36.数组

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

36.1 创建方法

In [4]:
```
MyArray1 = np.arange(1,20)
MyArray1
```

Out[4]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19])
```

In [5]:
```
range(1,10,2)
```

Out[5]:
```
range(1, 10, 2)
```

In [6]:
```
list(range(1,10,2))
```

Out[6]:
```
[1, 3, 5, 7, 9]
```

In [7]:
```
np.arange(1,10,2)
```

Out[7]:
```
array([1, 3, 5, 7, 9])
```

In [8]:
```
MyArray2=np.array([1,2,3,4,3,5])
MyArray2
```

Out[8]:
```
array([1, 2, 3, 4, 3, 5])
```

In [9]:
```
np.array(range(1,10,2))
```

Out[9]:
```
array([1, 3, 5, 7, 9])
```

In [10]:
```
MyArray3=np.zeros((5,5))
MyArray3
```

Out[10]:
```
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 [11]:
```
MyArray4=np.ones((5,5))

MyArray4
```

Out[11]:
```
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 [12]:
```
np.full((3,5),2)
```

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

In [13]:
```
rand=np.random.RandomState(30)
MyArray5=rand.randint(0,100,[3,5])
MyArray5
```

Out[13]:
```
array([[37, 37, 45, 45, 12],
       [23,  2, 53, 17, 46],
       [ 3, 41,  7, 65, 49]])
```

36.2 主要特征

In [15]:
```
import numpy as np
MyArray4=np.zeros(shape=(2,15),dtype=np.int)
MyArray4
```

Out[15]:
```
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]])
```

In [16]:
```
np.ones((3,5),dtype=float)
```

Out[16]:
```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

In [17]:
```
np.ones([3,5],dtype=float)
```

Out[17]:
```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

36.3 切片与读取

In [19]:
```
import numpy as np
myArray=np.array(range(1,10))

myArray
```

Out[19]:
```
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [20]:
```
myArray=np.arange(1,10)
myArray
```

Out[20]:
```
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [21]:
```
myArray[0]
```

Out[21]:
```
1
```

In [22]:
```
myArray[-1]
```

Out[22]:
```
9
```

In [23]:
```
import numpy as np
myArray=np.array(range(0,10))
```

```
print("myArray=",myArray)
print("myArray[1:9:2]=",myArray[1:9:2])
print("myArray[:9:2]=",myArray[:9:2])

print("myArray[::2]=",myArray[::2])
print("myArray[:]=",myArray[:])
print("myArray[:8:]=",myArray[:8:])
print("myArray[:8]=",myArray[0:8])
print("myArray[4::]=",myArray[4::])
print("myArray[9:1:-2]=",myArray[9:1:-2])
print("myArray[::-2]=",myArray[::-2])
print("myArray[[2,5,6]]=",myArray[[2,5,6]])
print("myArray[myArray>5]=",myArray[myArray>5])
```

```
myArray= [0 1 2 3 4 5 6 7 8 9]
myArray[1:9:2]= [1 3 5 7]
myArray[:9:2]= [0 2 4 6 8]
myArray[::2]= [0 2 4 6 8]
myArray[:]= [0 1 2 3 4 5 6 7 8 9]
myArray[:8:]= [0 1 2 3 4 5 6 7]
myArray[:8]= [0 1 2 3 4 5 6 7]
myArray[4::]= [4 5 6 7 8 9]
myArray[9:1:-2]= [9 7 5 3]
myArray[::-2]= [9 7 5 3 1]
myArray[[2,5,6]]= [2 5 6]
myArray[myArray>5]= [6 7 8 9]
```

In [24]:
```
myArray[0:2]
```

Out[24]: array([0, 1])

In [25]:
```
myArray[1:5:2]
```

Out[25]: array([1, 3])

In [26]:
```
myArray[::2]
```

Out[26]: array([0, 2, 4, 6, 8])

In [27]:
```
myArray[::-2]
```

Out[27]: array([9, 7, 5, 3, 1])

In [28]:
```
myArray
```

Out[28]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [29]:
```
myArray=np.array(range(1,11))
myArray
```

Out[29]: array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

In [30]:
```
myArray[1,3,6]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-30-13b1cd8a6af6> in <module>()
      1 #【注意】初学者容易出现的问题
      2
----> 3 myArray[1,3,6]
```

```
4
5  #【注意】报错信息为【IndexError: too many indices for array（索引的维度过多的错误信息）】
```

IndexError: too many indices for array

In [31]:
```
myArray[[1,3,6]]
```
Out[31]: array([2, 4, 7])

In [32]:
```
myArray
```
Out[32]: array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

In [33]:
```
myArray[:,np.newaxis]
```
Out[33]:
```
array([[ 1],
       [ 2],
       [ 3],
       [ 4],
       [ 5],
       [ 6],
       [ 7],
       [ 8],
       [ 9],
       [10]])
```

In [34]:
```
myArray[:,np.newaxis].shape
```
Out[34]: (10, 1)

In [35]:
```
myArray2=np.arange(1,21).reshape([5,4])
myArray2
```
Out[35]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [13, 14, 15, 16],
       [17, 18, 19, 20]])
```

In [36]:
```
myArray2[[2,4],3]
```
Out[36]: array([12, 20])

In [37]:
```
x=[2,4]
myArray2[x,3]
```
Out[37]: array([12, 20])

36.4 浅拷贝与深拷贝

In [38]:
```
import numpy as np
myArray1=np.array(range(0,10))
myArray2=myArray1
myArray2[1]=100
myArray1
```
Out[38]: array([0, 100,   2,   3,   4,   5,   6,   7,   8,   9])

In [39]:
```
import numpy as np
myArray1=np.array(range(0,10))
myArray2=myArray1.copy()
myArray2[1]=200
myArray1
```

Out[39]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

36.5 形状与重构

In [40]:
```
import numpy as np
MyArray5=np.arange(1,21)

MyArray5
```

Out[40]: array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20])

In [41]:
```
MyArray5.shape
```

Out[41]: (20,)

In [42]:
```
MyArray6=MyArray5.reshape(4,5)
MyArray6
```

Out[42]: array([[1, 2, 3, 4, 5],
[6, 7, 8, 9, 10],
[11, 12, 13, 14, 15],
[16, 17, 18, 19, 20]])

In [43]:
```
MyArray5.shape
```

Out[43]: (20,)

In [44]:
```
MyArray5
```

Out[44]: array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20])

In [45]:
```
MyArray5.reshape(5,4)
```

Out[45]: array([[1, 2, 3, 4],
[5, 6, 7, 8],
[9, 10, 11, 12],
[13, 14, 15, 16],
[17, 18, 19, 20]])

In [46]:
```
MyArray5.reshape(5,5)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-46-8920a583f59a> in <module>()
----> 1 MyArray5.reshape(5,5)
      2
      3 #报错：ValueError: cannot reshape array of size 20 into shape (5,5)，原因分析：reshape的前提是“可以reshape”。
      4

ValueError: cannot reshape array of size 20 into shape (5,5)
```

```
In [47]: MyArray5
```

```
Out[47]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
                18, 19, 20])
```

```
In [48]: MyArray5.resize(4,5)
         MyArray5
```

```
Out[48]: array([[ 1,  2,  3,  4,  5],
                [ 6,  7,  8,  9, 10],
                [11, 12, 13, 14, 15],
                [16, 17, 18, 19, 20]])
```

```
In [49]: MyArray5.swapaxes(0,1)
```

```
Out[49]: array([[ 1,  6, 11, 16],
                [ 2,  7, 12, 17],
                [ 3,  8, 13, 18],
                [ 4,  9, 14, 19],
                [ 5, 10, 15, 20]])
```

```
In [50]: MyArray5
```

```
Out[50]: array([[ 1,  2,  3,  4,  5],
                [ 6,  7,  8,  9, 10],
                [11, 12, 13, 14, 15],
                [16, 17, 18, 19, 20]])
```

```
In [51]: MyArray5=MyArray5.swapaxes(0,1)
         MyArray5
```

```
Out[51]: array([[ 1,  6, 11, 16],
                [ 2,  7, 12, 17],
                [ 3,  8, 13, 18],
                [ 4,  9, 14, 19],
                [ 5, 10, 15, 20]])
```

```
In [52]: MyArray5.flatten()
```

```
Out[52]: array([ 1,  6, 11, 16,  2,  7, 12, 17,  3,  8, 13, 18,  4,  9, 14, 19,  5,
                10, 15, 20])
```

```
In [53]: MyArray5.tolist()
```

```
Out[53]: [[1, 6, 11, 16],
          [2, 7, 12, 17],
          [3, 8, 13, 18],
          [4, 9, 14, 19],
          [5, 10, 15, 20]]
```

```
In [54]: MyArray5.astype(np.float)
```

```
Out[54]: array([[ 1.,  6., 11., 16.],
                [ 2.,  7., 12., 17.],
                [ 3.,  8., 13., 18.],
                [ 4.,  9., 14., 19.],
                [ 5., 10., 15., 20.]])
```

```
In [55]: MyArray5
```

Out[55]:
```
array([[ 1,  6, 11, 16],
       [ 2,  7, 12, 17],
       [ 3,  8, 13, 18],
       [ 4,  9, 14, 19],
       [ 5, 10, 15, 20]])
```

36.6 属性计算

In [56]:
```
np.rank(MyArray5)
```

```
C:\Anaconda\lib\site-packages\ipykernel_launcher.py:3: VisibleDeprecatio
nWarning: `rank` is deprecated; use the `ndim` attribute or function ins
tead. To find the rank of a matrix see `numpy.linalg.matrix_rank`.
  This is separate from the ipykernel package so we can avoid doing impo
rts until
```

Out[56]: 2

In [57]:
```
np.ndim(MyArray5)
```

Out[57]: 2

In [58]:
```
MyArray5.ndim
```

Out[58]: 2

In [59]:
```
np.shape(MyArray5)
```

Out[59]: (5, 4)

In [60]:
```
MyArray5.shape
```

Out[60]: (5, 4)

In [61]:
```
MyArray5.size
```

Out[61]: 20

In [62]:
```
type(MyArray5)
```

Out[62]: numpy.ndarray

36.7 ndarray的计算

In [63]:
```
MyArray5*10
```

Out[63]:
```
array([[ 10,  60, 110, 160],
       [ 20,  70, 120, 170],
       [ 30,  80, 130, 180],
       [ 40,  90, 140, 190],
       [ 50, 100, 150, 200]])
```

In [64]:
```
x=np.array([11,12,13,14,15,16,17,18])
x1,x2,x3=np.split(x,[3,5])
print(x1,x2,x3)
```

```
[11 12 13] [14 15] [16 17 18]
```

In [65]:
```
upper,lower=np.vsplit(MyArray5.reshape(4,5),[2])
print("上半部分为\n",upper)
print("\n\n下半部分为\n",lower)
```

```
上半部分为
[[ 1  6 11 16  2]
 [ 7 12 17  3  8]]


下半部分为
[[13 18  4  9 14]
 [19  5 10 15 20]]
```

In [66]:
```
np.concatenate((lower,upper),axis=0)
```

Out[66]:
```
array([[13, 18,  4,  9, 14],
       [19,  5, 10, 15, 20],
       [ 1,  6, 11, 16,  2],
       [ 7, 12, 17,  3,  8]])
```

In [67]:
```
np.vstack([upper,lower])
```

Out[67]:
```
array([[ 1,  6, 11, 16,  2],
       [ 7, 12, 17,  3,  8],
       [13, 18,  4,  9, 14],
       [19,  5, 10, 15, 20]])
```

In [68]:
```
np.hstack([upper,lower])
```

Out[68]:
```
array([[ 1,  6, 11, 16,  2, 13, 18,  4,  9, 14],
       [ 7, 12, 17,  3,  8, 19,  5, 10, 15, 20]])
```

In [69]:
```
np.add(MyArray5,1)
```

Out[69]:
```
array([[ 2,  7, 12, 17],
       [ 3,  8, 13, 18],
       [ 4,  9, 14, 19],
       [ 5, 10, 15, 20],
       [ 6, 11, 16, 21]])
```

36.8 ndarray的元素类型

In [70]:
```
import numpy as np
np.zeros(10,dtype="int16")
```

Out[70]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int16)
```

In [71]:
```
np.zeros(10,dtype="float")
```

Out[71]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

In [72]:
```
a1=np.array([1,2,3,None])
a1
```

Out[72]:
```
array([1, 2, 3, None], dtype=object)
```

In [73]:
```
a1=np.array([1,2,3,None,np.nan])
a1
```

```
Out[73]: array([1, 2, 3, None, nan], dtype=object)
```

36.8 插入与删除

```
In [74]:
import numpy as np
myArray1=np.array([11,12,13,14,15,16,17,18])
np.delete(myArray1,2)
```

```
Out[74]: array([11, 12, 14, 15, 16, 17, 18])
```

```
In [75]:
np.insert(myArray1,1,88)
```

```
Out[75]: array([11, 88, 12, 13, 14, 15, 16, 17, 18])
```

36.9 缺失值处理

```
In [76]:
np.isnan(myArray)
```

```
Out[76]: array([False, False, False, False, False, False, False, False, False,
                False])
```

```
In [77]:
np.any(np.isnan(myArray))
```

```
Out[77]: False
```

```
In [78]:
np.all(np.isnan(myArray))
```

```
Out[78]: False
```

```
In [79]:
MyArray=np.array([1,2,3,np.nan])
np.nansum(MyArray)
```

```
Out[79]: 6.0
```

```
In [80]:
np.sum(MyArray)
```

```
Out[80]: nan
```

36.10 ndarray的广播规则

```
In [81]:
import numpy as np
A1=np.array(range(1,10)).reshape([3,3])
A1
```

```
Out[81]: array([[1, 2, 3],
                [4, 5, 6],
                [7, 8, 9]])
```

```
In [82]:
A2=np.array([10,10,10])
A2
```

```
Out[82]: array([10, 10, 10])
```

```
In [83]:
A1+A2
```

```
Out[83]: array([[11, 12, 13],
```

```
       [14, 15, 16],
       [17, 18, 19]])
```

In [84]:
```
A3=np.arange(10).reshape(2,5)

A3
```

Out[84]:
```
array([[0, 1, 2, 3, 4],
       [5, 6, 7, 8, 9]])
```

In [85]:
```
A4=np.arange(16).reshape(4,4)
A4
```

Out[85]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15]])
```

In [86]:
```
A3+A4
  #报错
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-86-0fe8480883de> in <module>()
----> 1 A3+A4
      2   #报错：ValueError: operands could not be broadcast together with shapes (2,5) (4,4)

ValueError: operands could not be broadcast together with shapes (2,5) (4,4)
```

36.11 ndarray的排序

In [87]:
```
import numpy as np
myArray=np.array([11,18,13,12,19,15,14,17,16])
myArray
```

Out[87]:
```
array([11, 18, 13, 12, 19, 15, 14, 17, 16])
```

In [88]:
```
np.sort(myArray)
```

Out[88]:
```
array([11, 12, 13, 14, 15, 16, 17, 18, 19])
```

In [89]:
```
np.argsort(myArray)
```

Out[89]:
```
array([0, 3, 2, 6, 5, 8, 7, 1, 4], dtype=int64)
```

In [90]:
```
MyArray=np.array([[21, 22, 23, 24,25],
       [35,  34,33, 32, 31],
       [ 1, 2,  3, 100, 4]])
```

In [91]:
```
np.sort(MyArray,axis=1)
```

Out[91]:
```
array([[ 21,  22,  23,  24,  25],
       [ 31,  32,  33,  34,  35],
       [  1,   2,   3,   4, 100]])
```

In [92]:
```
np.sort(MyArray,axis=0)
```

Out[92]:
```
array([[  1,   2,   3,  24,   4],
```

```
[ 21,  22,  23,  32,  25],
[ 35,  34,  33, 100,  31]])
```