How to create a basic array

```
import numpy as np
```

```
np.zeros(2)
```

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

```
np.ones(2)
```

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

```
np.empty(2)
```

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

```
np.arange(10)
```

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
np.arange(2,10,2)
```

```
array([2, 4, 6, 8])
```

```
np.linspace(0,20,num=5) # linear space
```

```
array([ 0.,  5., 10., 15., 20.])
```

```
np.linspace(0,25,5) # define without num also
```

```
array([ 0.  ,  6.25, 12.5 , 18.75, 25.  ])
```

```
x=np.ones(2,dtype=np.int64) #specify which data type you want using
the dtype keyword.
```

```
x
```

```
array([1, 1], dtype=int64)
```

```
points=5
c=np.linspace(0,10,points)
```

```
c
```

```
array([ 0. ,  2.5,  5. ,  7.5, 10. ])
```

Adding, removing, and sorting elements

```
arr=np.array([4,56,88,93,32,87,2,1,15])
```

```
np.sort(arr)
```

```
array([ 1,  2,  4, 15, 32, 56, 87, 88, 93])
```

```
np.argsort(arr) #  index wise it display
```

```
array([7, 6, 0, 8, 4, 1, 5, 2, 3], dtype=int64)
```

Concatenate

```
a=np.array([1,2,3])
b=np.array([3,4,5])  #1D array

np.concatenate((a,b))
```

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

```
aa=([[1,2,3],[55,76,87]])
bb=([[3,4,5],[10,20,30]]) #2D array

np.concatenate((aa,bb),axis=1)
```

```
array([[ 1,  2,  3,  3,  4,  5],
       [55, 76, 87, 10, 20, 30]])
```

```
import numpy as np
x = np.array([-5, 9 , 20 , 25, -3, 5, 16, 10,-8])
x[(x >= -5) & (x <= 15)] *= -1
print(x)
```

```
[  5  -9  20  25   3  -5 -16 -10  -8]
```

```
import numpy as np
a = np.array([[34, 28,55], [8, 56, 3], [77, 87, 19]])
print(a.transpose()[-2,-2])
```

```
56
```

```
import numpy as np

def get_elements(arr):
    '''
    INPUT: arr -> 1D numpy array

    OUTPUT elements -> tuple of first and last element.
    '''

    first_element =np.array[0:]

    last_element =np.arrary[:-1]
```

```
    return (first_element, last_element)

import numpy as np
def seq(start, length, step):
    sequence =np.arange(length)*step+start
    return sequence

def rotate_img(mat):
    return np.flip(mat.T, axis=1)

import numpy as np
arr = np.array([1, 2, 3, 4])
print(arr[2] + arr[-2])
```

```
6
```

```
arr = np.array([1,2,3,4,5,6,7,8])
arr[::2] = range(10,50,10)
print(arr)
```

```
[10  2 20  4 30  6 40  8]
```

Transpose is a (Equivalent function)

```
a=np.array([[1,2,3],[67,87,43]])

a
```

```
array([[ 1,  2,  3],
       [67, 87, 43]])
```

```
a.transpose()
```

```
array([[ 1, 67],
       [ 2, 87],
       [ 3, 43]])
```

```
a.transpose((1,0)) # swapping the axis 0 means  rows and 1 means
columns
```

```
array([[ 1, 67],
       [ 2, 87],
       [ 3, 43]])
```

```
b=np.array([[1,2,3],[4,5,7]])

b.transpose()
```

```
array([[1, 4],
       [2, 5],
       [3, 7]])

b.T # .T is a shorthand

array([[1, 4],
       [2, 5],
       [3, 7]])

elements=np.arange(30).reshape((5,6))

elements=np.arange(2,11).reshape((3,3))

elements

array([[ 2,  3,  4],
       [ 5,  6,  7],
       [ 8,  9, 10]])

a=np.arange(6)

a

array([0, 1, 2, 3, 4, 5])

np.reshape(a, (1,6),order='A')

array([[0, 1, 2, 3, 4, 5]])

# Reverse Array
element=np.array([1,2,4,5,6,7,8,9])

reverse=np.flip(element
                )

print('reversed array is',reverse)

reversed array [9 8 7 6 5 4 2 1]

a = np.array([11, 11, 12, 13, 14, 15, 16, 17, 12, 13, 11, 14, 18, 19,
20])

r=np.unique(a)

r

array([11, 12, 13, 14, 15, 16, 17, 18, 19, 20])
```


Convert 1D array into 2D array

```
c=np.array([1,2,3,4,5])
```

```
c.shape
```

```
(5,)
```

```
c2=a[np.newaxis, :] # Row vector
```

```
c2.shape
```

```
(1, 2, 3)
```

```
c2
```

```
array([[[ 1,  2,  3],
        [67, 87, 43]]])
```

```
c3=c[: , np.newaxis]# column vector
```

```
c3.shape
```

```
(5, 1)
```

```
c3
```

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

```
a=np.expand_dims(c,axis=1)
```

```
a.shape
```

```
(5, 1)
```

```
d=np.expand_dims(c,axis=0)
```

```
d.shape
```

```
(1, 5)
```

Indexing and slicing

```
elements=np.array([1,2,3,4,5])
```

```
elements
```

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

```
elements[1]
```

```
2
```

```
elements[0:3]
```

```
array([1, 2, 3])
```

```
elements[-1] # negitive indixing
```

```
5
```

```
 elements[:]
```

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

```
elements[-2:]
```

```
array([4, 5])
```

```
print(a[a<5])  # print the array values less than 5
```

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

```
a = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
```

```
more=(a >= 7)
print(a[more])
```

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

```
divisible=a[a%2==0]
print(divisible)
```

```
[ 2  4  6  8 10 12]
```

```
notdivi=a[a%2=1]
print(notdivi)
```

```
[ 2  4  6  8 10 12]
```

```
notdivi=a[a%2==1]
print(notdivi)
```

```
[ 1  3  5  7  9 11]
```

```
c=a[(a>2)&(a<10)] #wheterh condition satisfy or not using & operator
c
```

```
array([3, 4, 5, 6, 7, 8, 9])
```

```
d=a[(a>2)|(a<10)] # #wheterh condition satisfy or not using | operator
d
```

```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

```
five=[(a>5) |(a==5)]
```

```
five
```

```
[array([[False, False, False, False],
        [ True,  True,  True,  True],
        [ True,  True,  True,  True]])]
```

```
b=np.nonzero(a<5)
print(b)
```

```
(array([0, 0, 0, 0], dtype=int64), array([0, 1, 2, 3], dtype=int64))
```

```
# Vstack
```

```
a1=np.array([[1,1],
             [2,2]])

a2=np.array([[3,3],
           [4,4]])

np.vstack((a1,a2))
```

```
array([[1, 1],
       [2, 2],
       [3, 3],
       [4, 4]])
```

```
#hstack
np.hstack((a1,a2))
```

```
array([[1, 1, 3, 3],
       [2, 2, 4, 4]])
```

```
x=np.arange(1,25).reshape(2,12)

x
```

```
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]])
```

```
np.hsplit(x,3)  # split this array into three equally shaped arrays
```

```
[array([[ 1,  2,  3,  4],
        [13, 14, 15, 16]]),
 array([[ 5,  6,  7,  8],
        [17, 18, 19, 20]]),
 array([[ 9, 10, 11, 12],
        [21, 22, 23, 24]])]
```

```
np.hsplit(x,(3,5))
```

```
[array([[ 1,  2,  3],
        [13, 14, 15]]),
 array([[ 4,  5],
        [16, 17]]),
```

```
 array([[ 6,  7,  8,  9, 10, 11, 12],
        [18, 19, 20, 21, 22, 23, 24]])]
```

```
a = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
```

```
b1=a[0, :]
```

```
b1
```

```
array([1, 2, 3, 4])
```

```
b1[0]=99
```

```
b1
```

```
array([99,  2,  3,  4])
```

```
a
```

```
array([[99,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```

```
b2=a.copy()
```

```
b2
```

```
array([[99,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```