpy as np

from numpy import pi

print(np.linspace(0, 2, 9))

x = np.linspace(0,2\*pi, 100)

print(x, "\n")

f = np.sin(x)

print(f, "\n")

import numpy as np

array\_1 = np.arange(6)

print(array\_1, "\n")

array\_2 = np.arange(12).reshape(4,3)

print(array\_2, "\n")

array\_3 = np.arange(24).reshape(2,3,4)

print(array\_3, "\n")

import numpy as np

print(np.arange(10000),"\n")

print(np.arange(10000).reshape(100,100), "\n")

import numpy as np

a = np.array( [20,30,40,50] )

b = np.arange( 4 )

print(b)

c = a-b

print(c)

print(b\*\*2)

print(10\*np.sin(a))

print(a<35)

import numpy as np

A = np.array( [[1,1],

               [0,1]])

B = np.array( [[2,0],

               [3,4]])

print(A \* B)

print(A @ B)

print(A.dot(B))

import numpy as np

a = np.ones((2,3), dtype=int)

b = np.random.random((2,3))

a \*= 3

print(a)

b += a

print(b)

import numpy as np

from numpy import pi

a = np.ones(3, dtype=np.int32)

b = np.linspace(0, pi, 3)

print(b.dtype.name)

c = a+b

print(c)

print(c.dtype.name)

d = np.exp(c\*1j)

print(d)

print(d.dtype.name)

import numpy as np

a = np.random.random((2,3))

print("a =", a, "\n")

print("sum a = ",a.sum(), "\n")

print("Min a = ", a.min(), "\n")

print("Max a = ", a.max(), "\n")

b = np.arange(12).reshape(3,4)

print("b = ",b,"\n")

print("sum b = ",b.sum(axis=0), "\n")

print("Min b = ",b.min(axis=1), "\n")

print("cumsum b = ", b.cumsum(axis=1), "\n")

import numpy as np

B = np.arange(3)

print("B = ", B, "\n")

print("ex B = ", np.exp(B),"\n")

print("sqrt B = ", np.sqrt(B),"\n")

C= np.array([2., -1., 4.])

print("add B,C =", np.add(B, C), "\n")

import numpy as np

a = np.arange(10)\*\*3

print("a =  ",a,"\n")

print("a[2] =  ",a[2],"\n")

print("a[2:5] =  ",a[2:5],"\n")

a[:6:2]= -1000

print("a =  ",a,"\n")

print("a[: :-1] =  ",a[ : :-1],"\n")

for i in a:

    print(i\*\*(1/3.))

import numpy as np

def f(x,y):

    return 10\*x+y

b = np.fromfunction(f,(5,4), dtype=int)

print("b = ", b, "\n")

print("b[2,3] = ", b[2,3], "\n")

print("b[0:5, 1] = ", b[0:5, 1], "\n")

print("b[ : ,1] = ", b[ : ,1], "\n")

print("b[1:3, :] = ", b[1:3, : ], "\n")

import numpy as np

c = np.array( [[[  0,  1,  2],

                [ 10,  12, 13]],

                [[100,101,102],

                [110,112,113]]])

print("c[-1] = ", c[-1], "\n")

print("c.shape = ",c.shape,"\n")

print("c[1,...] = ", c[1,...], "\n")

print("c[...,2] = ", c[...,2], "\n")

for row in c:

    print(row)

for element in c.flat:

    print(element)

import numpy as np

a = np.floor(10\*np.random.random((3,4)))

print("a =",a,"\n")

print("a.shape = ", a.shape,"\n")

print("a.ravel() =", a.ravel(), "\n")

print("a.reshape(6,2) = ",a.reshape(6,2),"\n")

print("a.T = ", a.T,"\n")

print("a.T.shape = ",a.T.shape,"\n")

print("a.shape = ", a.shape, "\n")

import numpy as np

a = np.floor(10\*np.random.random((3,4)))

print("a =", a,"\n")

print("a.resizee((2,6)) = ", a.resize((2,6)),"\n")

print("a =", a,"\n")

print("a.resjape(3,-1) = ", a.reshape(3,-1),"\n")

import numpy as np

from numpy import newaxis

a = np.floor(10\*np.random.random((2,2)))

print("a = ",a,"\n")

b = np.floor(10\*np.random.random((2,2)))

print("b = ",b,"\n")

print("vstack((a,b)) = ", np.vstack((a,b)),"\n")

print("hstack((a,b)) = ", np.hstack((a,b)),"\n")

print("a = ",np.column\_stack((a,b)),"\n")

a = np.array([4.,2.])

b = np.array([3.,8.])

print("column\_stack((a,b)) =", np.column\_stack((a,b)),"\n")

print("hstack((a,b)) = ", np.hstack((a,b)),"\n")

print("a[:,newaxis] = ", a[:,newaxis],"\n")

print("column\_stack((a[:,newaxis],b[:,newaxis])) = ", np.column\_stack((a[:,newaxis],b[:,newaxis])),"\n")

print("hstack((a[:,newaxis],b[:,newaxis])) =", np.hstack((a[:,newaxis],b[:,newaxis])),"\n")

import numpy as np

a = np.floor(10\*np.random.random((2,12)))

print("a = ", a,"\n")

print("hslit(a,3) = ", np.hsplit(a,3),"\n")

print("hsplit(a,(3,4)) = ", np.hsplit(a,(3,4)), "\n")

import numpy as np

a = np.arange(12)

b = a

print(b is a)

b.shape = 3,4

print(a.shape)

def f(x):

    print(id(x))

print(id(a))

f(a)

import numpy as np

a = np.arange(12)

c = a.view()

print(c is a, "\n")

print(c.base is a, "\n")

print(c.flags.owndata, "\n")

c.shape = 2,6

print(a.shape, "\n")

c[0,4] = 1234

print(a,"\n")

s = a[1:3]

s[:] = 10

print(a,"\n")

import numpy as np

a = np.arange(12)

d = a.copy()

print(d is a,"\n")

print(d.base is a, "\n")

d[0] = 9999

print(a, "\n")

a = np.arange(int(1e8))

b = a[:100].copy()

del a