

```
1 import pandas as pd
```

```
2 s1 = pd.Series([1, 2], name='A')  
s1
```

```
2 0    1  
   1    2  
   Name: A, dtype: int64
```

```
3 s2 = pd.Series([1, 2], name='B')  
s2
```

```
3 0    1  
   1    2  
   Name: B, dtype: int64
```

```
4 # 合并两个Series  
pd.concat([s1, s2])
```

```
4 0    1  
   1    2  
   0    1  
   1    2  
   dtype: int64
```

```
5 # 设置索引不重复  
pd.concat([s1, s2],  
          ignore_index=True)
```

```
5 0    1  
   1    2  
   2    1  
   3    2  
   dtype: int64
```

```
6 # axis默认为0也就是上下拼接 我们设置为1左右拼接  
pd.concat([s1, s2],  
          axis=1)
```

```
6
```

	A	B
0	1	1
1	2	2

```
7 # 读取Excel 有多个sheet_name 结果为dict  
df_dict = pd.read_excel(  
    'concat_0.xlsx',  
    sheet_name=None  
)  
df_dict
```

```
7 OrderedDict([('1月',  
               月份  货号  销量  收入  
               0    1  aaa  10  100  
               1    1  bbb  10  100),  
              ('2月',  
               月份  货号  销量  收入  
               0    2  aaa  20  200  
               1    2  bbb  20  200),  
              ('3月',  
               月份  货号  销量  收入
```

```
0 3 aaa 30 300
1 3 bbb 30 300))
```

```
8 df_dict['1月']
```

8

	月份	货号	销量	收入
0	1	aaa	10	100
1	1	bbb	10	100

```
9 df_dict['2月']
```

9

	月份	货号	销量	收入
0	2	aaa	20	200
1	2	bbb	20	200

```
10 df_dict['3月']
```

10

	月份	货号	销量	收入
0	3	aaa	30	300
1	3	bbb	30	300

```
11 # 拼接三个DF
pd.concat([
    df_dict['1月'],
    df_dict['2月'],
    df_dict['3月'],
], ignore_index=True)
```

11

	月份	货号	销量	收入
0	1	aaa	10	100
1	1	bbb	10	100
2	2	aaa	20	200
3	2	bbb	20	200
4	3	aaa	30	300
5	3	bbb	30	300

```
12 # 拼接三个DF设置索引
df = pd.concat([
    df_dict['1月'],
    df_dict['2月'],
    df_dict['3月'],
], keys=['1月', '2月', '3月'])
df
```

12

		月份	货号	销量	收入
1月	0	1	aaa	10	100
	1	1	bbb	10	100
2月	0	2	aaa	20	200
	1	2	bbb	20	200
3月	0	3	aaa	30	300
	1	3	bbb	30	300

13 df.loc['2月']

13

	月份	货号	销量	收入
0	2	aaa	20	200
1	2	bbb	20	200

14 # 直接传入dict 进行拼接 结果是一样的
df1 = pd.concat(df_dict)
df1

14

		月份	货号	销量	收入
1月	0	1	aaa	10	100
	1	1	bbb	10	100
2月	0	2	aaa	20	200
	1	2	bbb	20	200
3月	0	3	aaa	30	300
	1	3	bbb	30	300

15 # 查看结果是否相同 发现全部为True
df == df1

15

		月份	货号	销量	收入
1月	0	True	True	True	True
	1	True	True	True	True
2月	0	True	True	True	True
	1	True	True	True	True
3月	0	True	True	True	True
	1	True	True	True	True

```
16 # 读取Excel 指定第0列为索引
df_dict = pd.read_excel(
    'concat_1.xlsx',
    sheet_name=None,
    index_col=[0]
)
df_dict
```

```
16 OrderedDict([('销量',
                  销量
                  货号
                  bbb 2
                  aaa 1),
                ('成本',
                  成本
                  货号
                  aaa 10
                  bbb 20),
                ('库存',
                  库存
                  货号
                  bbb 200
                  ccc 300)])
```

```
17 # 单独取出是一个DF
df_dict['销量']
```

17

	销量
货号	
bbb	2
aaa	1

```
18 # 左右合并 默认并集
df = pd.concat([
    df_dict['销量'],
    df_dict['成本'],
    df_dict['库存'],
], axis=1)
```

d:\users\wyx\appdata\local\programs\python\python37\lib\site-packages\ipykernel_launcher.py:6: FutureWarning: of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

```
19 # 左右合并 指定join 为交集
pd.concat([
    df_dict['销量'],
    df_dict['成本'],
    df_dict['库存'],
], axis=1, join='inner')
```

19

	销量	成本	库存
货号			
bbb	2	20	200

20 df

20

	销量	成本	库存
aaa	1.0	10.0	NaN
bbb	2.0	20.0	200.0
ccc	NaN	NaN	300.0

```
21 # DF和Series进行合并 会有重复索引
pd.concat([df, df['库存']],
          axis=1)
```

21

	销量	成本	库存	库存
aaa	1.0	10.0	NaN	NaN
bbb	2.0	20.0	200.0	200.0
ccc	NaN	NaN	300.0	300.0

```
22 # 检查是否有重复索引 有的话抛出异常
pd.concat([df, df['库存']],
          axis=1,
          verify_integrity=True)
```

ValueError

Traceback (most recent call last)

```
<ipython-input-22-3ec4224b11e9> in <module>
```

```
2 pd.concat([df, df['库存']],
```

```
3         axis=1,
```

```
----> 4         verify_integrity=True)
```

```
d:\users\wyx\appdata\local\programs\python\python37\lib\site-packages\pandas\core\reshape\concat.py in
```

```
226         keys=keys, levels=levels, names=names,
```

```
227         verify_integrity=verify_integrity,
```

```
--> 228         copy=copy, sort=sort)
```

```
229     return op.get_result()
```

```
230
```

```
d:\users\wyx\appdata\local\programs\python\python37\lib\site-packages\pandas\core\reshape\concat.py in
```

```
379     self.copy = copy
```

```
380
```

```
--> 381     self.new_axes = self._get_new_axes()
```

```
382
```

```
383     def get_result(self):
```

```
d:\users\wyx\appdata\local\programs\python\python37\lib\site-packages\pandas\core\reshape\concat.py in
```

```
459         new_axes[i] = ax
```

```
460
```

```
--> 461     new_axes[self.axis] = self._get_concat_axis()
```

```
462     return new_axes
```

```
463
```

```
d:\users\wyx\appdata\local\programs\python\python37\lib\site-packages\pandas\core\reshape\concat.py in
```

```
517         self.levels, self.names)
```

```
518
```

```
--> 519     self._maybe_check_integrity(concat_axis)
```

```
520
```

```
521     return concat_axis
```

```
d:\users\wyx\appdata\local\programs\python\python37\lib\site-packages\pandas\core\reshape\concat.py in
    526         overlap = concat_index[concat_index.duplicated()].unique()
    527         raise ValueError('Indexes have overlapping values: '
--> 528                             '{overlap!s}'.format(overlap=overlap))
    529
    530
```

ValueError: Indexes have overlapping values: Index(['库存'], dtype='object')