BISHOP SCOTT BOYS' SCHOOL

(Affiliated to CBSE, New Delhi) Affiliation No.: 330726, School Campus: Chainpur, Jaganpura, By-Pass, Patna 804453.
Phone Number: 7061717782, 9798903550. ,
Web: www.bishopscottboysschool.com Email: info@bishopscottboysschool.com

ONLINE STUDY MATERIAL

SUBJEC-COMPUTER

SESSION-2020-21

CLASS-XII

TOPIC:PYTHON PANDAS-II(ASSIGNMENT)

DAY-1

Q1. What is difference between **iloc** and **loc** with respect to a DataFrame **?**

Ans-
(i) loc - To access row(s) and/or a combination of rows and columns **,** we use following syntax **:**

<DataFrameObject>**.loc[<startrow> : <endrow> , <startcolumn> : <endcolumn>]**

Examples

(i) To access a row , just give the row name/label as this **:**

<DF object>**.loc[**<row label> **, :]**

e.g.,

>>> **df5.loc[**'Delhi' **, :]**

OUTPUT

Population	10927986
Hospitals	189
Schools	7916

(ii) iloc - Sometimes your dataframe object does not contain row or column labels or even you m ay not remember them. In such cases , you can extract subset from dataframe using the row and c olumn numeric index/ position , but this time you will use i**loc instead of loc** means integer locat ion.

Syntax

<DF object>**.**iloc[<start row index> **:** <end row index> **,**<start col index> **:** <end column index>]

e.g.,

>>> df5**.**iloc[0 **:** 2 **,** 1 **:** 3]

	Hospitals	**Schools**
Delhi	189	7916
Mumbai	208	8508

Q2. What is difference between **iat** and **at** with respect to a DataFrame **?**

Ans-

You can use at or iat attributes with DF object as shown below **:**

Use	To
<DF object>**.**at[<row label> **,** <col label>]	Accessing a single value for a row/column label pair.
<DF object>**.iat**[<row index no**.** > **,** <col index no**.**>]	Access a single value for a row/column pair by integer position.

Examples

>>> df5**.at[‘**Chennai**’, ‘**Schools**’]**

OUTPUT

7617

>>> df5**.iat[** 3**,** 2**]**

OUTPUT
7617

DAY-2

Q3. Given **:**

```
import pandas as pd

d = { 'one' : pd.Series( [ 1. , 2. , 3. ] , index = ['a' , 'b' , 'c'] ) ,

     'two' : pd.Series( [ 1. , 2. , 3. , 4.] , index = ['a' , 'b' , 'c', 'd'] ) }

df = pd.DataFrame(d)

df1 = pd.DataFrame(d , index = ['d' , 'b' , 'a'] )

df2 = pd.DataFrame(d , index = ['d' , 'a'] , columns = ['two', 'three' ] )

print(df)

print(df1)

print(df2)
```

What will Python show the result as if you execute above code **?**

Ans-

```
import pandas as pd

d = { 'one' : pd.Series( [ 1. , 2. , 3. ] , index = ['a' , 'b' , 'c'] ) ,

     'two' : pd.Series( [ 1. , 2. , 3. , 4.] , index = ['a' , 'b' , 'c', 'd'] ) }

df = pd.DataFrame(d)

df1 = pd.DataFrame(d , index = ['d' , 'b' , 'a'] )

df2 = pd.DataFrame(d , index = ['d' , 'a'] , columns = ['two', 'three' ] )
```

```
print(df)
```

```
   one  two
a  1.0  1.0
b  2.0  2.0
c  3.0  3.0
d  NaN  4.0
```

print(df1)

```
   one  two
d  NaN  4.0
b  2.0  2.0
a  1.0  1.0
```

print(df2)

```
   two   three
d  4.0   NaN
a  1.0   NaN
```

Q4. How would you add a new column namely ‘val’ to a dataframe df that has 10 rows in it and has columns as ‘Item’ , ‘Qty’ , ‘Price’ **?** You can choose to put any values of your choice.

Ans-

```
import pandas as pd

dict1 = { 'Item' : [ 'Pencil' , 'Pen' , 'Eraser' , 'Sharpner', 'Scale' , 'Copy', 'Sticker' , 'Gum', 'Book','Stapler'] ,
          'Qty' : [6 , 8 , 5 , 20 , 60, 40, 30, 20, 10, 5] ,
          'Price' : [610 , 508 , 611 , 457 , 500 , 500, 100, 200, 300, 400]
                    }

df = pd.DataFrame(dict1)
print(df)
```

```
       Item  Qty  Price
0    Pencil    6    610
1       Pen    8    508
2    Eraser    5    611
3  Sharpner   20    457
4     Scale   60    500
5      Copy   40    500
6   Sticker   30    100
7       Gum   20    200
8      Book   10    300
9   Stapler    5    400
```

df['val']=1000

print(df)

```
      Item  Qty  Price   val
0   Pencil    6    610 1000
1      Pen    8    508 1000
2   Eraser    5    611 1000
3 Sharpner   20    457 1000
4    Scale   60    500 1000
5     Copy   40    500 1000
6  Sticker   30    100 1000
7      Gum   20    200 1000
8     Book   10    300 1000
9  Stapler    5    400 1000
```

DAY-3

Q5. Write code statements for a dataframe **df** for the following **:**

(a) delete an existing column from it.
(b) delete rows from 3 to 6 from it.
(c) check if the dataframe has any missing values.
(d) fill all missing values with 999 in it.

Ans-

Suppose a DataFrame namely **df** that stores the aid by NGOs for different states **:**

import pandas as pd

```
dict1 = { 'Toys' : [7916 , 8508 , 7226 , 7616 , 6000, 5000] ,
          'Books' : [6189 , 8208 , 5000 , 5200 , 6500, 4000] ,
          'Uniform' : [610 , 508 , 611 , 457 , 500 , 500],
          'Shoes' : [8810 , 6798 , 9611 , 6457 , 7000 , 8000 ]
        }
```

df = pd.DataFrame(dict1)

	Toys	Books	Uniforms	Shoes
0	7916	6189	610	8810
1	8508	8208	508	6798
2	7226	6149	611	9611
3	7617	6157	457	6457
4	6000	6500	500	7000
5	5000	4000	500	8000

(a) delete an existing column from

it. Ans-

```
del df['Shoes']
print(df)
```

OUTPUT

```
   Toys  Books  Uniform
0  7916   6189      610
1  8508   8208      508
2  7226   5000      611
3  7616   5200      457
4  6000   6500      500
5  5000   4000      500
```

(b) delete rows from 3 to 6 from it.

.

Ans-

```
df.drop([3,4,5])
print(df)
```

OUTPUT

```
   Toys  Books  Uniform
0  7916   6189      610
1  8508   8208      508
2  7226   5000      611
```

(c) check if the dataframe has any missing values.

.

Ans- Question (c) and (d) is from next chapter chapter-3 so will be answered in next couple of weeks.

Q6. Given a dataframe df as shown below **:**

	age	**favorite_color**	**marks**
Riya Sen	20	blue	88
Asma Tauhid	19	red	92
Gurkirat Kaur	22	yellow	95
Daniel Thames	21	green	70

Write statements to add a column **'**eligible**'** with default value as 'yes'

Ans-

```
import pandas as pd

dict1 = { 'age' : [20 , 19 , 22 , 21 ] ,
          'favorite_color' : ['blue' , 'red' , 'yellow' , 'green'] ,
          'marks' : [88 , 92 , 95 , 70],
                }

df = pd.DataFrame(dict1 , index = ['Riya Sen' , 'Asma Tauhid' , 'Gurkirat Kaur' , 'Daniel
Tham es'] )
df["eligible"] = 'yes'
print(df)
```

OUTPUT

```
               age favorite_color marks eligible
Riya Sen        20           blue     88      yes
Asma Tauhid     19            red     92      yes
Gurkirat Kaur   22         yellow     95      yes
Daniel Thames   21          green     70      yes
```

DAY-4+5

Q7. Create a dataframe namely **cdf** from a dictionary containing values in form of two series given below **:**

Series 1 :		**Series 2 :**	
a	**1**	**a**	**1**
b	**2**	**b**	**2**
c	**3**	**c**	**3**
		d	**4**

The column names of the dataframe should be **‘**Ser1**’ ,** ‘Ser2**’.**

Ans-

```
import pandas as pd

Series1 = pd.Series( [1, 2, 3] , index = [‘a’, ‘b’, ‘c’])
Series2 = pd.Series( [1, 2, 3, 4] , index = [‘a’, ‘b’, ‘c’, ‘d’])


dict1 = { ‘Ser1’ : Series1 , ‘Ser2’ : Series2 }

cdf = pd.DataFrame(dict1)

print(cdf)
```

OUTPUT

```
    Ser1  Ser2
a    1.0     1
b    2.0     2
c    3.0     3
d    NaN     4
```

Q8. Consider the following dataframe ndf as shown below **:**

	Column1	**Column2**	**Column3**	**Res**
T1	62**.**893165	100**.**0	60**.**00	True
T2	94**.**734483	100**.**0	59**.**22	True
T3	49**.**090140	100**.**0	46**.**04	False
T4	38**.**487265	85**.**4	58**.**62	False

What will be the output produced by following statements ?

(i) print(ndf**.**iat[3 **,** 2] **,** ndf**.**iat[2 **,** 3])
(ii) print(ndf**.**loc[**:** , 'Column3' **:**])
(iii) print(ndf**.**iloc[**:** 2 **,** 2 **:**])

Ans-

```
import pandas as pd

        dict1 = { 'Column1' : [62.893165 , 94.734483 , 49.090140 , 38.487265 ] ,
                  'Column2' : [100.0 , 100.0 , 100.0 , 85.4] ,
                  'Column3' : [60.00 , 59.22 , 46.04 , 58.62],
                  'Res' : [ 'True' , 'True' , 'False' , 'False' ]
                }

ndf = pd.DataFrame(dict1 , index = ['T1' , 'T2' , 'T3' , 'T4'] )
```

(i) print(ndf**.**iat[3 **,** 2] **,** ndf**.**iat[2 **,** 3])

OUTPUT

```
58.62   False
```

(ii) print(ndf**.**loc[**:** , 'Column3' **:**])

OUTPUT

```
      Column3    Res
T1    60.00      True
T2    59.22      True
T3    46.04      False
T4    58.62      False
```

(iii) print(ndf.iloc[: 2 , 2 :])

OUTPUT

	Column3	Res
T1	60.00	True
T2	59.22	True

DAY-5

Practice all the solved in example in chapter-3 PLOTTING BAR GRAPH AND CHART