The Pandas DataFrame is a Two-dimensional, tabular data, that uses the DataFrame() method to create a DataFrame. It also uses different built-in attributes and methods for basic functionalities. In this lesson, let us such attributes and methods in 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

pandas.DataFrame.dtypes is used to n the dtypes in the DataFrame.

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 Python Pandas:


```
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
RowA    Amit     1     95
RowB    John     4     70
RowC   Jacob     3     80
RowD   David     5     60
RowE   Steve     2     90

Datatypes:
Student    object
Rank        int64
            int64
  : object
```

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 Python Pand


```
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
```

Output

```
Student Records

      Student  Rank  Marks
RowA    Amit     1     95
RowB    John     4     70
RowC   Jacob     3     80
RowD   David     5     60
RowE   Steve     2     90

Number of Dimensions:
```

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 Python Panda


```
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

      Student  Rank  Marks
RowA    Amit     1     95
RowB    John     4     70
RowC   Jacob     3     80
RowD   David     5     60
RowE   Steve     2     90

Number of Elements:
```

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 implem the shape attribute in 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
print("\nDimensionality:\n", res.shape)
```

Output

```
Student Records

      Student  Rank  Marks
RowA    Amit     1     95
RowB    John     4     70
RowC   Jacob     3     80
RowD   David     5     60
RowE   Steve     2     90

sionality:
3)
```

index

The **pandas.DataFrame.index** is used to return the index of the DataFrame.

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

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

the index attribute in Python Pandas.

```python
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=['Row

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

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

Men

Output

```
Student Records

      Student  Rank  Marks
RowA    Amit     1     95
RowB    John     4     70
RowC   Jacob     3     80
RowD   David     5     60
RowE   Steve     2     90

DataFrame Index:
 Index(['RowA', 'RowB', 'RowC', 'RowD', '
```

T

The **pandas.DataFrame.T** is used to Transpose the rows and columns.

Let us now see an example to implement

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 Python Pandas:


```
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)
```

Output

```
Student Records

      Student  Rank  Marks
RowA    Amit     1     95
RowB    John     4     70
RowC   Jacob     3     80
RowD   David     5     60
RowE   Steve     2     90

Transpose:
          RowA  RowB   RowC   RowD   RowE
Student  Amit  John  Jacob  David  Steve
            1     4      3      5      2
           95    70     80     60     90
```

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 Python Pandas:


```
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  Rank  Marks
RowA    Amit     1     95
RowB    John     4     70
RowC   Jacob     3     80
RowD   David     5     60
RowE  Nathan     6     55
RowF   Steve     2     90

First 5 rows:
      Student  Rank  Marks
        Amit     1     95
        John     4     70
       Jacob     3     80
       David     5     60
```

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 Python Pandas:


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

Output

```
Student Records

       Student  Rank  Marks
RowA     Amit     1     95
RowB     John     4     70
RowC    Jacob     3     80
RowD    David     5     60
RowE   Nathan     6     55
RowF    Steve     2     90

Last 5 rows:
       Student  Rank  Marks
RowB     John     4     70
RowC    Jacob     3     80
RowD    David     5     60
RowE   Nathan     6     55
```