In [43]:

```
import numpy as np
 #Array
a=np.array([1,2,3])
print("Array :" , a)
```

```
Array : [1 2 3]
```

In [46]:

```
b=np.arange(0,10,2) #create an array with evenly space values
print("Array b:", b)
```

```
Array b: [0 2 4 6 8]
```

In [47]:

```
d=np.zeros((2,3))
print("Arrays d:\n",d) #craete an array filled with zeros
```

```
Arrays d:
 [[0. 0. 0.]
 [0. 0. 0.]]
```

In [49]:

```
e=np.ones((3,2), dtype=int)
print("Arrays d:\n",e) #craete an array filled with ones
```

```
Arrays d:
 [[1 1]
 [1 1]
 [1 1]]
```

In [56]:

```
f=np.eye(4) #Create identity matrix
print("Identity matrix:\n", f)
```

```
Identity matrix:
 [[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]
```

Array manipulation functions

Reshape array

In [58]:

```
a1=np.array([1,2,3])
reshaped=np.reshape(a1, (1,3))
print("Reshaped array :", reshaped)
```

```
Reshaped array : [[1 2 3]]
```

In [ ]:

```
#Flatten an array
```

In [22]:

```
f1=np.array([[1,2],[3,4]])
flattened=np.ravel(f1)
print('flattened', flattened)
```

```
flattened [1 2 3 4]
```

In [ ]:

```
#tranpose array
```

In [59]:

```
e1=np.array([[1,2],[3,4]])
transposed=np.transpose(e1)
```

In [ ]:

```
[[1,2]
 [2,4]]
```

In [61]:

```
#Stacked array vertically
a2=np.array([1,2])
b2=np.array([3,4])
stacked=np.vstack([a2,b2]) #hstack for horizontal
print("stacked arrays :\n", stacked)
```

```
stacked arrays :
 [[1 2]
 [3 4]]
```

In [62]:

```
#Stacked array vertically
a2=np.array([1,2])
b2=np.array([3,4])
stacked=np.hstack([a2,b2]) #hstack for horizontal
print("stacked arrays :\n", stacked)
```

```
stacked arrays :
 [1 2 3 4]
```

Mathematical function

In [63]:

```
g=np.array([1,2,3,4]) #Add 2 arrays
added=np.add(g,2)
print("added 2 g: ", added)
```

```
added 2 g:  [3 4 5 6]
```

In [25]:

```
#Squared 2 elements
squared =np.power(g,2)
print("squareg g: ", squared)
```

```
square  [ 1  4  9 16]
```

In [26]:

```
print(a1)
print(g)
```

```
[1 2 3]
[1 2 3 4]
```

In [27]:

```
print(a1)
print(a)
```

```
[1 2 3]
[1 2 3]
```

In [28]:

```
dot_product=np.dot(a1,a)
print("dot product",dot_product)
```

```
dot product 14
```

In [ ]:

```
#statistics
```

In [65]:

```
#Mean
s=np.array([1,2,3,4])
mean=np.mean(s)
print("mean of array:",mean)
```

```
mean of array: 2.5
```

In [64]:

```
#Standrad deviation of an array
std_dev=np.std(s)
print("stanard deviation:", std_dev)
```

```
stanard deviation: 1.118033988749895
```

In [32]:

```
#Minimun elemnet of an array
minimum=np.min(s)
print(minimum)
```

```
1
```

In [33]:

```
#maximum value
maximum=np.max(s)
print("maximum" , maximum)
```

```
maximum 4
```

In [35]:

```
#Random sampling
random=np.random.rand(3)
print('rand', random)
```

```
rand [0.78978651 0.50590872 0.23298738]
```

In [37]:

```
np.random.seed(0)
random_vals=np.random.rand(3)
print("Random values:" , random_vals)
```

```
random values [0.5488135  0.71518937 0.60276338]
```

In [66]:

```
#Random integers
rand_ints=np.random.randint(0,10, size=5)
print("Random Integers:" ,rand_ints)
```

```
Random Integers: [5 0 3 3 7]
```

In [41]:

```
#boolean and logical
logical_test=np.array([True,False,True])
all_true=np.all(logical_test)
print("elelemnts:",all_true)
```

```
elelemnts: False
```

In [42]:

```
set_a=np.array([1,2,3,4])
set_b=np.array([3,4,5,6])
intersection=np.intersect1d(set_a,set_b)
print(intersection)
```

```
[3 4]
```

In [67]:

```
#Union of 2 arrays
union=np.union1d(set_a,set_b)
print("union of a and b", union)
```

```
union of a and b [1 2 3 4 5 6]
```

In [ ]: