

## Pandas

- Pandas stand for python data analysis library
- It is a free and open source
- It is an excellent for data processing and analysing real world data

## Pandas and data structures

- 1.Series:
  - Used to create one dimensional array with named index
  - Ex:A column in an excel sheet
- 2.Data frame:
  - Two dimensional,size mutable hetrogenous tabular data
  - Ex:A table in an excel sheet

```
In [1]: 1 import pandas as pd
```

```
In [2]: 1 pd.__version__
```

...

## Series

```
In [4]: 1 # creating series using list
        2
        3 s=pd.Series([10,20,30,40,50])
        4 print(s)
```

...

```
In [5]: 1 # creating series using list
        2 s=pd.Series([10,20,30,40,50])
        3 print(s)
        4 print(type(s))
        5
```

...

```
In [7]: 1 # creating series using tuple
        2 s1=pd.Series((11,12,13,14,15))
        3 print(s1)
        4 print(type(s1))
```

...

```
In [8]: 1 # creating series by using the numpy array
        2
        3 import numpy as np
        4 ar=np.array([12,36,78,65,45])
        5 s=pd.Series(ar)
        6 print(s,type(s))
```

...

```
In [9]: 1 # getting series index
        2 s.index
```

...

```
In [10]: 1 # to get series values
         2 s.values
```

...

```
In [16]: 1 # changing the index
         2 s.index=['a','b','c','d','e']
         3 s
```

...

```
In [17]: 1 # creating series by using range()
         2
         3 s2=pd.Series(range(5,100,10))
         4 s2
```

...

```
In [18]: 1 # accessing values using index
         2
         3 s2[0]
```

...

```
In [19]: 1 s2[1]
```

...

```
In [20]: 1 print(s2[1])
         2 print(s2[2])
         3 print(s2[3])
         4 print(s2[4])
```

...

```
In [22]: 1 # creating a series by using data type complex
         2
         3 s=pd.Series([1,2,3,4,5],dtype="complex")
         4 s
```

...

```
In [23]: 1 # creating series along with required index
         2
         3 s=pd.Series([11,12,13,14,15],index=['a','b','c','d','e'])
         4 s
```

...

```
In [26]: 1 # using comprehension
         2 s=pd.Series([i**2 for i in range(1,11)],index=range(1,11))
         3 s
```

...

## data frame

```
In [27]: 1 # creating data frame using list
         2
         3 df=pd.DataFrame([10,20,30,40,50])
         4 print(df)
         5 print(type(df))
```

...

```
In [29]: 1 # creating data frame using series
         2 s=pd.Series([11,67,85,35,90])
         3 df=pd.DataFrame(s)
         4 print(s)
```

...

```
In [30]: 1 # creating data frame using nested list
         2
         3 n1=[[10,20,30],[40,50,60]]
         4 df=pd.DataFrame(n1)
         5 df
```

...

```
In [31]: 1 # to get index
         2
         3 df.index
```

...

```
In [32]: 1 # to get column index
         2
         3 df.columns
```

...

```
In [33]: 1 # to get values from data frame
         2
         3 df.values
```

...

```
In [34]: 1 d={'emp':pd.Series(['emp1','emp2','emp3'],index=[1,2,3]),
2         'year':pd.Series([2000,2002,2004],index=[1,2,3])}
3 df1=pd.DataFrame(d)
4 df1
```

...

```
In [37]: 1 d2={'emp':pd.Series(['emp1','emp2','emp4'],index=[1,2,3]),
2         'dept':pd.Series(['hr','finance','op'],index=[1,2,3])}
3 df2=pd.DataFrame(d2)
4 df2
```

...

```
In [38]: 1 # combining two data frames
2 pd.merge(df1,df2)
```

...

```
In [39]: 1 pd.merge(df1,df2,how="left")
```

...

```
In [40]: 1 pd.merge(df1,df2,how="right")
```

...

```
In [42]: 1 # creating student dictionary name,roll number,branch
2
3 s={'name':['Pavani','Akhila','Sindhu'],
4    'roll number':[475,468,422],
5    'branch':['IT','CSE','ECE']}
6
7 df=pd.DataFrame(s)
8 df
```

...

```
In [43]: 1 df.columns
```

...

```
In [44]: 1 df.index
```

...

```
In [45]: 1 df.values
```

...

```
In [46]: 1 # to get top 1 record
2 df.head(1)
```

...

```
In [47]: 1 # to get top 2 record  
        2 df.head(2)
```

...

```
In [48]: 1 # df.head() -> gives top 5 records
```

```
In [49]: 1 # Last record from bottom  
        2 df.tail(1)
```

