

## ch02.Series和DataFrame的索引与切片

```
In [2]: import numpy as np
import pandas as pd
pd.__version__

Out[2]: '1.2.4'
```

### 一、单级索引的Series

参数个数：1个  
参数类型：一个标签，多个标签的列表，切片

```
In [4]: arr = np.random.randn(5)
sel=pd.Series(arr, index = ['a','b','c','d','e'])
sel

Out[4]: a    0.157149
b   -1.306186
c   -0.177110
d   -0.794344
e   -1.101039
dtype: float64

In [14]: sel['c'],'-----分-----隔-----符-----',sel[['b','d']], '-----分-----隔-----符-----',sel['b':'d']

Out[14]: (-0.1771104404528647,
'-----分-----隔-----符-----',
b   -1.306186
d   -0.794344
dtype: float64,
'-----分-----隔-----符-----',
b   -1.306186
c   -0.177110
d   -0.794344
dtype: float64)

In [17]: sel[2],'-----分-----隔-----符-----',sel[[1,3]], '-----分-----隔-----符-----',sel[1:3]

Out[17]: (-0.1771104404528647,
'-----分-----隔-----符-----',
b   -1.306186
d   -0.794344
dtype: float64,
'-----分-----隔-----符-----',
b   -1.306186
c   -0.177110
dtype: float64)
```

### 二、两级索引的Series

参数个数：2个  
参数类型：第一个参数控制第一级的索引，类型完全同单级Series，意思也一样

```
In [20]: se2=pd.Series(data=np.random.randn(4), index=[list("aabb"), [1,2,'字符',2]])
se2

Out[20]: a    1   -0.941015
         2   -2.046029
b  字符    0.675569
         2    1.160345
dtype: float64

In [25]: se2['a',2],'-----分-----隔-----符-----',se2['b','字符'], '-----分-----隔-----符-----'    ## 取单个元素，注意2是数字，不是字符，不用加引号

Out[25]: (-2.046028681992297,
'-----分-----隔-----符-----',
0.6755689076477567,
'-----分-----隔-----符-----')
```

```
In [24]: se2['a'],se2[:,2]    ##取a所在的两行，以及2所在的两行，由于多行，返回的是单级Series对象

Out[24]: (1   -0.941015
2   -2.046029
dtype: float64,
a   -2.046029
b    1.160345
dtype: float64)
```

### 三、单级行索引的DataFrame

- [ ]操作符

参数个数：1个  
参数类型：

对行操作：单个标签，多个标签组成的列表  
对列操作：整数切片，标签切片

区分对行索引还是对列索引，就是看参数是列表还是切片  
由于这种方法不能同时对行和列操作，所以这里不过多介绍，直接跳过

- .loc[ ]方法

参数个数：2个  
参数类型：每个参数的类型完全同Series的单个参数类型，但只能是标签索引，不能是数字索引

注：其实参数个数可以是1个，可以是2个，按顺序分别控制行索引（如果是多级也遵循这个顺序）

```
In [26]: df1=pd.DataFrame(np.random.randint(0,10,[4,2]),index=['A','B','C','D'],columns=['a','b'])
df1
```

Out[26]:

	a	b
A	4	4
B	7	9
C	1	4
D	1	4

```
In [31]: df1.loc['A':'C','a'],df1.loc[['B','D'],'a'],df1.loc['A'],df1.loc['A',:]
```

Out[31]:

```
(A    4
 B    7
 C    1
 Name: a, dtype: int32,
 B    7
 D    1
 Name: a, dtype: int32,
 a    4
 b    4
 Name: A, dtype: int32,
 a    4
 b    4
 Name: A, dtype: int32)
```

- .iloc[ ]方法

参数个数：2个  
参数类型：同上，但只能是标签索引，不能是数字索引

```
In [33]: df1.iloc[1:3,0]
```

Out[33]:

```
B    7
C    1
Name: a, dtype: int32
```

#### 四、两级行索引的DataFrame

- .loc[ ]方法

参数个数：3个  
每个参数类型：单个标签，标签列表，标签切片  
分别控制行索引的第一级，第二级，列索引,其中前两个必须括起来

```
In [34]: header=pd.Index(['体力值','种族值','战斗力'],name='属性')
list1=['男','男','男','男','女','女','女']
list2=['百里守约','项羽','猪八戒','蔡徐坤','上官婉儿','小乔','蔡文姬']
index_col=pd.MultiIndex.from_arrays([list1,list2],names=['性别','姓名'])
df2=pd.DataFrame(data=np.random.randint(1,100,[7,3]),index=index_col,columns=header)
df2
```

Out[34]:

	属性	体力值	种族值	战斗力
性别	姓名			
男	百里守约	61	7	56
	项羽	12	61	20
	猪八戒	71	77	67
	蔡徐坤	1	22	83
女	上官婉儿	94	86	13
	小乔	5	25	2
	蔡文姬	25	6	53

In [38]:

df2.loc[('男','蔡徐坤'),'战斗力']

Out[38]:

83

In [40]:

df2.loc[(['男','女'],['猪八戒','蔡徐坤','小乔']),['种族值','战斗力']]

Out[40]:

属性		种族值	战斗力
性别	姓名		
男	猪八戒	77	67
	蔡徐坤	22	83
女	小乔	25	2