

```
In [1]: import numpy as np
import pandas
import matplotlib
from pandas import DataFrame, Series
from matplotlib import pyplot as plt
```

2、DataFrame

In []:

DataFrame是一个【表格型】的数据结构。DataFrame由按一定顺序排列的多列数据组成。设计初衷是将Series的使用场景从一维扩展到多维。DataFrame既有行索引，也有列索引。

- 行索引: index
- 列索引: columns
- 值: values

1) DataFrame的创建

最常用的方法是传递一个字典来创建。DataFrame以字典的键作为每一【列】的名称，以字典的值（一个数组）作为每一列。

此外，DataFrame会自动加上每一行的索引。

使用字典创建的DataFrame后，则columns参数将不可被使用。

同Series一样，若传入的列与字典的键不匹配，则相应的值为NaN。

- 使用ndarray创建DataFrame

```
In [2]: DataFrame(data=np.random.randint(1,100,size=(3,3)),index=['a','b','c'],columns=['A','B','C'])
```

Out[2]:

	A	B	C
a	6	1	15
b	8	57	33
c	71	29	73

DataFrame 的属性有 values ,index, shape, columns,

```
In [3]: dic={
    'name':['张三','李四','王五'],
    'python':[99,99,99],
    'java':[98,98,98],
}
```

```
In [4]: DataFrame(data=dic,)
```

Out[4]:

	name	python	java
0	张三	99	98
1	李四	99	98
2	王五	99	98

根据以下考试成绩表，创建一个DataFrame，命名为df:

	张三	李四
语文	150	0
数学	150	0
英语	150	0
理综	300	0

```
In [5]: df = DataFrame(data=[[150,0],[150,0],[150,0],[150,0]], index=['语文','数学','英语','理综'], columns=['张三','李四'])
```

2) DataFrame的索引

(1) 对列进行索引

- 通过类似字典的方式 `df['q']`
- 通过属性的方式 `df.q`

可以将DataFrame的列获取为一个Series。返回的Series拥有原DataFrame相同的索引，且name属性也已经设置好了，就是相应的列名。

```
In [6]: df.张三
```

```
Out[6]: 语文    150  
        数学    150  
        英语    150  
        理综    150  
        Name: 张三, dtype: int64
```

```
In [7]: df['张三']
```

```
Out[7]: 语文    150  
        数学    150  
        英语    150  
        理综    150  
        Name: 张三, dtype: int64
```

```
In [8]: # 修改列索引  
        df.columns = ['zhangsan','lisi']
```

```
In [9]: df
```

```
Out[9]:
```

	zhangsan	lisi
语文	150	0
数学	150	0
英语	150	0
理综	150	0

```
In [10]: # 获取前两列  
df[['lisi', 'zhangsan']]
```

```
Out[10]:
```

	lisi	zhangsan
语文	0	150
数学	0	150
英语	0	150
理综	0	150

(2) 对行进行索引

- 使用.loc[]加index来进行行索引
- 使用.iloc[]加整数来进行行索引

同样返回一个Series, index为原来的columns。

```
In [11]: df.loc['数学']
```

```
Out[11]: zhangsan    150  
         lisi        0  
         Name: 数学, dtype: int64
```

```
In [12]: df.iloc[1]
```

```
Out[12]: zhangsan    150  
        lisi        0  
        Name: 数学, dtype: int64
```

```
In [13]: df.loc[['语文', '英语']]
```

```
Out[13]:
```

	zhangsan	lisi
语文	150	0
英语	150	0

(3) 对元素索引的方法

- 使用列索引
- 使用行索引(`iloc[3,1]` or `loc['C','q']`) 行索引在前，列索引在后

```
In [14]: df
```

```
Out[14]:
```

	zhangsan	lisi
语文	150	0
数学	150	0
英语	150	0
理综	150	0

```
In [15]: df.loc['理综', 'zhangsan'] # loc['横', '列']
```

```
Out[15]: 150
```

切片:

【注意】 直接用中括号时：

- 索引表示的是列索引
- 切片表示的是行切片

```
In [16]: df[0:2] #切行
```

Out[16]:

	zhangsan	lisi
语文	150	0
数学	150	0

```
In [17]: df.iloc[:,0:1] #切列
```

Out[17]:

	zhangsan
语文	150
数学	150
英语	150
理综	150

```
In [18]: df.loc[:, 'zhangsan': 'lisi'] # 切列
```

Out[18]:

	zhangsan	lisi
语文	150	0
数学	150	0
英语	150	0
理综	150	0

3) DataFrame的运算

(1) DataFrame之间的运算

同Series一样:

- 在运算中自动对齐不同索引的数据
- 如果索引不对应, 则补NaN

```
In [19]: #假设张三考试数学被发现作弊, 要记为0分, 如何实现?  
df.loc['数学', 'zhangsan'] = 0
```

```
In [20]: df
```

Out[20]:

	zhangsan	lisi
语文	150	0
数学	0	0
英语	150	0
理综	150	0

```
In [21]: df.loc[1, 'zhangsan'] = 1 # 有则修改 没有则添加
```

```
In [22]: df
```

Out[22]:

	zhangsan	lisi
语文	150.0	0.0
数学	0.0	0.0
英语	150.0	0.0
理综	150.0	0.0
1	1.0	NaN

```
In [23]: df.iloc[1,0] = 0.1 # 张三数学成绩修改 01
```

```
In [24]: df
```

Out[24]:

	zhangsan	lisi
语文	150.0	0.0
数学	0.1	0.0
英语	150.0	0.0
理综	150.0	0.0
1	1.0	NaN

```
In [25]: # lisi 所有成绩 加100  
df['lisi'] = df['lisi']+100
```

```
In [26]: df
```

Out[26]:

	zhangsan	lisi
语文	150.0	100.0
数学	0.1	100.0
英语	150.0	100.0
理综	150.0	100.0
1	1.0	NaN


```
In [27]: # 所有人 所有成绩加10  
df = df+10  
df
```

Out[27]:

	zhangsan	lisi
语文	160.0	110.0
数学	10.1	110.0
英语	160.0	110.0
理综	160.0	110.0
1	11.0	NaN

In []: