

In [1]:

Out[1]: list

```
In [1]: a =[1,5,8,6]
         type(a)
```

Out[1]: list

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

```
In [27]: import numpy as np
         # abcd.__version__
```

```
In [28]: arr0 = np.array(5)
```

```
In [29]: print(arr0)
```

5

```
In [30]: type(arr0)
```

Out[30]: numpy.ndarray

```
In [31]: arr0.ndim
```

Out[31]: 0

```
In [32]: arr1 = np.array([5,6,9,8])
         print(arr1)
         type(arr1)
         arr1.ndim
```

[5 6 9 8]

Out[32]: 1

```
In [33]: a1 =np.array([5,6,9,9])
         print(a1)
         print(type(a1))
         print(a1.ndim)
```

[5 6 9 9]  
<class 'numpy.ndarray'>  
1

```
In [34]: a2 = np.array(['a', 's', 'f', 8])
print(a2)
print(type(a2))
```

```
['a' 's' 'f' '8']
<class 'numpy.ndarray'>
```

```
In [35]: arr2 = np.array([
    [5,8,9,6],
    [8,9,5,6]
])
print(arr2)
print(type(arr2))
```

```
[[5 8 9 6]
 [8 9 5 6]]
<class 'numpy.ndarray'>
```

```
In [36]: arr3 = np.array([[[5,9,6,8]]])
arr3.ndim
```

```
Out[36]: 3
```

```
In [37]: import pandas
```

```
In [38]: import numpy as np
a = np.array(7)
print(a)
print(type(a))
print(a.ndim)
```

```
7
<class 'numpy.ndarray'>
0
```

```
In [39]: b = np.array([8,9,5,6,5])
print(b)
print(type(b))
print(b.ndim)
```

```
[8 9 5 6 5]
<class 'numpy.ndarray'>
1
```

```
In [40]: my_list = []
c = np.array(my_list)
print(type(c))
print(type(my_list))

<class 'numpy.ndarray'>
<class 'list'>
```

```
In [41]: for i in range(1,5):
x = int(input('elements:'))
my_list.append(x)
d = np.array(my_list)
print(d)
print(type(d))
print(my_list)
```

```
elements:52
[52]
<class 'numpy.ndarray'>
[52]
elements:52
[52 52]
<class 'numpy.ndarray'>
[52, 52]
elements:52
[52 52 52]
<class 'numpy.ndarray'>
[52, 52, 52]
elements:63
[52 52 52 63]
<class 'numpy.ndarray'>
[52, 52, 52, 63]
```

```
In [42]: e = np.zeros(4)
print(e)
```

```
[0. 0. 0. 0.]
```

```
In [43]: f = np.ones(6)
print(f)
```

```
[1. 1. 1. 1. 1. 1.]
```

```
In [44]: g = np.arange(9)
print(g)
print(type(g))
```

```
[0 1 2 3 4 5 6 7 8]
<class 'numpy.ndarray'>
```

```
In [45]: #Spacial
j = np.linspace(0,10, num= 2)
j
```

```
Out[45]: array([ 0., 10.])
```

```
In [46]: k = np.random.rand(4)
k
```

```
Out[46]: array([0.5064164 , 0.65588167, 0.13683876, 0.86935419])
```

```
In [47]: k = np.random.rand(4)
k
```

```
Out[47]: array([0.26570139, 0.51425191, 0.9318007 , 0.30995579])
```

```
In [48]: #( - -> +1)
l =np.random.randn(4)
l
```

```
Out[48]: array([-0.01454436, -2.34098551, -0.22423949, -1.24888809])
```

```
In [49]: m =np.random.rand(4)
m
```

```
Out[49]: array([0.70275407, 0.83779804, 0.34202579, 0.70970157])
```

```
In [50]: #(start, stop, count)
n =np.random.randint(4,50,6)
n
```

```
Out[50]: array([27, 49, 29, 38, 25, 14])
```

```
In [51]: #(min ,max , count )
o = np.linspace(100,1000, num = 3)
o
```

```
Out[51]: array([ 100.,  550., 1000.])
```

```
In [52]: o.dtype
```

```
Out[52]: dtype('float64')
```

```
In [53]: m.dtype
```

```
Out[53]: dtype('float64')
```

```
In [54]: n.dtype
```

```
Out[54]: dtype('int32')
```

```
In [55]: l.dtype
```

```
Out[55]: dtype('float64')
```

```
In [56]: j.dtype
```

```
Out[56]: dtype('float64')
```

```
In [57]: c.dtype
```

```
Out[57]: dtype('float64')
```

```
In [58]: e.dtype
```

```
Out[58]: dtype('float64')
```

```
In [59]: r = np.array(['Seema', 'Manisha', 'Pooja'])  
r
```

```
Out[59]: array(['Seema', 'Manisha', 'Pooja'], dtype='<U7')
```

```
In [60]: s = np.array(['A'])  
s.dtype
```

```
Out[60]: dtype('<U1')
```

```
In [61]: t = np.array([True])  
t.dtype
```

```
Out[61]: dtype('bool')
```

```
In [62]: u = np.array([456])  
u.dtype
```

```
Out[62]: dtype('int32')
```

```
In [63]: v = np.array([-56845])  
v.dtype
```

```
Out[63]: dtype('int32')
```

```
In [64]: w = np.array([2+3j])  
w.dtype
```

```
Out[64]: dtype('complex128')
```

```
In [65]: x = np.array(['a', 's', 'w', 'f', 5, 6, 9, 5, False])  
x.dtype
```

```
Out[65]: dtype('<U11')
```

```
In [66]: y = np.array([5, 6, 8, 9], dtype= np.int_)  
print(y)  
print(y.dtype)
```

```
[5 6 8 9]  
int32
```

```
In [67]: y = np.array([5, 6, 8, 9], dtype= np.float_)  
print(y)  
print(y.dtype)
```

```
[5. 6. 8. 9.]  
float64
```

```
In [68]: z = np.array([5, 8, 6, 8.5], dtype = np.int_)  
print(z)  
print(z.dtype)
```

```
[5 8 6 8]  
int32
```

```
In [69]: a = np.float32(y)  
print(a)  
print(a.dtype)
```

```
[5. 6. 8. 9.]  
float32
```

```
In [70]: # List $ array-> datatype()
```

```
In [71]: a = []  
type(a)
```

```
Out[71]: list
```

```
In [72]: b = [i**2 for i in range(1, 15)]  
b
```

```
Out[72]: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196]
```

```
In [84]: %timeit [i**2 for i in range(1, 15)]
```

```
929 ns ± 3.68 ns per loop (mean ± std. dev. of 7 runs, 1,000,000 loops each)
```

```
In [85]: import numpy as np
%timeit np.arange(1,15)**2
```

1.12  $\mu$ s  $\pm$  32.9 ns per loop (mean  $\pm$  std. dev. of 7 runs, 1,000,000 loops each)

```
In [86]: %timeit [x**3 for x in range(1,20)]
```

1.36  $\mu$ s  $\pm$  24.9 ns per loop (mean  $\pm$  std. dev. of 7 runs, 1,000,000 loops each)

```
In [142]: import numpy as np
```

```
In [143]: %timeit np.arange(1,20)**3
```

1.84  $\mu$ s  $\pm$  29.5 ns per loop (mean  $\pm$  std. dev. of 7 runs, 1,000,000 loops each)

```
In [89]: %timeit np.arange(1,15)
```

521 ns  $\pm$  0.946 ns per loop (mean  $\pm$  std. dev. of 7 runs, 1,000,000 loops each)

```
In [90]: np.arange(1,15)**2
```

```
Out[90]: array([ 1,  4,  9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169,
                196])
```

```
In [91]: a =np.array([
            [8,2],[5,6],[6,5]
        ])
a.ndim
```

```
Out[91]: 2
```

```
In [92]: d =[[[5,8,9]]]
d[0][0][0]
```

```
Out[92]: [5, 8, 9]
```

```
In [93]: b =np.array([[[[[]]]])
b.ndim
```

```
Out[93]: 3
```

```
In [94]: c =[[5,6,2,5, 'Seema']]
c
```

```
Out[94]: [[5, 6, 2, 5, 'Seema']]
```

```
In [95]: d =[[[5,9,8],['a','b','c']]]  
d[0][0][1][0]
```

```
Out[95]: 'a'
```

```
In [96]: e =np.array([[[[8,5,8]]]])  
e.ndim
```

```
Out[96]: 4
```

```
In [97]: f =np.array([[[[[[[[8,5,8,5]]]]]]]])  
f.ndim
```

```
Out[97]: 10
```

```
In [98]: g = np.arange(3,3)  
g  
g.ndim
```

```
Out[98]: 1
```

```
In [99]: h =np.zeros((3,3))  
h
```

```
Out[99]: array([[0., 0., 0.],  
                [0., 0., 0.],  
                [0., 0., 0.]])
```

```
In [100]: i =np.ones((3,3))  
i
```

```
Out[100]: array([[1., 1., 1.],  
                [1., 1., 1.],  
                [1., 1., 1.]])
```

```
In [101]: j = np.eye(4)  
j
```

```
Out[101]: array([[1., 0., 0., 0.],  
                [0., 1., 0., 0.],  
                [0., 0., 1., 0.],  
                [0., 0., 0., 1.]])
```



```
In [102]: k =np.eye(4,6)
print(k)
k.ndim
```

```
[[1.  0.  0.  0.  0.  0.]
 [0.  1.  0.  0.  0.  0.]
 [0.  0.  1.  0.  0.  0.]
 [0.  0.  0.  1.  0.  0.]]
```

Out[102]: 2

```
In [103]: l =np.random.rand(3,4)
print(l)
print(type(l))
```

```
[[0.98811348 0.88331941 0.42343603 0.90413059]
 [0.91006411 0.69058905 0.95334515 0.00936687]
 [0.43209044 0.01285039 0.76401316 0.23125042]]
<class 'numpy.ndarray'>
```

```
In [104]: m =np.random.randint((2,10,2,2,2))
m
```

Out[104]: array([0, 8, 1, 0, 0])

```
In [105]: a =[np.random.randint(2,8)]
b = np.array(a)
b.ndim
```

Out[105]: 1

```
In [106]: a = np.array([2,5,2,5,8,9,6,8])
print(a.ndim)
# dim 2d (m*n), 3d (x*y)
b = a.reshape(2,4)
print(b)
print(b.ndim)
```

```
1
[[2 5 2 5]
 [8 9 6 8]]
2
```

```
In [107]: a = np.array([2,5,2,5,8,9,6,8,6,9,5,8])
print("original dim:",a.ndim)
b = a.reshape(2,3,2)
print(b)
print('reshape dim:',b.ndim)
```

```
original dim: 1
[[[2 5]
  [2 5]
  [8 9]]

  [[6 8]
  [6 9]
  [5 8]]]
reshape dim: 3
```

```
In [108]: g= [[[[100,200,300],['ad','bd'],[[[5,8,2,4],[45,78,5],['a','b','e']]]]]]
a
```

```
Out[108]: array([2, 5, 2, 5, 8, 9, 6, 8, 6, 9, 5, 8])
```

```
In [109]: # shape
import numpy
a = numpy.array([[5,8,8,8],[2,5,4,5]])
print(a)
print(a.shape)
print()
```

```
[[5 8 8 8]
 [2 5 4 5]]
(2, 4)
```

```
In [110]: b =numpy.array([[[5,8,9,6,4]]])
print(b.shape)
print(b)
```

```
(1, 1, 5)
[[[5 8 9 6 4]]]
```

```
In [111]: d =numpy.array([[3,5,8],[5,8,6],[8,9,6]])
d.shape
#d.ndim
```

```
Out[111]: (3, 3)
```

```
In [112]: e = numpy.array([[
    [5,8,2],
    [6,9,7],
    [8,9,6],
    [8,9,3]
]])
print(e.shape)
print(e)
print(e.ndim)
```

