

```

import numpy as np

a=np.array([1,2,3,4,5]) #to create array
print(a)

[1 2 3 4 5]

print(f"array a:{a}") #formating
array a:[1 2 3 4 5]

print(f"ndim:{a.ndim}") #dimension of array
print(f"size:{a.size}") #give size of array
print(f"shape:{a.shape}") #shpae of array
print(f"dtype:{a.dtype}") #datatype of array
print(f"itemsize:{a.itemsize}") #give size of single element

ndim:1
size:5
shape:(5,)
dtype:int32
itemsize:4

b=np.array([[1,2,3],[4,5,6]]) #create two dimension array
print(b)

[[1 2 3]
 [4 5 6]]

print(f"ndim:{b.ndim}")
print(f"size:{b.size}")
print(f"shape:{b.shape}")
print(f"dtype:{b.dtype}")
print(f"itemsize:{b.itemsize}")

ndim:2
size:6
shape:(2, 3)
dtype:int32
itemsize:4

c=np.array([1,2,3,4,5])
d=np.add(a,c) #add data of array
e=np.divide(a,c) #divide data of array
f=np.absolute(a,c) #absolute value of data
g=np.subtract(a,c) #subtract data
h=np.remainder(a,c) #give reminder value
i=np.power(a,c) #power of data
j=np.mod(a,c) #mode of data
print(d)
print(e)
print(f)

```

```

print(g)
print(h)
print(i)
print(j)

[ 2  4  6  8 10]
[1. 1. 1. 1. 1.]
[1 2 3 4 5]
[0 0 0 0 0]
[0 0 0 0 0]
[  1  4 27 256 3125]
[0 0 0 0 0]

a=np.array(['dhruva','dhyey','devansi'])
c=np.array(['madhavi','remansi','aayush'])
print(f"ndim:{a.ndim}")
print(f"size:{a.size}")
print(f"shape:{a.shape}")
print(f"dtype:{a.dtype}")
print(f"itemsize:{a.itemsize}")

ndim:1
size:3
shape:(3,)
dtype:<U7
itemsize:28

z=np.zeros((2,2)) #create 2x2 array with zeros
print(z)
z=np.ones((2,4)) #create 2x4 array with all value as 1
print(z)
z=np.empty((2,2)) #create 2x2 array with empty value
print(z)
z=np.arange(2,28,2) #create array between lower and upper limit
print(z)
z=np.linspace(1,5,3) #crete 3 element array between 1 to 5
print(z)

[[0. 0.]
 [0. 0.]]
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]]
[[0. 0.]
 [0. 0.]]
[ 2  4  6  8 10 12 14 16 18 20 22 24 26]
[1. 3. 5.]

a=np.array([[2,3,4],[6,7,8]])
print(a)
sliced_arr=a[1,2] #slice array
print(sliced_arr)

```

```

sliced_arr=a[:2,::2]
print(sliced_arr)

[[2 3 4]
 [6 7 8]]
8
[[2 4]
 [6 8]]

i=a[[1,0],[0,1]] #create array based on given index
print(i)

[6 3]

print(a.sum()) #add all element of array
print(np.sqrt(a)) #find sqrt of data

30
[[1.41421356 1.73205081 2.          ]
 [2.44948974 2.64575131 2.82842712]]

b=a+6 #add specific value in existing vale
print(b)

[[ 8  9 10]
 [12 13 14]]

m1=np.array([[1,2,3],[4,5,6],[7,8,9]])
m2=np.array([[2,3,4],[5,6,7],[8,9,2]])
m3=np.dot(m1,m2) #multiply two matrix
print(m3)

[[ 36  42  24]
 [ 81  96  63]
 [126 150 102]]

print(np.all(m1))
bool=np.array([[True,True],[True,False]])
print(np.all(bool)) #use to check and condition
print(np.all(bool,axis=0)) #check condition row wise
print(np.all(bool,axis=1)) #check condition column wise

True
False
[ True False]
[ True False]

bool=np.array([[True,True],[True,False]])
print(np.any(bool)) #use to check or condition
print(np.any(bool,axis=0)) #check condition row wise
print(np.any(bool,axis=1)) #check condition column wise

```

```

True
[ True  True]
[ True  True]

a=np.arange(9)
print(a)
re=a.reshape(3,3) #to reshape the existing array
print(re)

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

print(m1)
print(m2)
print(m3)

[[1 2 3]
 [4 5 6]
 [7 8 9]]
[[2 3 4]
 [5 6 7]
 [8 9 2]]
[[ 36  42  24]
 [ 81  96  63]
 [126 150 102]]

h=np.vstack((m1,m2,m3)) #merge array vertically
print(h)
h=np.hstack((m1,m2,m3)) #merge array horizontally
print(h)

[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [ 2  3  4]
 [ 5  6  7]
 [ 8  9  2]
 [ 36 42 24]
 [ 81 96 63]
 [126 150 102]]
[[ 1  2  3  2  3  4 36 42 24]
 [ 4  5  6  5  6  7 81 96 63]
 [ 7  8  9  8  9  2 126 150 102]]

hsplit=np.hsplit(h,3)#split array vertically
print(hsplit)
vsplit=np.vsplit(h,3)#split array horizontally
print(vsplit)

```

```

[array([[1, 2, 3],
        [4, 5, 6],
        [7, 8, 9]]), array([[2, 3, 4],
        [5, 6, 7],
        [8, 9, 2]]), array([[ 36,  42,  24],
        [ 81,  96,  63],
        [126, 150, 102]])]
[array([[ 1,  2,  3,  2,  3,  4, 36, 42, 24]]), array([[ 4,  5,  6,
 5,  6,  7, 81, 96, 63]]), array([[ 7,  8,  9,  8,  9,  2, 126,
150, 102]])]

print("sqrt:", np.sqrt(b))
print("sin:", np.sin(b))
print("cos:", np.cos(b))
print("log:", np.log(b))
print("sum:", np.sum(b))
print("std:", np.std(b))

sqrt: [[2.82842712 3.         3.16227766]
 [3.46410162 3.60555128 3.74165739]]
sin: [[ 0.98935825  0.41211849 -0.54402111]
 [-0.53657292  0.42016704  0.99060736]]
cos: [[-0.14550003 -0.91113026 -0.83907153]
 [ 0.84385396  0.90744678  0.13673722]]
log: [[2.07944154 2.19722458 2.30258509]
 [2.48490665 2.56494936 2.63905733]]
sum: 66
std: 2.160246899469287

```