In [1]:
```python
import numpy as nm
a = nm.array([1,2,3,4,5,6,7,8,9])
print(a)
```
```
[1 2 3 4 5 6 7 8 9]
```

In [2]:
```python
import numpy as np

print(np.__version__)
```
```
1.20.3
```

In [3]:
```python
print(type(a))
```
```
<class 'numpy.ndarray'>
```

In [10]:
```python
# zero dimensional array
a = np.array(42)
print(a)
```
```
42
```

In [7]:
```python
# one dimensional array
a  =np.array([1,2,3,4,5])
print(a)
```
```
[1 2 3 4 5]
```

In [8]:
```python
# two dimensional array
b = np.array([[1,2,3],[4,5,6]])
print(b)
```
```
[[1 2 3]
 [4 5 6]]
```

In [9]:
```python
# three dimensional array
c = np.array([[[1,2,3],[4,5,6],[1,2,3],[4,5,6]]])
print(c)
```
```
[[[1 2 3]
  [4 5 6]
  [1 2 3]
  [4 5 6]]]
```

In [11]:
```python
#checking no. of dimensions
a = np.array(42)
b = np.array([1,2,3,4,5])
c = np.array([[1,2,3],[4,5,6]])
d = np.array([[[1,2,3],[4,5,6]],[[1,2,3],[4,5,6]]])

print(a.ndim)
print(b.ndim)
print(c.ndim)
print(d.ndim)
```
```
0
1
2
3
```

In [13]:
```python
# higher dimensional arrays
e = np.array([1,2,3,4], ndmin=5)
print(e)
print('number of dimensions:',e.ndim)
```
```
[[[[[1 2 3 4]]]]]
number of dimensions: 5
```

In [14]:
```python
# access array elements
e = np.array([1,2,3,4])
print(e[0])
```
```
1
```

In [15]:
```python
e = np.array([1,2,3,4])
print(e[1])
```
```
2
```

In [16]:
```python
e = np.array([1,2,3,4])
print(e[2] + e[3])
```
```
7
```

In [17]:
```python
# access 2d arrays
f = np.array([[1,2,3,4,5],[6,7,8,9,10]])
print('2nd element on 1st row:', f[0,1])
```
```
2nd element on 1st row: 2
```

In [18]:
```python
f = np.array([[1,2,3,4,5],[6,7,8,9,10]])
print('5th element on 2nd row:', f[1,4])
```
```
5th element on 2nd row: 10
```

In [19]:
```python
g = np.array([[[1,2,3],[4,5,6]],[[7,8,9],[10,11,12]]])
print(g[0,1,2])
```
```
6
```

In [20]:
```python
# negative indexing
f = np.array([[1,2,3,4,5],[6,7,8,9,10]])
print('Last element from 2nd dim:', f[1,-1])
```
```
Last element from 2nd dim: 10
```

In [21]:
```python
# slicing
h = np.array([1,2,3,4,5,6,7])
print(h[1:5])
```
```
[2 3 4 5]
```

In [22]:
```python
print(h[4:])
```
```
[5 6 7]
```

In [23]:
```python
print(h[:4])
```
```
[1 2 3 4]
```

In [24]:
```python
print(h[-3:-1])
```
```
[5 6]
```

In [25]:
```python
print(h[1:5:2])
```
```
[2 4]
```

In [26]:
```python
print(h[::2])
```
```
[1 3 5 7]
```

In [27]:
```python
# slicing 2d arrays
i = np.array([[1,2,3,4,5],[6,7,8,9,10]])
print(i[1, 1:4])
```
```
[7 8 9]
```

In [28]:
```python
print(i[0:2, 2])
```
```
[3 8]
```

In [29]:
```python
print(i[0:2, 1:4])
```
```
[[2 3 4]
 [7 8 9]]
```

In [30]:
```python
# shape of an array
j = np.array([[1,2,3,4],[5,6,7,8]])
print(j.shape)
```
```
(2, 4)
```

In [31]:
```python
k = np.array([1,2,3,4], ndmin=5)
print(k)
print('shape of array :', k.shape)
```
```
[[[[[1 2 3 4]]]]]
shape of array : (1, 1, 1, 1, 4)
```

In [32]:
```python
# reshaping arrays from 1d to 2d
l = np.array([1,2,3,4,5,6,7,8,9,10,11,12])
newl = l.reshape(4,3)
print(newl)
```
```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

In [33]:
```python
# reshaping from 1d to 3d
newll = l.reshape(2,3,2)
print(newll)
```
```
[[[ 1  2]
  [ 3  4]
  [ 5  6]]

 [[ 7  8]
  [ 9 10]
  [11 12]]]
```

In [34]:
```python
# linspace
l = np.linspace(2,15,5)
print(l)
```
```
[ 2.    5.25  8.5  11.75 15.  ]
```

In [36]:
```python
# ravel
m = np.array([[1,2],[3,4]])
print(m)
n = m.ravel()
print(n)
```
```
[[1 2]
 [3 4]]
[1 2 3 4]
```

In [37]:
```python
# mathematical functions
o = np.array([1,2,3,4,5])
print(o.min())
print(o.max())
print(o.sum())
```
```
1
5
15
```

In [38]:
```python
# matrix of random numbers
p = np.random.random((3,4))
print(p)
```
```
[[0.30421695 0.15993897 0.09386098 0.66303771]
 [0.54187961 0.41972818 0.9768206  0.20095328]
 [0.38243525 0.42431232 0.96905612 0.99571246]]
```

In [41]:
```python
q = np.floor(p)
print(q)
```
```
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

In [42]:
```python
r = np.ceil(p)
print(r)
```
```
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
```

In [43]:
```python
s = np.std(p)
print(s)
```
```
0.31015192790136464
```

In [44]:
```python
# vertical and horizontal stacking
x = np.array([(1,2,3),(3,4,5)])
y = np.array([(6,7,8),(9,10,11)])
print(np.vstack((x,y)))
```
```
[[ 1  2  3]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

In [45]:
```python
print(np.hstack((x,y)))
```
```
[[ 1  2  3  6  7  8]
 [ 3  4  5  9 10 11]]
```

In [46]:
```python
# splitting
arr = np.array([1,2,3,4,5,6])
newarr = np.array_split(arr,3)
print(newarr)
```
```
[array([1, 2]), array([3, 4]), array([5, 6])]
```

In [47]:
```python
print(newarr[0])
print(newarr[1])
print(newarr[2])
```
```
[1 2]
[3 4]
[5 6]
```

In [48]:
```python
# splitting 2d arrays
arrr = np.array([[1,2],[3,4],[5,6],[7,8],[9,10],[11,12]])
newarrr = np.array_split(arrr,3)
print(newarrr)
```
```
[array([[1, 2],
       [3, 4]]), array([[5, 6],
       [7, 8]]), array([[ 9, 10],
       [11, 12]])]
```

In [50]:
```python
bar = np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12],[13,14,15],[16,17,18
newbar = np.array_split(bar,3)
print(newbar)
```
```
[array([[1, 2, 3],
       [4, 5, 6]]), array([[ 7,  8,  9],
       [10, 11, 12]]), array([[13, 14, 15],
       [16, 17, 18]])]
```

In [51]:
```python
newbar = np.array_split(bar, 3,axis=1)
print(newbar)
```
```
[array([[ 1],
       [ 4],
       [ 7],
       [10],
       [13],
       [16]]), array([[ 2],
       [ 5],
       [ 8],
       [11],
       [14],
       [17]]), array([[ 3],
       [ 6],
       [ 9],
       [12],
       [15],
       [18]])]
```

In [52]:
```python
newbar = np.hsplit(arr,3)
print(newbar)
```
```
[array([1, 2]), array([3, 4]), array([5, 6])]
```

In [53]:
```python
newbar = np.resize(bar,(2,5))
print(newbar)
```
```
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]]
```

In [54]:
```python
newbar = np.append(a,[5,6])
print(newbar)
```
```
[42  5  6]
```

In [55]:
```python
# unique
newbar = np.unique(bar)
print(newbar)
```
```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18]
```

In [56]:
```python
# insert
newbar = np.array([[1,1,1],[2,2,2],[3,3,3],[4,4,4]])
print(newbar)
```
```
[[1 1 1]
 [2 2 2]
 [3 3 3]
 [4 4 4]]
```

In [57]:
```python
# broadcasting
newbar = np.array([[1,2],[3,4]])
print(newbar)
```
```
[[1 2]
 [3 4]]
```

In [58]:
```python
crwd = newbar+1
print(crwd)
```
```
[[2 3]
 [4 5]]
```

In [60]:
```python
crwd1 = newbar/2
print(crwd1)
```
```
[[0.5 1. ]
 [1.5 2. ]]
```

In [61]:
```python
crwd2 = newbar*3
print(crwd2)
```
```
[[ 3  6]
 [ 9 12]]
```

In [62]:
```python
crwd4 = newbar-4
print(crwd4)
```
```
[[-3 -2]
 [-1  0]]
```

In [ ]:
```python

```