In [3]:

```
import numpy as np
a=np.arange(0,60,10).reshape(-1,1)+np.arange(0,6)
a
```

Out[3]:

```
array([[ 0,  1,  2,  3,  4,  5],
       [10, 11, 12, 13, 14, 15],
       [20, 21, 22, 23, 24, 25],
       [30, 31, 32, 33, 34, 35],
       [40, 41, 42, 43, 44, 45],
       [50, 51, 52, 53, 54, 55]])
```

In [4]:

```
a[(0,1,2,3),(1,2,3,4)]
```

Out[4]:

```
array([ 1, 12, 23, 34])
```

In [6]:

```
a[3:,[1,2,5]]#第四行至第六行，然后取第二，三，六列
```

Out[6]:

```
array([[31, 32, 35],
       [41, 42, 45],
       [51, 52, 55]])
```

In [7]:

```
mask=np.array([1,0,1,0,0,1],dtype=np.bool)
a[mask,2]
```

Out[7]:

```
array([ 2, 22, 52])
```

In [11]:

```
mask1=np.array([1,0,1,0,0,1])
mask2=[True,False,True,False,False,True]
a[mask1,2]
```

Out[11]:

```
array([12,  2, 12,  2,  2, 12])
```

In [12]:

```
a[mask2,2]
```

Out[12]:

```
array([ 2, 22, 52])
```

In [13]:

```
a[mask1,2]
```

Out[13]:

```
array([12,  2, 12,  2,  2, 12])
```

In [14]:

```
a[mask2,2]
```

Out[14]:

```
array([ 2, 22, 52])
```

In [15]:

```
a[[1,2],:]
```

Out[15]:

```
array([[10, 11, 12, 13, 14, 15],
       [20, 21, 22, 23, 24, 25]])
```

In [16]:

```
a[[1,2]]
```

Out[16]:

```
array([[10, 11, 12, 13, 14, 15],
       [20, 21, 22, 23, 24, 25]])
```

In [17]:

```
x=np.array([[0,1],[2,3]])
y=np.array([[-1,-2],[-3,-4]])
a[x,y]
```

Out[17]:

```
array([[ 5, 14],
       [23, 32]])
```

In [18]:

```
palette=np.array([[0,0,0],[255,0,0],[0,255,0],[0,0,255],[255,255,255]])
image=np.array([[0,1,2,0],[0,3,4,0]])
palette[image]
```

Out[18]:

```
array([[[  0,   0,   0],
        [255,   0,   0],
        [  0, 255,   0],
        [  0,   0,   0]],

       [[  0,   0,   0],
        [  0,   0, 255],
        [255, 255, 255],
        [  0,   0,   0]]])
```

内存结构

In [26]:

```
a=np.array([[0,1,2],[2,3,4],[5,6,7]],dtype=np.float32)
a
```

Out[26]:

```
array([[ 0.,  1.,  2.],
       [ 2.,  3.,  4.],
       [ 5.,  6.,  7.]], dtype=float32)
```

In [23]:

```
b=a[::2,::2]
```

In [24]:

```
b
```

Out[24]:

```
array([[ 0.,  2.],
       [ 5.,  7.]], dtype=float32)
```

In [25]:

```
b.strides
```

Out[25]:

```
(24, 8)
```

In [27]:

```
a.strides
```

Out[27]:

```
(12, 4)
```

In [29]:

```
c=np.array([[1,2,3],[4,5,6],[7,8,9]],dtype=np.float32,order="F")
c.strides
```

Out[29]:

```
(4, 12)
```

In [32]:

```
print(a.flags)#C_CONTINGUOUS:数据存储区域是否是C语言格式的连续空间。F-CONTIGUOUS:数据存储空
```

```
  C_CONTIGUOUS : True
  F_CONTIGUOUS : False
  OWNDATA : True
  WRITEABLE : True
  ALIGNED : True
  UPDATEIFCOPY : False
```

In [33]:

```
a.T.flags
```

Out[33]:

```
  C_CONTIGUOUS : False
  F_CONTIGUOUS : True
  OWNDATA : False
  WRITEABLE : True
  ALIGNED : True
  UPDATEIFCOPY : False
```

