CLASS NOTES	
Class: XII	**Date: 18-09-2020**
Subject: **Informatics Practices**	**Topic: Python Pandas I (Revision)**

Python Pandas I (Revision)

Some Facts:

A. PANDAS has derived its name from "**PAN**el **DA**ta **S**ystem".
B. The main author of Pandas is "**Wes McKinney**".
C. It is used as Python's library for data analysis.
D. It is an open source.
E. It offers high performance, easy to use data structure and data analysis tools.
F. In order to use Pandas we need to import by using "import pandas as pd".

Pandas data structure

Data structure: It refers to specialized way of storing data so as to apply a specific type of functionality on them.

Out of the many data structure two basic data structures – Series and DataFrame are universally popular for their dependability.

Difference Between Series and DataFrame

Property	Series	DataFrame
Dimensions	1-Dimentional	2-Dimentional
Type of Data	Homogeneous – All elements must be of same data type.	Heterogeneous – DataFrame objects can have different data types.
Mutability	Size- Immutable: Size of a series object, once created, cannot change.	Size – Mutable: We can drop or add the elements in an existing dataframe object.

Series Data Structure:

It represents a one dimensional array of indexed data.

It has two main components:

1. An array of actual data.
2. An associated array of indexes or data labels.

The index is used to access individual data values.

Example:

Index	Data
0	32
1	89
2	92
3	23

Index	Data
'A'	101
'B'	894
'C'	912
'D'	253

Creating Series Object

Make sure that you have imported pandas and numpy modules.

To create an empty series just issue the following command:

<Series Object>=pandas.Series()

To create a non-empty Series object, you need to specify arguments for data and indexes as per following syntax :

<Series Object>=pd.Series(data, index=idx)

data: It can be a python sequence, an ndarray, a python dictionary or a scalar value.

idx: It is a valid numpy datatype.

series() method with arguments:

Syntax is

<Series Object> = pandas.Series(data, index=idx)

1. Creating series using List:

```
Ser2 = pd.Series([12,23,34,45,67])
>>> print(Ser2)
0   12
1   23
2   34
3   45
4   67
```

Creating a series using range() method.

```
>>> ser3=pd.Series(range(4))
>>> print(ser3)
0   0
1   1
2   2
3   3
```

The default index can be changed also. Following example explains :

```
>>> ser3.index= ['One','Two','Three','Four']
>>> print(ser3)
One     0
Two     1
Three   2
Four    3
```

To create a series with explicitly defined index values.

```
>>> ser3=pd.Series([12,23,34,45,67], index = ['A','B','C','D','E'])
>>> print(ser3)
A   12
B   23
C   34
D   45
E   67
```

Accessing single or multiple values based on index from Series:

```
>>> ser3 ['B']
```

```
23
>>> ser3[['A','C','D']]
A    12
C    34
D    45
>>> ser3[['B','D','A']]
B    23
D    45
A    12
```

To perform indexing, slicing and accessing data from a series:

```
ser3 =
>>> print(ser3[1])          To extract the second value
23
>>> print(ser3[:3])         To extract the first three values
A    12
B    23
C    34
>>> print(ser3[-3:])        To extract the last three values
C    34
D    45
E    67
```

To create a series from a scalar or constant value:

```
>>> ser4 = pd.Series(25 , index = ['a','b','c','d'])
>>> print(ser4)
a    25
b    25
c    25
d    25
>>> ser5=pd.Series(25, index=[0,1,2,3,4])
>>> print(ser5)
0    25
1    25
2    25
3    25
4    25
```

Creating a series with range() and for loop:

```
>>> ser6=pd.Series(range(1,20,4), index=[x for x in 'abcde'])
>>> print(ser6)
a     1
b     5
c     9
d    13
e    17
```

Creating a series from two different list:

```
>>> rollno=[1,2,3,4,5]
>>> marks=[12,13,16,18,20]
>>> ser7=pd.Series(marks, index=rollno)
>>> print(ser7)
```

```
1    12
2    13
3    16
4    18
5    20
```

Creating a series from Dictionary :

```
>>> ser8=pd.Series({1:34,2:43,3:31,4:55})
>>> print(ser8)
    1    34
    2    43
    3    31
    4    55
```

Extracting values from a series using head() and tail() functions:

```
>>> print(ser8.head(2))
```
First two values from the series

```
Rollno
1    34
2    43
```

>>> print(ser8.tail(2)) **Last two values from the series**

```
Rollno
3    31
4    55
```

Extracting value using condition : Assume that we have a series ser7

```
>>> ser7
1    12
2    13
3    16
4    18
5    20
```

>>> ser7<17 **Condition to extract the values below 17(Vectorised)**

```
1    True
2    True
3    True
4    False
5    False
```

>>> ser7[ser7<17] **Condition to extract the values below 17(Filter)**

```
1    12
2    13
3    16
```

To delete an element usinf drop() method:

```
>>> ser7
1    12
2    13
3    16
4    18
5    20
```

>>> ser7.drop(2)

```
1    12
3    16
4    18
5    20
```

Adding NaN values in a Series object

```
import pandas as pd
import numpy as np
s1=pd.Series([13, np.NaN,34,np.NaN, 23])
print(s1)
```

Output:

```
0    13.0
1     NaN
2    34.0
3     NaN
4    23.0
```

Creating DataFrame:

Creating DataFrame from series:

```
import pandas as pd

Maths = pd.Series({1:34,2:45,3:23,4:44,5:20})
Science=pd.Series({1:22,2:34,3:43,4:13,5:25})

Df3=pd.DataFrame({'Maths-Marks': Maths, 'Science-Marks': Science })
print(Df3)
```

Out Put:

```
>>> Maths = pd.Series({1:34,2:45,3:23,4:44,5:20})
>>> Science=pd.Series({1:22,2:34,3:43,4:13,5:25})
>>> Df3=pd.DataFrame({'Maths-Marks': Maths, 'Science-Marks': Science })
>>> print(Df3)
   Maths-Marks  Science-Marks
1           34             22
2           45             34
3           23             43
4           44             13
5           20             25
```

Creating DataFrame from Dictionary

```
stu={'Name':['Babita','Keshav','Gunjan','Reshma'], 'Eng':[44,45,76,89], 'Maths':[76,56,45,55],
     'IP':[35,56,70,65]}
import pandas as pd
print(stu)
Studf=pd.DataFrame(stu)
print(Studf)
```

Output Screen

```
>>> stu={'Name':['Babita','Keshav','Gunjan','Reshma'], 'Eng':[44,45,76,89], 'Maths':[76,56,45,55], 'IP':[35,56,70,65]}
>>> print(stu)
{'Name': ['Babita', 'Keshav', 'Gunjan', 'Reshma'], 'Eng': [44, 45, 76, 89], 'Maths': [76, 56, 45, 55],
 'IP': [35, 56, 70, 65]}
>>> Studf=pd.DataFrame(stu)
>>> print(Studf)
    Name  Eng  Maths  IP
0  Babita   44     76    35
1  Keshav  45     56    56
2  Gunjan  76     45    70
3  Reshma 89     55    65
```

To sort the values of data frame in DESCENDING ORDER

```
>>> print(Df3.sort_values(by=['Maths-Marks'], ascending=False))
   Maths-Marks  Science-Marks
2           45             34
4           44             13
1           34             22
3           23             43
5           20             25
```

To Create an indexed DataFrame:

```
import pandas as pd

stu={‘Name’:[‘Babita’,’Keshav’,’Gunjan’,’Reshma’], ‘Eng’:[44,45,76,89], ‘Maths’:[76,56,45,55], ‘IP’:[35,56,70,65]}

print(stu)       # Dictionaries of Series has been generated now

>>> Studf2=pd.DataFrame(stu,index=['S01','S02','S03','S04'])
>>> Studf2

        Name     Eng  Maths  IP
S01  Babita     44     76     35
S02  Keshav     45     56     56
S03  Gunjan     76     45     70
S04  Reshma     89     55     65
```