

```
In [1]: #import the library  
import pandas as pd
```

```
In [2]: #print the version of pandas  
print(pd.__version__)
```

1.3.4

```
In [3]: # create pandas series using scalar values  
ser1=pd.Series([15,6,7,4])  
print(ser1)  
print(type(ser1))
```

```
0    15  
1     6  
2     7  
3     4  
dtype: int64  
<class 'pandas.core.series.Series'>
```

```
In [4]: ser1=pd.Series([15,6,7,4,6.7])  
print(ser1)  
print(type(ser1))
```

```
0    15.0  
1     6.0  
2     7.0  
3     4.0  
4     6.7  
dtype: float64  
<class 'pandas.core.series.Series'>
```

```
In [5]: # implicit index  
#change index- explicit index  
ser1=pd.Series([15,6,7,4,6.7], index=[1,2,3,4,5])  
print(ser1)  
print(type(ser1))
```

```
1    15.0  
2     6.0  
3     7.0  
4     4.0  
5     6.7  
dtype: float64  
<class 'pandas.core.series.Series'>
```

```
In [6]: # create a series of some elements,  
# give them A, B, C  
ser1=pd.Series([15,6,7,4,6.7], index=["A","B","C","D","E"])  
print(ser1)  
print(type(ser1))
```

```
A    15.0
B     6.0
C     7.0
D     4.0
E     6.7
dtype: float64
<class 'pandas.core.series.Series'>
```

```
In [7]: ser2 = pd.Series(['Name1', 'Name2', 'Name3', 'Name4'], index=['A', 'B', 'C', 'D'])
print(ser2)
print(type(ser2))
```

```
A    Name1
B    Name2
C    Name3
D    Name4
dtype: object
<class 'pandas.core.series.Series'>
```

```
In [8]: ser1 = pd.Series([15, 6, 7, 4, 6.7], index=range(1, 6))
print(ser1)
print(type(ser1))
```

```
1    15.0
2     6.0
3     7.0
4     4.0
5     6.7
dtype: float64
<class 'pandas.core.series.Series'>
```

```
In [9]: # create pandas series using list
list1 = [5, 6, 7, 8, 9]
ser2 = pd.Series(list1)
ser2
```

```
Out[9]: 0    5
1    6
2    7
3    8
4    9
dtype: int64
```

```
In [10]: # create pandas series using list
list1 = [5, 6, 7, 8, 9]
index = [1, 2, 3, 4, 5]
ser2 = pd.Series(list1, index)
ser2
```

```
Out[10]: 1    5
2    6
3    7
4    8
5    9
dtype: int64
```

```
In [13]: # create a series using a list & assign roman number as index
list1 = [5, 6, 7, 8, 9]
index = ['I', 'II', 'III', 'IV', 'V']
ser2 = pd.Series(list1, index)
ser2
```

```
Out[13]: I      5  
        II     6  
        III    7  
        IV     8  
        V      9  
        dtype: int64
```

```
In [14]: # array create series using numpy array  
import numpy as np  
country=np.array(['India','US','Japan','France'])  
country
```

```
Out[14]: array(['India', 'US', 'Japan', 'France'], dtype='<U6')
```

```
In [17]: ser3=pd.Series(country)  
ser3
```

```
Out[17]: 0      India  
        1        US  
        2      Japan  
        3      France  
        dtype: object
```

```
In [18]: # Create a Series using dictionary  
dict1={'id':12,'Name':'Aman','Salary':10000}  
dict1
```

```
Out[18]: {'id': 12, 'Name': 'Aman', 'Salary': 10000}
```

```
In [19]: ser3=pd.Series(dict1)  
ser3
```

```
Out[19]: id      12  
        Name    Aman  
        Salary  10000  
        dtype: object
```

```
In [20]: # create pandas series using list  
list1=[5,6,7,8,9]  
index=[1,2,3,4,5]  
ser2=pd.Series(list1,index)  
ser2
```

```
Out[20]: 1      5  
        2      6  
        3      7  
        4      8  
        5      9  
        dtype: int64
```

```
In [21]: #Check the attributes of series  
print(ser2.values)  
print(type(ser2.values))  
print(type(ser2))  
print(ser2.index)  
print(ser2.dtype)
```

```
[5 6 7 8 9]
```

```
ndarray'>
```

```
<class 'pandas.core.series.Series'>  
Int64Index([1, 2, 3, 4, 5], dtype='int64')  
int64
```

```
In [22]: gdp=pd.Series([123,543,654,124,789,633],index=['US','India','Japan','UK','China','Russia'])
```

```
In [23]: gdp['US']
```

```
Out[23]: 123
```

```
In [24]: gdp[0:4]
```

```
Out[24]: US      123  
India    543  
Japan    654  
UK       124  
dtype: int64
```

```
In [25]: # iloc & loc method  
#iloc- indexed loction, with implicit index  
#print first 4 countries name & GDP  
gdp.iloc[0:4]
```

```
Out[25]: US      123  
India    543  
Japan    654  
UK       124  
dtype: int64
```

```
In [26]: # loc method works labels or explicit index  
gdp.loc['US':'Japan']
```

```
Out[26]: US      123  
India    543  
Japan    654  
dtype: int64
```

```
In [27]: gdp.iloc[3:6].values
```

```
Out[27]: array([124, 789, 633], dtype=int64)
```

```
In [30]: first=pd.Series([3,4,5,6,7],index=["A","B","C","D","E"])  
second=pd.Series([8,7,6,5,4],index=["A","B","C","D","E"])  
first
```

```
Out[30]: A      3  
B      4  
C      5  
D      6  
E      7  
dtype: int64
```

```
In [31]: second
```

```
Out[31]: A      8  
B      7
```

```
D    5
E    4
dtype: int64
```

```
In [32]: # Addition
print(first.add(second))
```

```
A    11
B    11
C    11
D    11
E    11
dtype: int64
```

```
In [33]: first+second
```

```
Out[33]: A    11
B    11
C    11
D    11
E    11
dtype: int64
```

```
In [34]: first-second
```

```
Out[34]: A    -5
B    -3
C    -1
D     1
E     3
dtype: int64
```

```
In [35]: first*second
```

```
Out[35]: A    24
B    28
C    30
D    30
E    28
dtype: int64
```

```
In [37]: first=pd.Series([3,4,5,6,7],index=["A","B","C","D","E"])
second=pd.Series([8,7,6,5,9],index=["A","B","C","F","G"])
first+second
```

```
Out[37]: A    11.0
B    11.0
C    11.0
D     NaN
E     NaN
F     NaN
G     NaN
dtype: float64
```

```
In [38]: #NaN is not a number
```

```
In [39]: # Operation
print(first.mean())
print(first.median())
```

```
print(first.mode()) # in panda we can do mode
print(first.max())
print(first.min())
```

```
5.0
5.0
0    3
1    4
2    5
3    6
4    7
dtype: int64
7
3
```

```
In [41]: first=pd.Series([3,4,5,6,7,3])
print(first.mode())
```

```
0    3
dtype: int64
```

DataFrame Multidim Array Hetrogenous data Apply powerful data operation data wrangling task

```
In [42]: # Create a DataFrame using Series
Ser1=pd.Series(['India', 'France', 'Japan'])
df1=pd.DataFrame(Ser1)
df1
```

```
Out[42]:
```

	0
0	India
1	France
2	Japan

```
In [43]: df2=pd.DataFrame({'Country':ser1, 'Population':ser2})
df2
```

```
Out[43]:
```

	Country	Population
1	15.0	5
2	6.0	6
3	7.0	7
4	4.0	8
5	6.7	9

```
In [46]: # Create a DataFrame using Series
Ser1=pd.Series(['India', 'France', 'Japan'])
ser2=pd.Series([100,200,300])

df2=pd.DataFrame({'Country':Ser1, 'Population':ser2})
df2
```

```
Out[46]:
```

	Country	Population
0	India	100
1	France	200

	Country	Population
2	Japan	300

In [47]:

```
# Attributes  
print(df2.index)  
print(df2.columns)  
print(type(df2))
```

```
RangeIndex(start=0, stop=3, step=1)  
Index(['Country', 'Population'], dtype='object')  
<class 'pandas.core.frame.DataFrame'>
```

In []: