

EX.NO: 03

DATE : 02/03/2024

## PANDAS – DATAFRAME- BASICS

### Aim:

To install pandas package and do the basic DataFrame operations

### Description:

1. Declaring empty DataFrame
2. Declare and print the DataFrame series
3. Do the basic functions such as extracting one column and one row, adding one row

### Programs:

#### 1. Declaring and printing the DataFrame series

```
import pandas as pd

emp = pd.Series(['Parker', 'John', 'Smith', "William"])
Id = pd.Series([102, 104, 203, 501])
frame = {'EMP': emp, 'ID': Id}
result = pd.DataFrame(frame)
print('\n Series to Data Frame\n')
print(result)
```

### OUTPUT :

```
Series to Data Frame

   EMP  ID
0  Parker  102
1   John  104
2  Smith  203
3 William  501
```

## 2. Performing Basic Functions

### i) EXTRACTING ONE COLUMN

```
#EXTRACTING ONE COLUMN

print("\n Extracting one Column: \n")
print( result['EMP'])
```

#### OUTPUT:

```
Extracting one Column:

0    Parker
1      John
2     Smith
3   William
Name: EMP, dtype: object
```

### ii) ADDING NEW COLUMN

```
#ADDING NEW COLUMN

result['AGE']=pd.Series([35,24,40,38])
print('\n',result)
```

#### OUTPUT:

	EMP	ID	AGE
0	Parker	102	35
1	John	104	24
2	Smith	203	40
3	William	501	38

### iii) EXTRACTING THE THIRD ROW

```
#EXTRACTING THE THIRD ROW

print('\n',result.loc[2])
```

#### OUTPUT:

```
EMP    Smith
ID      203
AGE      40
Name: 2, dtype: object
```

#### iv) SLICING THE ROWS

```
#SLICING THE ROWS
print('\n Slice the rows: \n', result[1:3])
```

#### OUTPUT:

```
Slice the rows:
      EMP  ID  AGE
1  John  104   24
2  Smith 203   40
```

#### v) ADDING AND DELETING ROWS

```
#ADDING AND DELETING ROWS
d2 = pd.DataFrame([[ 'Dale',123],[ 'Mark',143]],columns=[ 'EMP','ID'])
print("\n Adding new row values :\n", result.append(d2))
print("\n Deleting particular row:\n", result.drop(1))
```

#### OUTPUT:

```
Adding new row values :
      EMP  ID  AGE
0  Parker  102  35.0
1   John  104  24.0
2   Smith  203  40.0
3 William  501  38.0
0    Dale  123   NaN
1    Mark  143   NaN

Deleting particular row:
      EMP  ID  AGE
0  Parker  102   35
2   Smith  203   40
3 William  501   38
```

**EX.NO: 03****DATE:**02/03/2024

## **PANDAS –DATAFRAME – FUNCTIONS**

### **Aim:**

To perform Pandas DataFrame Functions

### **Description:**

1. Apply sum and square root functions through Numpy
2. Perform min and max operations through aggregation functions
3. Add new column using assign function
4. Do sort and merge functions

### **Programs:**

#### **i) FINDING SQUAREROOT OF A DATAFRAME**

```
import pandas as pd
import numpy as np

print("DATAFRAME APPLY FUNCTION:\n")
info = pd.DataFrame([[2,7]]*3, columns=['P','Q'])
print("\n Original DataFrame:\n",info)
print("\n Square root of DataFrame:\n", info.apply(np.sqrt))
```

### **OUTPUT:**

```
DATAFRAME APPLY FUNCTION:
```

```
Original DataFrame:
```

```
   P  Q
0  2  7
1  2  7
2  2  7
```

```
Square root of DataFrame:
```

```
   P      Q
0  1.414214  2.645751
1  1.414214  2.645751
2  1.414214  2.645751
```

## ii) DATAFRAME ASSIGN FUNCTIONS

```
#DATAFRAME ASSIGN FUNCTIONS

print("\nDataFrame Assign Function:\n")
d2 = pd.DataFrame([[ 'Dale', 123], [ 'Mark', 143]], columns=[ 'EMP', 'ID'])
print("\nOriginal DataFrame:\n", d2)
d2[ 'AGE']=[35,40]
```

### OUTPUT:

```
DataFrame Assign Function:
```

```
Original DataFrame:
```

	EMP	ID
0	Dale	123
1	Mark	143

## iii) ADDING NEW COLUMNS

```
#ADDING NEW COLUMNS

print("\nAdding new column:\n", d2)
d = d2.assign(sex=[ 'Male', 'Male'])
print("\nAdding new column:\n", d)
```

### OUTPUT:

```
Adding new column:
```

	EMP	ID	AGE
0	Dale	123	35
1	Mark	143	40

```
Adding new column:
```

	EMP	ID	AGE	sex
0	Dale	123	35	Male
1	Mark	143	40	Male

## iv) DATAFRAME SORT FUNCTIONS

```
#DATAFRAME SORT FUNCTIONS

print("\nDataFrame sort functions:\n")
info= pd.DataFrame(np.random.randn(5,2), index=[3,2,0,4,1], columns=[ 'col3', 'col4'])
print(info)
info2 = info.sort_index()
print('\nSort index:\n',info2)
info3 = info.sort_values(by='col3')
print("\nSort Values:\n", info3)
```

**OUTPUT:**

DataFrame sort functions:

	col3	col4
3	-0.023617	-0.118871
2	0.842075	-0.148991
0	-1.479072	0.316568
4	-0.987557	1.520186
1	0.336717	0.389722

Sort index:

	col3	col4
0	-1.479072	0.316568
1	0.336717	0.389722
2	0.842075	-0.148991
3	-0.023617	-0.118871
4	-0.987557	1.520186

Sort Values:

	col3	col4
0	-1.479072	0.316568
4	-0.987557	1.520186
3	-0.023617	-0.118871
1	0.336717	0.389722
2	0.842075	-0.148991

**v) DATAFRAME MERGE FUNCTION**

```
#DATAFRAME MERGE FUNCTION

print("\nDataFrame Merge Functions:\n")
left = pd.DataFrame({'id':[1,2,3,4,5],
                     'Name':['Alex','Amy','Allen','Alice','Ayoung'],
                     'Sub':['Sub1','Sub2','Sub4','Sub6','Sub5']})
right = pd.DataFrame({'id':[1,2,3,4,5],
                      'Name':['Billy','Brian','Bran','Bryce','Betty'],
                      'Sub':['Sub2','Sub4','Sub3','Sub6','Sub5']})

print(left)
print('\n',right)
print('\n',pd.merge(left,right,on='id'))
```

**OUTPUT:**

3	4	Alice	Sub6
4	5	Ayoung	Sub5

	id	Name	Sub
0	1	Billy	Sub2
1	2	Brian	Sub4
2	3	Bran	Sub3
3	4	Bryce	Sub6
4	5	Betty	Sub5

DataFrame Merge Functions:

	id	Name	Sub
0	1	Alex	Sub1
1	2	Amy	Sub2
2	3	Allen	Sub4

	id	Name_x	Sub_x	Name_y	Sub_y
0	1	Alex	Sub1	Billy	Sub2
1	2	Amy	Sub2	Brian	Sub4
2	3	Allen	Sub4	Bran	Sub3
3	4	Alice	Sub6	Bryce	Sub6
4	5	Ayoung	Sub5	Betty	Sub5

**EX.NO:** 03**DATE:** 02/03/2024**PANDAS SERIES****Aim:**

To perform Pandas Series

**Description:**

1. Create Series from array
2. Create Series from dictionary
3. Create Series using scalar value
4. Create Series using index
5. Know the size, dimension, shape and index of Series

**Programs:****i) CREATING SERIES FROM ARRAY**

```
import pandas as pd
import numpy as np
#Creating Series from Array
arr=np.array(['p','a','n','d','a','s'])
a=pd.Series(arr)
print(a)
```

**OUTPUT:**

```
0    p
1    a
2    n
3    d
4    a
5    s
dtype: object
```

**ii) CREATING SERIES FROM DICTIONARY**

```
#Series from dictionary
arr={'x':0.,'y':1.,'z':2.}
b=pd.Series(arr)
print("\n\nSeries from dictionary:\n")
print(b)
```

**OUTPUT:**

```
Series from dictionary:
```

```
x    0.0
y    1.0
z    2.0
dtype: float64
```

**iii) CREATING SERIES USING SCALAR**

```
#Series using scalar
x=pd.Series(4,index=[0,1,2,3])
print("\nSeries using scalar\n")
print(x)
```

**OUTPUT:**

```
Series using scalar
```

```
0    4
1    4
2    4
3    4
dtype: int64
```

**RESULT:**

This program was executed