...

```
In [50]: 1 # Last 2 records from bottom  
        2 df.tail(2)
```

...

```
In [51]: 1 # tail(5) -> gives bottom 5 records from data frames
```

```
In [52]: 1 # to access particular column values  
        2 df['name']
```

...

```
In [53]: 1 # to access more than 1 column values  
        2  
        3 df[['name', 'branch']]
```

...

```
In [54]: 1 df
```

...

```
In [55]: 1 # to add a new column  
        2  
        3 df['marks']=[75,68,22]  
        4 df
```

...

```
In [56]: 1 # indexing  
        2 # iloc -> integer based indexing  
        3 # loc -> both integer and string based indexing
```

```
In [57]: 1 df.iloc[1,0]
```

...

```
In [58]: 1 df.loc[0, 'name']
```

...

```
In [59]: 1 # renaming the column name
        2 df.rename(columns={'name':'Name'},inplace=True)
        3 df
```

...

```
In [60]: 1 # renaming all column names
        2 df.columns=['Name','Rollnum','Branch','Marks']
        3 df
```

...

```
In [64]: 1 # changing particular value
        2 df.loc[0,'Name']='Akhi'
        3 df
```

...

```
In [65]: 1 df
```

...

```
In [67]: 1 df.loc[4]=['jyo',58,'mech',16]
        2 df
```

...

```
In [72]: 1 # to change index
        2 df.index=['a','b','c','d']
        3 df
```

...

```
In [68]: 1 # to delete data from DataFrame
        2
        3 # drop
        4 # axis=0 row
        5 # axis=1 column
```

```
In [73]: 1 df.drop('a',axis=0,inplace=True)
        2 df
```

...

```
In [74]: 1 df.drop('d',axis=0,inplace=True)
        2 df
```

...

```
In [75]: 1 # to delete a column 'marks'
        2
        3 df.drop('Marks',axis=1,inplace=True)
        4 df
```

...

```
In [76]: 1 # to delete all columns
        2 df.drop(df.columns,axis=1,inplace=True)
```

```
In [77]: 1 df
```

...

## File I/O

```
In [12]: 1 import pandas as pd
        2 df=pd.read_csv("C://Users//apssdc//Desktop//475//Salary_Data.csv")
        3 df
```

...

```
In [13]: 1 df.shape # no of rows and columns
```

...

```
In [14]: 1 df.head() # to get top 5 rows
```

...

```
In [15]: 1 df.tail() # to get bottom 5 records
```

...

```
In [16]: 1 df.describe()
```

...

```
In [17]: 1 df.columns
```

...

```
In [18]: 1 df.index
```

...

```
In [19]: 1 df.values
```

...

```
In [21]: 1 df.sample()
```

...

```
In [22]: 1 df.sample(10)
```

...

```
In [24]: 1 df
```

...

```
In [25]: 1 df.loc[2, 'Education Level']
```

...

```
In [26]: 1 # to update any value
2 df.loc[2, 'Education Level'] = 'PHD'
3 df
```

...

```
In [27]: 1 # to get the count of male, female, others
2
3 df['Gender']
```

...

```
In [29]: 1 # to get the count of male, female, others
2
3 df['Gender'].value_counts()
```

...

```
In [30]: 1 # to get the no of job titles
2
3 df['Job Title']
```

...

```
In [31]: 1 # to get the no of job titles
2
3 df['Job Title'].value_counts()
```

...

```
In [33]: 1 # to get the records of data scientists
2
3 df['Job Title'] == 'Data Scientist'
```

...

```
In [34]: 1 # to get the records of data scientists
2
3 df[df['Job Title'] == 'Data Scientist']
```

...

```
In [35]: 1 # to get the maximum salary record
2
3 df['Salary']
```

...

```
In [37]: 1 # to get the maximum salary record
2
3 df[df['Salary'] == max(df['Salary'])]
```

...



```
In [38]: 1 # sorting values based on salary
        2
        3 df.sort_values('Salary',ascending=False)
```

...

```
In [40]: 1 # unique
        2
        3 df['Job Title'].unique
```

...

```
In [42]: 1 # unique
        2
        3 len(df['Job Title'].unique())
```

...

```
In [45]: 1 # statistics
        2
        3 df.min()
```

...

```
In [43]: 1 df.min()
```

...

```
In [44]: 1 df.max()
```

...

```
In [46]: 1 df.mean()
```

...

```
In [47]: 1 df.std()
```

...