

```
In [1]: import numpy as np
```

```
In [92]: #zeros is a function which we can create n dimensional elements is called zeros
```

```
In [93]: a=np.zeros((4,3))  
a
```

```
Out[93]: array([[0., 0., 0.],  
               [0., 0., 0.],  
               [0., 0., 0.],  
               [0., 0., 0.]])
```

```
In [94]: a=np.zeros((4,3),dtype=int)  
a
```

```
Out[94]: array([[0, 0, 0],  
               [0, 0, 0],  
               [0, 0, 0],  
               [0, 0, 0]])
```

```
In [95]: #ones
```

```
In [96]: a=np.ones((4,3),dtype=int)  
a
```

```
Out[96]: array([[1, 1, 1],  
               [1, 1, 1],  
               [1, 1, 1],  
               [1, 1, 1]])
```

```
In [ ]: #full
```

```
In [150]: a=np.full((4,3),81)  
a
```

```
Out[150]: array([[81., 81., 81.],  
               [81., 81., 81.],  
               [81., 81., 81.],  
               [81., 81., 81.]])
```

```
In [ ]: #eye
```

```
In [98]: a=np.eye(10)  
a
```

```
Out[98]: array([[1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],  
               [0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],  
               [0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],  
               [0., 0., 0., 1., 0., 0., 0., 0., 0., 0.],  
               [0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],  
               [0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],  
               [0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],  
               [0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],  
               [0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],  
               [0., 0., 0., 0., 0., 0., 0., 0., 0., 1.]])
```

```
In [2]: a=np.eye(10,dtype=float)  
a
```

```
Out[2]: array([[1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],  
               [0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],  
               [0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],  
               [0., 0., 0., 1., 0., 0., 0., 0., 0., 0.],  
               [0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],  
               [0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],  
               [0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],  
               [0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],  
               [0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],  
               [0., 0., 0., 0., 0., 0., 0., 0., 0., 1.]])
```

```
In [4]: np.diag(a)
```

```
Out[4]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

```
In [ ]: #diag
```

```
In [4]: a=np.diag([10,11,12])  
a
```

```
Out[4]: array([[10,  0,  0],  
              [ 0, 11,  0],  
              [ 0,  0, 12]])
```

