

```
In [1]: import numpy as np
import pandas as pd
pd.__version__
```

Out[1]: '1.2.4'

```
In [2]: df = pd.DataFrame(np.random.randn(10, 4))
df
```

Out[2]:

	0	1	2	3
0	-0.395800	-0.139479	0.751977	2.028461
1	-0.587758	-0.433575	-1.114608	-0.136803
2	-1.087555	0.467530	0.285237	-2.578197
3	0.382419	0.667569	0.981114	-1.019622
4	0.930965	-0.512963	-0.143839	-0.424685
5	-0.850639	-1.801774	-0.457765	-0.245743
6	0.240210	-0.422405	-1.021124	0.051285
7	-0.092449	0.326511	1.029859	2.218536
8	0.498064	-0.443838	-0.041216	1.322280
9	0.111124	0.347739	-0.862645	1.644943

一、拆分

```
In [3]: p1 = df[:3]
p2 = df[3:7]
p3 = df[7:]
p1, p2, p3
```

Out[3]:

(	0	1	2	3
0	-0.395800	-0.139479	0.751977	2.028461
1	-0.587758	-0.433575	-1.114608	-0.136803
2	-1.087555	0.467530	0.285237	-2.578197,
	0	1	2	3
3	0.382419	0.667569	0.981114	-1.019622
4	0.930965	-0.512963	-0.143839	-0.424685
5	-0.850639	-1.801774	-0.457765	-0.245743
6	0.240210	-0.422405	-1.021124	0.051285,
	0	1	2	3
7	-0.092449	0.326511	1.029859	2.218536
8	0.498064	-0.443838	-0.041216	1.322280
9	0.111124	0.347739	-0.862645	1.644943)

二、用concat实现多张表上下或左右拼接，按列表中的先后顺序拼接多张表

pandas.concat(objs, axis=0)

- objs：一个序列，序列元素为Series/DataFrame/Panel等。你也可以传入一个字典，此时字典的键将作为keys参数。
- axis：指定拼接沿着哪个轴。可以为0/'index'，表示沿着 0 轴拼接。可以为1/'columns'，表示沿着 1轴拼接。

```
In [4]: df1=pd.concat([p1,p2,p3])    ##参数是包含多个DataFrame的列表
df1
```

Out[4]:

	0	1	2	3
0	-0.395800	-0.139479	0.751977	2.028461
1	-0.587758	-0.433575	-1.114608	-0.136803
2	-1.087555	0.467530	0.285237	-2.578197
3	0.382419	0.667569	0.981114	-1.019622
4	0.930965	-0.512963	-0.143839	-0.424685
5	-0.850639	-1.801774	-0.457765	-0.245743
6	0.240210	-0.422405	-1.021124	0.051285
7	-0.092449	0.326511	1.029859	2.218536
8	0.498064	-0.443838	-0.041216	1.322280
9	0.111124	0.347739	-0.862645	1.644943

```
In [5]: df2=pd.concat([p1,p2,p3],axis=1)
df2
```

Out[5]:

	0	1	2	3	0	1	2	3	0	1	2	3
0	-0.395800	-0.139479	0.751977	2.028461	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	-0.587758	-0.433575	-1.114608	-0.136803	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	-1.087555	0.467530	0.285237	-2.578197	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	0.382419	0.667569	0.981114	-1.019622	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	0.930965	-0.512963	-0.143839	-0.424685	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	-0.850639	-1.801774	-0.457765	-0.245743	NaN	NaN	NaN	NaN
6	NaN	NaN	NaN	NaN	0.240210	-0.422405	-1.021124	0.051285	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-0.092449	0.326511	1.029859	2.218536
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.498064	-0.443838	-0.041216	1.322280
9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.111124	0.347739	-0.862645	1.644943

三、用merge实现两张表左右拼接，可以指定表1在左边还是右边，但是仅仅两张表

DataFrame.merge(right, how='inner', on=None)

参数：

- right：另一个DataFrame对象
- how：指定连接类型。可以为：
  - 'left': 左连接。只使用左边DataFrame的连接键
  - 'right': 右连接。只使用右边DataFrame的连接键
- on：一个label或者label list。它指定用作连接键的列的label。并且必须在两个DataFrame中这些label都存在。如果它为None，则默认使用两个DataFrame的列label的交集。你可以通过left\_on/right\_on分别指定两侧DataFrame对齐的连接键。

In [6]:

```
idx = pd.Index(['a','c','d','f'],name='idx1')
df = pd.DataFrame({'col1':[1,2,3,4], 'col2':[2,3,4,5], 'col3':[3,4,5,6]},index=idx)
df2 = pd.DataFrame({'col1':[4,3,2,1], 'col3':[4,5,6,3], 'col4':[1,3,3,2]},index=['a','f','g','i'])
df,df2
```

Out[6]:

	col1	col2	col3
idx1			
a	1	2	3
c	2	3	4
d	3	4	5
f	4	5	6,
	col1	col3	col4
a	4	4	1
f	3	5	3
g	2	6	3
i	1	3	2)

In [7]:

```
df.merge(df2,how='left',on='col1'), ##由于how='left'，所以表1在左边，col3_x是表1的col3
```

Out[7]:

	col1	col2	col3_x	col3_y	col4
0	1	2	3	3	2
1	2	3	4	6	3
2	3	4	5	5	3
3	4	5	6	4	1,)

In [8]:

```
df.merge(df2,how='right',on='col1')
```

Out[8]:

	col1	col2	col3_x	col3_y	col4
0	4	5	6	4	1
1	3	4	5	5	3
2	2	3	4	6	3
3	1	2	3	3	2