import numpy as np  
arr = np.array([1, 2, 3, 4, 5])  
print(arr)  
print(type(arr))

Practice : Slicing:

` 1

Trignometric function:

# Python code to demonstrate trigonometric function

import numpy as np

# create an array of angles

angles = np.array([0, 30, 45, 60, 90, 180])

# conversion of degree into radians

# using deg2rad function

radians = np.deg2rad(angles)

# sine of angles

print('Sine of angles in the array:')

sine\_value = np.sin(radians)

print(np.sin(radians))

# inverse sine of sine values

print('Inverse Sine of sine values:')

print(np.rad2deg(np.arcsin(sine\_value)))

# hyperbolic sine of angles

print('Sine hyperbolic of angles in the array:')

sineh\_value = np.sinh(radians)

print(np.sinh(radians))

# inverse sine hyperbolic

print('Inverse Sine hyperbolic:')

print(np.sin(sineh\_value))

# hypot function demonstration

base = 4

height = 3

print('hypotenuse of right triangle is:')

print(np.hypot(base, height))

Statistical:

# Python code demonstrate statistical function

import numpy as np

# construct a weight array

weight = np.array([50.7, 52.5, 50, 58, 55.63, 73.25, 49.5, 45])

# minimum and maximum

print('Minimum and maximum weight of the students: ')

print(np.amin(weight), np.amax(weight))

# range of weight i.e. max weight-min weight

print('Range of the weight of the students: ')

print(np.ptp(weight))

# percentile

print('Weight below which 70 % student fall: ')

print(np.percentile(weight, 70))

# mean

print('Mean weight of the students: ')

print(np.mean(weight))

# median

print('Median weight of the students: ')

print(np.median(weight))

# standard deviation

print('Standard deviation of weight of the students: ')

print(np.std(weight))

# variance

print('Variance of weight of the students: ')

print(np.var(weight))

# average

print('Average weight of the students: ')

print(np.average(weight))

Bit:

# Python code to demonstrate bitwise-function

import numpy as np

# construct an array of even and odd numbers

even = np.array([0, 2, 4, 6, 8, 16, 32])

odd = np.array([1, 3, 5, 7, 9, 17, 33])

# bitwise\_and

print('bitwise\_and of two arrays: ')

print(np.bitwise\_and(even, odd))

# bitwise\_or

print('bitwise\_or of two arrays: ')

print(np.bitwise\_or(even, odd))

# bitwise\_xor

print('bitwise\_xor of two arrays: ')

print(np.bitwise\_xor(even, odd))

# invert or not

print('inversion of even no. array: ')

print(np.invert(even))

# left\_shift

print('left\_shift of even no. array: ')

print(np.left\_shift(even, 1))

# right\_shift

print('right\_shift of even no. array: ')

print(np.right\_shift(even, 1))