# Create an array of 10 zeros

**import** numpy **as** np arr**=**np**.**zeros(10) arr

In [1]:

Out[1]:

array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])

# Create an array of 10 ones

arr**=**np**.**ones(10) arr

In [2]:

Out[2]:

In [4]:

arr**=**np**.**ones(10)**\***5 arr

array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])

Out[4]:

array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])

# Create an array of the integer from 10 to 50

**for** i **in** range(10,51): array **=** np**.**array(i) print(array,end**=**" ")

In [8]:

10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

# Create an array of all the even interger from 10 to 50

**for** i **in** range(10,51):

**if**(i**%2**==0):

array**=**np**.**array(i) print(array,end**=**" ")

In [10]:

10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50

# Create a 3X3 matfrix with values ranging form 0 to 8

array**=**np**.**arange(0,9)**.**reshape(3,3) print(array)

In [12]:

[[0 1 2]

[3 4 5]

[6 7 8]]

# Create 3X3 identify matrix

array**=**np**.**eye(3) array

In [15]:

Out[15]:

array([[1., 0., 0.],

[0., 1., 0.],

[0., 0., 1.]])

# use numpy to generate a random number 0 and 1

array **=** np**.**random**.**randint(0,1) array

In [16]:

Out[16]: 0

# use numpy to generate an array of 25 random numbers sampled from a standard normal distribution

**import** numpy **as** np array**=**np**.**random**.**rand(5,5) array

In [17]:

Out[17]:

|  |  |  |  |
| --- | --- | --- | --- |
| array([[0.74256594, 0.37246014, | 0.37282874, | 0.98365189, | 0.72580023], |
| [0.13950812, 0.0317423 , | 0.86388023, | 0.21167192, | 0.99822987], |
| [0.31732952, 0.36014321, | 0.69775346, | 0.57950007, | 0.29345113], |
| [0.83493934, 0.15723232, | 0.59184564, | 0.22546064, | 0.41096483], |
| [0.66702403, 0.66266544, | 0.65503963, | 0.01393382, | 0.0250321 ]]) |

# Createb a following matrix

array **=** np**.**linspace(0,1,100) array **=** array**.**reshape(10,10) array

In [20]:

Out[20]:

array([[0. , 0.01010101, 0.02020202, 0.03030303, 0.04040404,

# Create an array of 20 linearly spaced points between o and 1

array**=**np**.**linspace(0,1,20) array

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0.05050505, | 0.06060606, | 0.07070707, | 0.08080808, | 0.09090909], |
| [0.1010101 , | 0.11111111, | 0.12121212, | 0.13131313, | 0.14141414, |
| 0.15151515, | 0.16161616, | 0.17171717, | 0.18181818, | 0.19191919], |
| [0.2020202 , | 0.21212121, | 0.22222222, | 0.23232323, | 0.24242424, |
| 0.25252525, | 0.26262626, | 0.27272727, | 0.28282828, | 0.29292929], |
| [0.3030303 , | 0.31313131, | 0.32323232, | 0.33333333, | 0.34343434, |
| 0.35353535, | 0.36363636, | 0.37373737, | 0.38383838, | 0.39393939], |
| [0.4040404 , | 0.41414141, | 0.42424242, | 0.43434343, | 0.44444444, |
| 0.45454545, | 0.46464646, | 0.47474747, | 0.48484848, | 0.49494949], |
| [0.50505051, | 0.51515152, | 0.52525253, | 0.53535354, | 0.54545455, |
| 0.55555556, | 0.56565657, | 0.57575758, | 0.58585859, | 0.5959596 ], |
| [0.60606061, | 0.61616162, | 0.62626263, | 0.63636364, | 0.64646465, |
| 0.65656566, | 0.66666667, | 0.67676768, | 0.68686869, | 0.6969697 ], |
| [0.70707071, | 0.71717172, | 0.72727273, | 0.73737374, | 0.74747475, |
| 0.75757576, | 0.76767677, | 0.77777778, | 0.78787879, | 0.7979798 ], |
| [0.80808081, | 0.81818182, | 0.82828283, | 0.83838384, | 0.84848485, |
| 0.85858586, | 0.86868687, | 0.87878788, | 0.88888889, | 0.8989899 ], |
| [0.90909091, | 0.91919192, | 0.92929293, | 0.93939394, | 0.94949495, |
| 0.95959596, | 0.96969697, | 0.97979798, | 0.98989899, | 1. ]]) |

In [21]:

Out[21]:

array([0. , 0.05263158, 0.10526316, 0.15789474, 0.21052632,

0.26315789, 0.31578947, 0.36842105, 0.42105263, 0.47368421,

0.52631579, 0.57894737, 0.63157895, 0.68421053, 0.73684211,

0.78947368, 0.84210526, 0.89473684, 0.94736842, 1. ])

# Create now will be given matrix, and asked to be replicate the resulting matrix

mat**=**np**.**arange(1,26)**.**reshape(5,5) mat

In [24]:

Out[24]:

In [25]:

Out[25]:

In [26]:

Out[26]:

In [27]:

Out[27]:

array([[ 1, 2, 3, 4, 5],

[ 6, 7, 8, 9, 10],

[11, 12, 13, 14, 15],

[16, 17, 18, 19, 20],

[21, 22, 23, 24, 25]])

mat[2:,1:]

array([[12, 13, 14, 15],

[17, 18, 19, 20],

[22, 23, 24, 25]])

mat[0:3,1:2]

array([[ 2],

[ 7],

[12]])

mat[4:5,0:5]

array([[21, 22, 23, 24, 25]])

# Get the sum all values in matrix

x**=**np**.**sum(mat) x

In [28]:

Out[28]:

325

# Get the standard deviation of the values in mat

x**=**np**.**std(mat) x

In [29]:

Out[29]:

7.211102550927978

# Get the sum of all coulmns in matrix

mat**=**mat**.**sum(axis**=**0) mat

In [30]:

Out[30]:

In [ ]:

array([55, 60, 65, 70, 75])