```
In [101]: np.diag(a)
```

```
Out[101]: array([10, 11, 12])
```

```
In [102]: a=np.eye(10,dtype=float)  
np.diag(a)
```

```
Out[102]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

```
In [103]: #arange
```

```
In [104]: a=np.arange(10,21)
```

```
In [105]: a.shape
```

```
Out[105]: (11,)
```

```
In [106]: a
```

```
Out[106]: array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20])
```

```
In [107]: #step size
```

```
In [108]: np.arange(10,21,2)
```

```
Out[108]: array([10, 12, 14, 16, 18, 20])
```

```
In [109]: for y in range(1,10,3):  
           print(y)
```

```
1  
4  
7
```

```
In [110]: x="abcdef"
          x[::-1]
```

```
Out[110]: 'fedcba'
```

```
In [111]: for y in x[::-1]:
          print(y)
```

```
f
e
d
c
b
a
```

```
In [112]: for y in x:
          print(y)
```

```
a
b
c
d
e
f
```

```
In [113]: for y in x[::-1]:
          print(y)
```

```
f
e
d
c
b
a
```

```
In [114]: #linespace
```

```
In [115]: np.linspace(1,10,5)
```

```
Out[115]: array([ 1. ,  3.25,  5.5 ,  7.75, 10. ])
```

```
In [116]: np.linspace(1,8,3)
```

```
Out[116]: array([1. , 4.5, 8. ])
```

```
In [117]: np.linspace(1,8)
```

```
Out[117]: array([1.          , 1.14285714, 1.28571429, 1.42857143, 1.57142857,
 1.71428571, 1.85714286, 2.          , 2.14285714, 2.28571429,
 2.42857143, 2.57142857, 2.71428571, 2.85714286, 3.          ,
 3.14285714, 3.28571429, 3.42857143, 3.57142857, 3.71428571,
 3.85714286, 4.          , 4.14285714, 4.28571429, 4.42857143,
 4.57142857, 4.71428571, 4.85714286, 5.          , 5.14285714,
 5.28571429, 5.42857143, 5.57142857, 5.71428571, 5.85714286,
 6.          , 6.14285714, 6.28571429, 6.42857143, 6.57142857,
 6.71428571, 6.85714286, 7.          , 7.14285714, 7.28571429,
 7.42857143, 7.57142857, 7.71428571, 7.85714286, 8.          ])
```

```
In [118]: np.linspace(1,8,3,dtype=int)
```

```
Out[118]: array([1, 4, 8])
```

```
In [ ]: #random
```

```
In [119]: np.random.random((2,3))
```

```
Out[119]: array([[0.73835572, 0.53322247, 0.36899536],
 [0.67832334, 0.97583968, 0.73739115]])
```

```
In [120]: np.random.random((2,3))
```

```
Out[120]: array([[0.64894374, 0.49302928, 0.06128197],
 [0.71639133, 0.87473148, 0.22303129]])
```

```
In [121]: a=np.array([1,2,3,4,5])
```

```
In [122]: a+2
```

```
Out[122]: array([3, 4, 5, 6, 7])
```

```
In [123]: a=np.array([1,2,3,4,5])
```

```
In [124]: a*2
```

```
Out[124]: array([ 2,  4,  6,  8, 10])
```

```
In [125]: a.shape
```

```
Out[125]: (5,)
```

```
In [6]: import time
```

```
In [7]: x=time.time()
y=np.arange(1,1000000)*2
z=time.time()
print(z-x)
```

```
0.011001110076904297
```

```
In [8]: l=[]
for y in range(1,1000000):
    l.append(y*2)
z=time.time()
print(z-x)
```

```
1.6864938735961914
```

```
In [129]: #access the elements inside an n_dimensional array
```

```
In [130]: #n=0
```

```
In [131]: #n=1
```

```
In [132]: x=np.arange(1,9)
```

```
In [133]: x
```

```
Out[133]: array([1, 2, 3, 4, 5, 6, 7, 8])
```

```
In [134]: x[0]
```

```
Out[134]: 1
```

```
In [135]: x[-1]
```

```
Out[135]: 8
```

```
In [136]: x[:]
```

```
Out[136]: array([1, 2, 3, 4, 5, 6, 7, 8])
```

```
In [137]: #n=2
```

```
In [10]: z=np.random.random((3,4))
```

```
In [11]: z
```

```
Out[11]: array([[0.72998054, 0.87750143, 0.29928021, 0.01756708],  
                [0.19348855, 0.0219769 , 0.07884348, 0.34516104],  
                [0.27306488, 0.53953957, 0.87796977, 0.70946849]])
```

```
In [13]: z[1,2]
```

```
Out[13]: 0.07884348488890291
```

```
In [14]: z[2,3]
```

```
Out[14]: 0.7094684888921877
```

```
In [15]: z[0,0]
```

```
Out[15]: 0.7299805439889024
```

```
In [16]: x=6.3
```

```
In [72]: type(x)
```

```
Out[72]: float
```

```
In [1]: #practice
```

