



AniPython

连接/合并表格

`join()` 函数 左右合并


```
DataFrame.join(  
    self,  
    other, # Series/DataFrame/list of DataFrame  
    how = 'left', # {'left', 'right', 'outer', 'inner'}  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```



```
DataFrame.join(  
    self,  
    other,  
    how = 'left',  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```



```
DataFrame.join(  
    self,  
    other = Series,  
    how = 'left',  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```

	销量
AAA	1
BBB	2

AAA	10
BBB	20
Name:成本 dtype: int64	


```
DataFrame.join(  
    self,  
    other = Series,  
    how = 'left',  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```

	销量
AAA	1
BBB	2

AAA	10
BBB	20
Name:成本 dtype: int64	



	成本
AAA	10
BBB	20


```
DataFrame.join(  
    self,  
    other = Series,  
    how = 'left',  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```

	销量
AAA	1
BBB	2

	成本
AAA	10
BBB	20


```
DataFrame.join(  
    self,  
    other = Series,  
    how = 'left',  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```

	销量	成本
AAA	1	10
BBB	2	20


```
DataFrame.join(  
    self,  
    other = DataFrame,  
    how = 'left',  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```

	销量
AAA	1
BBB	2

	成本
AAA	10
BBB	20


```
DataFrame.join(  
    self,  
    other = DataFrame,  
    how = 'left',  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```

	销量	成本
AAA	1	10
BBB	2	20


```
DataFrame.join(  
    self,  
    other = DataFrame,  
    how = 'left',  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```

	销量
AAA	1
BBB	2

	成本
BBB	20
CCC	30


```
DataFrame.join(  
    self,  
    other = DataFrame,  
    how = 'outer',  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```

	销量
AAA	1
BBB	2

	成本
BBB	20
CCC	30


```
DataFrame.join(  
    self,  
    other = DataFrame,  
    how = 'outer',  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```

	销量
AAA	1
BBB	2

	成本
BBB	20
CCC	30

	销量	成本
AAA	1.0	NaN
BBB	2.0	20.0
CCC	NaN	30.0

outer


```
DataFrame.join(  
    self,  
    other = DataFrame,  
    how = 'left',  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```

	销量
AAA	1
BBB	2

	成本
BBB	20
CCC	30

left

	销量	成本
AAA	1.0	NaN
BBB	2.0	20.0
CCC	NaN	30.0

outer


```
DataFrame.join(  
    self,  
    other = DataFrame,  
    how = 'right',  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```

	销量
AAA	1
BBB	2

	成本
BBB	20
CCC	30

	销量	成本
AAA	1.0	NaN
BBB	2.0	20.0
CCC	NaN	30.0

left

right

outer


```
DataFrame.join(  
    self,  
    other = DataFrame,  
    how = 'inner',  
    lsuffix = '',  
    rsuffix = '',  
    sort = False  
) -> <DataFrame>
```

	销量
AAA	1
BBB	2

	成本
BBB	20
CCC	30

	销量	成本
AAA	1.0	NaN
BBB	2.0	20.0
CCC	NaN	30.0

Diagram illustrating the result of an inner join operation. The resulting DataFrame contains only the rows where both '销量' (Sales) and '成本' (Cost) are present. The row for 'BBB' is highlighted, showing '销量' as 2.0 and '成本' as 20.0. The row for 'AAA' has '销量' as 1.0 and '成本' as NaN. The row for 'CCC' has '销量' as NaN and '成本' as 30.0. The 'left' DataFrame (销量) is highlighted in yellow, the 'right' DataFrame (成本) is highlighted in purple, and the 'inner' join result is highlighted in green. The 'outer' join result is highlighted in red.