In [34]:

```
a=np.array([[1,2,3],[4,5,6],[7,8,9]],dtype=np.float32)
b=a.view(np.uint32)
c=a.view(np.uint8)
```

In [35]:

```
b
```

Out[35]:

```
array([[1065353216, 1073741824, 1077936128],
       [1082130432, 1084227584, 1086324736],
       [1088421888, 1090519040, 1091567616]], dtype=uint32)
```

In [36]:

```
c
```

Out[36]:

```
array([[  0,   0, 128,  63,   0,   0,   0,  64,   0,   0,  64,  64],
       [  0,   0, 128,  64,   0,   0, 160,  64,   0,   0, 192,  64],
       [  0,   0, 224,  64,   0,   0,   0,  65,   0,   0,  16,  65]],
dtype=uint8)
```

In [37]:

```
a[0,0]=3.24
```

In [38]:

```
b
```

Out[38]:

```
array([[1078942761, 1073741824, 1077936128],
       [1082130432, 1084227584, 1086324736],
       [1088421888, 1090519040, 1091567616]], dtype=uint32)
```

In [39]:

```
c
```

Out[39]:

```
array([[ 41,  92,  79,  64,   0,   0,   0,  64,   0,   0,  64,  64],
       [  0,   0, 128,  64,   0,   0, 160,  64,   0,   0, 192,  64],
       [  0,   0, 224,  64,   0,   0,   0,  65,   0,   0,  16,  65]],
dtype=uint8)
```

In [41]:

```
b[0,0]
```

Out[41]:

```
1078942761
```

In [42]:

```
number=np.linspace(0.1,10,100)
y=number.astype(np.float32)
x2=y*0.5
i=y.view(np.int32)
i[:]=0x5f3759df-(i>>1)
y=y*(1.5-x2*y*y)
np.max(np.abs(1/np.sqrt(number)-y))
```

Out[42]:

```
0.0050456140410597428
```

ufunc函数

In [45]:

```
x=np.linspace(0,6,10)#linspace产生一个从0到6的随机数组
y=np.sin(x)#np.sin是一个ufanc函数
y
```

Out[45]:

```
array([ 0.        ,  0.6183698 ,  0.9719379 ,  0.90929743,  0.4572726
3,
       -0.19056796, -0.7568025 , -0.99895492, -0.81332939, -0.2794155
])
```

In [47]:

```
t=np.sin(x,out=x)#
t is x
```

Out[47]:

```
True
```

四则运算

In [49]:

```
a=np.arange(0,4)
b=np.arange(1,5)
a
```

Out[49]:

```
array([0, 1, 2, 3])
```

In [50]:

```
b
```

Out[50]:

```
array([1, 2, 3, 4])
```

In [52]:

```
c=np.add(a,b)#add()函数计算两个数组之和
c
```

Out[52]:

```
array([1, 3, 5, 7])
```

In [53]:

```
np.add(a,b,c)#指定了第三个参数out，则不产生新数组，而直接将结果保存到新数组
c
```

Out[53]:

```
array([1, 3, 5, 7])
```

In [54]:

```
np.add(a,b,b)
```

Out[54]:

```
array([1, 3, 5, 7])
```

In [64]:

```
a=np.arange(5)
b=np.arange(4,9,1)
```

In [65]:

```
a
```

Out[65]:

```
array([0, 1, 2, 3, 4])
```

In [66]:

```
b
```

Out[66]:

```
array([4, 5, 6, 7, 8])
```

In [70]:

```
np.add(a,b)
```

Out[70]:

```
array([ 4,  6,  8, 10, 12])
```

In [72]:

```
a=1
```

In [74]:

```
b=3
```

In [78]:

```
c=5
```

In [79]:

```
#计算x=a*b+c，np.add(a,b,y)相当于y+=b
x=a*b
x+=c
```

In [81]:

```
x
```

Out[81]:

```
8
```

In [82]:

```
x=a*b+c#比上面多一次内存分配
```

In [83]:

```
x
```

Out[83]:

8

In [ ]:

```

```