How numpy is faster than python

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

In [3]:
```
d=np.arange(1000000)
```

In [4]:
```
%time for i in range(1,10): r=[x*2 for x in l]
```
```
Wall time: 1.65 s
```

In [4]:
```
%time for i in range(1,10): r=[x*2 for x in l]
```
```
Wall time: 1.65 s
```

In [5]:
```
%time for i in range(1,10): r=d*2
```
```
Wall time: 55 ms
```

A dimension array of object

In [6]:
```
x=np.zeros((3,3))
x
```
Out[6]:
```
array([[0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.]])
```

In [7]:
```
x=np.zeros((5,5))
```

In [8]:
```
x
```
Out[8]:
```
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.]])
```

In [9]:
```
y=np.ones((4,4))
```

In [10]:
```
y
```
Out[10]:
```
array([[1., 1., 1., 1.],
       [1., 1., 1., 1.],
       [1., 1., 1., 1.],
       [1., 1., 1., 1.]])
```

In [11]:
```
y=np.ones((6,6))
```

In [12]:
```
y
```
Out[12]:
```
array([[1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1.]])
```

In [13]:
```
m=np.empty((3,4))#empty can return any thing
m
```
Out[13]:
```
array([[4.52784699e-312, 2.47032823e-322, 0.00000000e+000,
        0.00000000e+000],
       [0.00000000e+000, 4.47032019e-038, 4.51331725e-090,
        5.88362146e-062],
       [1.00567961e-047, 2.34393997e-056, 3.99910963e+252,
        4.93432906e+257]])
```

In [14]:
```
m=np.empty((4,5))#empty can return any thing
```

In [15]:
```
m
```
Out[15]:
```
array([[4.52774044e-312, 6.27463370e-322, 0.00000000e+000,
        0.00000000e+000, 0.00000000e+000],
       [5.30276956e+180, 1.57076922e-076, 4.57753266e-071,
        4.26362806e-086, 3.35809988e-143],
       [6.01433264e+175, 6.93885958e+218, 5.56218858e+180,
        3.94356143e+180, 4.25698516e+174],
       [1.71141910e-052, 1.54474561e+185, 4.79125231e-037,
        5.83031528e-144, 1.50008929e+248]])
```

In [16]:
```
l=[1,2,3,4]
a=np.array(l)
```

In [17]:
```
a
```
Out[17]:
```
array([1, 2, 3, 4])
```

In [18]:
```
lst=[25,50,57,78]
p=np.array(lst)
p
```
Out[18]:
```
array([25, 50, 57, 78])
```

In [19]:
```
n=np.arange(1,100,10)
n
```
Out[19]:
```
array([ 1, 11, 21, 31, 41, 51, 61, 71, 81, 91])
```

In [20]:
```
n.shape
```
Out[20]:
```
(10,)
```

In [21]:
```
x=np.zeros((10,10))
x
```
Out[21]:
```
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.],
       [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.],
       [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.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])
```

In [22]:
```
x.shape
```
Out[22]:
```
(10, 10)
```

In [23]:
```
a=x.reshape(2,2,5,5)
```

In [24]:
```
a
```
Out[24]:
```
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.],
         [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.],
         [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.],
         [0., 0., 0., 0., 0.],
         [0., 0., 0., 0., 0.]]]])
```

In [25]:
```
x=np.ones((2,6))
x
```
Out[25]:
```
array([[1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1.]])
```

In [26]:
```
x.reshape((3,2,2))
```
Out[26]:
```
array([[[1., 1.],
        [1., 1.]],

       [[1., 1.],
        [1., 1.]],

       [[1., 1.],
        [1., 1.]]])
```

Arithmatic with ndarray

In [27]:
```
a=np.random.randn((5))
a
```
Out[27]:
```
array([-0.80546897, -0.2464746 ,  0.20089386,  0.24402186,  0.26691438])
```

In [28]:
```
b=np.random.randn((5))
b
```
Out[28]:
```
array([ 1.14575789e+00,  6.07458489e-01, -1.66147102e+00,  4.05760297e-01,
       -1.26643808e-03])
```

In [29]:
```
a+b
```
Out[29]:
```
array([ 0.34028893,  0.36098389, -1.46057716,  0.64978215,  0.26564794])
```

In [30]:
```
a-b
```
Out[30]:
```
array([-1.95122686, -0.85393309,  1.86236487, -0.16173844,  0.26818081])
```

In [31]:
```
a*b
```
Out[31]:
```
array([-9.22872429e-01, -1.49723088e-01, -3.33779324e-01,  9.90143808e-02,
       -3.38030529e-04])
```

In [32]:
```
b*2
```
Out[32]:
```
array([ 2.29151579e+00,  1.21491698e+00, -3.32294203e+00,  8.11520595e-01,
       -2.53287616e-03])
```

In [33]:
```
a>0
```
Out[33]:
```
array([False, False,  True,  True,  True])
```

In [34]:
```
a[a>0]
```
Out[34]:
```
array([0.20089386, 0.24402186, 0.26691438])
```

In [35]:
```
a.dtype
```
Out[35]:
```
dtype('float64')
```

In [36]:
```
a.ndim
```
Out[36]:
```
1
```

Indexing and slicing

In [37]:
```
x=np.array([1,10,3,5,2])
```

In [38]:
```
x
```
Out[38]:
```
array([ 1, 10,  3,  5,  2])
```

In [39]:
```
x[4]
```
Out[39]:
```
2
```

In [40]:
```
x[x>2]
```
Out[40]:
```
array([10,  3,  5])
```

In [41]:
```
x[[3,1,2]]
```
Out[41]:
```
array([ 5, 10,  3])
```

In [42]:
```
x=np.random.randn(6,6)
x
```
Out[42]:
```
array([[ 1.31967796e-01, -1.38823382e-02, -1.16978359e+00,
         6.77169410e-01, -7.99218146e-01, -2.57661746e+00],
       [ 3.32232522e-01,  1.83009452e+00, -8.22620294e-01,
        -6.61600131e-01,  1.81846744e+00, -1.12623284e-01],
       [ 9.48863727e-04,  4.65812204e-01,  8.97956717e-01,
        -3.32840829e-02, -2.99498692e+00,  6.66919949e-01],
       [ 5.65108115e-01,  1.27150858e+00,  1.87657354e-03,
        -8.56927407e-01,  8.26814633e-01, -9.77708329e-02],
       [-3.05893257e-01, -8.88211153e-01, -2.94981807e-01,
         4.58856132e-01, -4.98740739e-01,  1.44967139e+00],
       [ 5.33602195e-01, -2.36922691e-01, -1.28248085e-01,
         1.22055143e+00, -5.72488304e-01, -1.09899051e+00]])
```

In [43]:
```
x[0]
```
Out[43]:
```
array([ 0.1319678 , -0.01388234, -1.16978359,  0.67716941, -0.79921815,
       -2.57661746])
```

sclicing

In [44]:
```
import numpy as np
x
```
Out[44]:
```
array([[ 1.31967796e-01, -1.38823382e-02, -1.16978359e+00,
         6.77169410e-01, -7.99218146e-01, -2.57661746e+00],
       [ 3.32232522e-01,  1.83009452e+00, -8.22620294e-01,
        -6.61600131e-01,  1.81846744e+00, -1.12623284e-01],
       [ 9.48863727e-04,  4.65812204e-01,  8.97956717e-01,
        -3.32840829e-02, -2.99498692e+00,  6.66919949e-01],
       [ 5.65108115e-01,  1.27150858e+00,  1.87657354e-03,
        -8.56927407e-01,  8.26814633e-01, -9.77708329e-02],
       [-3.05893257e-01, -8.88211153e-01, -2.94981807e-01,
         4.58856132e-01, -4.98740739e-01,  1.44967139e+00],
       [ 5.33602195e-01, -2.36922691e-01, -1.28248085e-01,
         1.22055143e+00, -5.72488304e-01, -1.09899051e+00]])
```

In [45]:
```
x[1:4]
```
Out[45]:
```
array([[ 3.32232522e-01,  1.83009452e+00, -8.22620294e-01,
        -6.61600131e-01,  1.81846744e+00, -1.12623284e-01],
       [ 9.48863727e-04,  4.65812204e-01,  8.97956717e-01,
        -3.32840829e-02, -2.99498692e+00,  6.66919949e-01],
       [ 5.65108115e-01,  1.27150858e+00,  1.87657354e-03,
        -8.56927407e-01,  8.26814633e-01, -9.77708329e-02]])
```

In [46]:
```
x[0:5:2]
```
Out[46]:
```
array([[ 1.31967796e-01, -1.38823382e-02, -1.16978359e+00,
         6.77169410e-01, -7.99218146e-01, -2.57661746e+00],
       [ 9.48863727e-04,  4.65812204e-01,  8.97956717e-01,
        -3.32840829e-02, -2.99498692e+00,  6.66919949e-01],
       [-3.05893257e-01, -8.88211153e-01, -2.94981807e-01,
         4.58856132e-01, -4.98740739e-01,  1.44967139e+00]])
```

In [47]:
```
x[2:,1:4]
```
Out[47]:
```
array([[ 0.4658122 ,  0.89795672, -0.03328408],
       [ 1.27150858,  0.00187657, -0.85692741],
       [-0.88821115, -0.29498181,  0.45885613],
       [-0.23692269, -0.12824808,  1.22055143]])
```

In [48]:
```
import numpy as np
x=np.ones((5,5))
x
```
Out[48]:
```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

In [49]:
```
x[2:-2,2:-2]=4
x
```
Out[49]:
```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 4., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

In [50]:
```
x
```
Out[50]:
```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 4., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

In [51]:
```
x=np.ones((5,5))
x
```
Out[51]:
```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

In [52]:
```
x[1:-1,1:-1]=0
x
```
Out[52]:
```
array([[1., 1., 1., 1., 1.],
       [1., 0., 0., 0., 1.],
       [1., 0., 0., 0., 1.],
       [1., 0., 0., 0., 1.],
       [1., 1., 1., 1., 1.]])
```

Element wise array function

In [53]:
```
x=np.array([4,6,7,2,3])
np.sqrt(x)
```
Out[53]:
```
array([2.        , 2.44948974, 2.64575131, 1.41421356, 1.73205081])
```

In [54]:
```
np.power(x,3)
```
Out[54]:
```
array([ 64, 216, 343,   8,  27], dtype=int32)
```

In [55]:
```
y=[7,5,2,8,7]
```

In [56]:
```
np.maximum(y,x)
```
Out[56]:
```
array([7, 6, 7, 8, 7])
```

np.where()

In [57]:
```
salary=np.array([0,-1,10000,25000])
np.where(salary<=0,30000,salary)
```
Out[57]:
```
array([30000, 30000, 10000, 25000])
```

In [58]:
```
np.where(salary<=0,"NOT OK","OK")
```
Out[58]:
```
array(['NOT OK', 'NOT OK', 'OK', 'OK'], dtype='<U6')
```

MATHEMATICAL &STATISTICAL METHOD

In [59]:
```
x=np.array([10,9,7,10,7])
```

In [60]:
```
x
```
Out[60]:
```
array([10,  9,  7, 10,  7])
```

In [61]:
```
x.mean()
```
Out[61]:
```
8.6
```

In [62]:
```
x.cumsum()
```
Out[62]:
```
array([10, 19, 26, 36, 43], dtype=int32)
```

In [63]:
```
x.cumprod()
```
Out[63]:
```
array([   10,    90,   630,  6300, 44100], dtype=int32)
```

In [64]:
```
y=x>6
y
```
Out[64]:
```
array([ True,  True,  True,  True,  True])
```

In [65]:
```
y.sum()
```
Out[65]:
```
5
```

In [66]:
```
y.any()
```
Out[66]:
```
True
```

In [67]:
```
y.all()
```
Out[67]:
```
True
```

In [68]:
```
y=x>7
```

In [69]:
```
y
```
Out[69]:
```
array([ True,  True, False,  True, False])
```

In [70]:
```
y.sum()
```
Out[70]:
```
3
```

In [71]:
```
y.any()
```
Out[71]:
```
True
```

In [72]:
```
y.all()
```
Out[72]:
```
False
```

In [73]:
```
x.sort()
```

In [74]:
```
x
```
Out[74]:
```
array([ 7,  7,  9, 10, 10])
```

In [75]:
```
np.unique(x)
```
Out[75]:
```
array([ 7,  9, 10])
```

In [1]:
```
l=range(1000000)
```

In [4]:
```
%time for i in range(1,10): r=[x*2 for x in l]
```
```
Wall time: 1.65 s
```

In [4]:
```
%time for i in range(1,10): r=[x*2 for x in l]
```
```
Wall time: 1.65 s
```

In [4]:
```
%time for i in range(1,10): r=[x*2 for x in l]
```
```
Wall time: 1.65 s
```