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

```
In [8]: x=np.array([1,2,3,4,5],dtype=float,ndmin=4)
x
```

```
Out[8]: array([[[[1., 2., 3., 4., 5.]]]])
```

```
In [17]: x=np.zeros((4,4,3),dtype=int)
x
```

```
Out[17]: array([[[0, 0, 0],
                  [0, 0, 0],
                  [0, 0, 0],
                  [0, 0, 0]],

                [[0, 0, 0],
                  [0, 0, 0],
                  [0, 0, 0],
                  [0, 0, 0]],

                [[0, 0, 0],
                  [0, 0, 0],
                  [0, 0, 0],
                  [0, 0, 0]],

                [[0, 0, 0],
                  [0, 0, 0],
                  [0, 0, 0],
                  [0, 0, 0]]])
```



```
In [15]: x=np.zeros((4,3),dtype=int)
x
```

```
Out[15]: array([[0, 0, 0],
               [0, 0, 0],
               [0, 0, 0],
               [0, 0, 0]])
```

```
In [22]: x=np.ones((4,4,2),dtype=complex)
x
```

```
Out[22]: array([[[1.+0.j, 1.+0.j],
                 [1.+0.j, 1.+0.j],
                 [1.+0.j, 1.+0.j],
                 [1.+0.j, 1.+0.j]],

                [[1.+0.j, 1.+0.j],
                 [1.+0.j, 1.+0.j],
                 [1.+0.j, 1.+0.j],
                 [1.+0.j, 1.+0.j]],

                [[1.+0.j, 1.+0.j],
                 [1.+0.j, 1.+0.j],
                 [1.+0.j, 1.+0.j],
                 [1.+0.j, 1.+0.j]],

                [[1.+0.j, 1.+0.j],
                 [1.+0.j, 1.+0.j],
                 [1.+0.j, 1.+0.j],
                 [1.+0.j, 1.+0.j]])])
```

```
In [26]: x=np.full((8,6),34,dtype=float)
x
```

```
Out[26]: array([[34., 34., 34., 34., 34., 34.],
               [34., 34., 34., 34., 34., 34.],
               [34., 34., 34., 34., 34., 34.],
               [34., 34., 34., 34., 34., 34.],
               [34., 34., 34., 34., 34., 34.],
               [34., 34., 34., 34., 34., 34.],
               [34., 34., 34., 34., 34., 34.],
               [34., 34., 34., 34., 34., 34.]])
```

```
In [34]: x=np.eye((3,3))
x
```

```
-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_4956\3883218915.py in <module>
----> 1 x=np.eye((3,3))
      2 x

~\anaconda3\lib\site-packages\numpy\lib\twodim_base.py in eye(N, M, k, dtype, order, like)
    212     if M is None:
    213         M = N
--> 214     m = zeros((N, M), dtype=dtype, order=order)
    215     if k >= M:
    216         return m

TypeError: 'tuple' object cannot be interpreted as an integer
```

```
In [39]: x=np.eye((3))
print(x)
print(x.shape)
```

```
[[1.  0.  0.]
 [0.  1.  0.]
 [0.  0.  1.]]
(3, 3)
```

```
In [52]: x=np.diag((5,6,70))
x
```

```
Out[52]: array([[ 5,  0,  0],
                [ 0,  6,  0],
                [ 0,  0, 70]])
```

```
In [55]: x[0,0]
```

```
Out[55]: 5
```

```
In [58]: np.diag(x)
```

```
Out[58]: array([ 5,  6, 70])
```

```
In [59]: x=np.eye((3,3))  
x
```

```
-----  
TypeError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_4956\3883218915.py in <module>  
----> 1 x=np.eye((3,3))  
      2 x  
  
~\anaconda3\lib\site-packages\numpy\lib\twodim_base.py in eye(N, M, k, dtype, order, like)  
    212     if M is None:  
    213         M = N  
--> 214     m = zeros((N, M), dtype=dtype, order=order)  
    215     if k >= M:  
    216         return m  
  
TypeError: 'tuple' object cannot be interpreted as an integer
```

```
In [61]: a=np.arange(10,21,2)  
a
```

```
Out[61]: array([10, 12, 14, 16, 18, 20])
```

```
In [62]: a.shape
```

```
Out[62]: (6,)
```

```
In [63]: a.ndim
```

```
Out[63]: 1
```

```
In [65]: x="abcdef"  
x[::]
```

```
Out[65]: 'abcdef'
```

```
In [78]: x="abcdef"
x[::-1]
```

```
Out[78]: 'fedcba'
```

```
In [67]: x="abcdef"
x[-1]
```

```
Out[67]: 'f'
```

```
In [77]: x="abcdef"
x[::]
```

```
Out[77]: 'a'
```

```
In [149]: np.linspace(1,7,dtype=float)
```

```
Out[149]: array([1.          , 1.12244898, 1.24489796, 1.36734694, 1.48979592,
 1.6122449  , 1.73469388, 1.85714286, 1.97959184, 2.10204082,
 2.2244898  , 2.34693878, 2.46938776, 2.59183673, 2.71428571,
 2.83673469, 2.95918367, 3.08163265, 3.20408163, 3.32653061,
 3.44897959, 3.57142857, 3.69387755, 3.81632653, 3.93877551,
 4.06122449, 4.18367347, 4.30612245, 4.42857143, 4.55102041,
 4.67346939, 4.79591837, 4.91836735, 5.04081633, 5.16326531,
 5.28571429, 5.40816327, 5.53061224, 5.65306122, 5.7755102  ,
 5.89795918, 6.02040816, 6.14285714, 6.26530612, 6.3877551  ,
 6.51020408, 6.63265306, 6.75510204, 6.87755102, 7.          ])
```

```
In [ ]: np.zeros((4,5))
np.ones((4,3))
np.eye(10)
np.diag([2,4,6])
np.linspace(1,8)
np.full((5,6),21)
np.arange(1,6)
```

```
In [164]: np.arange(1,6)
```

```
Out[164]: array([1, 2, 3, 4, 5])
```

In []: