

pandas 操作 dataframe 使用 drop() 函数 删除一行或一列

作者: [ITCrowed](#)

原文链接: <https://ld246.com/article/1572803096859>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



drop()函数的用法: `DataFrame.drop(labels=None,axis=0, index=None, columns=None, inplace=False)`

默认参数 `axis=0`, 表示对行进行操作, 如需对列进行操作需要更改默认参数为 `axis=1`,

默认参数 `inplace=False`, 表示该删除操作不改变原数据, 而是返回一个执行删除操作后的新 dataframe, 如需直接在原数据上进行删除操作, 需要更改默认参数为 `inplace=True`

参数说明

labels	就是要删除的行列的名字, 用列表给定
axis 指定axis=1;	默认为0, 指删除行, 因此删除columns时
index	直接指定要删除的行
columns	直接指定要删除的列
inplace=False, 而是返回一个执行删除操作后的 新dataframe;	默认该删除操作不改变原数据
inplace=True, , 删除后无法返回。	则会直接在原数据上进行删除操

因此, 删除行列有两种方式:

1. labels=None,axis=0 的组合
2. index或columns直接指定要删除的行或列

删除Series的一个元素

```
series = Series([4.5,7.2,-5.3,3.6], index=['A','B','C','D'])
```

```
series .drop('C')
output:
A 4.5
B 7.2
D 3.6
```

删除DataFrame的行或列

```
>>> df = pd.DataFrame(np.arange(12).reshape(3,4), columns=['A', 'B', 'C', 'D'])
>>> df
   A  B  C  D
0  0  1  2  3
1  4  5  6  7
2  8  9 10 11
```

#Drop columns, 两种方法等价

```
>>> df.drop(['B', 'C'], axis=1)
```

```
   A  D
0  0  3
1  4  7
2  8 11
```

```
>>> df.drop(columns=['B', 'C'])
```

```
   A  D
0  0  3
1  4  7
2  8 11
```

第一种方法下删除column一定要指定axis=1,否则会报错, 如下

```
>>> df.drop(['B', 'C'])
```

```
ValueError: labels ['B' 'C'] not contained in axis
```

#Drop rows

```
>>> df.drop([0, 1])
```

```
   A  B  C  D
2  8  9 10 11
```

```
>>> df.drop(index=[0, 1])
```

```
   A  B  C  D
2  8  9 10 11
```

说明

.drop() 返回的是一个新对象, 原对象并不会被改变。