# python numpy

## python list

In [4]:
```
import numpy as np
```

In [5]:
```
np.__version__
```

Out[5]:
```
'1.14.2'
```

In [7]:
```
li = list(range(10))
li
```

Out[7]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [8]:
```
li[3]
```

Out[8]:
```
3
```

In [9]:
```
li[3] = 52
```

In [10]:
```
li
```

Out[10]:
```
[0, 1, 2, 52, 4, 5, 6, 7, 8, 9]
```

In [11]:
```
li[5] = 'love'
li
```

Out[11]:
```
[0, 1, 2, 52, 4, 'love', 6, 7, 8, 9]
```

## python array

In [13]:
```
import array
```

In [14]:
```
arr = array.array('i', range(10))
```

In [15]:
```
arr
```

Out[15]:
```
array('i', [0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [16]:
```
arr[3] = 33
arr
```

Out[16]:
```
array('i', [0, 1, 2, 33, 4, 5, 6, 7, 8, 9])
```

In [17]:
```
# arr[5] = 'love'     TypeError: an integer is required (got type s
tr)
```

```
---------------------------------------------------------
TypeError                 Traceback (most recent call last)
<ipython-input-17-091c800b417e> in <module>()
----> 1 arr[5] = 'love'

TypeError: an integer is required (got type str)
```

In [18]:
```
arr2 = array.array('f', range(10))
```

In [19]:
```
arr2
```

Out[19]:
```
array('f', [0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0])
```

## numpy.ndarray

In [20]:
```
nparray = np.array(list(range(10)))
```

In [21]:
```
nparray
```

Out[21]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [22]:
```
type(nparray)
```

Out[22]:
```
numpy.ndarray
```

In [23]:
```
nparray.dtype
```

Out[23]:
```
dtype('int64')
```

In [24]:
```
nparray[3]
```

Out[24]:
```
3
```

In [25]:
```
nparray[3] = 33
```

In [26]:
```
nparray
```

Out[26]:
```
array([ 0,  1,  2, 33,  4,  5,  6,  7,  8,  9])
```

In [27]:
```
nparray[5] = 'love'
```

```
---------------------------------------------------------
ValueError                Traceback (most recent call last)
<ipython-input-27-61dc208c1817> in <module>()
----> 1 nparray[5] = 'love'

ValueError: invalid literal for int() with base 10: 'love'
```

In [28]:
```
nparray[5] = 55.8
```

In [29]:
```
nparray
```

Out[29]:
```
array([ 0,  1,  2, 33,  4, 55,  6,  7,  8,  9])
```

In [30]:
```
nparray.dtype
```

Out[30]:
```
dtype('int64')
```

In [31]:
```
nparray2 = np.array([1, 2, 3.0])
```

In [32]:
```
nparray2
```

Out[32]:
```
array([1., 2., 3.])
```

In [33]:
```
nparray2.dtype
```

Out[33]:
```
dtype('float64')
```

In [36]:
```
nparray3 = np.array([1,2,3], dtype = float)
nparray3
```

Out[36]:
```
array([1., 2., 3.])
```

In [37]:
```
nparray3.dtype
```

Out[37]:
```
dtype('float64')
```

## python array 与 numpy.ndarray 性能比较

In [40]:
```
def python_test(n):
    a = [i**2 for i in range(n)]
    b = [i**3 for i in range(n)]
    c = []
    for i in range(n):
        c.append(a[i] + b[i])
    return c
```

In [41]:
```
python_test(10)
```

Out[41]:
```
[0, 2, 12, 36, 80, 150, 252, 392, 576, 810]
```

In [44]:
```
def numpy_test(n):
    a = np.arange(n)**2
    b = np.arange(n)**3
    c = a + b
    return c
```

In [45]:
```
numpy_test(10)
```

Out[45]:
```
array([  0,   2,  12,  36,  80, 150, 252, 392, 576, 810])
```

In [46]:
```
%time res1 = python_test(10000000)
```

```
CPU times: user 11 s, sys: 1.43 s, total: 12.4 s
Wall time: 12.8 s
```

In [47]:
```
%time res2 = numpy_test(10000000)
```

```
CPU times: user 135 ms, sys: 141 ms, total: 276 ms
Wall time: 289 ms
```