

## Day23\_pandas

- Pandas stands for Python Data Analysis Module
- Pandas deals with the following two data structures –
  - Series
  - DataFrame

### Series

Series is a one-dimensional array like structure with homogeneous data. For example, the following series is a collection of integers 10, 23, 56, ...

10	23	56	17	52	61	73	90	26	72
----	----	----	----	----	----	----	----	----	----

### Key Points

- Homogeneous data
- Size Immutable
- Values of Data Mutable

### DataFrame

DataFrame is a two-dimensional array with heterogeneous data. For example,

Name	Age	Gender	Rating
Steve	32	Male	3.45
Lia	28	Female	4.6
Vin	45	Male	3.9
Katie	38	Female	2.78

The table represents the data of a sales team of an organization with their overall performance rating. The data is represented in rows and columns. Each column represents an attribute and each row represents a person.

### Data Type of Columns

The data types of the four columns are as follows –

Column	Type
Name	String
Age	Integer
Gender	String
Rating	Float

### Key Points

- Heterogeneous data
- Size Mutable
- Data Mutable

## pandas.Series

A pandas Series can be created using the following constructor –

```
pandas.Series(data, index, dtype)
```

The parameters of the constructor are as follows –

Sr.No	Parameter & Description
1	data: data takes various forms like ndarray, list, constants
2	Index: Index values must be unique and hashable, same length as data. Default np.arange(n) if no index is passed.
3	dtype: dtype is for data type. If None, data type will be inferred

A series can be created using various inputs like –

- Array
- Dict

- Scalar value or constant

Example:

```
import numpy as np
arr = np.array([14,18,24])
#array([14, 18, 24])
```



```
import pandas as pd
sr1 = pd.Series(arr)
print(sr1)
```

*Size of the data*  
*np.arange(3)* → 0, 1, 2

```
0 14
1 18
2 24
dtype: int32
```

```
sr1 = pd.Series(arr,dtype='float32')
print(sr1)
```

*self* }

```
0 14.0
1 18.0
2 24.0
dtype: float32
```

```
sr1 = pd.Series(arr,index=[101,102,103],dtype='float32')
print(sr1)
```

```
101 14.0
102 18.0
103 24.0
dtype: float32
```

```
type(sr1)
#pandas.core.series.Series
```

```
import pandas as pd
s = pd.Series()
print(s)
```

```
#Series([], dtype: float64)
```

```
import pandas as pd
import numpy as np
data = np.array(['a','b','c','d'])
s = pd.Series(data)
print (s)
```

```
0 a
1 b
2 c
3 d
dtype: object
```

```
import pandas as pd
import numpy as np
data = np.array(['a','b','c','d'])
s = pd.Series(data,index=[100,101,102,103])
```

```
print (s)
```

```
import pandas as pd
```

```
import numpy as np
```

```
data = {
```

```
    'a' : 0.,
```

```
    'b' : 1.,
```

```
    'c' : 2.
```

```
}
```

```
s = pd.Series(data)
```

```
print (s)
```

```
#Key of dictionary becomes index
```

```
a 0.0
```

```
b 1.0
```

```
c 2.0
```

```
dtype: float64
```

```
data = {'a' : 0.,
```

```
        'b' : 1.,
```

```
        'c' : 2.}
```

```
s = pd.Series(data,index=['b','c','d','a'])
```

```
print(s)
```

```
b 1.0
```

```
c 2.0
```

```
d NaN
```

```
a 0.0
```

```
dtype: float64
```

```
import numpy as np
```

```
s = pd.Series(5, index=[0, 1, 2, 3])
```

```
print (s)
```

```
0 5
```

```
1 5
```

```
2 5
```

```
3 5
```

```
dtype: int64
```

```
s = pd.Series([1,2,3,4,5],
```

```
              index = ['a','b','c','d','e'])
```

```
print(s)
```

0	a	1	-5
1	b	2	-4
2	c	3	-3
3	d	4	-2
4	e	5	-1

```
dtype: int64
```

```
#retrieve the first element
```

```
print(s[0])
```

```
#1
```

```
#retrieve the last element
```

```
print(s[-1])
```

```
#5
```

```
s = pd.Series([1,2,3,4,5],index = ['a','b','c','d','e'])
print(s)
a 1
b 2
c 3
d 4
e 5
dtype: int64
```

```
#retrieve the first three element
print(s[:3])
a 1
b 2
c 3
dtype: int64
```

```
s = pd.Series([1,2,3,4,5],index = ['a','b','c','d','e'])

#retrieve the last three element
print(s[-3:])

c 3
d 4
e 5
dtype: int64
```

```
s = pd.Series([1,2,3,4,5],index = ['a','b','c','d','e'])
print(s)
a 1
b 2
c 3
d 4
e 5
dtype: int64
#retrieve multiple elements
print( s[['a','c','d']])

a 1
c 3
d 4
dtype: int64
```

### pandas.DataFrame

A pandas DataFrame can be created using the following constructor –

```
pandas.DataFrame( data, index, columns, dtype)
```

The parameters of the constructor are as follows –

Sr.No	Parameter & Description
1	data: data takes various forms like ndarray, series, map, lists, dict, constants and also another DataFrame.
2	index: For the row labels, the Index to be used for the resulting frame is Optional Default np.arange(n) if no index is passed.
3	columns: For column labels, the optional default syntax is np.arange(n). This is only true if no index is passed.
4	dtype: Data type of each column.

## Examples

```
df = pd.DataFrame()
print(df)
```

```
Empty DataFrame
Columns: []
Index: []
```

```
type(df)
```

```
#pandas.core.frame.DataFrame
```

```
data = np.array([[1,2,3,4,5],
                 [5,4,3,2,1]])
```

```
df = pd.DataFrame(data)
print(df)
```

```
0 1 2 3 4
0 1 2 3 4
1 5 4 3 2 1
```

```
data = [['Alex',10],
        ['Bob',12],
        ['Clarke',13]]
```

```
df = pd.DataFrame(data,columns=['Name','Age'])
print (df)
```

```
   Name  Age
0  Alex   10
1  Bob   12
2  Clarke 13
```

```
data = {'Name':['Tom', 'Jack', 'Steve', 'Ricky'],
        'Age':[28,34,29,42]}
```

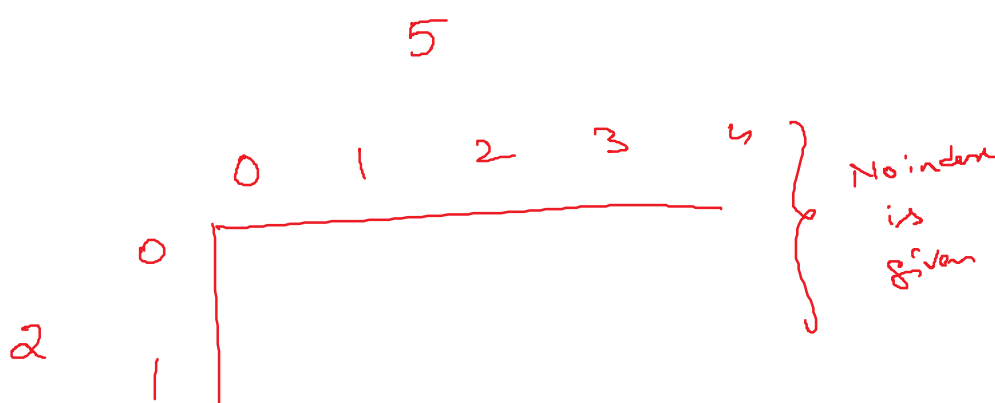
```
df = pd.DataFrame(data)
print( df)
```

```
   Name  Age
0  Tom   28
1  Jack  34
2  Steve 29
3  Ricky 42
```

```
data = {'Name':['Tom', 'Jack', 'Steve', 'Ricky'],
        'Age':[28,34,29,42]}
```

```
df = pd.DataFrame(data, index=['rank1','rank2','rank3','rank4'])
print(df)
```

```
   Name  Age
rank1  Tom  28
rank2  Jack 34
rank3  Steve 29
rank4  Ricky 42
```



```
data = [{'a': 1, 'b': 2},
        {'a': 5, 'b': 10, 'c': 20}]
df = pd.DataFrame(data)
print(df)
```

	a	b	c
1	1	2	NaN
5	5	10	20

```
a b c
0 1 2 NaN
1 5 10 20.0
```

```
data = [{'a': 1, 'b': 2},
        {'a': 5, 'b': 10, 'c': 20}]
```

```
#With two column indices, values same as dictionary keys
df1 = pd.DataFrame(data,
                   index=['first', 'second'],
                   columns=['a', 'b'])
```

	a	b
first	1	2
second	5	10

```
#With two column indices with one index with other name
df2 = pd.DataFrame(data,
                   index=['first', 'second'],
                   columns=['a', 'b1'])
```

	a	b1
first	1	NaN
second	5	NaN

```
print(df1)
a b
first 1 2
second 5 10
```

```
print(df2)
a b1
first 1 NaN
second 5 NaN
```

```
d = {
    'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']),
    'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])
}
```

```
df = pd.DataFrame(d)
print(df)
one two
a 1.0 1
b 2.0 2
c 3.0 3
d NaN 4
```