jupyter NumPy-python Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Code

In [2]:
```
'''numpy is a python library, its stands for Numerical Python(numpy)
its consisting of multi-dimensionsal array objects'''

!pip install numpy
import numpy as np
```

```
Requirement already satisfied: numpy in c:\users\hp\anaconda3\lib\site-packages (1.20.1)
```

In [13]:
```
a=np.array([2,5,88,45])
a
```

Out[13]:
```
array([ 2,  5, 88, 45])
```

In [42]:
```
#dimensional-array like 2d,3d

a1=np.array(5)
a=np.array([2])
b=np.array([24,36,45])
c=np.array([[24,36,45],[55,88,66]])
d=np.array([[[24,36,45],[55,88,66],[29,39,49]]])
e=np.array([[[24,36,45],[55,88,66]],[[11,22,33],[77,89,99]]])
```

In [43]:
```
print('a1 dimension:',a1.ndim)
print('a dimension:', a.ndim)
print('b dimension:', b.ndim)
print('c dimension:', c.ndim)
print('d dimension:', d.ndim)
print('e dimension:', e.ndim)
```

```
a1 dimension: 0
a dimension: 1
b dimension: 1
c dimension: 2
d dimension: 3
e dimension: 3
```

```
c dimension: 2
d dimension: 3
e dimension: 3
```

In [50]:

```
## find - shape of array

a1=np.array(5)
a=np.array([2])
b=np.array([24,36,45])
c=np.array([[24,36,45],[55,88,66]])
d=np.array([[[24,36,45],[55,88,66],[29,39,49]]])
e=np.array([[[24,36,45],[55,88,66]],[[11,22,33],[77,89,99]]])

print(a1)
print('a1 shape:',a1.shape)
print('----------')
print(a)
print('a shape:', a.shape)
print('----------')
print(b)
print('b shape:', b.shape)
print('-----------')
print(c)
print('c shape:', c.shape)
print('-----------')
print(d)
print('d shape:', d.shape)
print('-------------')
print(e)
print('e shape:', e.shape)
```

```
5
a1 shape: ()
----------
[2]
a shape: (1,)
----------
[24 36 45]
b shape: (3,)
-----------
[[24 36 45]
 [55 88 66]]
c shape: (2, 3)
-----------
[[[24 36 45]
  [55 88 66]
  [29 39 49]]]
d shape: (1, 3, 3)
-------------
[[[24 36 45]
  [55 88 66]]

 [[11 22 33]
  [77 89 99]]]
e shape: (2, 2, 3)
```

In [60]:
```
## data type

a1=np.array(5)
a=np.array([2])
b=np.array([24,36,45])
c=np.array([[24,36,45],[55,88,66]])
d=np.array([[[24,36,45],[55,88,66],[29,39,49]]])
e=np.array([[[24,36,45],[55,88,66]],[[11,22,33],[77,89,99]]])

print('a1 type:',a1.dtype)
print('a type:', a.dtype)
print('b type:', b.dtype)
print('c type:', c.dtype)
print('d type:', d.dtype)
print('e type:', e.dtype)
```

```
a1 type: int32
a type: int32
b type: int32
c type: int32
d type: int32
e type: int32
```

In [63]:
```
type(c)  # type func
```

Out[63]: numpy.ndarray

In [64]:
```
type(d)
```

Out[64]: numpy.ndarray

In [73]:
```
# length
len(b)
```

Out[73]: 3

In [74]:
```
len(e)
```

Out[74]: 2

Out[74]: 2

In [78]:
```
1 np.arange(12,24) # range
```

Out[78]:
```
array([12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23])
```

In [79]:
```
1 np.arange(12,24,3)
```

Out[79]:
```
array([12, 15, 18, 21])
```

In [89]:
```
1 np.eye(6,dtype=float) # eye function passing parameter
```

Out[89]:
```
array([[1., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 1., 0., 0.],
       [0., 0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 0., 1.]])
```

In [101]:
```
1 np.eye(5,7,dtype=float)
```

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

In [104]:
```
1 np.zeros((5,6))
```

Out[104]:
```
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.]])
```

In [105]:
```
1 np.ones((4,5))
```

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

In [105]:
```
1 np.ones((4,5))
```

Out[105]:
```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

In [109]:
```
1 np.full((4,6),4)
```

Out[109]:
```
array([[4, 4, 4, 4, 4, 4],
       [4, 4, 4, 4, 4, 4],
       [4, 4, 4, 4, 4, 4],
       [4, 4, 4, 4, 4, 4]])
```

In [118]:
```
1 d=(24,25,26,27,28) ## diagonal function array
2 np.diag(d)
```

Out[118]:
```
array([[24,  0,  0,  0,  0],
       [ 0, 25,  0,  0,  0],
       [ 0,  0, 26,  0,  0],
       [ 0,  0,  0, 27,  0],
       [ 0,  0,  0,  0, 28]])
```

In [143]:
```
1 import random   # random generate integer
2 np.random.random(3)[2]
```

Out[143]:
```
0.3988726505824248
```

In [156]:
```
1 np.random.rand(4,3)
```

Out[156]:
```
array([[0.14635282, 0.10153919, 0.25619568],
       [0.49906035, 0.9714526 , 0.44474371],
       [0.41101535, 0.19550629, 0.56378519],
       [0.53453225, 0.52147374, 0.0928892 ]])
```

In [163]:
```
r=np.arange(12,28)
r
```

Out[163]:
```
array([12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27])
```

In [164]:
```
r.shape
```

Out[164]:
```
(16,)
```

In [167]:
```
r1=r.reshape(4,4) # covert 1x16 to 4x4
r1
```

Out[167]:
```
array([[12, 13, 14, 15],
       [16, 17, 18, 19],
       [20, 21, 22, 23],
       [24, 25, 26, 27]])
```

In [168]:
```
r1.shape  #  shape
```

Out[168]:
```
(4, 4)
```

In [173]:
```
# convert old dimension
r2=r1.ravel()
r2
```

Out[173]:
```
array([12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27])
```

In [175]:
```
r2.shape
```

Out[175]:
```
(16,)
```

In [177]:
```
r
```

Out[177]:
```
array([12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27])
```

In [218]:
```
k=np.array([[[2,3,4],[5,6,7],[8,9,10],[45,2,8]]])
k.shape
```

Out[218]: (1, 4, 3)

In [220]:
```
k1=np.arange(15)
k1
```

Out[220]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14])

In [223]:
```
k2=k1
k2
```

Out[223]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14])

In [226]:
```
k2[5]=25
print(k2)
print(k1)
```

```
[ 0  1  2  3  4 25  6  7  8  9 10 11 12 13 14]
[ 0  1  2  3  4 25  6  7  8  9 10 11 12 13 14]
```

In [234]:
```
print(np.shares_memory(k1,k2))
print(id(k1))
print(id(k2))
```

```
True
2351235265808
2351235265808
```

In [252]:
```
k3=k1.view()
np.shares_memory(k1,k3)
```

Out[252]: True

array([12, 18])

In [218]:
```
k=np.array([[[2,3,4],[5,6,7],[8,9,10],[45,2,8]]])
k.shape
```

Out[218]: (1, 4, 3)

In [220]:
```
k1=np.arange(15)
k1
```

Out[220]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14])

In [223]:
```
k2=k1
k2
```

Out[223]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14])

In [226]:
```
k2[5]=25
print(k2)
print(k1)
```

```
[ 0  1  2  3  4 25  6  7  8  9 10 11 12 13 14]
[ 0  1  2  3  4 25  6  7  8  9 10 11 12 13 14]
```

In [234]:
```
print(np.shares_memory(k1,k2))
print(id(k1))
print(id(k2))
```

```
True
2351235265808
2351235265808
```

In [252]:
```
k3=k1.view()
np.shares_memory(k1,k3)
```

Out[252]: True

In [236]:
```
k3[4]=77
print(k1)
print(k3)
```

```
[ 0  1  2  3 77 25  6  7  8  9 10 11 12 13 14]
[ 0  1  2  3 77 25  6  7  8  9 10 11 12 13 14]
```

In [238]:
```
k4=np.copy(k1)
k4
```

Out[238]:
```
array([ 0,  1,  2,  3, 77, 25,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```

In [246]:
```
k4[-3]=0
print(k1)
print(k4)
```

```
[ 0  1  2  3 77 25  6  7  8  9 10 11 12 13 14]
[ 0  1  2  3 77 25  6  7  8  9 10 11  0 13 14]
```

In [250]:
```
np.shares_memory(k1,k4)

```

Out[250]:
```
False
```

In [268]:
```
m=np.array([[24,25,26],[27,28,29],[30,31,38]])
m
```

Out[268]:
```
array([[24, 25, 26],
       [27, 28, 29],
       [30, 31, 38]])
```

In [269]:
```
m<28
```

Out[269]:
```
array([[ True,  True,  True],
       [ True, False, False],
       [False, False, False]])
```

jupyter NumPy-python Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Code

In [269]:
```
1 m<28
```

Out[269]:
```
array([[ True,  True,  True],
       [ True, False, False],
       [False, False, False]])
```

In [270]:
```
1 m>28
```

Out[270]:
```
array([[False, False, False],
       [False, False,  True],
       [ True,  True,  True]])
```

In [275]:
```
1 m[m>28]
```

Out[275]:
```
array([29, 30, 31, 38])
```

In [276]:
```
1 m[(m>28) & (m<34)]
```

Out[276]:
```
array([29, 30, 31])
```

In [277]:
```
1 k1==k4
```

Out[277]:
```
array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True, False,  True,  True])
```

In [278]:
```
1 k1==k3
```

Out[278]:
```
array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True])
```

In [280]:
```
1 a
```

Out[280]:
```
array([[24, 25, 26],
       [27, 28, 29],
       [30, 31, 38]])
```

In [281]:
```
1 a.transpose()
```

Out[281]:
```
array([[24, 27, 30],
       [25, 28, 31],
       [26, 29, 38]])
```

In [283]:
```
1 print(k1)
2 print(k4)
3 np.vstack((k1,k4))
```

```
[ 0  1  2  3 77 25  6  7  8  9 10 11 12 13 14]
[ 0  1  2  3 77 25  6  7  8  9 10 11  0 13 14]
```

Out[283]:
```
array([[ 0,  1,  2,  3, 77, 25,  6,  7,  8,  9, 10, 11, 12, 13, 14],
       [ 0,  1,  2,  3, 77, 25,  6,  7,  8,  9, 10, 11,  0, 13, 14]])
```

In [284]:
```
1 np.hstack((k1,k4))
```

Out[284]:
```
array([ 0,  1,  2,  3, 77, 25,  6,  7,  8,  9, 10, 11, 12, 13, 14,  0,  1,
        2,  3, 77, 25,  6,  7,  8,  9, 10, 11,  0, 13, 14])
```

In [287]:
```
1 np.insert(k1,7,60)
```

Out[287]:
```
array([ 0,  1,  2,  3, 77, 25,  6, 60,  7,  8,  9, 10, 11, 12, 13, 14])
```

In [288]:
```
1 np.insert(k1,7,k4)
```

Out[288]:
```
array([ 0,  1,  2,  3, 77, 25,  6,  0,  1,  2,  3, 77, 25,  6,  7,  8,  9,
       10, 11,  0, 13, 14,  7,  8,  9, 10, 11, 12, 13, 14])
```

In [291]:
```
1 print(k1)        # index 5th 25 deleted
2 np.delete(k1,5)
```

```
[ 0  1  2  3 77 25  6  7  8  9 10 11 12 13 14]
```

Out[291]:
```
array([ 0,  1,  2,  3, 77,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```

jupyter NumPy-python Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Code

In [292]:
```
a
```

Out[292]:
```
array([[24, 25, 26],
       [27, 28, 29],
       [30, 31, 38]])
```

In [294]:
```
np.sin(a)
```

Out[294]:
```
array([[-0.90557836, -0.13235175,  0.76255845],
       [ 0.95637593,  0.27090579, -0.66363388],
       [-0.98803162, -0.40403765,  0.29636858]])
```

In [296]:
```
np.tan(a)
```

Out[296]:
```
array([[-2.1348967 , -0.13352641,  1.17875355],
       [-3.2737038 , -0.2814296 ,  0.88714284],
       [-6.4053312 , -0.44169557,  0.31030966]])
```

In [297]:
```
np.exp(a)
```

Out[297]:
```
array([[2.64891221e+10, 7.20048993e+10, 1.95729609e+11],
       [5.32048241e+11, 1.44625706e+12, 3.93133430e+12],
       [1.06864746e+13, 2.90488497e+13, 3.18559318e+16]])
```

In [298]:
```
np.sum(a)
```

Out[298]:
```
258
```

In [300]:
```
np.sum(a,axis=0)
```

Out[300]:
```
array([81, 84, 93])
```

In [299]:
```
np.sum(a,axis=1)
```

Out[299]:
```
array([75, 84, 99])
```

In [301]:
```
a
```

Out[301]:
```
array([[24, 25, 26],
```

jupyter NumPy-python Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Code

In [301]:
```
1 a
```

Out[301]:
```
array([[24, 25, 26],
       [27, 28, 29],
       [30, 31, 38]])
```

In [302]:
```
1 np.mean(a)
```

Out[302]:
```
28.666666666666668
```

In [305]:
```
1 np.median(a)
```

Out[305]:
```
28.0
```

In [307]:
```
1 np.std(a)
```

Out[307]:
```
3.9440531887330774
```

In [308]:
```
1 np.max(a)
```

Out[308]:
```
38
```

In [310]:
```
1 k1
```

Out[310]:
```
array([ 0,  1,  2,  3, 77, 25,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```

In [309]:
```
1 np.sort(k1)
```

Out[309]:
```
array([ 0,  1,  2,  3,  6,  7,  8,  9, 10, 11, 12, 13, 14, 25, 77])
```

In [311]:
```
1 np.where(k1==10)
```

Out[311]:
```
(array([10], dtype=int64),)
```

In [312]:
```
1 np.where(k1%5==0)
```

Out[312]:
```
(array([ 0,  5, 10], dtype=int64),)
```

```
In [302]: 1 np.mean(a)
Out[302]: 28.666666666666668

In [305]: 1 np.median(a)
Out[305]: 28.0

In [307]: 1 np.std(a)
Out[307]: 3.9440531887330774

In [308]: 1 np.max(a)
Out[308]: 38

In [310]: 1 k1
Out[310]: array([ 0,  1,  2,  3, 77, 25,  6,  7,  8,  9, 10, 11, 12, 13, 14])

In [309]: 1 np.sort(k1)
Out[309]: array([ 0,  1,  2,  3,  6,  7,  8,  9, 10, 11, 12, 13, 14, 25, 77])

In [311]: 1 np.where(k1==10)
Out[311]: (array([10], dtype=int64),)

In [312]: 1 np.where(k1%5==0)
Out[312]: (array([ 0,  5, 10], dtype=int64),)

In [317]: 1 np.where(k1>7,k1,0)
Out[317]: array([ 0,  0,  0,  0, 77, 25,  0,  0,  8,  9, 10, 11, 12, 13, 14])
```