**Numpy on jupyter Notebook:**

a. Import the "numpy" library as "np".

Import numpy as np

×

. . .

b. Create an array of shape (2, 3, 4) of zeros.

a = np.zeros((2,3,4))

print(a)

×

. . .

c. Create an array of shape (2, 3, 4) of ones

a = np.ones((2,3,4))

print(a)

×

. . .

d. Create an array with values 0 to 999 using the "np.arange" function

a = np.arange(0,1000)

print(a)

×

e. Create an array from the list [2, 3.2, 5.5, -6.4, -2.2, 2.4] and assign it to the variable "a"

a = np.array([2,3.2,5.5,-6.4,-2.2,2.4])

print(a)

×

f. Do you know what a[1] will equal? Print it to see

print(a[1])

×

g. Try printing a[1:4] to see what that equals

print(a[1:4])

×

h. Create a 2-D array from the following list and assign it to the variable "a":

[[2, 3.2, 5.5, -6.4, -2.2, 2.4],

[1, 22, 4, 0.1, 5.3, -9],

[3, 1, 2.1, 21, 1.1, -2]]

b = np.array([[2,3.2,5.5,-6.4,-2.2,2.4],[1,22,4,0.1,5.3,-9],[3,1,2.1,21,1.1,-2]])

print(b)

×

i. Can you guess what the following slices are equal to? Print them to check your understanding.

a[:, 3]

a[1:4, 0:4]

a[1:, 2]

print(b[:,3])

print(b[1:3,0:4])

print(b[1:,2])

×

j. Create a 2-D array of shape (2, 4) containing two lists (range(4), range(10, 14))

and assign it to the variable "arr".Print the shape of the array. Print the size of the array. Print the maximum and minimum of the array

arr = np.array([range(0,4),range(10,14)])

print(arr)

print("Shape: ",arr.shape)

print("size: ",arr.size)

×

. . .

print("Maximum Value: ",np.max(arr))

×

. . .

print("Minimum Value: ",np.min(arr))

×

k. Continue to use the array "arr" as defined above.Print the array re-shaped to (2, 2, 2).Print the array transposed.Print the array flattened to a single dimension. Print the array converted to floats.

arr = arr.reshape((2,2,2))

print(np.transpose(arr))

print(arr.astype(float))

×

l. Create an an array counting from 1 to 20 inclusive

a = np. arange(1,21)

print(a)

×

m. The array of multiples of 3 greater than 0 and less than 30

c = np.arange(9)+1

c = c\*3

print(c)

×

n. The array of 8 equally spaced floats x where 0 ≤ x ≤ 1

d = np.linspace(0,1,8)

print(d)

×

o. Use np.arange and reshape to create the array A = [[1 2 3 4] [5 6 7 8]]

a = np.arange(8)+1

A = a.reshape(2,4)

print(A)

×

p. Use np.array to create the array B = [1 2]

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

print(B)

×

q. Use broadcasting to add B to A to create the final array A + B

a = np.arange(8)+1

A = a.reshape(2,4)

print(A)

×

. . .

B = 1

c = A+B

print(c)×