1. Create a 4X2 integer array and Prints its attributes

Attributes like:

1. The shape of an array.-2
2. Array dimensions.
3. The Length of each element of the array in bytes.

import numpy

firstArray = numpy.empty([4,2])

print(firstArray)

print("Printing numpy array Attributes")

print("1> Array Shape is: ", firstArray.shape)

print("2>. Array dimensions are ", firstArray.ndim)

print("3>. Length of each element of array in bytes is ", firstArray.itemsize)

[[0.00000000e+000 0.00000000e+000]

[0.00000000e+000 0.00000000e+000]

[0.00000000e+000 1.29049947e-320]

[1.22439157e-311 0.00000000e+000]]

Printing numpy array Attributes

1> Array Shape is: (4, 2)

2>. Array dimensions are 2

3>. Length of each element of array in bytes is 8

1. Create a 5X2 integer array from a range between 100 to 200 such that the difference between each element is 10

import numpy as np

array=np.arange(100,200,10,)

array=array.reshape(5,2)

print(array)

[[100 110]

[120 130]

[140 150]

[160 170]

[180 190]]

#### Add the following two NumPy arrays

import numpy

arrayOne = numpy.array([[5, 6, 9], [21 ,18, 27]])

arrayTwo = numpy.array([[15 ,33, 24], [4 ,7, 1]])

array=arrayOne + arrayTwo

print(array)

[[20 39 33]

[25 25 28]]

1. Create an 8X3 integer array from a range between 10 to 34 such that the difference between each element is 1 and then Split the array into four equal-sized sub-arrays.

Expected Output:

Creating 8X3 array using numpy.arange

[[10 11 12]

[13 14 15]

[16 17 18]

[19 20 21]

[22 23 24]

[25 26 27]

[28 29 30]

[31 32 33]]

Dividing 8X3 array into 4 sub array

[array([[10, 11, 12],[13, 14, 15]]),

array([[16, 17, 18],[19, 20, 21]]),

array([[22, 23, 24],[25, 26, 27]]),

array([[28, 29, 30],[31, 32, 33]])]

import numpy as np

array=np.arange(10,34,1)

abc=array.reshape(8,3)

print(abc)

[[10 11 12]

[13 14 15]

[16 17 18]

[19 20 21]

[22 23 24]

[25 26 27]

[28 29 30]

[31 32 33]]

array1=np.split(array,4)

print(array1)

[array([[10, 11, 12], [13, 14, 15]]),

array([[16, 17, 18],[19, 20, 21]]),

array([[22, 23, 24], [25, 26, 27]]),

array([[28, 29, 30], [31, 32, 33]])]

4Write a NumPy program to generate six random integers between 10 and 30.   
Expected Output:  
[20 28 27 17 28 29]

import numpy as np

x=np.random.randint(10,20,size=6)

print(x)

[10 19 16 15 18 11]

1. Write a NumPy program to generate five random numbers from the normal distribution.

Expected Output:  
[-0.43262625 -1.10836787 1.80791413 0.69287463 -0.53742101]

import numpy as np

x=np.random.normal(size=5)

print(x)

[-3.18711194 1.98747462 -0.82592802 -1.140237 0.42366513]

1. Write a NumPy program to create a 3x3x3 array with random values.

import numpy as np

x=np.random.random((3,3,3))

print(x)

[[[0.87401066 0.60584219 0.3771377 ]

[0.57654124 0.77664094 0.01634945]

[0.51445166 0.08741763 0.07741953]]

[[0.93221872 0.53808224 0.52584985]

[0.5394492 0.76430677 0.71036074]

[0.10913014 0.08231621 0.62397936]]

[[0.54190253 0.17806571 0.74332471]

[0.18844405 0.8031751 0.62036467]

[0.69816501 0.70287123 0.59605866]]]

1. Write a NumPy program to normalize a 3x3 random matrix.

import numpy as np

x= np.random.random((3,3))

print("Original Array:")

print(x)

xmax, xmin = x.max(), x.min()

x = (x - xmin)/(xmax - xmin)

print("After normalization:")

print(x)

Original Array:

[[0.05227822 0.5825975 0.41694171]

[0.21512085 0.10683812 0.08274312]

[0.70388057 0.67105072 0.21164193]]

After normalization:

[[0. 0.8138695 0.55964115]

[0.24991105 0.08373188 0.04675382]

[1. 0.94961674 0.24457202]]

1. Write a NumPy program to shuffle numbers between 0 and 10 (inclusive).

import numpy as np

x = np.arange(0,10)

np.random.shuffle(x)

print(x)

[3 9 5 1 0 4 8 7 6 2]

1. Write a NumPy program to create a 5x5 array with random values and find the minimum and maximum values.

import numpy as np

x=np.random.random((5,5))

print(x)

print(x.max())

print(x.min())

[[0.52850301 0.87705212 0.57729788 0.1461655 0.26653388]

[0.48504007 0.59452141 0.94809335 0.40705196 0.43124761]

[0.55347584 0.26384682 0.95432951 0.70350933 0.43054281]

[0.84084875 0.99240577 0.6318883 0.19660499 0.88085561]

[0.56710083 0.78991276 0.21967527 0.58196157 0.98320027]]

0.9919564163014233

0.026676838486655363