Pandas-DataFrame And Series

Pandas is a powerful data manipulation library in Python, widely used for data analysis and data cleaning. It provides two primary data structures: Series and DataFrame. A Series is a one-dimensional array-like object, while a DataFrame is a two-dimensional, size-mutable, and potentially heterogeneous tabular data structure with labeled axes (rows and columns).

In [1]:
```
import pandas as pd
data=[1,2,3,4,5]
series=pd.Series(data)
print('Series \n', series)
```

```
Series
 0   1
1   2
2   3
3   4
4   5
dtype: int64
```

Here in left side 0 1 2 3 4 are index of 1 2 3 4 5. this all can be changed by using dictionary type

In [19]:
```
#Create a Series from dictionary
series_dict={'a':1,'b':2,'c':3}
series_dict=pd.Series(data)
print(series_dict)
```

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

In [23]:
```
data=[10,20,30]
index=['a','b','c']
pd.Series(data, index=index)
```

Out[23]:
```
a   10
b   20
c   30
dtype: int64
```

In [3]:
```
## Dataframe
## create a Dataframe from a dictionary oof list
data={
'Name': ['Krish', 'John', 'Jack'],
'Age': [25,30,45],
'City': ['Bangalore', 'New York', 'Florida']
}
df=pd.DataFrame(data)
print(df)
print(type(df))
```

```
    Name  Age       City
0  Krish   25  Bangalore
1   John   30   New York
2   Jack   45    Florida
<class 'pandas.core.frame.DataFrame'>
```

In [4]:
```
## Create a Data frame from a List of Dictionaries
data=[
{'Name': 'Krish', 'Age': 32, 'City': 'Bangalore'},
{'Name': 'John', 'Age': 34, 'City': 'Kolkata'},
{'Name': 'Bappy', 'Age': 45, 'City': 'Mumbai'},
{'Name': 'Jack', 'Age': 18, 'City': 'Bhopal'}
]
df=pd.DataFrame(data)
print(df)
print(type(df))
```

```
    Name  Age       City
0  Krish   32  Bangalore
1   John   34    Kolkata
2  Bappy   45     Mumbai
3   Jack   18      Bhopal
<class 'pandas.core.frame.DataFrame'>
```

In [47]:
```
df=pd.read_csv('industry.csv')
df.head(5)
```

Out[47]:

	Industry
0	Accounting/Finance
1	Advertising/Public Relations
2	Aerospace/Aviation
3	Arts/Entertainment/Publishing
4	Automotive

In [6]:
```
df.tail(5)
```

Out[45]:

	Industry
38	Skilled Labor
39	Technology
40	Telecommunications
41	Transportation/Logistics
42	Other

In [80]:
```
df
```

Out[80]:

	Industry
0	Accounting/Finance
1	Advertising/Public Relations
2	Aerospace/Aviation
3	Arts/Entertainment/Publishing
4	Automotive
5	Banking/Mortgage
6	Business Development
7	Business Opportunity
8	Clerical/Administrative
9	Construction/Facilities
10	Consumer Goods
11	Customer Service
12	Education/Training
13	Energy/Utilities
14	Engineering
15	Government/Military
16	Green
17	Healthcare
18	Hospitality/Travel
19	Human Resources
20	Installation/Maintenance
21	Insurance
22	Internet
23	Job Search Aids
24	Law Enforcement/Security
25	Legal
26	Management/Executive
27	Manufacturing/Operations
28	Marketing
29	Non-Profit/Volunteer
30	Pharmaceutical/Biotech
31	Professional Services
32	QA/Quality Control
33	Real Estate
34	Restaurant/Food Service
35	Retail
36	Sales
37	Science/Research
38	Skilled Labor
39	Technology
40	Telecommunications
41	Transportation/Logistics
42	Other

In [15]:
```
## Create a Data frame from a List of Dictionaries
data=[
{'Name': 'Krish', 'Age': 32, 'City': 'Bangalore'},
{'Name': 'John', 'Age': 34, 'City': 'Kolkata'},
{'Name': 'Bappy', 'Age': 45, 'City': 'Mumbai'},
{'Name': 'Jack', 'Age': 18, 'City': 'Bhopal'}
]
df=pd.DataFrame(data)
print(df)
print(type(df))
```

```
    Name  Age       City
0  Krish   32  Bangalore
1   John   34    Kolkata
2  Bappy   45     Mumbai
3   Jack   18      Bhopal
<class 'pandas.core.frame.DataFrame'>
```

In [16]:
```
df['Name']
```

Out[16]:
```
0    Krish
1     John
2    Bappy
3     Jack
Name: Name, dtype: object
```

In [64]:
```
df.iloc[3][0]
```

Out[64]:
```
'Jack'
```

In [18]:
```
df.iloc[0][2]
```

```
C:\Users\ANKIT\AppData\Local\Temp\ipykernel_16044\3241268714.py:1: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  df.iloc[3][0]
```

Out[18]:
```
'Jack'
```

In [19]:
```
df.iloc[0][2]
```

```
C:\Users\ANKIT\AppData\Local\Temp\ipykernel_16044\2529637664.py:1: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  df.iloc[3][2]
```

