10:44 | 26.2KB/s

The Series in Pandas is a one-dimensional array that uses the **Series()** method to create a Series, but it also uses different built-in attributes and methods for basic functionalities. In this lesson, let us see such attributes and methods i Men
Python Pandas for Series:

- **dtype:** Return the dtype.
- **ndim:** Return the Number of dimensions
- **size:** Return the number of elements.
- **name:** Return the name of the Series.
- **hasnans:** Returns True if NaNs are in the series.
- **index:** The index of the series
- **head():** Return the first n rows.
- **tail():** Return the last n rows.
- **info():** Display the Summary of the series

Let us understand them one by one:

dtype

The **pandas.series.dtype** is used to return the datatype of the Series.

× Python

the type attribute in ... Python Pandas:

dtype

The **pandas.series.dtype** is used to return the datatype of the Series.

Let us now see an example to implement the type attribute in Python Pandas:

```
import pandas as pd

# Data to be stored in the Pandas Series
data = [10, 20, 40, 80, 100]

# Create a Series using the Series() meth

s = pd.Series(data)

# Display the Series
print("Series: \n", s)

# Datatype
print("\nSeries Datatype: ", s.dtype)
```

Output

```
Series:
0      10
1      20
2      40
3      80
4     100
dtype: int64

Series Datatype:  int64
```

ndim

The **pandas.series.ndim** is used to return the number of dimensions of the Series.

Let us now see an example to implement

10:44 | 6.8KB/s

ndim

The **pandas.series.ndim** is used to return the number of dimensions of the Series.

Let us now see an example to implement the ndim attribute in Python Panc

Men

```
import pandas as pd

# Data to be stored in the Pandas Series
data = [10, 20, 40, 80, 100]

# Create a Series using the Series() meth
s = pd.Series(data)

# Display the Series
print("Series: \n", s)

# Dimensions
print("\nSeries Dimensions: ", s.ndim)
```

Output

```
Series:
0     10
1     20
2     40
3     80
4    100
dtype: int64

Series Dimensions:  1
```

size

The **pandas.series.size** is used to return umber of elements in the Pandas Series

10:44 | 60.7KB/s

size

The **pandas.series.size** is used to return the number of elements in the Pandas Series.

Let us now see an example to implement the size attribute in Python Panda

Menu

```
import pandas as pd

# Data to be stored in the Pandas Series
data = [10, 20, 40, 80, 100]

# Create a Series using the Series() meth
s = pd.Series(data)

# Display the Series
print("Series: \n", s)

# Return the number of elements in the Se
print("\nSeries Size: ", s.size)
```

Output

```
Series:
0     10
1     20
2     40
3     80
4    100
dtype: int64

Series Size:  5
```

name

andas.series.name is used to return the name of the Series in Pandas.

10:44 | 1.4KB/s

name

The **pandas.series.name** is used to return the name of the Series in Pandas.

Let us now see an example to implement the name attribute in Python Pandas:

Menu

```
import pandas as pd

# Data to be stored in the Pandas Series
data = [10, 20, 40, 80, 100]

# Create a Series using the Series() meth
# We have set the Series name using the n
s = pd.Series(data, name ="MyNumberSeries

# Display the Series
print("Series: \n", s)

# Return the name of the Series
print("\nSeries Name: ", s.name)
```

Output

```
Series:
0      10
1      20
2      40
3      80
4     100
Name: MyNumberSeries, dtype: int64

Series Name:  MyNumberSeries
```

hasnans

The **pandas.series.hasnans** attribute ns True if NaNs are in the Pandas Series.

10:44 | 0.4KB/s

hasnans

The **pandas.series.hasnans** attribute returns True if NaNs are in the Pandas Series.

Let us now see an example to implement the hasnans attribute in Python Pandas:

Men

```
import pandas as pd
import numpy as np

# Data to be stored in the Pandas Series
data = [10, 20, 40, 80, 100, np.NaN]

# Create a Series using the Series() meth
s = pd.Series(data)

# Display the Series
print("Series: \n", s)

# Check whether the Series has NaNs
print("\nDoes the Series has NaN? ", s.ha
```

Output

```
Series:
0     10.0
1     20.0
2     40.0
3     80.0
4    100.0
5      NaN
dtype: float64

Does the Series has NaN?  True
```

ex

10:44 | 0.1KB/s

index

The **pandas.series.index** attribute is used to display the index of the Pandas Series.

Discover related topics

C++

Men

C++ Programming >

C Program >

C Programming >

Java >

Let us now see an example to implement the index attribute in Python Pandas:

```
import pandas as pd

# Data to be stored in the Pandas Series
data = [10, 20, 40, 80, 100]

# Create a Series using the Series() meth
s = pd.Series(data, index=["RowA", "RowB"

# Display the Series
print("Series (with custom index labels)

# Return the index of the Series
      ("\nSeries Index: ", s.index)
```

Output

10:44 | 2.1KB/s

head()

The **pandas.series.head()** method is used to return the first n rows of the Pandas Series.

Let us now see an example to implement the head() method in Python Panc Men

```
import pandas as pd

# Data to be stored in the Pandas Series
data = [10, 20, 40, 80, 100, 200, 300, 50

# Create a Series using the Series() meth
s = pd.Series(data, index=["RowA", "RowB"

# Display the Series
print("Series (with custom index labels):

# Return the first n rows.
# The 5 is default for n
print("\nThe first 5 rows of the series:\
```

Output

```
Series (with custom index labels):
RowA      10
RowB      20
RowC      40
RowD      80
RowE     100
RowF     200
RowG     300
RowH     500
dtype: int64

The first 5 rows of the series:
RowA      10
          20
          40
          80
         100
```

tail()

The **pandas.series.tail()** method is used to return the last n rows of the Pandas Series.

Let us now see an example to implement the tail() method in Python Panda

```
import pandas as pd

# Data to be stored in the Pandas Series
data = [10, 20, 40, 80, 100, 200, 300, 50

# Create a Series using the Series() meth
s = pd.Series(data, index=["RowA", "RowB"

# Display the Series
print("Series (with custom index labels):

# Return the last n rows.
# The 5 is default for n
print("\nThe last 5 rows of the series:\n
```

Output

```
Series (with custom index labels):
RowA      10
RowB      20
RowC      40
RowD      80
RowE     100
RowF     200
RowG     300
RowH     500
dtype: int64

The last 5 rows of the series:
RowD      80
         100
         200
         300
RowH     500
```

10:44 | 0.4KB/s

info()

The **pandas.series.info()** method is used to display the Summary of the Pandas Series.

Let us now see an example to implement the info() method in Python Pand

Men

```
import pandas as pd

# Data to be stored in the Pandas Series
data = [10, 20, 40, 80, 100, 200, 300, 50

# Create a Series using the Series() meth
s = pd.Series(data, index=["RowA", "RowB"

# Display the Series
print("Series (with custom index labels):

# Return the summary of the series
print("\nSeries Summary:\n", s.info())
```

Output

```
Series (with custom index labels):
RowA      10
RowB      20
RowC      40
RowD      80
RowE     100
RowF     200
RowG     300
RowH     500
dtype: int64
<class 'pandas.core.series.Series'>
Index: 8 entries, RowA to RowH
Series name: None
Non-Null Count  Dtype
  ----------    -----
     -null      int64
     s: int64(1)
     y usage: 128.0+ bytes
```