

DataFrame

A pandas DataFrame is a two (or more) dimensional data structure – basically a table with rows and columns. The columns have names and the rows have indexes. For instance, the price can be the name of a column and 2,3 the price values.

A picture of a Pandas DataFrame is shown alongside:

	Item Price	
0	A	2
1	B	3

In general, you could say that the Pandas DataFrame consists of three main components: the data, the index, and the columns.

A pandas DataFrame can be created using the following constructor `pandas.DataFrame (data, index, columns, dtype, copy)`.

The parameters of the constructor are as follows:

Create a DataFrame from Dictionary

When DataFrame is created by using Dictionary, keys of dictionary are set as columns of DataFrame. You can change the order of columns and store specified columns. If you try to change the column name, NaN will be displayed.

Note: Column name values must be same as dictionary keys

Q. Write python code to create a Dictionary Dic to store roll, name and marks of 3 students. Convert Dic into DataFrame df and display the DataFrame df.

Ans.

```
import pandas as pd
Dic={'roll':(1,2,3),'name':('a','b','c'),'marks':(24,53,66)}
df=pd.DataFrame(Dic)
print (df)
```

Output:

	roll	name	marks
0	1	a	24
1	2	b	53
2	3	c	66

Q. Write python code to create the DataFrame emp using dictionary:

	Name	Salary
101	Rohan	20000
102	Aman	25000

Ans.

```
import pandas as pd

emp=
pd.DataFrame({'Name':['Rohan','Aman'],'Salary':[20000,25000]},
             index=[101,102]);

print(emp)
```

or

```
import pandas as pd

Dic= {'Name':['Rohan','Aman'],'Salary':[20000,25000]}

emp = pd.DataFrame(Dic,index=[101,102]);

print(emp)
```

➤ **Iterating in Pandas DataFrame**

Iteration is a general term for taking each item of something one after another. In Pandas DataFrame, we can iterate an element in two ways:

(i) Iterating over rows:

There are three functions to iterate over rows as follows:

- `iterrows()` : It returns the iterator yielding each index value along with a series containing the data in each row.
- `iteritems()` : It iterates over each column as key, value pair with label as key and column value as series object.
- `itertuples()`: In DataFrame, it returns a tuple for each row. The first element of the tuple will be the row's corresponding index value, while the remaining value are the rows values.

(ii) Iterating over columns

In order to iterate over columns, we need to create a list of DataFrame columns and then iterating through that list to pull out the DataFrame columns.

➤ **Operations on rows and columns**

As we know, DataFrame is a two-dimensional data structure means data is arranged in a tabular format like rows and columns, some basic operations can be performed like adding, deleting, selecting and renaming. These operations are as follows:

(i) Addition

- To add a column in Pandas DataFrame, a new list as a column can be declared and add to an existing DataFrame.
- To add a row in Pandas DataFrame, we can concat the old DataFrame with new one.

(ii) Selection

- To select a column in Pandas DataFrame, we can either access the columns by calling them by their column names.
- To retrieve rows from a DataFrame, a special method is used named DataFrame.loc[]. Rows can also be selected by passing integer location to iloc[] method.

(iii) Deletion

- To delete a column from Pandas DataFrame, drop() method is used. Columns are deleted by dropping columns with column names.
- To delete a row from Pandas DataFrame, drop() method is used. Rows are deleted by dropping rows by index label.

➤ **Head and Tail functions**

head() and tail() methods or functions are used to view a small sample of a DataFrame object. These functions are described below

(i) head():

This function returns the first n rows for the object based on position. It is useful for quick testing if your object has the right type of data in it.

Syntax DataFrame.head (n=5)

Parameters: n-is an integer value, number of rows to be returned where default value is 5.

Return DataFrame with top n rows

Q. Give the output:

```
import pandas as pd
Dic={'empno':(101,102,103,104,105,106),'grade':('a','b','a','c','b','c'),
    'dept': ('sales','pur','mar','sales','pur','mar')}
df=pd.DataFrame(Dic)
print(df.head(3))
```

Output:

	empno	grade	dept
0	101	a	sales
1	102	b	pur
2	103	a	mar

Q. Give the output

Ans.

```
import pandas as pd
Dic={'empno':(101,102,103,104,105,106),'grade':('a','b','a','c','b','c'),'
dept': ('sales','pur','mar','sales','pur','mar')}
df=pd.DataFrame(Dic)
print(df.head()[['empno','dept']])
```

Output:

	empno	dept
0	101	sales
1	102	pur
2	103	mar
3	104	sales
4	105	pur

(ii) tail():

This function returns last n rows from the object based on position. It is useful for quickly verifying data. e.g. after sorting

Syntax: DataFrame.tail (n=5)

Q. Give the output

```
import pandas as pd
Dic={'empno':(101,102,103,104,105,106),'grade':('a','b','a','c','b','c'),'
dept': ('sales','pur','mar','sales','pur','mar')}
```

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

Ans.

	empno	grade	dept
1	102	b	pur
2	103	a	mar
3	104	c	sales
4	105	b	pur
5	106	c	mar

➤ **Indexing using Labels**

Indexing in Pandas means simply selecting particular rows and columns of a DataFrame. Indexing can also be known as subset selection.

It is common operation to pick out one of the DataFrame's columns to work on.

To select a column by its label, we use the `.loc[]` function.

Pandas DataFrame.loc attribute access a group of rows and columns by label(s) or a boolean array in the given DataFrame.

Syntax: DataFrame.loc

loc takes two single/list/range operators separated by ','.

The first one indicates the row and the second one indicates columns.

➤ **Boolean Indexing:**

It helps us to select the data from the DataFrames using a boolean vector.

We need a DataFrame with a boolean index to use the boolean indexing.

In boolean indexing, we can filter a data in four ways:

- Accessing a DataFrame with a boolean index
- Applying a boolean mask to a DataFrame
- Masking data based on column value
- Masking data based on index value