

```
In [147]: # Series
```

```
In [148]: # pip install pandas
```

```
In [149]: import pandas as pd
```

```
In [150]: s = pd.Series([12, 8, 19, 17])  
s
```

```
Out[150]: 0    12  
         1     8  
         2    19  
         3    17  
         dtype: int64
```

```
In [151]: s = pd.Series([12, 8, 19, 17], index=['ali', 'taha', 'sara', 'omid'])  
s
```

```
Out[151]: ali     12  
         taha     8  
         sara    19  
         omid    17  
         dtype: int64
```

```
In [152]: s = s.reindex(['ali', 'taha', 'mahsa', 'omid'])  
s
```

```
Out[152]: ali      12.0  
         taha      8.0  
         mahsa    NaN  
         omid     17.0  
         dtype: float64
```

```
In [153]: s[1]
```

```
Out[153]: 8.0
```

```
In [154]: s['mahsa']
```

```
Out[154]: nan
```

```
In [155]: s['taha']
```

```
Out[155]: 8.0
```

```
In [156]: ▶ s
```

```
Out[156]: ali      12.0  
         taha      8.0  
         mahsa     NaN  
         omid     17.0  
         dtype: float64
```

```
In [157]: ▶ s['mahsa'] = 20
```

```
In [158]: ▶ s
```

```
Out[158]: ali      12.0  
         taha      8.0  
         mahsa     20.0  
         omid     17.0  
         dtype: float64
```

```
In [159]: ▶ s.index
```

```
Out[159]: Index(['ali', 'taha', 'mahsa', 'omid'], dtype='object')
```

```
In [160]: ▶ s.values
```

```
Out[160]: array([12.,  8., 20., 17.])
```

```
In [161]: ▶ s[1:3]
```

```
Out[161]: taha      8.0  
         mahsa     20.0  
         dtype: float64
```

```
In [162]: ▶ s[:2]
```

```
Out[162]: ali      12.0  
         taha      8.0  
         dtype: float64
```

```
In [163]: ▶ s.iloc[:2]
```

```
Out[163]: ali      12.0  
         taha      8.0  
         dtype: float64
```

```
In [164]: ▶ s.index.name = 'name'  
s
```

```
Out[164]: name  
ali      12.0  
taha     8.0  
mahsa    20.0  
omid     17.0  
dtype: float64
```

```
In [165]: ▶ s.name = 'grade'  
s
```

```
Out[165]: name  
ali      12.0  
taha     8.0  
mahsa    20.0  
omid     17.0  
Name: grade, dtype: float64
```

```
In [166]: ▶ s = s.drop('omid')  
s
```

```
Out[166]: name  
ali      12.0  
taha     8.0  
mahsa    20.0  
Name: grade, dtype: float64
```

```
In [167]: ▶ s = s.drop(['ali', 'taha'])  
s
```

```
Out[167]: name  
mahsa    20.0  
Name: grade, dtype: float64
```

```
In [168]: ▶ myser = pd.Series([12, 4, 5, 7, 2], index=['a', 'b', 'c', 'd', 'e'])  
myser
```

```
Out[168]: a      12  
b       4  
c       5  
d       7  
e       2  
dtype: int64
```

```
In [169]: ▶ myser.pop('c')
```

```
Out[169]: 5
```

```
In [170]: ▶ myser
```

```
Out[170]: a    12  
         b     4  
         d     7  
         e     2  
         dtype: int64
```

```
In [171]: ▶ import numpy as np  
s = pd.Series([12, 4, 5,np.NaN, 7, 2],index=['a', 'b', 'c', 'd', 'e', 'f'])  
s
```

```
Out[171]: a    12.0  
         b     4.0  
         c     5.0  
         d    NaN  
         e     7.0  
         f     2.0  
         dtype: float64
```

```
In [172]: ▶ s.isna()
```

```
Out[172]: a    False  
         b    False  
         c    False  
         d     True  
         e    False  
         f    False  
         dtype: bool
```

```
In [173]: ▶ s.notna()
```

```
Out[173]: a     True  
         b     True  
         c     True  
         d    False  
         e     True  
         f     True  
         dtype: bool
```

```
In [174]: ▶ s
```

```
Out[174]: a    12.0  
         b     4.0  
         c     5.0  
         d    NaN  
         e     7.0  
         f     2.0  
         dtype: float64
```

```
In [175]: ▶ s.isin([5])
```

```
Out[175]: a    False  
         b    False  
         c     True  
         d    False  
         e    False  
         f    False  
         dtype: bool
```

```
In [176]: ▶ s
```

```
Out[176]: a    12.0  
         b     4.0  
         c     5.0  
         d     NaN  
         e     7.0  
         f     2.0  
         dtype: float64
```

```
In [177]: ▶ sc = s.copy()
```

```
In [178]: ▶ s['b']=89  
         s
```

```
Out[178]: a    12.0  
         b    89.0  
         c     5.0  
         d     NaN  
         e     7.0  
         f     2.0  
         dtype: float64
```

```
In [179]: ▶ sc
```

```
Out[179]: a    12.0  
         b     4.0  
         c     5.0  
         d     NaN  
         e     7.0  
         f     2.0  
         dtype: float64
```

In [180]: `s`

```
Out[180]: a    12.0  
         b    89.0  
         c     5.0  
         d    NaN  
         e     7.0  
         f     2.0  
         dtype: float64
```

In [181]: `s.sort_values()`

```
Out[181]: f     2.0  
         c     5.0  
         e     7.0  
         a    12.0  
         b    89.0  
         d    NaN  
         dtype: float64
```

In [182]: `s.sort_values(ascending=False)`

```
Out[182]: b    89.0  
         a    12.0  
         e     7.0  
         c     5.0  
         f     2.0  
         d    NaN  
         dtype: float64
```

In [183]: `s`

```
Out[183]: a    12.0  
         b    89.0  
         c     5.0  
         d    NaN  
         e     7.0  
         f     2.0  
         dtype: float64
```

In [184]: `s.rank()`

```
Out[184]: a     4.0  
         b     5.0  
         c     2.0  
         d    NaN  
         e     3.0  
         f     1.0  
         dtype: float64
```

```
In [185]: ▶ s = pd.Series([12, 4, 5,np.NaN, 7, 2],index=['a', 'b', 'c', 'a', 'e', 'b'])  
s
```

```
Out[185]: a    12.0  
         b     4.0  
         c     5.0  
         a    NaN  
         e     7.0  
         b     2.0  
         dtype: float64
```

```
In [186]: ▶ s['b']
```

```
Out[186]: b    4.0  
         b    2.0  
         dtype: float64
```

```
In [187]: ▶ s.index.is_unique
```

```
Out[187]: False
```

```
In [188]: ▶ s.describe()
```

```
Out[188]: count    5.000000  
         mean     6.000000  
         std      3.807887  
         min      2.000000  
         25%      4.000000  
         50%      5.000000  
         75%      7.000000  
         max     12.000000  
         dtype: float64
```

```
In [189]: ▶ s.count()
```

```
Out[189]: 5
```

```
In [190]: ▶ s.quantile(0.5)
```

```
Out[190]: 5.0
```

```
In [191]: ▶ s.quantile([0.25,0.75])
```

```
Out[191]: 0.25    4.0  
         0.75    7.0  
         dtype: float64
```

In [192]: `s`

```
Out[192]: a    12.0
          b     4.0
          c     5.0
          a    NaN
          e     7.0
          b     2.0
          dtype: float64
```

In [193]: `s >= 5`

```
Out[193]: a    True
          b   False
          c    True
          a   False
          e    True
          b   False
          dtype: bool
```

In [194]: `s.where(s >= 5)`

```
Out[194]: a    12.0
          b    NaN
          c     5.0
          a    NaN
          e     7.0
          b    NaN
          dtype: float64
```

In [195]: `myser = pd.Series(['a', 'b', 'a', 'c', 'd'])`  
`myser`

```
Out[195]: 0    a
          1    b
          2    a
          3    c
          4    d
          dtype: object
```

In [196]: `myser.duplicated()`

```
Out[196]: 0    False
          1    False
          2     True
          3    False
          4    False
          dtype: bool
```



```
In [197]: myser.drop_duplicates(keep='last')
```

```
Out[197]: 1    b
          2    a
          3    c
          4    d
          dtype: object
```

```
In [198]: s = pd.Series([1, 2, 3, 4])
          s
```

```
Out[198]: 0    1
          1    2
          2    3
          3    4
          dtype: int64
```

```
In [199]: s.add_prefix('item_')
```

```
Out[199]: item_0    1
          item_1    2
          item_2    3
          item_3    4
          dtype: int64
```

```
In [200]: s.add_suffix('_item')
```

```
Out[200]: 0_item    1
          1_item    2
          2_item    3
          3_item    4
          dtype: int64
```

```
In [201]: a = pd.Series([1, 10, 3])
          b = pd.Series([4, 5, 6])
```

```
In [202]: a + b
```

```
Out[202]: 0    5
          1   15
          2    9
          dtype: int64
```

```
In [203]: a.add(b)
```

```
Out[203]: 0    5
          1   15
          2    9
          dtype: int64
```

```
In [204]: ▶ a.subtract(b)
```

```
Out[204]: 0   -3  
          1    5  
          2   -3  
          dtype: int64
```

```
In [205]: ▶ a.multiply(b)
```

```
Out[205]: 0    4  
          1   50  
          2   18  
          dtype: int64
```

```
In [206]: ▶ a.pow(b)
```

```
Out[206]: 0    1  
          1  100000  
          2    729  
          dtype: int64
```

```
In [207]: ▶ a.divide(b)
```

```
Out[207]: 0    0.25  
          1    2.00  
          2    0.50  
          dtype: float64
```

```
In [208]: ▶ a.mod(b)
```

```
Out[208]: 0    1  
          1    0  
          2    3  
          dtype: int64
```

```
In [209]: ▶ a = pd.Series([1, 10, 3],index=['a','b','c'] )  
          b = pd.Series([4, 5, 6],index=['a','b','d'] )
```

```
In [210]: ▶ a.add(b)
```

```
Out[210]: a    5.0  
          b   15.0  
          c    NaN  
          d    NaN  
          dtype: float64
```

```
In [211]: ▶ a.add(b, fill_value=0)
```

```
Out[211]: a    5.0  
         b   15.0  
         c    3.0  
         d    6.0  
         dtype: float64
```

```
In [212]: ▶ # eq , ne , gt , ge , lt , le
```

```
In [213]: ▶ s1 = pd.Series([8, 2, 12, 6, 5, 4])  
         s2 = pd.Series([20, 2, 7, 6, 2, 17])
```

```
In [214]: ▶ s1.eq(s2)
```

```
Out[214]: 0    False  
         1     True  
         2    False  
         3     True  
         4    False  
         5    False  
         dtype: bool
```

```
In [215]: ▶ s1.ne(s2)
```

```
Out[215]: 0     True  
         1    False  
         2     True  
         3    False  
         4     True  
         5     True  
         dtype: bool
```

```
In [216]: ▶ s1.gt(s2)
```

```
Out[216]: 0    False  
         1    False  
         2     True  
         3    False  
         4     True  
         5    False  
         dtype: bool
```

```
In [217]: ▶ s1.lt(s2)
```

```
Out[217]: 0    True
          1    False
          2    False
          3    False
          4    False
          5     True
          dtype: bool
```

```
In [218]: ▶ # argmax , argmin() , idxmin() , idxmax()
```

```
In [219]: ▶ score = pd.Series({'Java' : 15, 'C++' : 20, 'Python' : 12 , 'Pascal' : 9})
```

```
In [220]: ▶ score
```

```
Out[220]: Java      15
          C++      20
          Python   12
          Pascal    9
          dtype: int64
```

```
In [221]: ▶ score.argmax()
```

```
Out[221]: 1
```

```
In [222]: ▶ score.argmin()
```

```
Out[222]: 3
```

```
In [223]: ▶ score.idxmin()
```

```
Out[223]: 'Pascal'
```

```
In [224]: ▶ score.idxmax()
```

```
Out[224]: 'C++'
```

```
In [225]: ▶ # cumsum , cumprod (cumulative)
```

```
In [226]: ▶ s = pd.Series([3, 2, np.nan, 5, 0])  
s
```

```
Out[226]: 0    3.0  
1    2.0  
2    NaN  
3    5.0  
4    0.0  
dtype: float64
```

```
In [227]: ▶ s.cumsum()
```

```
Out[227]: 0    3.0  
1    5.0  
2    NaN  
3   10.0  
4   10.0  
dtype: float64
```

```
In [228]: ▶ s.cumprod()
```

```
Out[228]: 0    3.0  
1    6.0  
2    NaN  
3   30.0  
4    0.0  
dtype: float64
```

```
In [229]: ▶ # value_counts
```

```
In [230]: ▶ s = pd.Series(['c', 'a', 'd', 'a', 'a', 'c', 'b', 'b', 'c', 'c'])  
s
```

```
Out[230]: 0    c  
1    a  
2    d  
3    a  
4    a  
5    c  
6    b  
7    b  
8    c  
9    c  
dtype: object
```

```
In [231]: ▶ s.value_counts()
```

```
Out[231]: c    4  
         a    3  
         b    2  
         d    1  
         dtype: int64
```

```
In [232]: ▶ pd.value_counts(s.values, sort=False)
```

```
Out[232]: a    3  
         c    4  
         d    1  
         b    2  
         dtype: int64
```

```
In [233]: ▶ # unique
```

```
In [234]: ▶ s.unique()
```

```
Out[234]: array(['c', 'a', 'd', 'b'], dtype=object)
```

```
In [235]: ▶ pd.Series(['c','a','d','a','a','c','b','b','c','c']).unique()
```

```
Out[235]: array(['c', 'a', 'd', 'b'], dtype=object)
```

```
In [236]: ▶ # append
```

```
In [237]: ▶ s1 = pd.Series([8, 2, 12, 6, 5, 4])  
         s2 = pd.Series([20, 2, 7, 6, 2, 17] , index=[6,7,8,9,10,11])
```

```
In [238]: ▶ s1.append(s2)
```

```
Out[238]: 0      8  
         1      2  
         2     12  
         3      6  
         4      5  
         5      4  
         6     20  
         7      2  
         8      7  
         9      6  
        10      2  
        11     17  
         dtype: int64
```

```
In [239]: ▶ # combine
```

```
In [240]: ▶ s1 = pd.Series({'ali' : 16 , 'sara' : 17})  
          s2 = pd.Series({'ali' : 19 , 'sara' : 15 , 'taha' : 18})
```

```
In [241]: ▶ s1.combine(s2, max)
```

```
Out[241]: ali      19.0  
          sara      17.0  
          taha      NaN  
          dtype: float64
```

```
In [242]: ▶ s1.combine(s2, max, fill_value=0)
```

```
Out[242]: ali      19  
          sara      17  
          taha      18  
          dtype: int64
```

```
In [243]: ▶ # apply
```

```
In [244]: ▶ s = pd.Series([10, 5, 100])  
          s
```

```
Out[244]: 0      10  
          1       5  
          2     100  
          dtype: int64
```

```
In [245]: ▶ s.apply(np.log10)
```

```
Out[245]: 0      1.00000  
          1      0.69897  
          2      2.00000  
          dtype: float64
```

```
In [246]: ▶ s
```

```
Out[246]: 0      10  
          1       5  
          2     100  
          dtype: int64
```

```
In [247]: ▶ def f(x):  
          return x**2  
          s.apply(f)
```

```
Out[247]: 0      100  
          1      25  
          2    10000  
          dtype: int64
```

```
In [248]: ▶ lam = lambda x: x**2
          s.apply(lam)
```

```
Out[248]: 0      100
          1       25
          2    10000
          dtype: int64
```

```
In [249]: ▶ s
```

```
Out[249]: 0      10
          1       5
          2     100
          dtype: int64
```

```
In [250]: ▶ def myfunc(x, y):
          return x - y

          s.apply(myfunc, args=(2,))
```

```
Out[250]: 0      8
          1      3
          2     98
          dtype: int64
```

```
In [251]: ▶ s
```

```
Out[251]: 0      10
          1       5
          2     100
          dtype: int64
```

```
In [252]: ▶ def f(r, **kwargs):
          for i in kwargs:
              r += kwargs[i]
          return r

          s.apply(f, x=3, y=2)
```

```
Out[252]: 0      15
          1      10
          2     105
          dtype: int64
```

```
In [253]: ▶ # transform
```



```
In [254]: ▶ s = pd.Series([10, 4, 9])
          s.transform([np.sqrt, np.log10])
```

Out[254]:

	sqrt	log10
0	3.162278	1.000000
1	2.000000	0.602060
2	3.000000	0.954243

```
In [255]: ▶ # agg : aggregate
```

```
In [256]: ▶ s = pd.Series([10, 4, 9, 2, 18, 6])
          s.agg(['min', 'max'])
```

Out[256]:

min	2
max	18

dtype: int64

```
In [257]: ▶ # nLargest
```

```
In [258]: ▶ data = {'a' : 6, 'b' : 3, 'c' : 8, 'd' : 5, 'e' : 9, 'f' : 3, 'g':5, 'h' : 4, 'i' : 5}
          s = pd.Series(data)
          s
```

Out[258]:

a	6
b	3
c	8
d	5
e	9
f	3
g	5
h	4
i	5

dtype: int64

```
In [259]: ▶ s.nlargest()
```

Out[259]:

e	9
c	8
a	6
d	5
g	5

dtype: int64

```
In [260]: ▶ s.nlargest(4, keep='last')
```

```
Out[260]: e    9  
         c    8  
         a    6  
         i    5  
         dtype: int64
```

```
In [261]: ▶ s.nlargest(4, keep='all')
```

```
Out[261]: e    9  
         c    8  
         a    6  
         d    5  
         g    5  
         i    5  
         dtype: int64
```

```
In [262]: ▶ s.nsmallest()
```

```
Out[262]: b    3  
         f    3  
         h    4  
         d    5  
         g    5  
         dtype: int64
```

```
In [263]: ▶ # groupby
```

```
In [264]: ▶ i = ['BMW', 'BMW', 'Benz', 'Benz']  
         d = [220, 180, 230, 200]
```

```
In [265]: ▶ s = pd.Series(d, index=i, name='MaxSpeed')  
         s
```

```
Out[265]: BMW    220  
         BMW    180  
         Benz   230  
         Benz   200  
         Name: MaxSpeed, dtype: int64
```

```
In [266]: ▶ s.groupby(i).max()
```

```
Out[266]: BMW    220  
         Benz   230  
         Name: MaxSpeed, dtype: int64
```

```
In [267]: ▶ s.groupby(i).mean()
```

```
Out[267]: BMW      200  
Benz      215  
Name: MaxSpeed, dtype: int64
```

```
In [268]: ▶ # between
```

```
In [269]: ▶ s = pd.Series([15, 9, 18, 20])  
s.between(10,20)
```

```
Out[269]: 0      True  
1      False  
2      True  
3      True  
dtype: bool
```

```
In [270]: ▶ # dropna
```

```
In [271]: ▶ s = pd.Series([15, np.nan, 9, 18, np.nan, 20])  
s
```

```
Out[271]: 0      15.0  
1       NaN  
2       9.0  
3      18.0  
4       NaN  
5      20.0  
dtype: float64
```

```
In [272]: ▶ s.dropna(inplace=True)
```

```
In [273]: ▶ s
```

```
Out[273]: 0      15.0  
2       9.0  
3      18.0  
5      20.0  
dtype: float64
```

```
In [274]: ▶ # to_numpy
```

```
In [275]: ▶ s = pd.Series([15, 9, 18, 20])  
arr = s.to_numpy()  
arr
```

```
Out[275]: array([15,  9, 18, 20], dtype=int64)
```

```
In [276]: ▶ type(s)
```

```
Out[276]: pandas.core.series.Series
```

```
In [277]: ▶ type(arr)
```

```
Out[277]: numpy.ndarray
```

```
In [278]: ▶ # to_dict
```

```
In [279]: ▶ s = pd.Series([15, 9, 18, 20])  
d = s.to_dict()  
d
```

```
Out[279]: {0: 15, 1: 9, 2: 18, 3: 20}
```

```
In [280]: ▶ s
```

```
Out[280]: 0    15  
         1     9  
         2    18  
         3    20  
         dtype: int64
```

```
In [281]: ▶ type(d)
```

```
Out[281]: dict
```

```
In [282]: ▶ # replcae
```

```
In [283]: ▶ s = pd.Series([15, 9, 18, 20])  
s.replace(9, 10)
```

```
Out[283]: 0    15  
         1    10  
         2    18  
         3    20  
         dtype: int64
```

```
In [284]: ▶ # repeat
```

```
In [285]: ▶ s = pd.Series([15, 9, 18, 20])  
s.repeat(3)
```

```
Out[285]: 0    15  
          0    15  
          0    15  
          1     9  
          1     9  
          1     9  
          2    18  
          2    18  
          2    18  
          3    20  
          3    20  
          3    20  
dtype: int64
```

```
In [286]: ▶ # MultiIndex
```

```
In [287]: ▶ lst = [['BMW', 'BMW', 'Benz', 'Benz'], ['A', 'B', 'A', 'B']]  
mi = pd.MultiIndex.from_arrays(lst, names=('Machine', 'Class'))  
d = [220, 180, 230, 200]  
s = pd.Series(d, index=mi)  
s
```

```
Out[287]: Machine  Class  
BMW           A      220  
              B      180  
Benz          A      230  
              B      200  
dtype: int64
```

```
In [288]: ▶ s.groupby(level='Machine').max()
```

```
Out[288]: Machine  
BMW      220  
Benz     230  
dtype: int64
```

```
In [289]: ▶ s.groupby(level=0).max()
```

```
Out[289]: Machine  
BMW      220  
Benz     230  
dtype: int64
```

```
In [290]: ▶ s.groupby(level='Class').max()
```

```
Out[290]: Class  
A      230  
B      200  
dtype: int64
```

```
In [291]: ▶ s.groupby(level=1).max()
```

```
Out[291]: Class  
A      230  
B      200  
dtype: int64
```

دانشگاه شهید مدنی آذربایجان  
برنامه نویسی پیشرفته با پایتون  
امین گلزاری اسکویی  
۱۴۰۰-۱۴۰۱

[Codes and Projects \(click here\)](https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Advanced-2021) (<https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Advanced-2021>) [slides and videos \(click here\)](https://drive.google.com/drive/folders/1Dx3v7fD1QBWL-MNP2hd7ilxaRbeALkKA) (<https://drive.google.com/drive/folders/1Dx3v7fD1QBWL-MNP2hd7ilxaRbeALkKA>)