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

```
In [113]: f =numpy.array([3,6,8,9,5])
f[1]
f[-2]
```

```
Out[113]: 9
```

```
In [114]: g =numpy.array([
    [8,6,9,5],
    [5,8,9,6]
])
g[1,2]
g [-2,-1]
```

```
Out[114]: 5
```

```
In [115]: h = numpy.array([[
    [5,6,2],
    [6,9,8]
]])
h.ndim
h.shape
h[0]
h[0,1]
```

```
Out[115]: array([6, 9, 8])
```

```
In [116]: a = numpy.array([[4,5,6,8],[4,8,2,6]])
b = numpy.array([[5,6,3,5],[8,9,6,5]])
a+b
numpy.add(a,b)
```

```
Out[116]: array([[ 9, 11,  9, 13],
                  [12, 17,  8, 11]])
```

```
In [117]: a = numpy.array([[4,5,6,8],[8,9,5,8]])  
b = numpy.array([[5,6,3,5],[8,6,9,5]])  
a*b
```

```
Out[117]: array([[20, 30, 18, 40],  
                [64, 54, 45, 40]])
```

```
In [118]: a = numpy.array([4,5,6,8])  
b = numpy.array([5,6,3,5])  
a/b
```

```
Out[118]: array([0.8          , 0.83333333, 2.          , 1.6          ])
```

```
In [119]: a = numpy.array([4,5,6,8])  
b = numpy.array([5,6,3,5])  
a%b
```

```
Out[119]: array([4, 5, 0, 3])
```

```
In [120]: a = numpy.array([[4,5,6,8],[8,9,5,8]])  
b = numpy.array([[5,6,3,5],[8,6,9,5]])  
a-b
```

```
Out[120]: array([[ -1,  -1,   3,   3],  
                [  0,   3,  -4,   3]])
```

```
In [121]: b-a
```

```
Out[121]: array([[ 1,  1, -3, -3],  
                [ 0, -3,  4, -3]])
```

```
In [122]: a*b
```

```
Out[122]: array([[20, 30, 18, 40],  
                [64, 54, 45, 40]])
```

```
In [123]: numpy.reciprocal(a)
```

```
Out[123]: array([[0, 0, 0, 0],  
                [0, 0, 0, 0]])
```

```
In [124]: s =numpy.array([1,8,9,14])  
1/s  
numpy.reciprocal(s)
```

```
Out[124]: array([1, 0, 0, 0])
```

```
In [125]: print(numpy.max(s))  
          print(numpy.argmax(s))
```

```
14  
3
```

```
In [126]: numpy.sqrt(s)
```

```
Out[126]: array([1.          , 2.82842712, 3.          , 3.74165739])
```

```
In [127]: numpy.sin(s)
```

```
Out[127]: array([0.84147098, 0.98935825, 0.41211849, 0.99060736])
```

```
In [128]: numpy.cos(s)
```

```
Out[128]: array([ 0.54030231, -0.14550003, -0.91113026,  0.13673722])
```

```
In [129]: numpy.reciprocal(s)
```

```
Out[129]: array([1, 0, 0, 0])
```

```
In [130]: print(numpy.min(s))  
          print(numpy.argmin(s))
```

```
1  
0
```

```
In [131]: f = numpy.array([5,8,9,6,5,8,2,-6])  
          print(numpy.argmin(f))  
          print(numpy.argmax(f))
```

```
7  
2
```

```
In [132]: numpy.cumsum(f)
```

```
Out[132]: array([ 5, 13, 22, 28, 33, 41, 43, 37])
```

```
In [133]: import numpy  
          c = numpy.array([5,8,9,6])  
          c.nbytes  
          c.ndim
```

```
Out[133]: 1
```

```
In [134]: g =c.reshape(2,2)
g
```

```
Out[134]: array([[5, 8],
                [9, 6]])
```

```
In [135]: e = numpy.array([2,5,6,8,9,2,6,3])
f =e.reshape(1,4,2)
f
```

```
Out[135]: array([[[2, 5],
                  [6, 8],
                  [9, 2],
                  [6, 3]]])
```

```
In [136]: g =numpy.array([2,8,9,6,4,8])
h=g.reshape(2,3)
h
```

```
Out[136]: array([[2, 8, 9],
                [6, 4, 8]])
```

```
In [137]: h.ndim
```

```
Out[137]: 2
```

```
In [138]: g.reshape(2,1,3)
```

```
Out[138]: array([[[2, 8, 9]],
                [[6, 4, 8]])]
```

```
In [139]: #broadcasting

a =numpy.array([4,2,5,1])
b=numpy.array([5])
a+b
```

```
Out[139]: array([ 9,  7, 10,  6])
```

```
In [140]: # broadcast error -> dimension same + similar/shape
a =numpy.array([[1,2],[5,6]])
print(a.shape)
b =numpy.array([[2,6],[8,9]])
print(b.shape)
#a+b
```

```
(2, 2)
(2, 2)
```

```
In [144]: # around: rounding off  
np.around
```

```
Out[144]: <function around at 0x00000223CD7A8630>
```

```
In [147]: import numpy  
a = numpy.array([[4,2,5,1],  
                 [4,2,5,1],  
                 [4,2,5,1],  
                 [4,2,5,1]])  
np.amin(a)
```

```
Out[147]: 1
```