In [100]:

```
np.cos(data)
```

Out[100]:

```
array([ 0.28366219, -0.9899925 ,  0.96017029, -0.41614684,  0.75390225,
        0.28366219,  0.96017029, -0.91113026, -0.41614684, -0.91113026])
```

14th August 2021

vstack ==> Vertical stack

In [1]:

```
import numpy as np
```

In [2]:

```
arr1 = np.array([1,2,3])
arr2 = np.array([4,5,6])
```

In [3]:

```
arr1
```

Out[3]:

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

In [4]:

```
arr2
```

Out[4]:

```
array([4, 5, 6])
```

In [6]:

```
np.vstack((arr1,arr2))
```

Out[6]:

```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [ ]:

```

```

In [7]:

```
np.hstack((arr1,arr2))
```

Out[7]:

```
array([1, 2, 3, 4, 5, 6])
```

In [ ]:

```

```

generate random numbers in numpy

```
loc => The center of distribution
scale => standard deviation
size => number of items needed
```

In [8]:

```
np.random.normal(5,0.5,10)
```

Out[8]:

```
array([4.68673309, 3.97328146, 4.35045772, 5.53252203, 4.68165753,
       5.15732502, 5.46665867, 6.06308174, 4.92646927, 4.437834  ])
```

In [9]:

```
np.arange(10,21)
```

Out[9]:

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

In [10]:

```
np.linspace(1.0,5.0,10)
```

Out[10]:

```
array([1.        , 1.44444444, 1.88888889, 2.33333333, 2.77777778,
       3.22222222, 3.66666667, 4.11111111, 4.55555556, 5.        ])
```

In [ ]:

```

```

In [11]:

```
np.logspace(3,4,5)
```

Out[11]:

```
array([ 1000.        ,  1778.27941004,  3162.27766017,  5623.4132519 ,
       10000.        ])
```

In [ ]:

```

```

In [12]:

```
np.random.rand(3,3)
```

Out[12]:

```
array([[0.41578307, 0.8388365 , 0.22881436],
       [0.38284802, 0.10812475, 0.9260067 ],
       [0.2094045 , 0.15473044, 0.08255211]])
```

In [ ]:

```

```

In [22]:

```
np.random.randint(10,21,3)
```

Out[22]:

```
array([11, 18, 19])
```

In [ ]:

```

```

In [23]:

```
np.random.random_sample(10)
```

Out[23]:

```
array([0.19726802, 0.91746964, 0.59283665, 0.95956448, 0.51323291,
       0.75564914, 0.60747668, 0.81175719, 0.93497635, 0.57289313])
```

In [24]:

```
np.random.random(10)
```

Out[24]:

```
array([0.10625133, 0.1087037 , 0.88951939, 0.35566021, 0.19940294,
       0.99875057, 0.5525282 , 0.29926559, 0.97268628, 0.68197957])
```

repeat

In [13]:

```
arr = [10,20,30]
```

In [14]:

```
np.repeat(arr,3)
```

Out[14]:

```
array([10, 10, 10, 20, 20, 20, 30, 30, 30])
```

In [ ]:

```

```

In [15]:

```
np.tile(arr,3)
```

Out[15]:

```
array([10, 20, 30, 10, 20, 30, 10, 20, 30])
```

In [ ]:

```

```

In [ ]:

```

```

String array

In [16]:

```
str_arr = np.array(['Mumbai','Pune','Chennai'])
```

In [17]:

```
str_arr
```

Out[17]:

```
array(['Mumbai', 'Pune', 'Chennai'], dtype='<U7')
```

In [ ]:

```

```

identity matrix

In [18]:

```
arr = andint(10,21,3)np.eye(3)
```

In [19]:

```
arr
```

Out[19]:

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

In [ ]:

```

```

In [20]:

```
arr = np.array([[10,20,30],[40,50,60]])
```

In [21]:

```
np.info(arr)
```

```
class:  ndarray
shape:  (2, 3)
strides:  (24, 8)
itemsize:  8
aligned:  True
contiguous:  True
fortran:  False
data pointer: 0x565069783db0
byteorder:  little
byteswap:  False
type: int64
```

In [ ]:

```

```

permutation

In [25]:

```
arr = np.arange(10)
```

In [26]:

```
arr
```

Out[26]:

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

In [27]:

```
np.random.permutation(arr)
```

Out[27]:

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

In [ ]:

```

```

In [29]:

```
my_arr = np.arange(10,20).reshape(5,2)
```

In [30]:

```
my_arr
```

Out[30]:

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

In [31]:

```
my_arr.max()
```

Out[31]:

```
19
```

In [32]:

```
my_arr.min()
```

Out[32]:

```
10
```

In [33]:

```
my_arr.argmax()
```

Out[33]:

```
9
```

In [34]:

```
my_arr.argmin()
```

Out[34]:

```
0
```

In [42]:

```
my_arr.shape
```

Out[42]:

```
(5, 2)
```

In [35]:

```
np.sum(my_arr)
```

Out[35]:

```
145
```

In [37]:

```
np.sum(my_arr,axis=0) #sum col wise
```

Out[37]:

```
array([70, 75])
```

In [ ]:

```
 
```

In [41]:

```
np.sum(my_arr,axis=1) #row wise
```

Out[41]:

```
array([21, 25, 29, 33, 37])
```

In [43]:

```
arr1
```

Out[43]:

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

In [44]:

```
arr2
```

Out[44]:

```
array([4, 5, 6])
```

In [ ]:

```
 
```

In [45]:

```
np.minimum(arr1,arr2)
```

Out[45]:

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

In [46]:

```
np.maximum(arr1,arr2)
```

Out[46]:

```
array([4, 5, 6])
```

In [ ]:

```
 
```

In [ ]:

```
 
```

In [47]:

```
arr1 = np.random.rand(10)*10
```

In [48]:

```
arr1
```

Out[48]:

```
array([8.47815198, 7.17802705, 0.85469956, 8.55702182, 9.15059731,
       2.02647361, 6.9317298 , 4.52147187, 5.32298816, 1.644119  ])
```

In [ ]:

```
 
```

In [49]:

```
arr1.mean()
```

Out[49]:

```
5.46652801464039
```

In [ ]:

```
 
```

In [54]:

```
np.median(arr1)
```

Out[54]:

```
6.127358979124971
```

In [ ]:

In [55]:

```
np.std(arr1)
```

Out[55]:

```
2.93081770409358
```

In [ ]:

In [57]:

```
np.var(arr1)
```

Out[57]:

```
8.589692414628365
```

In [ ]:

In [58]:

```
arr1 = np.arange(10,20).reshape(5,2)
```

In [59]:

```
arr1
```

Out[59]:

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

In [60]:

```
np.cumsum(arr1)
```

Out[60]:

```
array([ 10,  21,  33,  46,  60,  75,  91, 108, 126, 145])
```

In [ ]:

In [61]:

```
np.cumsum(arr1,axis=0) #on the basis of col
```

Out[61]:

```
array([[10, 11],
       [22, 24],
       [36, 39],
       [52, 56],
       [70, 75]])
```

In [63]:

```
np.cumsum(arr1,axis=1) # on the basis of rows
```

Out[63]:

```
array([[10, 21],
       [12, 25],
       [14, 29],
       [16, 33],
       [18, 37]])
```

In [65]:

```
arr1
```

Out[65]:

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

In [64]:

```
np.cumprod(arr1)
```

Out[64]:

```
array([          10,          110,         1320,        17160,
             240240,      3603600,     57657600,    980179200,
        17643225600, 335221286400])
```

In [ ]:

```

```

filter array data

In [66]:

```
arr = np.arange(16).reshape(4,4)
```

In [67]:

```
arr
```

Out[67]:

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15]])
```

In [ ]:

```

```

In [68]:

```
arr > 6
```

Out[68]:

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

In [69]:

```
arr[arr>6]
```

Out[69]:

```
array([ 7,  8,  9, 10, 11, 12, 13, 14, 15])
```

In [ ]:

```

```

In [72]:

```
np.where(arr>6,"Yes","No")
```

Out[72]:

```
array([['No', 'No', 'No', 'No'],
       ['No', 'No', 'No', 'Yes'],
       ['Yes', 'Yes', 'Yes', 'Yes'],
       ['Yes', 'Yes', 'Yes', 'Yes']], dtype='<U3')
```

In [ ]:

```

```

In [73]:

```
arr = np.random.rand(5,5)
```

In [74]:

```
arr
```

Out[74]:

```
array([[0.71208519, 0.99064124, 0.94950856, 0.65406004, 0.4429267 ],
       [0.73919625, 0.0348003 , 0.88708116, 0.90732353, 0.289696  ],
       [0.25970957, 0.20674533, 0.87321095, 0.87095669, 0.63964905],
       [0.71767035, 0.64746299, 0.08983889, 0.97617487, 0.49288666],
       [0.52308887, 0.051347  , 0.86201432, 0.67364729, 0.65678467]])
```

np.sort

In [75]:

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

Out[75]:

```
array([[0.4429267 , 0.65406004, 0.71208519, 0.94950856, 0.99064124],
       [0.0348003 , 0.289696  , 0.73919625, 0.88708116, 0.90732353],
       [0.20674533, 0.25970957, 0.63964905, 0.87095669, 0.87321095],
       [0.08983889, 0.49288666, 0.64746299, 0.71767035, 0.97617487],
       [0.051347  , 0.52308887, 0.65678467, 0.67364729, 0.86201432]])
```

In [77]:

```
arr
```

Out[77]:

```
array([[0.71208519, 0.99064124, 0.94950856, 0.65406004, 0.4429267 ],
       [0.73919625, 0.0348003 , 0.88708116, 0.90732353, 0.289696  ],
       [0.25970957, 0.20674533, 0.87321095, 0.87095669, 0.63964905],
       [0.71767035, 0.64746299, 0.08983889, 0.97617487, 0.49288666],
       [0.52308887, 0.051347  , 0.86201432, 0.67364729, 0.65678467]])
```

In [76]:

```
np.sort(arr,axis=0)
```

Out[76]:

```
array([[0.25970957, 0.0348003 , 0.08983889, 0.65406004, 0.289696  ],
       [0.52308887, 0.051347  , 0.86201432, 0.67364729, 0.4429267 ],
       [0.71208519, 0.20674533, 0.87321095, 0.87095669, 0.49288666],
       [0.71767035, 0.64746299, 0.88708116, 0.90732353, 0.63964905],
       [0.73919625, 0.99064124, 0.94950856, 0.97617487, 0.65678467]])
```

In [78]:

```
np.sort(arr,axis=1)
```

Out[78]:

```
array([[0.4429267 , 0.65406004, 0.71208519, 0.94950856, 0.99064124],
       [0.0348003 , 0.289696  , 0.73919625, 0.88708116, 0.90732353],
       [0.20674533, 0.25970957, 0.63964905, 0.87095669, 0.87321095],
       [0.08983889, 0.49288666, 0.64746299, 0.71767035, 0.97617487],
       [0.051347  , 0.52308887, 0.65678467, 0.67364729, 0.86201432]])
```

In [ ]:

```

```

In [79]:

```
np.argsort(arr,axis=0)
```

Out[79]:

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

In [80]:

```
np.argsort(arr,axis=1)
```

Out[80]:

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

In [ ]:

```

```

In [81]:

```
arr1 = np.array([1,2,3,4,5])
arr2 = np.array([3,4,5,6,7])
```

In [82]:

```
np.intersect1d(arr1,arr2)
```

Out[82]:

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

In [ ]:

```

```

In [83]:

```
np.union1d(arr1,arr2)
```

Out[83]:

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

In [ ]:

```

```

In [84]:

```
np.isin(arr1,arr2)
```

Out[84]:

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

In [ ]:

```

```

In [85]:

```
arr1
```

Out[85]:

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

In [86]:

```
arr2
```

Out[86]:

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

In [87]:

```
np.dot(arr1,arr2)
```

Out[87]:

```
85
```

copy

In [100]:

```
a = np.arange(6)
```

In [89]:

```
a
```

Out[89]:

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

In [90]:

```
id(a)
```

Out[90]:

```
139667710374512
```

In [ ]:

```

```

In [91]:

```
b = a
```

In [92]:

```
b
```

Out[92]:

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

In [94]:

```
id(b)
```

Out[94]:

```
139667710374512
```

In [ ]:

```

```

In [97]:

```
b.shape= 3,2
```

In [98]:

```
b
```

Out[98]:

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

In [99]:

```
a
```

Out[99]:

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

In [ ]:

```

```

In [101]:

```
a
```

Out[101]:

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

In [102]:

```
b = np.copy(a)
```

In [103]:

```
b
```

Out[103]:

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

In [ ]:

```

```

In [104]:

```
a
```

Out[104]:

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

In [106]:

```
id(a)
```

Out[106]:

```
139667713541104
```

In [107]:

```
id(b)
```

Out[107]:

```
139667713543696
```

In [ ]:

```

```

In [108]:

```
a.shape = 3,2
```

In [109]:

```
a
```

Out[109]:

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

In [110]:

```
b
```

Out[110]:

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

In [ ]:

```

```