PYTHON PANDAS NOTES (CBSE class XII Informatics Practices 2025-26)

By D K Meshvaniya (PGT CS, PM SHRI KV Jetpur)

pandas:

- → Powerful data analysis
- → 3 data structures: Series, DataFrame, Panel
- → Most famous Python package for data science
- → Named from "Panel Data", and "Python Data Analysis"
- → created by Wes McKinney in 2008
- → Open source

matplotlib

- → Powerful and versatile plotting library
- → Used for visualization
- → Can create static, animated and interacting visualization
- → Customization of visualization with color, font, line style etc.
- → Can create: Line chart, bar chart, pie chart, area chart, histogram, scatter chart
- created by John D. Hunter.
- → Open source

Series

- → Series is a data structure in Pandas
- → 1 Dimensional
- → Homogeneous
- → Default indexes assigned to each elements
- → Default indexes are 0,1,2,3...
- → Index can be set numerical or alphabetical keys
- → Numerical index overwrites the default index.
- → Created with pandas.Series() function
- → Accepts two arguments data and index.
- → Index argument is optional.
- → Size mutable
- → A one-dimensional labeled array capable of holding any data type
- → Can be created from List, Dict, Array, Scalar value etc.
- → Syntax:

<seriesname>=pandas.Series(<list/tupe/dictionary/scalar value/ndarray>,index=<list>)

→ Example: Creating series from list

S1=pandas.Series([10,20,30,40])

DataFrame

- → DataFrame is a data structure in Pandas
- → 2 Dimensional
- → Data stored in rows and columns
- → Rows and columns are labeled
- → Size mutable
- → Heterogeneous
- → Created with pandas.DataFrame() function
- → Many functionalities like max, min, average etc available
- → Syntax

<DataFramename>=pandas.DataFrame(<dict/list/Series>,column=<list>,index=<list>)

→ Example

df1=pandas.DataFrame({'Eco':[77,95],'IP':[88,96]})

1	20
2	30
3	40

Eco

77

95

88

96

10

PYTHON PANDAS NOTES	(CBSE class XII Informatics	Practices 2025-26), by D K Meshvaniya
---------------------	-----------------------------	---------------------------------------

Mathematical operations on Series

- → Adding a value to Series
 - Adding a value to series object will add that value to all the elements of Series.
 - Try what happens:

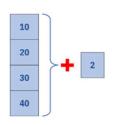
S1 series is 10,20,30,40.

S2=S1+2

- → Adding a Series to another Series
 - It will find the same indexes and perform operation
 - If index match not found, it will assign <u>NaN</u> to that index in result.
 - Try what happens:

S1 series is 10,20,30,40 with index a,b,c,d S2 series is 2,4,5,7 with index b,c,d,e

S3=S1+S2



a 10 b 20 c 30 d 40 b 2 c 4 d 5 e 7

head() function:

- → Returns top/first N records
- → Default value of N is 5
- → Return type is Series.
- → Ex: SeriesName.head(10) will return top/first 10 records in series form.

tail() function:

- → Returns bottom/last N records
- → Default value of N is 5
- → Return type is Series.
- → Ex: SeriesName.tail(10) will return bottom/last 10 records in series form.

Selection/slicing

- → We can get some selected elements of Series.
- → Index will be used
- → Ex: Series S1 has 10,20,30,40,50,60,70,80,90,100 with default index.

S1[index] will return element of given index.

S1[start:stop] will return elements from "start" to "stop" index.

S1[start:stop:step] will return elements from "start" to "stop" index with given step.

→ Series.loc[] can be used similarly.

Indexes in DataFrame

- → Default index of rows and column will be 0,1,2,...
- → If creating from Dictionary, then the keys of dictionary will become column titles.
- → Column names can be set using "column" argument.
- → "column" argument can accept list.
- → Rows indexes can be set using "index" argument.
- → "index" argument can accept list.
- → Rows index can be numerical or string.
- → If index are assigned numerical, the default index will be overwritten.

DataFrame can be created from

- (1) Dictionary of Series
 - Create some Series
 - Pass them to a dictionary
 - Pass the dictionary to DataFrame
 - Dictionary keys will become column titles.
 - Each Series will become a column in DataFrame.
- (2) List of Dictionaries
 - Create some dictionaries with same keys
 - Pass these dictionaries to a List
 - Pass the list to DataFrame.
 - Dictionary keys will become column titles.
 - Each dictionary will become a row in DataFrame.
- (3) Dictionary of Lists (Similar to "Dictionary of Series")
- (4) List of Lists
 - Create some lists
 - Pass these lists to a main list.
 - Pass the main list to DataFrame.
 - Each sub list will become rows of DataFrame.
 - Column title to be given manually.