In [9]:
```
import numpy as np
```

In [10]:
```
x=np.arange(0,6).reshape(3,2)
y=np.arange(0,6).reshape(3,2)
```

In [11]:
```
x
```

Out[11]:
```
array([[0, 1],
       [2, 3],
       [4, 5]])
```

In [12]:
```
y
```

Out[12]:
```
array([[0, 1],
       [2, 3],
       [4, 5]])
```

In [13]:
```
np.array_equal(x,y)
```

Out[13]:
```
True
```

In [14]:
```
x==y
```

Out[14]:
```
array([[ True,  True],
       [ True,  True],
       [ True,  True]])
```

In [15]:
```
x=np.arange(0,6).reshape(3,2)
y=np.arange(0,6).reshape(2,3)
```

In [16]:
```
x
```

Out[16]:
```
array([[0, 1],
       [2, 3],
       [4, 5]])
```

In [17]:
```
y
```

Out[17]:
```
array([[0, 1, 2],
       [3, 4, 5]])
```

In [18]:
```
x==y
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_6416\4034449525.py:1: DeprecationWarning: elementwise comparison fa
iled; this will raise an error in the future.
  x==y
```

Out[18]:
```
False
```

In [19]:
```
np.array_equal(x,y)
```

Out[19]:
```
False
```

In [20]:
```
#np.logical_and()
#np.logical_or()
```

In [21]:
```
x1=np.array([[1,2,0,4]])
x2=np.array([[0,5,0,7]])
```

In [22]:
```
np.logical_and(x1,x2)
```

Out[22]:
```
array([[False,  True, False,  True]])
```

In [23]:
```
bool(0)
```

Out[23]:
```
False
```

In [24]:
```
np.logical_or(x1,x2)
```

Out[24]:
```
array([[ True,  True, False,  True]])
```

In [25]:
```
#np.any()
#np.all()
```

In [26]:
```
x=np.arange(0,6).reshape(3,2)
```

In [27]:
```
x
```

Out[27]:
```
array([[0, 1],
       [2, 3],
       [4, 5]])
```

In [28]:
```
x%2==0
```

Out[28]:
```
array([[ True, False],
       [ True, False],
       [ True, False]])
```

In [29]:
```
np.all(x%2==0)#all should be true if not false
```

Out[29]:
```
False
```

In [30]:
```
np.any(x%2==0)#all smust not be true
```

Out[30]:
```
True
```

In [31]:
```
#
```

In [32]:
```
x1=np.arange(0,6).reshape(3,2)
x2=np.arange(0,6).reshape(3,2)
```

In [33]:
```
x1
```

Out[33]:
```
array([[0, 1],
       [2, 3],
       [4, 5]])
```

In [34]:
```
x2
```

Out[34]:
```
array([[0, 1],
       [2, 3],
       [4, 5]])
```

In [35]:
```
x3=x2.T
x3
```

Out[35]:
```
array([[0, 2, 4],
       [1, 3, 5]])
```

In [36]:
```
np.dot(x1,x3)
```

Out[36]:
```
array([[ 1,  3,  5],
       [ 3, 13, 23],
       [ 5, 23, 41]])
```

In [37]:
```
x2.T
```

Out[37]:
```
array([[0, 2, 4],
       [1, 3, 5]])
```

In [38]:
```
x1
```

Out[38]:
```
array([[0, 1],
       [2, 3],
       [4, 5]])
```

In [39]:
```
#sum()-- sum of all the values inside an array
```

In [40]:
```
y=np.arange(0,9).reshape(3,3)
```

In [41]:
```
y
```

Out[41]:
```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

In [42]:
```
np.sum(y)
```

Out[42]:
```
36
```

In [43]:
```
np.sum(y,axis=0)#0---columnwise sum
```

Out[43]:
```
array([ 9, 12, 15])
```

In [44]:
```
np.sum(y,axis=1)#1---rowwise sum
```

Out[44]:
```
array([ 3, 12, 21])
```

In [45]:
```
np.prod(y)
```

Out[45]:
```
0
```

In [46]:
```
np.prod(y,axis=0)#0---columnwise prod
```

Out[46]:
```
array([ 0, 28, 80])
```

In [47]:
```
np.prod(y,axis=1)#1---rowwise prod
```

Out[47]:
```
array([  0,  60, 336])
```

In [48]:
```
np.min(y)
```

Out[48]:
```
0
```

In [49]:
```
np.min(y,axis=0)#0---columnwise min
```

Out[49]:
```
array([0, 1, 2])
```

In [50]:
```
np.min(y,axis=1)#1---rowwise min
```

Out[50]:
```
array([0, 3, 6])
```

In [51]:
```
np.max(y,axis=0)#0---columnwise mix
```

Out[51]:
```
array([6, 7, 8])
```

In [52]:
```
#np.nan gives null values
#nan default datatype is float
```

In [53]:
```
y1=y.astype(float)
```

In [54]:
```
y
```

Out[54]:
```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

In [55]:
```
y1[1,1]=np.nan
y1
```

Out[55]:
```
array([[ 0.,  1.,  2.],
       [ 3., nan,  5.],
       [ 6.,  7.,  8.]])
```

In [56]:
```
y1.sum(y)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_6416\3379458123.py in <module>
----> 1 y1.sum(y)

~\anaconda3\lib\site-packages\numpy\core\_methods.py in _sum(a, axis, dtype, out, keepdims, initial, where)
     46 def _sum(a, axis=None, dtype=None, out=None, keepdims=False,
     47          initial=_NoValue, where=True):
---> 48     return umr_sum(a, axis, dtype, out, keepdims, initial, where)
     49
     50 def _prod(a, axis=None, dtype=None, out=None, keepdims=False,

TypeError: only integer scalar arrays can be converted to a scalar index
```

In [57]:
```
np.sum(y1)
```

Out[57]:
```
nan
```

In [58]:
```
np.nansum(y1)
```

Out[58]:
```
32.0
```

In [59]:
```
np.nanprod(y1)
```

Out[59]:
```
0.0
```

In [60]:
```
x=np.arange(0,9).reshape(3,3)
```

In [61]:
```
x
```

Out[61]:
```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

In [62]:
```
x[np.isnan(x)]=#doubt
```

```
  File "C:\Users\user\AppData\Local\Temp\ipykernel_6416\270144752.py", line 1
    x[np.isnan(x)]=#doubt
                   ^
SyntaxError: invalid syntax
```

In [63]:
```
#np.inf
```

In [64]:
```
y1[0,0]=np.inf#doubt
y1
```

Out[64]:
```
array([[inf,  1.,  2.],
       [ 3., nan,  5.],
       [ 6.,  7.,  8.]])
```

In [65]:
```
#cummulative
```

In [66]:
```
z=np.array([1,2,3,4])
```

In [67]:
```
np.cumsum(z)
```

Out[67]:
```
array([ 1,  3,  6, 10], dtype=int32)
```

In [68]:
```
np.cumprod(z)
```

Out[68]:
```
array([ 1,  2,  6, 24], dtype=int32)
```

In [69]:
```
z=np.array([[1,2,3,4],[1,2,3,4]])
```

In [70]:
```
np.cumsum(z)
```

Out[70]:
```
array([ 1,  3,  6, 10, 11, 13, 16, 20], dtype=int32)
```

In [71]:
```
np.cumprod(z)
```

Out[71]:
```
array([  1,   2,   6,  24,  24,  48, 144, 576], dtype=int32)
```

In [74]:
```
z
```

Out[74]:
```
array([[1, 2, 3, 4],
       [1, 2, 3, 4]])
```

In [75]:
```
np.cumsum(z,axis=0)#doubt row
```

Out[75]:
```
array([[1, 2, 3, 4],
       [2, 4, 6, 8]], dtype=int32)
```

In [76]:
```
np.cumsum(z,axis=1)#doubt col
```

Out[76]:
```
array([[ 1,  3,  6, 10],
       [ 1,  3,  6, 10]], dtype=int32)
```

In [77]:
```
np.cumprod(z,axis=0)#doubt
```

Out[77]:
```
array([[ 1,  2,  3,  4],
       [ 1,  4,  9, 16]], dtype=int32)
```

In [78]:
```
np.cumprod(z,axis=1)#doubt
```

Out[78]:
```
array([[ 1,  2,  6, 24],
       [ 1,  2,  6, 24]], dtype=int32)
```

In [80]:
```
z=np.array([[1,2,3,4],[2,4,5,4]])
z
```

Out[80]:
```
array([[1, 2, 3, 4],
       [2, 4, 5, 4]])
```

In [ ]:
```

```

In [ ]:

In [ ]:

In [105]:
```
#np.nancum
#np.nanprod
```

In [106]:
```
#copy
#view
```

In [107]:
```
#flateen()
```

In [108]:
```
z.flatten()#
```

Out[108]:
```
array([1, 2, 3, 4, 1, 2, 3, 4])
```

In [109]:
```
z
```

Out[109]:
```
array([[1, 2, 3, 4],
       [1, 2, 3, 4]])
```

In [110]:
```
z.ravel()#is also use to convert into 1D array but if it is view of orignal copy
```

Out[110]:
```
array([1, 2, 3, 4, 1, 2, 3, 4])
```

In [147]:
```
x7=np.array([1.3446,4.647])
x7
```

Out[147]:
```
array([1.3446, 4.647 ])
```

In [150]:
```
z=np.round(x7,2)
```

In [151]:
```
z
```

Out[151]:
```
array([1.34, 4.65])
```

In [152]:
```
x7
```

Out[152]:
```
array([1.3446, 4.647 ])
```

In [153]:
```
x8=np.ceil(x7)
x8
```

Out[153]:
```
array([2., 5.])
```

In [118]:
```
x8=np.floor(x7)
x8
```

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

In [119]:
```
np.log(x8)
```

Out[119]:
```
array([0.        , 1.38629436])
```

In [120]:
```
x9=np.array([0,1,2,3,-9,10])
```

In [121]:
```
np.sqrt(x9)
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_7568\2146963291.py:1: RuntimeWarning: invalid value encountered in
sqrt
  np.sqrt(x9)
```

Out[121]:
```
array([0.        , 1.        , 1.41421356, 1.73205081,        nan,
       3.16227766])
```

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

In [12]:
```
x=np.arange(11,21).reshape(5,2)
y=np.arange(11,21).reshape(5,2)
```

In [13]:
```
x
```

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

In [14]:
```
y
```

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

In [16]:
```
z=np.array_equal(x,y)
z
```

Out[16]:
```
True
```

In [19]:
```
np.array_equal(x,y)
```

Out[19]:
```
True
```

In [20]:
```
x==y
```

Out[20]:
```
array([[ True,  True],
       [ True,  True],
       [ True,  True],
       [ True,  True],
       [ True,  True]])
```

In [21]:
```
x%2==0
```

Out[21]:
```
array([[False,  True],
       [False,  True],
       [False,  True],
       [False,  True],
       [False,  True]])
```

In [22]:
```
x[x%2==0]
```

Out[22]:
```
array([12, 14, 16, 18, 20])
```

In [23]:
```
y%2==0
```

Out[23]:
```
array([[False,  True],
       [False,  True],
       [False,  True],
       [False,  True],
       [False,  True]])
```

In [24]:
```
y[y%2==0]
```

Out[24]:
```
array([12, 14, 16, 18, 20])
```

In [25]:
```
#
```

In [30]:
```
p=np.array([[2,4,6,8]])
q=np.array([[1,3,5,7]])
```

In [31]:
```
p
```

Out[31]:
```
array([[2, 4, 6, 8]])
```

In [33]:
```
p.reshape(2,2)
```

Out[33]:
```
array([[2, 4],
       [6, 8]])
```

In [34]:
```
q
```

Out[34]:
```
array([[1, 3, 5, 7]])
```

In [36]:
```
q.reshape(2,2)
```

Out[36]:
```
array([[1, 3],
       [5, 7]])
```

In [40]:
```
np.logical_and(p,q)
```

Out[40]:
```
array([ True,  True,  True,  True])
```

In [41]:
```
np.logical_or(p,q)
```

Out[41]:
```
array([ True,  True,  True,  True])
```

In [63]:
```
s=np.array([3,4,5,0,3,0])
t=np.array([11,3,9,1,3,1])
```

In [64]:
```
s
```

Out[64]:
```
array([3, 4, 5, 0, 3, 0])
```

In [65]:
```
t
```

Out[65]:
```
array([11,  3,  9,  1,  3,  1])
```

In [66]:
```
np.logical_and(s,t)
```

Out[66]:
```
array([ True,  True,  True, False,  True, False])
```

In [67]:
```
np.logical_or(s,t)
```

Out[67]:
```
array([ True,  True,  True,  True,  True,  True])
```

In [68]:
```
np.any(s,t)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_5464\4044031781.py in <module>
----> 1 np.any(s,t)

<__array_function__ internals> in any(*args, **kwargs)

~\anaconda3\lib\site-packages\numpy\core\fromnumeric.py in any(a, axis, out, keepdims, where)
   2356
   2357     """
-> 2358     return _wrapreduction(a, np.logical_or, 'any', axis, None, out,
   2359                           keepdims=keepdims, where=where)
   2360

~\anaconda3\lib\site-packages\numpy\core\fromnumeric.py in _wrapreduction(obj, ufunc, method, axis, dtype, ou
t, **kwargs)
     84                 return reduction(axis=axis, out=out, **passkwargs)
     85
---> 86     return ufunc.reduce(obj, axis, dtype, out, **passkwargs)
     87
     88

TypeError: only integer scalar arrays can be converted to a scalar index
```

In [69]:
```
np.all(s%2==0)
```

Out[69]:
```
False
```

In [70]:
```
np.any(s%2==0)
```

Out[70]:
```
True
```

In [71]:
```
np.array(t%2==0)
```

Out[71]:
```
array([False, False, False, False, False, False])
```

In [72]:
```
np.all(t%2==0)
```

Out[72]:
```
False
```

In [73]:
```
np.any(t%2==0)
```

Out[73]:
```
False
```

In [74]:
```
t
```

Out[74]:
```
array([11,  3,  9,  1,  3,  1])
```

In [87]:
```
u=np.array([2,4,6,1,3,9]).reshape(2,3)
v=np.array([3,9,6,2,4,6]).reshape(3,2)
u
```

Out[87]:
```
array([[2, 4, 6],
       [1, 3, 9]])
```

In [88]:
```
v
```

Out[88]:
```
array([[3, 9],
       [6, 2],
       [4, 6]])
```

In [89]:
```
np.dot(u,v)
```

Out[89]:
```
array([[54, 62],
       [57, 69]])
```

In [90]:
```
np.sum(u)
```

Out[90]:
```
25
```

In [91]:
```
np.sum(v)
```

Out[91]:
```
30
```

In [94]:
```
np.sum(u,axis=0)
```

Out[94]:
```
array([ 3,  7, 15])
```

In [95]:
```
np.sum(u,axis=1)
```

Out[95]:
```
array([12, 13])
```

In [96]:
```
np.prod(v,axis=0)
```

Out[96]:
```
array([ 72, 108])
```

In [97]:
```
np.prod(v,axis=1)
```

Out[97]:
```
array([27, 12, 24])
```

In [102]:
```
x=np.sum(u,axis=0),np.sum(u,axis=1)
x
```

Out[102]:
```
(array([ 3,  7, 15]), array([12, 13]))
```

In [103]:
```
np.min(u)
```

Out[103]:
```
1
```

In [104]:
```
 np.min(u,axis=0)
```

Out[104]:
```
array([1, 3, 6])
```

In [105]:
```
np.min(u,axis=1)
```

Out[105]:
```
array([2, 1])
```

In [106]:
```
np.prod(u,axis=0)
```

Out[106]:
```
array([ 2, 12, 54])
```

In [107]:
```
u
```

Out[107]:
```
array([[2, 4, 6],
       [1, 3, 9]])
```

In [108]:
```
np.prod(u,axis=1)
```

Out[108]:
```
array([48, 27])
```

In [117]:
```
t=np.arange(0,9).reshape(3,3)
s=np.arange(0,9).reshape(3,3)
```

In [120]:
```
t
```

Out[120]:
```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

In [121]:
```
t[1,1]=np.nan
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_5464\1009962919.py in <module>
----> 1 t[1,1]=np.nan

ValueError: cannot convert float NaN to integer
```

In [ ]:
```

```

In [128]:
```
t1=t.astype(float)
t1
```

Out[128]:
```
array([[0., 1., 2.],
       [3., 4., 5.],
       [6., 7., 8.]])
```

In [129]:
```
t1[1,1]=np.nan
t1[0,0]=np.nan
t1
```

Out[129]:
```
array([[nan,  1.,  2.],
       [ 3., nan,  5.],
       [ 6.,  7.,  8.]])
```

In [130]:
```
np.sum(t1)
```

Out[130]:
```
nan
```

In [131]:
```
np.nansum(t1)
```

Out[131]:
```
32.0
```

In [133]:
```
np.nanprod(t1)
```

Out[133]:
```
10080.0
```

In [114]:
```
x=np.arange(0,6).reshape(3,2)
y=np.arange(0,6).reshape(3,2)
```

In [ ]: