element comparisons

```
import numpy as np
```

```
a=np.array([1,2,5,4])
b=np.array([6,2,9,4])
print(a==b)
```

```
[False  True False  True]
```

```
print(a>b)
```

```
[False False False False]
```

```
print(a<b)
```

```
[ True False  True False]
```

array wise comparisons

```
print(np.array_equal(a,b))
```

```
False
```

```
c=np.array([1,2,5,4])
print(np.array_equal(a,c))
```

```
True
```

Logical Operations

```
a=np.array([1,0,0,1],dtype='bool')
b=np.array([0,1,0,1],dtype='bool')
print(np.logical_or(a,b))
```

```
[ True  True False  True]
```

```
print(np.logical_and(a,b))
```

```
[False False False  True]
```

```
print(np.logical_not(a))
```

```
[False  True  True False]
```

Transcdental operations

```
a=np.arange(5)+1
print(np.sin(a))
```

```
[ 0.84147098  0.90929743  0.14112001 -0.7568025  -0.95892427]
```

```
Start coding or generate with AI.
```

```
print(np.log(a))
```

```
[0.         0.69314718 1.09861229 1.38629436 1.60943791]
```

print(np.log(a))

```
print(np.exp(a))
```

```
[  2.71828183   7.3890561   20.08553692  54.59815003 148.4131591 ]
```

Double-click (or enter) to edit

Shape Mismatch

```
a=np.array([1,2,3,4])
b=np.array([5,10])
print(a+b)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-18-f8e2422298ab> in <cell line: 3>()
      1 a=np.array([1,2,3,4])
      2 b=np.array([5,10])
----> 3 print(a+b)

ValueError: operands could not be broadcast together with shapes (4,) (2,)
```

Basic Reductions

```
x=np.array([1,2,3,4])
print(np.sum(x))
```

```
10
```

```
y=np.array([[1,2],[3,4]])
print(y)
print("*"*100)
print(y.T)   #transpose
```

```
[[1 2]
 [3 4]]
****************************************************************************************************
[[1 3]
 [2 4]]
```

```
print(y.sum(axis=0))#column wise sum
```

```
[4 6]
```

```
print(y.sum(axis=1))#row wise sum
```

```
[3 7]
```

```
print(y.max())
```

```
4
```

```
print(y.argmax())
```

```
3
```

```
print(y.argmin())
```

```
0
```

logical reductions

```
print(np.all([True, False, False])) #logical and
```

```
False
```

```
print(np.any([True, False, False])) #logical or
```

```
True
```

```
a=np.zeros((50,50))
print(np.any(a!=0))
```

```
False
```

Statistics

```
x=np.arange(1,10)
print(np.mean(x))
```

```
5.0
```

```
print(np.median(x))
```

```
5.0
```

```
y=np.array([[1,2,3],[4,5,6]])
print(np.mean(y,axis=0)) #column wise mean
print(np.mean(y,axis=1))#row wise mean
```

```
[2.5 3.5 4.5]
[2. 5.]
```

```
print(np.std(x))
```

```
2.581988897471611
```

Example:

```
x=np.array([1,2,3,4,5])
print(x)
```

```
[1 2 3 4 5]
```

2.Matrix creation

```
import numpy as np
x=np.arange(2,11).reshape(3,3)
print(x)
```

```
[[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]]
```

```
import numpy as np
x = np.array([[4,6],[2,1]])
print(x)
print("*"*10)
print(np.sort(x, axis=0))
print(np.sort(x, axis=1))
```

```
[[4 6]
 [2 1]]
```

```
**********
[[2 1]
 [4 6]]
[[4 6]
 [1 2]]
```

```
x= np.array([[10, 20, 30] , [40, 50, 60]])
farr=x.flatten()
print(farr)
print(farr.flags)
farr.flags['C_CONTIGUOUS']
```

```
[10 20 30 40 50 60]
  C_CONTIGUOUS : True
  F_CONTIGUOUS : True
  OWNDATA : True
  WRITEABLE : True
  ALIGNED : True
  WRITEBACKIFCOPY : False

True
```

```
# dipslay all the dates for the month of march,2017s
import datetime

# Create a datetime object for March 1, 2017
date = datetime.datetime(2017, 3, 1)

# Iterate through the days of March
while date.month == 3:
  # Print the current date
  print(date.strftime("%Y-%m-%d"))

  # Increment the date by one day
  date += datetime.timedelta(days=1)
```

```
2017-03-01
2017-03-02
2017-03-03
2017-03-04
2017-03-05
2017-03-06
2017-03-07
2017-03-08
2017-03-09
2017-03-10
2017-03-11
2017-03-12
2017-03-13
2017-03-14
2017-03-15
2017-03-16
2017-03-17
2017-03-18
2017-03-19
```

```
        2017-03-20
        2017-03-21
        2017-03-22
        2017-03-23
        2017-03-24
        2017-03-25
        2017-03-26
        2017-03-27
        2017-03-28
        2017-03-29
        2017-03-30
        2017-03-31
```