

```
In [1]: import numpy as np
```

```
In [2]: import time
```

```
In [3]: import sys
```

```
In [4]: l=range(1000)
```

```
In [5]: print(sys.getsizeof(5)*len(l))
```

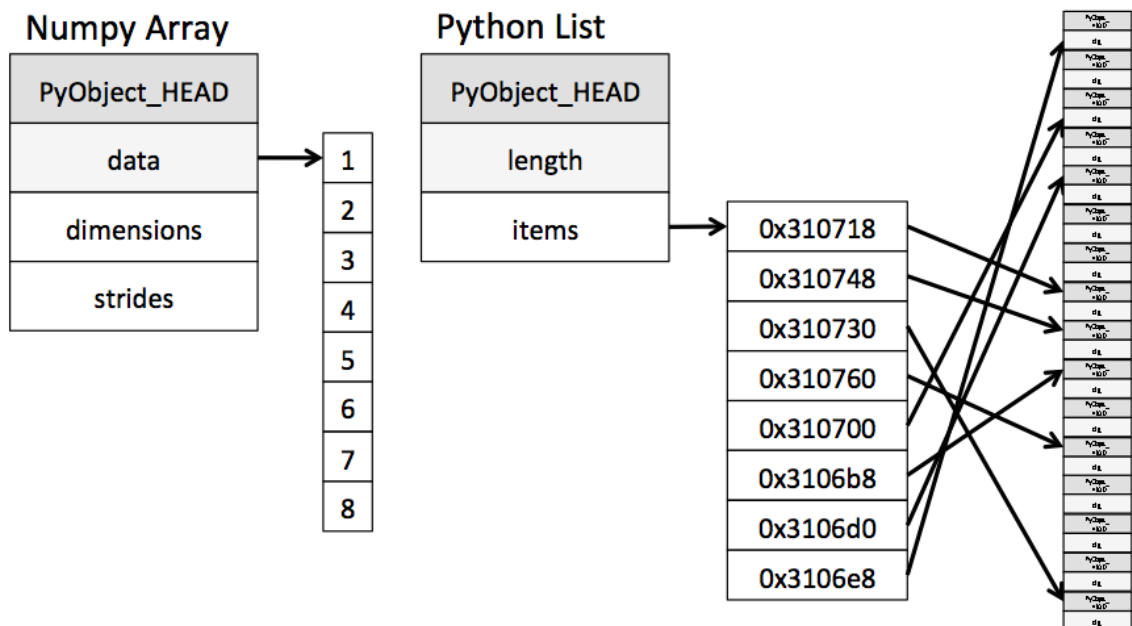
28000

```
In [6]: array = np.arange(1000)
```

```
In [7]: print(array.size*array.itemsize)
```

4000

array.size returns size of the array and array.itemsize returns size of one element
sys.getsizeof(anyone element) returns size of one element and len(l) returns list size



```
In [8]: l1=range(1000)
l2=range(1000)
```

```
In [9]: a1=np.arange(1000)
a2=np.arange(1000)
```

```
In [13]: start=time.time()
print(start)
```