Out[19]:
```
'Bangalore'
```

In [67]:
```
df.iloc[0]
```

Out[67]:
```
Name         Krish
Age             32
City    Bangalore
Name: 0, dtype: object
```

ACCESSING A SPECIFIED ELEMENT

In [68]:
```
df
```

Out[68]:

	Name	Age	City
0	Krish	32	Bangalore
1	John	34	Kolkata
2	Bappy	45	Mumbai
3	JAck	18	bhopal

In [71]:
```
df.at[2,'Name']
```

Out[71]:
```
'Bappy'
```

In [73]:
```
df.at[1,'Age']
```

Out[73]:
```
34
```

ACCESSING A SPECIFIED ELEMENT USING A LIST

In [82]:
```
df.iat[2,2]
```

Out[82]:
```
'Mumbai'
```

In [ ]:
```

```

DATA MANUPULATION WITH DATAFRAME

In [86]:
```
df
```

Out[86]:

	Name	Age	City
0	Krish	32	Bangalore
1	John	34	Kolkata
2	Bappy	45	Mumbai
3	JAck	18	bhopal

In [97]:
```
## adding a new column

df['Salary']=[50000,60000, 70000, 82000]
df
```

Out[97]:

	Name	Age	City	Salary
0	Krish	32	Bangalore	50000
1	John	34	Kolkata	60000
2	Bappy	45	Mumbai	70000
3	JAck	18	bhopal	82000

In [116]:
```
#remove a column
df.drop('Salary',axis=1)
```

Out[116]:

	Name	Age	City
0	Krish	32	Bangalore
1	John	34	Kolkata
2	Bappy	45	Mumbai
3	JAck	18	bhopal

AXIS CHECKS ROW OR COLUMN INDEX

0 = ROW INDEX (default)

1= COLUMN INDEX

In [106]:
```
df
```

Out[106]:

	Name	Age	City	Salary
0	Krish	32	Bangalore	50000
1	John	34	Kolkata	60000
2	Bappy	45	Mumbai	70000
3	JAck	18	bhopal	82000

Column didn't got deleted

By default it's not permanent. It don't effect the original database

To delete the column permanently we have to use "inplace= True" inside the function

In [117]:
```
#remove a column
df.drop('Salary',axis=1,inplace=True)
```

In [118]:
```
df
```

Out[118]:

	Name	Age	City
0	Krish	32	Bangalore
1	John	34	Kolkata
2	Bappy	45	Mumbai
3	JAck	18	bhopal

In [122]:
```
df['Age']=df['Age']+1
```

In [123]:
```
df
```

Out[123]:

	Name	Age	City
0	Krish	33	Bangalore
1	John	35	Kolkata
2	Bappy	46	Mumbai
3	JAck	19	bhopal

In [125]:
```
#not permanent
```

In [127]:
```
df.drop(0)
```

Out[127]:

	Name	Age	City
1	John	35	Kolkata
2	Bappy	46	Mumbai
3	JAck	19	bhopal

In [146]:
```
#permanent
df.drop(1,inplace=True)
```

In [147]:
```
df
```

Out[147]:

	Name	Age	City
2	Bappy	46	Mumbai
3	JAck	19	bhopal

In [134]:
```
df=pd.read_csv('cust.csv')
df.head(5)
```

Out[134]:

	Index	Customer Id	First Name	Last Name	Company	City	Country	Phone 1	Phone 2	Email	Subscription Date	Website
0	1	DD37Cf93aecA6Dc	Sheryl	Baxter	Rasmussen Group	East Leonard	Chile	229.077.5154	397.884.0519x718	zunigavanessa@smith.info	2020-08-24	http://www.stephenson.com/
1	2	1f07b829A4CAd010	Preston	Lozano	Vega-Gentry	East Jimmychester	Djibouti	5153435776	686-620-1820x944	vmata@colon.com	2021-04-23	http://www.hobbs.com/
2	3	6F9d879bDAfE5a6	Roy	Berry	Murillo-Perry	Isabelborough	Antigua and Barbuda	+1-539-402-0259	(496)978-3969x58947	beckycarr@hogan.com	2020-03-25	http://www.lawrence.com/
3	4	5Cef8BFA16c5e3c	Linda	Olsen	Dominguez, Mcmillan and Donovan	Bensonview	Dominican Republic	001-808-617-6467x12895	+1-813-324-8756	stanleyblackwell@benson.org	2020-06-02	http://www.good-lyons.com/
4	5	053d585Ab6b3159	Joanna	Bender	Martin, Lang and Andrade	West Priscilla	Slovakia (Slovak Republic)	001-234-203-0635x76146	001-199-446-3860x3486	colinalvarado@miles.net	2021-04-17	https://goodwin-ingram.com/

In [135]:
```
df.describe()
```

Out[135]:

	Index
count	100.000000
mean	50.500000
std	29.011492
min	1.000000
25%	25.750000
50%	50.500000
75%	75.250000
max	100.000000

In [ ]:
```

```