notes created by swapan chetri

Libraries in python

python library is a collection of functions and methods that allows us to perform many actions without writing our code.

pandas -> data manipulation

numpy -> numerical computing

matplotlib -> visualization

pandas library

-> pandas stand for "panel data" and it is the core library for Data Manipulation and Data Analysis.

-> It consists of Single and Multi dimensional data structures for data manipulation.

Single Dimensional -> Series Object

Multi Dimensional -> DataFrame

importing pandas

In [2]:

```
#if pandas is not installed try this in the cell(!pip install pandas)
import pandas as pd
```

pandas series object

In [6]:

```
s1=pd.Series([1,2,3,4,5])
s1
# it will print the labels/index at the first eg. 0,1,2,3,4.....
```

Out[6]:

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

In [7]:

```
# checking the type of s1
type(s1)
```

Out[7]:

```
pandas.core.series.Series
```

In [11]:

```
#changing index
s1=pd.Series([1,2,3,4,5],index=["a","b","c","d","e"])
s1
```

Out[11]:

```
a    1
b    2
c    3
d    4
e    5
dtype: int64
```

In [15]:

```
# series object from dictionary
s1=pd.Series({"Kundan":98,"Indraja":97,"Ayesha":96,"Jhansi":97})
s1
```

Out[15]:

```
Kundan     98
Indraja    97
Ayesha     96
Jhansi     97
dtype: int64
```

In [17]:

```
# changing the index position
s2=pd.Series({"Kundan":98,"Indraja":97,"Ayesha":96,"Jhansi":97},index=["Indraja","Jhansi","
s2
```

Out[17]:

```
Indraja    97
Jhansi     97
Kundan     98
Ayesha     96
dtype: int64
```

In [18]:

```
# extracting individual elements
# extracting the single element
s1=pd.Series([1,2,3,4,5,6,7,8,9])
s1[3]
```

Out[18]:

```
4
```

In [19]:

```
# extracting elements from back
s1[-3:]
```

Out[19]:

```
6    7
7    8
8    9
dtype: int64
```

In [20]:

```
# extracting a sequence of elements
s1[:4]
```

Out[20]:

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

In [21]:

```
# basic math operations on series object
# adding a scaler value to series elements
s1=pd.Series([1,2,3,4,5,6,7,8,9])
s1+5
```

Out[21]:

```
0     6
1     7
2     8
3     9
4    10
5    11
6    12
7    13
8    14
dtype: int64
```

In [23]:

```
# adding two series
s1=pd.Series([1,2,3,4,5,6,7,8,9])
s2=pd.Series([10,11,12,13,14,15,16,17,18,19])
s1+s2
```

Out[23]:

```
0    11.0
1    13.0
2    15.0
3    17.0
4    19.0
5    21.0
6    23.0
7    25.0
8    27.0
9     NaN
dtype: float64
```

pandas DataFrame

-> DataFrame is a 2-dimensional labelled data structure.

-> A DataFrame comprises of rows and columns.

In [31]:

```
#creating DataFrame using dictionary
students=pd.DataFrame({"Name":["Arghya Paul",
"Asira O",
"Ch. Jhansi Priya",
"Dungala Gangaraju",
"Gandeti Poornachandra rao",
"Gavara Anil",
"Gokada Rajeswari",
"Harsh Rajesh Tiwari"
],"Marks":[97,98,99,93,95,94,97,91]})
#printing students
students
```

Out[31]:

	Name	Marks
0	Arghya Paul	97
1	Asira O	98
2	Ch. Jhansi Priya	99
3	Dungala Gangaraju	93
4	Gandeti Poornachandra rao	95
5	Gavara Anil	94
6	Gokada Rajeswari	97
7	Harsh Rajesh Tiwari	91

DataFrame inbuilt functions

In [29]:

```
# head()
# returns the first five rows from the dataset
students.head()
```

Out[29]:

	Name	Marks
0	Arghya Paul	97
1	Asira O	98
2	Ch. Jhansi Priya	99
3	Dungala Gangaraju	93
4	Gandeti Poornachandra rao	95

In [32]:

```
#  tail()
# returns the last five rows from the dataset
students.tail()
```

Out[32]:

	Name	Marks
3	Dungala Gangaraju	93
4	Gandeti Poornachandra rao	95
5	Gavara Anil	94
6	Gokada Rajeswari	97
7	Harsh Rajesh Tiwari	91

In [36]:

```
# shape()
# returns the number of rows and columns from the dataset
students.shape
```

Out[36]:

```
(8, 2)
```

In [38]:

```
# describe()
# to know about dataframe (general information) just like mean, std....
students.describe()
```

Out[38]:

	Marks
count	8.000000
mean	95.500000
std	2.725541
min	91.000000
25%	93.750000
50%	96.000000
75%	97.250000
max	99.000000

to load dataset

In [64]:

```
# df=pd.read_csv("file_name.csv")
```

Extracting values/records from dataset

In [43]:

```
# .iloc[]
# first part is rows and the second part columns
students.iloc[0:3,0:2]
```

Out[43]:

	Name	Marks
0	Arghya Paul	97
1	Asira O	98
2	Ch. Jhansi Priya	99

In [48]:

```
# .loc[]
# we can pass the values of rows and columns with name
students.loc[0:5,("Name")]
# here 0 and 5 both are inclusive
```

Out[48]:

```
0                  Arghya Paul
1                      Asira O
2             Ch. Jhansi Priya
3            Dungala Gangaraju
4    Gandeti Poornachandra rao
5                  Gavara Anil
Name: Name, dtype: object
```

dropping the columns

In [52]:

```
# drops the columns
students.drop("Marks",axis=1)
```

Out[52]:

	Name
0	Arghya Paul
1	Asira O
2	Ch. Jhansi Priya
3	Dungala Gangaraju
4	Gandeti Poornachandra rao
5	Gavara Anil
6	Gokada Rajeswari
7	Harsh Rajesh Tiwari

In [54]:

```
# drop the rows
students.drop([1,2,3],axis=0)
```

Out[54]:

	Name	Marks
0	Arghya Paul	97
4	Gandeti Poornachandra rao	95
5	Gavara Anil	94
6	Gokada Rajeswari	97
7	Harsh Rajesh Tiwari	91

More pandas functions

In [63]:

```
# mean
students.mean()
```

Out[63]:

```
Marks    95.5
dtype: float64
```

In [62]:

```
# median
students.median()
```

Out[62]:

```
Marks    96.0
dtype: float64
```

In [61]:

```
# minimum
students["Marks"].min()
```

Out[61]:

```
91
```

In [60]:

```
# maximum
students["Marks"].max()
```

Out[60]:

```
99
```

In [65]:

```
# user defined functions on pandas DataFrame or we can pass inbuilt functions
#defining the functions
def add(a):
    return a*5
students[["Marks"]].apply(add)
```

Out[65]:

	Marks
0	485
1	490
2	495
3	465
4	475
5	470
6	485
7	455

In [68]:

```
#  value counts
students["Name"].value_counts()
```

Out[68]:

```
Asira O                      1
Gokada Rajeswari             1
Harsh Rajesh Tiwari          1
Dungala Gangaraju            1
Gavara Anil                  1
Arghya Paul                  1
Gandeti Poornachandra rao    1
Ch. Jhansi Priya             1
Name: Name, dtype: int64
```

In [69]:

```
# sort values ascending
students.sort_values(by="Marks")
```

Out[69]:

	Name	Marks
7	Harsh Rajesh Tiwari	91
3	Dungala Gangaraju	93
5	Gavara Anil	94
4	Gandeti Poornachandra rao	95
0	Arghya Paul	97
6	Gokada Rajeswari	97
1	Asira O	98
2	Ch. Jhansi Priya	99

In [70]:

```
# sort values descending
students.sort_values(by="Marks",ascending=False)
```

Out[70]:

	Name	Marks
2	Ch. Jhansi Priya	99
1	Asira O	98
0	Arghya Paul	97
6	Gokada Rajeswari	97
4	Gandeti Poornachandra rao	95
5	Gavara Anil	94
3	Dungala Gangaraju	93
7	Harsh Rajesh Tiwari	91

export dataframe to csv

In [ ]:

```
# here df is dataframe name, to_csv is function and "export.csv" is the name that we want t
# df.to_csv("export.csv",index=False)
```