

## Series

```
import pandas as pd
from pandas import Series, DataFrame
```

```
In [11]: obj = pd.Series([4, 7, -5, 3])
```

```
In [12]: obj
```

```
0    4
1    7
2   -5
3    3
dtype: int64
```

```
In [13]: obj.values
```

```
array([ 4,  7, -5,  3])
```

```
In [14]: obj.index
```

```
RangeIndex(start=0, stop=4, step=1)
```

```
In [15]: obj2 = pd.Series([4, 7, -5, 3], index=['d', 'b', 'a', 'c'])
```

```
In [16]: obj2
```

```
d    4
b    7
a   -5
c    3
dtype: int64
```

```
In [17]: obj2.index
```

```
Index(['d', 'b', 'a', 'c'], dtype='object')
```

```
In [18]: obj2['d']
```

```
4
```

```
In [19]: obj2['d'] = 6
```

```
In [20]: obj2[['c', 'a', 'd']]
```

```
c    3
a   -5
```

```
d      6
dtype: int64
```

```
In [21]: obj2[obj2 > 3]
```

```
d      6
b      7
dtype: int64
```

```
In [22]: obj2 * 2
```

```
d      12
b      14
a     -10
c       6
dtype: int64
```

```
In [24]: 'b' in obj2
```

```
True
```

```
In [25]: 'e' in obj2
```

```
False
```

```
In [26]: sdata = {'Ohio': 35000, 'Texas': 71000, 'Oregon': 16000, 'Utah': 5000}
```

```
In [27]: obj3 = pd.Series(sdata)
```

```
In [28]: obj3
```

```
Ohio      35000
Texas     71000
Oregon    16000
Utah       5000
dtype: int64
```

```
In [29]: states = ['California', 'Texas', 'Oregon', 'Utah', 'Ohio']
```

```
In [30]: obj4 = pd.Series(sdata, index=states)
```

```
In [31]: obj4
```

```
California    NaN
Texas         71000.0
Oregon        16000.0
Utah           5000.0
Ohio          35000.0
dtype: float64
```

```
In [32]: pd.isnull(obj4)
```

```

California    True
Texas         False
Oregon        False
Utah          False
Ohio          False
dtype: bool

```

In [33]: `pd.notnull(obj4)`

```

California    False
Texas         True
Oregon        True
Utah          True
Ohio          True
dtype: bool

```

In [37]: `obj3 + obj4`

```

California    NaN
Ohio          70000.0
Oregon        32000.0
Texas         142000.0
Utah          10000.0
dtype: float64

```

In [38]: `obj4.name = 'population'`

In [39]: `obj4.index.name = 'state'`

In [40]: `obj4`

```

state
California    NaN
Texas         71000.0
Oregon        16000.0
Utah          5000.0
Ohio          35000.0
Name: population, dtype: float64

```

In [41]: `obj`

```

0    4
1    7
2   -5
3    3
dtype: int64

```

In [42]: `obj.index = ['Bob', 'Steve', 'Jeff', 'Ryan']`

In [43]: `obj`

```

Bob         4
Steve       7
Jeff       -5

```

```
Ryan      3  
dtype: int64
```

## DataFrame

```
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada', 'Nevada'],  
        'year': [2000, 2001, 2002, 2001, 2002, 2003],  
        'pop': [1.5, 1.7, 3.6, 2.4, 2.9, 3.2]}  
frame = pd.DataFrame(data)  
frame
```

	state	year	pop
0	Ohio	2000	1.5
1	Ohio	2001	1.7
2	Ohio	2002	3.6
3	Nevada	2001	2.4
4	Nevada	2002	2.9
5	Nevada	2003	3.2

```
In [46]: frame.tail()
```

	state	year	pop
1	Ohio	2001	1.7
2	Ohio	2002	3.6
3	Nevada	2001	2.4
4	Nevada	2002	2.9
5	Nevada	2003	3.2

```
In [47]: pd.DataFrame(data, columns=['pop', 'state', 'year'])
```

```

In [48]: frame2 = pd.DataFrame(data, columns=['year', 'state', 'debt', 'pop'],
.....: index=['one', 'two', 'three', 'four',
.....: 'five', 'six'])

```

```
frame2
```

	year	state	debt	pop
<b>one</b>	2000	Ohio	NaN	1.5
<b>two</b>	2001	Ohio	NaN	1.7
<b>three</b>	2002	Ohio	NaN	3.6
<b>four</b>	2001	Nevada	NaN	2.4
<b>five</b>	2002	Nevada	NaN	2.9
<b>six</b>	2003	Nevada	NaN	3.2

```
frame2.columns
```

```
Index(['year', 'state', 'debt', 'pop'], dtype='object')
```

```
In [51]: frame2['state']
```

```

one      Ohio
two      Ohio
three    Ohio
four     Nevada
five     Nevada
six      Nevada
Name: state, dtype: object

```

```
In [52]: frame2.debt
```

```

one      NaN
two      NaN
three    NaN
four     NaN
five     NaN
six      NaN
Name: debt, dtype: object

```

```
In [53]: frame2.loc['three']
```

```

year      2002
state     Ohio
debt      NaN

```

```
pop      3.6
Name: three, dtype: object
```

```
In [54]: frame2['debt'] = 16.5
```

```
In [55]: frame2
```

	year	state	debt	pop
<b>one</b>	2000	Ohio	16.5	1.5
<b>two</b>	2001	Ohio	16.5	1.7
<b>three</b>	2002	Ohio	16.5	3.6
<b>four</b>	2001	Nevada	16.5	2.4
<b>five</b>	2002	Nevada	16.5	2.9
<b>six</b>	2003	Nevada	16.5	3.2

```
import numpy as np
```

```
In [56]: frame2['debt'] = np.arange(6.)
```

```
In [57]: frame2
```

	year	state	debt	pop
<b>one</b>	2000	Ohio	0.0	1.5
<b>two</b>	2001	Ohio	1.0	1.7
<b>three</b>	2002	Ohio	2.0	3.6
<b>four</b>	2001	Nevada	3.0	2.4
<b>five</b>	2002	Nevada	4.0	2.9
<b>six</b>	2003	Nevada	5.0	3.2

```
In [58]: val = pd.Series([-1.2, -1.5, -1.7], index=['two', 'four', 'five'])
```

```
In [59]: frame2['debt'] = val
```

```
In [60]: frame2
```

	year	state	debt	pop
<b>one</b>	2000	Ohio	NaN	1.5
<b>two</b>	2001	Ohio	-1.2	1.7
<b>three</b>	2002	Ohio	NaN	3.6
<b>four</b>	2001	Nevada	-1.5	2.4
<b>five</b>	2002	Nevada	-1.7	2.9
<b>six</b>	2003	Nevada	NaN	3.2

```
In [61]: frame2['eastern'] = frame2.state == 'Ohio'
frame2
```

	year	state	debt	pop	eastern
<b>one</b>	2000	Ohio	NaN	1.5	True
<b>two</b>	2001	Ohio	-1.2	1.7	True
<b>three</b>	2002	Ohio	NaN	3.6	True
<b>four</b>	2001	Nevada	-1.5	2.4	False
<b>five</b>	2002	Nevada	-1.7	2.9	False
<b>six</b>	2003	Nevada	NaN	3.2	False

```
In [63]: del frame2['eastern']
```

```
In [64]: frame2.columns
```

```
Index(['year', 'state', 'debt', 'pop'], dtype='object')
```

```
In [65]: pop = {'Nevada': {2001: 2.4, 2002: 2.9},
....: 'Ohio': {2000: 1.5, 2001: 1.7, 2002: 3.6}}
```

```
In [66]: frame3 = pd.DataFrame(pop)
```

```
In [67]: frame3
```

	Nevada	Ohio
<b>2001</b>	2.4	1.7
<b>2002</b>	2.9	3.6
<b>2000</b>	NaN	1.5

```
In [68]: frame3.T
```

```
In [69]: pd.DataFrame(pop, index=[2001, 2002, 2003])
```

```
In [70]: pdata = {'Ohio': frame3['Ohio'][1:2],
....: 'Nevada': frame3['Nevada'][2:3]}
```

```
In [71]: pd.DataFrame(pdata)
```

	Ohio	Nevada
<b>2000</b>	NaN	NaN
<b>2002</b>	3.6	NaN

```
In [72]: frame3.index.name = 'year'; frame3.columns.name = 'state'
In [73]: frame3
```

	state	Nevada	Ohio
year			
2001		2.4	1.7
2002		2.9	3.6
2000		NaN	1.5

```
In [74]: frame3.values
```

```
array([[2.4, 1.7],
       [2.9, 3.6],
       [nan, 1.5]])
```

## Index Objects

```
In [76]: obj = pd.Series(range(3), index=['a', 'b', 'c'])
In [77]: index = obj.index
In [78]: index
```

```
Index(['a', 'b', 'c'], dtype='object')
```

```
In [79]: index[1:]
```

```
Index(['b', 'c'], dtype='object')
```

```
In [87]: 'Ohio' in frame3.columns
```

```
True
```

```
In [88]: 2003 in frame3.index
```

```
False
```

```
In [89]: dup_labels = pd.Index(['foo', 'foo', 'bar', 'bar'])
In [90]: dup_labels
```

```
Index(['foo', 'foo', 'bar', 'bar'], dtype='object')
```

## Reindexing

```
In [91]: obj = pd.Series([4.5, 7.2, -5.3, 3.6], index=['d', 'b', 'a', 'c'])
```



```
In [92]: obj
```

```

d    4.5
b    7.2
a   -5.3
c    3.6
dtype: float64
```

```
In [93]: obj2 = obj.reindex(['a', 'b', 'c', 'd', 'e'])
```

```
In [94]: obj2
```

```

a   -5.3
b    7.2
c    3.6
d    4.5
e     NaN
dtype: float64
```

```
In [95]: obj3 = pd.Series(['blue', 'purple', 'yellow'], index=[0, 2, 4])
```

```
In [96]: obj3
```

```

0    blue
2   purple
4   yellow
dtype: object
```

```
In [97]: obj3.reindex(range(6), method='ffill')
```

```

0    blue
1    blue
2   purple
3   purple
4   yellow
5   yellow
dtype: object
```

```
import numpy as np
```

```
In [98]: frame = pd.DataFrame(np.arange(9).reshape((3, 3)), index=['a', 'c', 'd'], columns=['
```

```
In [99]: frame
```

	Ohio	Texas	California
a	0	1	2
c	3	4	5
d	6	7	8

```
In [100]: frame2 = frame.reindex(['a', 'b', 'c', 'd'])
```

```
In [101]: frame2
```

	Ohio	Texas	California
<b>a</b>	0.0	1.0	2.0
<b>b</b>	NaN	NaN	NaN
<b>c</b>	3.0	4.0	5.0
<b>d</b>	6.0	7.0	8.0

```
In [102]: states = ['Texas', 'Utah', 'California']
```

```
In [103]: frame.reindex(columns=states)
```

## Dropping Entries from an Axis

```
In [105]: obj = pd.Series(np.arange(5.), index=['a', 'b', 'c', 'd', 'e'])
```

```
In [106]: obj
```

```
a    0.0
b    1.0
c    2.0
d    3.0
e    4.0
dtype: float64
```

```
In [107]: new_obj = obj.drop('c')
```

```
In [108]: new_obj
```

```
a    0.0
b    1.0
d    3.0
e    4.0
dtype: float64
```

```
In [110]: data = pd.DataFrame(np.arange(16).reshape((4, 4)),
```

```
.....: index=['Ohio', 'Colorado', 'Utah', 'New York'],
```

```
.....: columns=['one', 'two', 'three', 'four'])
```

```
In [111]: data
```

	one	two	three	four
<b>Ohio</b>	0	1	2	3
<b>Colorado</b>	4	5	6	7
<b>Utah</b>	8	9	10	11
<b>New York</b>	12	13	14	15

```
In [112]: data.drop(['Colorado', 'Ohio'])
```

	one	two	three	four
<b>Utah</b>	8	9	10	11
<b>New York</b>	12	13	14	15

```
In [114]: data.drop(['two', 'four'], axis='columns')
```

	one	three
<b>Ohio</b>	0	2
<b>Colorado</b>	4	6
<b>Utah</b>	8	10
<b>New York</b>	12	14

```
obj.drop('b')
```

```
c    2.0
e    4.0
dtype: float64
```

## Indexing, Selection, and Filtering

```
In [117]: obj = pd.Series(np.arange(4.), index=['a', 'b', 'c', 'd'])
```

```
In [118]: obj
```

```
a    0.0
b    1.0
c    2.0
d    3.0
dtype: float64
```

```
In [121]: obj[1:3]
```

```
b    1.0
c    2.0
dtype: float64
```

```
In [123]: obj[[1, 3]]
```

```
b    1.0
d    3.0
dtype: float64
```

```
In [124]: obj[obj < 3]
```

```
a    0.0
b    1.0
c    2.0
dtype: float64
```

```
In [125]: obj['a':'c']
```

```
a    0.0
b    1.0
c    2.0
dtype: float64
```

```
In [126]: obj['b':'c'] = 5
```

```
In [127]: obj
```

```
a    0.0
b    5.0
c    5.0
d    3.0
dtype: float64
```

```
In [128]: data = pd.DataFrame(np.arange(16).reshape((4, 4)),
.....: index=['Ohio', 'Colorado', 'Utah', 'New York'],
.....: columns=['one', 'two', 'three', 'four'])
```

```
In [129]: data
```

	one	two	three	four
<b>Ohio</b>	0	1	2	3
<b>Colorado</b>	4	5	6	7
<b>Utah</b>	8	9	10	11
<b>New York</b>	12	13	14	15

```
In [130]: data['two']
```

```
Ohio    1
Colorado 5
Utah    9
New York 13
Name: two, dtype: int64
```

```
In [131]: data[['three', 'one']]
```

	three	one
<b>Ohio</b>	2	0
<b>Colorado</b>	6	4
<b>Utah</b>	10	8

In [132]: data[:2]

	one	two	three	four
<b>Ohio</b>	0	1	2	3
<b>Colorado</b>	4	5	6	7

In [133]: data[data['three'] > 5]

	one	two	three	four
<b>Colorado</b>	4	5	6	7
<b>Utah</b>	8	9	10	11
<b>New York</b>	12	13	14	15

In [134]: data < 5

	one	two	three	four
<b>Ohio</b>	True	True	True	True
<b>Colorado</b>	True	False	False	False
<b>Utah</b>	False	False	False	False
<b>New York</b>	False	False	False	False

In [135]: data[data < 5] = 0

In [136]: data

	one	two	three	four
<b>Ohio</b>	0	0	0	0
<b>Colorado</b>	0	5	6	7
<b>Utah</b>	8	9	10	11
<b>New York</b>	12	13	14	15

## Selection with loc and iloc

```
In [137]: data.loc['Colorado', ['two', 'three']]
```

```
two      5
three     6
Name: Colorado, dtype: int64
```

```
In [138]: data.iloc[2, [3, 0, 1]]
```

```
four     11
one       8
two       9
Name: Utah, dtype: int64
```

```
In [142]: data.iloc[:, :3][data.three > 5]
```

## Integer Indexes

```
ser = pd.Series(np.arange(3.))
ser
```

```
0    0.0
1    1.0
2    2.0
dtype: float64
```

```
ser[-1]
```

```
-----
ValueError                                Traceback (most recent call last)
/usr/local/lib/python3.6/dist-packages/pandas/core/indexes/range.py in get_loc(self,
key, method, tolerance)
-----
```

```
In [145]: ser2 = pd.Series(np.arange(3.), index=['a', 'b', 'c'])
```

```
In [146]: ser2[-1]
```

```
2.0
```

The above exception was the direct cause of the following exception:

```
In [147]: ser[:1]
```

```
0    0.0
dtype: float64
key, method, tolerance)
```

```
In [148]: ser.loc[:1]
```

```
0    0.0
1    1.0
dtype: float64
KeyError: -1
```

## Arithmetic and Data Alignment

```
In [150]: s1 = pd.Series([7.3, -2.5, 3.4, 1.5], index=['a', 'c', 'd', 'e'])
```

```
In [151]: s2 = pd.Series([-2.1, 3.6, -1.5, 4, 3.1],
```

```
.....: index=['a', 'c', 'e', 'f', 'g'])
```

```
In [152]: s1
```

```
a    7.3
c   -2.5
d    3.4
e    1.5
dtype: float64
```

```
s2
```

```
a   -2.1
c    3.6
e   -1.5
f    4.0
g    3.1
dtype: float64
```

```
In [154]: s1 + s2
```

```
a    5.2
c    1.1
d    NaN
e    0.0
f    NaN
```

```
g      NaN
dtype: float64
```

```
In [155]: df1 = pd.DataFrame(np.arange(9.).reshape((3, 3)), columns=list('bcd'),
.....: index=['Ohio', 'Texas', 'Colorado'])
In [156]: df2 = pd.DataFrame(np.arange(12.).reshape((4, 3)), columns=list('bde'),
.....: index=['Utah', 'Ohio', 'Texas', 'Oregon'])
In [157]: df1
```

	b	c	d
Ohio	0.0	1.0	2.0
Texas	3.0	4.0	5.0
Colorado	6.0	7.0	8.0

```
df2
```

	b	d	e
Utah	0.0	1.0	2.0
Ohio	3.0	4.0	5.0
Texas	6.0	7.0	8.0
Oregon	9.0	10.0	11.0

```
In [159]: df1 + df2
```

	b	c	d	e
Colorado	NaN	NaN	NaN	NaN
Ohio	3.0	NaN	6.0	NaN
Oregon	NaN	NaN	NaN	NaN
Texas	9.0	NaN	12.0	NaN
Utah	NaN	NaN	NaN	NaN

```
In [165]: df1 = pd.DataFrame(np.arange(12.).reshape((3, 4)),
.....: columns=list('abcd'))
In [166]: df2 = pd.DataFrame(np.arange(20.).reshape((4, 5)),
.....: columns=list('abcde'))
In [167]: df2.loc[1, 'b'] = np.nan
In [168]: df1
```



	a	b	c	d
0	0.0	1.0	2.0	3.0
1	4.0	5.0	6.0	7.0
2	8.0	9.0	10.0	11.0

df2

	a	b	c	d	e
0	0.0	1.0	2.0	3.0	4.0
1	5.0	NaN	7.0	8.0	9.0
2	10.0	11.0	12.0	13.0	14.0
3	15.0	16.0	17.0	18.0	19.0

In [170]: df1 + df2

	a	b	c	d	e
0	0.0	2.0	4.0	6.0	NaN
1	9.0	NaN	13.0	15.0	NaN
2	18.0	20.0	22.0	24.0	NaN
3	NaN	NaN	NaN	NaN	NaN

In [171]: df1.add(df2, fill\_value=0)

	a	b	c	d	e
0	0.0	2.0	4.0	6.0	4.0
1	9.0	5.0	13.0	15.0	9.0
2	18.0	20.0	22.0	24.0	14.0
3	15.0	16.0	17.0	18.0	19.0

In [174]: df1.reindex(columns=df2.columns, fill\_value=0)

## Operations between DataFrame and Series

```
In [175]: arr = np.arange(12.).reshape((3, 4))
```

```
In [176]: arr
```

```
array([[ 0.,  1.,  2.,  3.],
       [ 4.,  5.,  6.,  7.],
       [ 8.,  9., 10., 11.]])
```

```
In [177]: arr[0]
```

```
array([0., 1., 2., 3.])
```

```
In [178]: arr - arr[0]
```

```
array([[0., 0., 0., 0.],
       [4., 4., 4., 4.],
       [8., 8., 8., 8.]])
```

```
In [179]: frame = pd.DataFrame(np.arange(12.).reshape((4, 3)),
```

```
.....: columns=list('bde'),
```

```
.....: index=['Utah', 'Ohio', 'Texas', 'Oregon'])
```

```
In [180]: series = frame.iloc[0]
```

```
In [181]: frame
```

	b	d	e
<b>Utah</b>	0.0	1.0	2.0
<b>Ohio</b>	3.0	4.0	5.0
<b>Texas</b>	6.0	7.0	8.0
<b>Oregon</b>	9.0	10.0	11.0

```
In [182]: series
```

```
b    0.0
d    1.0
e    2.0
Name: Utah, dtype: float64
```

```
In [183]: frame - series
```

	<b>b</b>	<b>d</b>	<b>e</b>
<b>Utah</b>	0.0	0.0	0.0
<b>Ohio</b>	3.0	3.0	3.0
<b>Texas</b>	6.0	6.0	6.0

```
In [184]: series2 = pd.Series(range(3), index=['b', 'e', 'f'])
```

```
In [185]: frame + series2
```

	<b>b</b>	<b>d</b>	<b>e</b>	<b>f</b>
<b>Utah</b>	0.0	NaN	3.0	NaN
<b>Ohio</b>	3.0	NaN	6.0	NaN
<b>Texas</b>	6.0	NaN	9.0	NaN
<b>Oregon</b>	9.0	NaN	12.0	NaN

```
In [186]: series3 = frame['d']
```

```
In [187]: frame
```

	<b>b</b>	<b>d</b>	<b>e</b>
<b>Utah</b>	0.0	1.0	2.0
<b>Ohio</b>	3.0	4.0	5.0
<b>Texas</b>	6.0	7.0	8.0
<b>Oregon</b>	9.0	10.0	11.0

```
In [188]: series3
```

```
Utah      1.0
Ohio      4.0
Texas     7.0
Oregon    10.0
Name: d, dtype: float64
```

```
In [189]: frame.sub(series3, axis='index')
```

b d e

## Function Application and Mapping

```
In [190]: frame = pd.DataFrame(np.random.randn(4, 3), columns=list('bde'),
.....: index=['Utah', 'Ohio', 'Texas', 'Oregon'])
In [191]: frame
```

	b	d	e
<b>Utah</b>	0.800704	0.055542	-1.171911
<b>Ohio</b>	-1.587904	-0.706285	-1.309765
<b>Texas</b>	-0.976592	-0.381089	-1.638524
<b>Oregon</b>	-0.822151	-0.737718	0.266883

```
In [192]: np.abs(frame)
```

	b	d	e
<b>Utah</b>	0.800704	0.055542	1.171911
<b>Ohio</b>	1.587904	0.706285	1.309765
<b>Texas</b>	0.976592	0.381089	1.638524
<b>Oregon</b>	0.822151	0.737718	0.266883

```
In [193]: f = lambda x: x.max() - x.min()
In [194]: frame.apply(f)
```

```
b    2.388608
d    0.793259
e    1.905408
dtype: float64
```

```
In [195]: frame.apply(f, axis=0)
```

```
b    2.388608
d    0.793259
e    1.905408
dtype: float64
```

## Sorting and Ranking

```
In [201]: obj = pd.Series(range(4), index=['d', 'a', 'b', 'c'])
In [202]: obj.sort_index()
```

```

a    1
b    2
c    3
d    0
dtype: int64

```

```

In [203]: frame = pd.DataFrame(np.arange(8).reshape((2, 4)),
.....: index=['three', 'one'],
.....: columns=['d', 'a', 'b', 'c'])
In [204]: frame.sort_index()

```

	d	a	b	c
one	4	5	6	7
three	0	1	2	3

```

In [205]: frame.sort_index(axis=1)

```

	a	b	c	d
three	1	2	3	0
one	5	6	7	4

```

In [207]: obj = pd.Series([4, 7, -3, 2])
In [208]: obj.sort_values()

```

```

2    -3
3     2
0     4
1     7
dtype: int64

```

```

In [209]: obj = pd.Series([4, np.nan, 7, np.nan, -3, 2])
In [210]: obj.sort_values()

```

```

4    -3.0
5     2.0
0     4.0
2     7.0
1    NaN
3    NaN
dtype: float64

```

```

In [211]: frame = pd.DataFrame({'b': [4, 7, -3, 2], 'a': [0, 1, 0, 1]})
In [212]: frame

```

	b	a
0	4	0
1	7	1
2	-3	0
3	2	1

```
In [213]: frame.sort_values(by='b')
```

	b	a
2	-3	0
3	2	1
0	4	0
1	7	1

```
In [214]: frame.sort_values(by=['a', 'b'])
```

```
In [215]: obj = pd.Series([7, -5, 7, 4, 2, 0, 4])
```

```
In [216]: obj.rank()
```

0	6.5
1	1.0
2	6.5
3	4.5
4	3.0
5	2.0
6	4.5

dtype: float64

```
In [217]: obj.rank(method='first')
```

0	6.0
1	1.0
2	7.0
3	4.0
4	3.0
5	2.0
6	5.0

dtype: float64

```
In [219]: frame = pd.DataFrame({'b': [4.3, 7, -3, 2], 'a': [0, 1, 0, 1],
.....: 'c': [-2, 5, 8, -2.5]})
```

```
In [220]: frame
```

```
In [221]: frame.rank(axis='columns')
```

## Axis Indexes with Duplicate Labels

```
In [222]: obj = pd.Series(range(5), index=['a', 'a', 'b', 'b', 'c'])
```

```
In [223]: obj
```

```
In [224]: obj.index.is_unique
```

```
In [225]: obj['a']
```

```
In [227]: df = pd.DataFrame(np.random.randn(4, 3), index=['a', 'a', 'b', 'b'])
```

```
In [228]: df
```

```
In [229]: df.loc['b']
```