1709381532.4878507

```
In [18]: print (l1)
```

range(0, 1000)

```
In [ ]:
```

```
In [ ]:   
  
In [11]: result=[x+y for x,y in zip(l1,l2)]  
  
In [12]: print((time.time()-start)*1000)  
206025.8491039276  
  
In [14]: start=time.time()  
print(start)  
1709381580.0353758  
  
In [15]: result1=a1+a2  
  
In [16]: print((time.time()-start)*1000)  
42824.413776397705  
  
In [19]: np.arange(10)  
Out[19]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])  
  
In [20]: np.arange(10, dtype=float)  
Out[20]: array([0., 1., 2., 3., 4., 5., 6., 7., 8., 9.])  
  
In [21]: a=np.array([5,6,7])  
  
In [22]: a[0]  
Out[22]: 5  
  
In [23]: a=np.array([[1,2],[3,4],[5,6]])  
  
In [24]: a[0]  
Out[24]: array([1, 2])  
  
In [25]: a.ndim  
Out[25]: 2  
  
In [26]: a.itemsize  
Out[26]: 4  
  
In [27]: a.dtype  
Out[27]: dtype('int32')  
  
In [28]: a=np.array([[1,2],[3,4],[5,6]], dtype=np.float64)  
  
In [29]: a.itemsize  
Out[29]: 8
```

```
In [30]: a[0]
```

```
Out[30]: array([1., 2.])
```

```
In [31]: a
```

```
Out[31]: array([[1., 2.],  
               [3., 4.],  
               [5., 6.]])
```

```
In [32]: a.size
```

```
Out[32]: 6
```

```
In [33]: a.shape
```

```
Out[33]: (3, 2)
```

```
1 2 3 4 5 6
```

```
In [35]: a=np.array([[1,2],[3,4],[5,6]], dtype=complex)
```

```
In [36]: a
```

```
Out[36]: array([[1.+0.j, 2.+0.j],  
               [3.+0.j, 4.+0.j],  
               [5.+0.j, 6.+0.j]])
```

```
In [38]: np.zeros((3,4))
```

```
Out[38]: array([[0., 0., 0., 0.],  
               [0., 0., 0., 0.],  
               [0., 0., 0., 0.]])
```

```
In [39]: np.ones((3,4))
```

```
Out[39]: array([[1., 1., 1., 1.],  
               [1., 1., 1., 1.],  
               [1., 1., 1., 1.]])
```

```
In [40]: l=range(5)
```

```
In [41]: l[0]
```

```
Out[41]: 0
```

```
In [42]: l[1]
```

```
Out[42]: 1
```

```
In [43]: np.arange(1,5)
```

```
Out[43]: array([1, 2, 3, 4])
```

```
In [44]: l[4]
```

```
Out[44]: 4
```

```
In [46]: l[3]
```

Out[46]: 3

In [47]: `np.arange(1,5,2)`

Out[47]: `array([1, 3])`

In [49]: *#linear sequence of numbers*
`np.linspace(1,5,10)`

Out[49]: `array([1. , 1.44444444, 1.88888889, 2.33333333, 2.77777778,
 3.22222222, 3.66666667, 4.11111111, 4.55555556, 5.])`

In [50]: `np.linspace(1,5,5)`

Out[50]: `array([1., 2., 3., 4., 5.])`

In [51]: `a=np.array([[1,2],[3,4],[5,6]])`

In [52]: `a.shape`

Out[52]: `(3, 2)`

In [53]: `a.reshape(2,3)`

Out[53]: `array([[1, 2, 3],
 [4, 5, 6]])`

In [54]: `a.reshape(6,1)`

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

In [57]: `a`

Out[57]: `array([[1, 2],
 [3, 4],
 [5, 6]])`

In [58]: `a.min()`

Out[58]: `1`

In [59]: `a.max()`

Out[59]: `6`

In []:

In [55]: `a.ravel()`

Out[55]: `array([1, 2, 3, 4, 5, 6])`

In [56]: `a`

Out[56]: `array([[1, 2],
 [3, 4],
 [5, 6]])`

```
In [60]: a.sum()
```

```
Out[60]: 21
```

```
In [61]: #axis 0 represents columns  
# axis 1 represents rows  
a.sum(axis=0)
```

```
Out[61]: array([ 9, 12])
```

```
In [62]: a.sum(axis=1)
```

```
Out[62]: array([ 3,  7, 11])
```

```
In [63]: np.sqrt(a)
```

```
Out[63]: array([[1.          , 1.41421356],  
                [1.73205081, 2.          ],  
                [2.23606798, 2.44948974]])
```

```
In [64]: a
```

```
Out[64]: array([[1, 2],  
                [3, 4],  
                [5, 6]])
```

```
In [65]: np.std(a)
```

```
Out[65]: 1.707825127659933
```

```
In [ ]: a=np.array([[1,2],[3,4],[5,6]])
```

```
In [67]: b=np.array([[1,2],[3,4],[5,6]])
```

```
In [68]: a+b
```

```
Out[68]: array([[ 2,  4],  
                [ 6,  8],  
                [10, 12]])
```

```
In [70]: a*b
```

```
Out[70]: array([[ 1,  4],  
                [ 9, 16],  
                [25, 36]])
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

In []:

In []: