The Pandas DataFrame is a Twodimensional, tabular data, that uses the

DataFrame() method to create a

DataFrame. It also uses different built-in

Men

attributes and methods for basic functionalities. In this lesson, let us

such attributes and methods in Q Python Pandas for DataFrame:

- dtypes: Return the dtypes in the DataFrame
- ndim: Return the number of dimensions of the DataFrame
- size: Return the number of elements in the DataFrame.
- shape: Return the dimensionality of the DataFrame in the form of a tuple.
 - index: Return the index of the
- DataFrame
- T: Transpose the rows and columns
- head(): Return the first n rows.
- tail(): Return the last n rows.

Let us understand them one by one:

dtypes

andas.DataFrame.dtypes is used to h the dtypes in the DataFrame.

네네 🥞 🐠 10:36 | 13.6KB/s 🏵 🖪 M

dtypes

The **pandas.DataFrame.dtypes** is used to return the dtypes in the DataFrame.

Let us now see an example to implement

the dtypes attribute in Q Python Pandas:

Men

import pandas as pd

Dataset

data = {

'Student': ["Amit", "John", "Jacob", 'Rank': [1, 4, 3, 5, 2], 'Marks': [95, 70, 80, 60, 90]

}

Create a DataFrame using the DataFrame(res = pd.DataFrame(data, index=['RowA',

Display the Records print("Student Records\n\n", res)

Datatypes in the DataFrame
print("\nDatatypes:\n", res.dtypes)

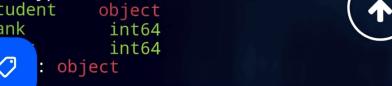
Output

Student Records Student Rank

Marks Amit RowA 1 95 RowB John 4 70 Jacob 80 David 60

RowC RowD RowE Steve 90

Datatypes: Student object Rank int64 int64



all all 🛜 🐠 10:37 | 0.3KB/s 🏵 🖪 m ndim

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

DataFrame.

Let us now see an example to implement the ndim attribute in Q Python Pand Men

import pandas as pd

Dataset

data = {
 'Student': ["Amit", "John", "Jacob",
 'Rank': [1, 4, 3, 5, 2],
 'Marks': [95, 70, 80, 60, 90] } # Create a DataFrame using the DataFrame(

res = pd.DataFrame(data, index=['RowA', # Display the Records print("Student Records\n\n", res)

Number of Dimensions in the DataFrame print("\nNumber of Dimensions:\n", res.nd

Student Rank Marks Amit 95 RowA 1 4 RowB John 70 Jacob RowC 80

Output Student Records

RowD David 60 RowE Steve 90 Number of Dimensions:



네네 (주) (90) 10:37 | 9.1KB/s 🏵 🖪 M

size

The **pandas.DataFrame.size** is used to return the number of elements in the

DataFrame. Let us now see an example to implement

the size attribute in Q <u>Python</u> Panda Men

import pandas as pd

Dataset

data = { - \
'Student': ["Amit", "John", "Jacob",
'Rank': [1, 4, 3, 5, 2],
'Marks': [95, 70, 80, 60, 90] }

Create a DataFrame using the DataFrame(res = pd.DataFrame(data, index=['RowA',

Display the Records
print("Student Records\n\n", res) # Number of elements in the DataFrame

print("\nNumber of Elements:\n", res.size

Output

Student Records Marks Rank Student 95 1 Amit RowA 70 4 John RowB 80 RowC

Jacob 60 David RowD 90 Steve RowE Number of Elements:



10:37 | 5.3KB/s 🏵 🚄 m

·네네 📚 🐠

shape

The **pandas.DataFrame.shape** is used to return the dimensionality of the

DataFrame in the form of a tuple.

Let us now see an example to impler

the shape attribute in Q Python Pan

import pandas as pd

Dataset

data = { 'Student': ["Amit", "John", "Jacob",
'Rank': [1, 4, 3, 5, 2],
'Marks': [95, 70, 80, 60, 90]

Create a DataFrame using the DataFrame(res = pd.DataFrame(data, index=['RowA',

Display the Records print("Student Records\n\n", res) # Return the dimensionality of the DataFr # Result in a Tuple form

Output

Student Records

Student Amit RowA RowB

Rank Marks 1 4 John Jacob David

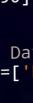
95

70

80 60

90

print("\nDimensionality:\n", res.shape)



Men

Steve RowE 0 sionality: 3)

RowC

RowD

```
10:37 | 2.8KB/s 🖰
The pandas.DataFrame.index is used to
eturn the index of the DataFrame.
 Discover related topics
                                     Men
 Find Number
 Java
 Python Dataframe
 Python
 Python Pandas
Let us now see an example to implement
```

the index attribute in Q <u>Python</u> Pandas: import pandas as pd # Dataset 'Student': ["Amit", "John", "Jacob", 'Rank': [1, 4, 3, 5, 2], 'Marks': [95, 70, 80, 60, 90] $data = {$ } # Create a DataFrame using the DataFrame res = pd.DataFrame(data, index=['Row.

splay the Records

("Student Records\n\n", res)

curn the index of the DataFrame print("\nDataFrame Index:\n". res.index)

```
import pandas as pd
# Dataset
data = {
     'Student': ["Amit", "John", "Jacob",
'Rank': [1, 4, 3, 5, 2],
'Marks': [95, 70, 80, 60, 90]
}
# Create a DataFrame using the DataF
res = pd.DataFrame(data, index=['Rov Men
# Display the Records
print("Student Records\n\n", res)
```

Return the index of the DataFrame
print("\nDataFrame Index:\n", res.index)

Rank Marks

1

4

Output

Student Records

RowD

RowE

T

Student Amit RowA

RowB John RowC Jacob John David

Steve

DataFrame Index:

Index(['RowA', 'RowB', 'RowC', 'RowD',

95

70

80 60

90

The **pandas.DataFrame.T** is used to

Transpose the rows and columns. now see an example to implement

41 41 📚 🐠 10:37 | 14.1KB/s 🏵 🕅 🖪

T

The pandas.DataFrame.T is used to

Transpose the rows and columns.

Let us now see an example to implement

the T attribute in \bigcirc Python Pandas:

Men

import pandas **as** pd

Dataset

data = {

'Student': ["Amit", "John", "Jacob", 'Rank': [1, 4, 3, 5, 2], 'Marks': [95, 70, 80, 60, 90] }

Create a DataFrame using the DataFrame(
res = pd.DataFrame(data, index=['RowA', '

Display the Records print("Student Records\n\n", res)

Return the Transpose print("\nTranspose:\n", res.T)

Student

Amit

John

Rank

4

1

5

4

70

60 90

3

80

Marks

95

70

80

RowD Jacob David

60

Output Student Records

RowA RowB

RowC Jacob

RowD David RowE Steve

Transpose:

RowA RowB RowC <u>Stud</u>ent Amit John 0 95

10:37 | 16.1KB/s 🌀 M 🖪 all all 🛜 🐠 head()

The **pandas.DataFrame.head()** is used to

return the first n rows.

Let us now see an example to implement the head() method in \bigcirc Python Pandas:

Men

import pandas as pd

Dataset data = {

'Student': ["Amit", "John", "Jacob", 'Rank': [1, 4, 3, 5, 6, 2], 'Marks': [95, 70, 80, 60, 55, 90]

Create a DataFrame using the DataFrame(res = pd.DataFrame(data, index=['RowA',

Display the Records

print("Student Records\n\n", res) # Return the first n rows

Default value of n is 5
print("\nFirst 5 rows:\n", res.head())

Output

Student Records

Student Amit RowB

> 5 rows: Student

> > Amit

John

Jacob

David

John Jacob

David Nathan Steve

Rank 1 4

6 Rank

95 70

Marks

Marks

95

70

80

RowA

RowC

RowD

RowE

RowF

First

KUWC

و الدالد 10:37 | 57.2KB/s 🏵 🕅 ◀

tail()

The pandas.DataFrame.tail() is used to return the last n rows.

Let us now see an example to implement

the tail() method in Q Python Pandas:

Men import pandas as pd # Dataset

 $data = {$

'Student': ["Amit", "John", "Jacob", 'Rank': [1, 4, 3, 5, 6, 2], 'Marks': [95, 70, 80, 60, 55, 90]

Create a DataFrame using the DataFrame(res = pd.DataFrame(data, index=['RowA',

Default value of n is 5 print("\nLast 5 rows:\n", res.tail())

Display the Records print("Student Records\n\n", res) # Return the last n rows

Last

ROWE

Output

Student Records

Student Rank Marks

RowA Amit 1 95 70

4

3 5 6

80

RowC Jacob RowD David RowE Nathan RowF Steve

> 5 rows: Student

> > John

Jacob

David

Mathan

John

RowB

4

60 55 90 Rank Marks

70

80

60

55