In [1]:

```
# import major libraries
import pandas as pd
import numpy as np
```

Two data types:

- Series --> column
- DataFrames --> Table

In [2]:

```
#series -- > column
#list
l1=[10,20,30,40]
pd.Series(l1)
```

Out[2]:

```
0    10
1    20
2    30
3    40
dtype: int64
```

In [3]:

```
# countries
countries = ['India','China','USA','Japan','Russia']
countries = pd.Series(countries)
countries
```

Out[3]:

```
0     India
1     China
2       USA
3     Japan
4    Russia
dtype: object
```

In [6]:

```
dict1={
    'Dunki':'SRK',
    'Sultan':'SK',
    'Sanju':'Ranbir kapoor',
    'PK':'AK',
    'Holiday':'Akshay Kumar'
}
movies = pd.Series(dict1)
movies
#labelled indexes
```

Out[6]:

```
Dunki                SRK
Sultan                SK
Sanju      Ranbir kapoor
PK                    AK
Holiday     Akshay Kumar
dtype: object
```

In [7]:

```
sub = ['Hindi','Englis','SST','Science']
```

```
marks=[np.nan,78,56,np.nan]
std=pd.Series(marks,index=sub,name='Vipul_Marks')
std
```

Out[7]:

```
Hindi       NaN
Englis     78.0
SST        56.0
Science     NaN
Name: Vipul_Marks, dtype: float64
```

In [9]:

```
#(saving Table in home of jupyter CSV=Comma Separated Values)
std.to_csv('std.csv')
```

In [10]:

```
#numpy arrays --> series
marks = pd.Series(np.random.randint(0,101,100),index=range(1,101,1)) #index --
marks
```

Out[10]:

```
1      88
2      74
3      78
4      90
5       3
       ..
96     20
97     19
98     15
99     86
100    93
Length: 100, dtype: int32
```

In [11]:

```
# attributes
# basic attributes
# index
print(marks.index)
print(countries.index)
print(movies.index)
```

```
RangeIndex(start=1, stop=101, step=1)
RangeIndex(start=0, stop=5, step=1)
Index(['Dunki', 'Sultan', 'Sanju', 'PK', 'Holiday'], dtype='object')
```

In [12]:

```
# values
marks.values
movies.values
```

Out[12]:

```
array(['SRK', 'SK', 'Ranbir kapoor', 'AK', 'Akshay Kumar'], dtype=object)
```

In [13]:

```
#dtype
marks.dtype
std.dtype
print(countries.dtype)
```

```
object
```

In [14]:
```
#name
marks.name
std.name
```
Out[14]:
```
'Vipul_Marks'
```
In [15]:
```
#shape
marks.shape
```
Out[15]:
```
(100,)
```
In [16]:
```
#size
marks.size
std.size
```
Out[16]:
```
4
```
In [17]:
```
#count function
marks.count()
std.count()
```
Out[17]:
```
np.int64(2)
```
In [18]:
```
#ndim
marks.ndim
```
Out[18]:
```
1
```
In [20]:
```
# isunique
countries.is_unique
marks.is_unique
```
Out[20]:
```
False
```
In [21]:
```
#empty
marks.empty
```
Out[21]:
```
False
```
In [22]:
```
#str
countries.str.upper()
```
Out[22]:
```
0     INDIA
1     CHINA
2       USA
3     JAPAN
4    RUSSIA
dtype: object
```
In [24]:
```
# function
# head
# tail
# sample
marks
# formula for all value shown 1 to 100(consider not you use it)  --> pd.setopt
marks.head(10) #top 5
marks.tail() #last 5
marks.sample(5) # random 5
```

```
Out[24]: 99    86
         50    21
         65    11
         95    75
         42    19
         dtype: int32
```

```
In [36]: marks.head(10) #top 10
```

```
Out[36]: 1     88
         2     74
         3     78
         4     90
         5      3
         6     19
         7     46
         8     84
         9     41
         10    30
         dtype: int32
```

```
In [37]: marks.tail() #last 5
```

```
Out[37]: 96     20
         97     19
         98     15
         99     86
         100    93
         dtype: int32
```

```
In [25]: #info
         #describe()
         marks.info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 100 entries, 1 to 100
Series name: None
Non-Null Count  Dtype
--------------  -----
100 non-null    int32
dtypes: int32(1)
memory usage: 532.0 bytes
```

```
In [26]: marks.describe()
```

Out[26]:
```
count    100.000000
mean      51.890000
std       29.570629
min        0.000000
25%       22.000000
50%       54.000000
75%       79.250000
max       96.000000
dtype: float64
```

In [28]:
```
#SELECTION AND FILTERATION --> IMPORTANT
```

In [29]:
```
marks[3] #indexing
```

Out[29]:
```
np.int32(78)
```

In [30]:
```
marks[3:5] # slicing
```

Out[30]:
```
4    90
5     3
dtype: int32
```

In [31]:
```
#loc --> labelled indexing
movies
```

Out[31]:
```
Dunki                 SRK
Sultan                 SK
Sanju      Ranbir kapoor
PK                     AK
Holiday     Akshay Kumar
dtype: object
```

In [32]:
```
movies.loc['Dunki':'PK']
```

Out[32]:
```
Dunki                SRK
Sultan                SK
Sanju     Ranbir kapoor
PK                    AK
dtype: object
```

In [33]:
```
#iloc --> index
countries[1:4:2] # 2 stpe size or gap
```

Out[33]:
```
1    China
3    Japan
dtype: object
```

In [34]:
```
#condition based
marks[marks<10]
```

Out[34]:

```
5     3
30    8
43    0
55    0
85    6
dtype: int32
```

In [38]:

```
# sorting methods
#sort_values
#sort_index
```

In [39]:

```
marks.sort_values()
```

Out[39]:

```
55     0
43     0
5      3
85     6
30     8
      ..
57    94
75    94
62    96
29    96
59    96
Length: 100, dtype: int32
```

In [41]:

```
marks=marks.sort_values(ascending=False)
marks
```

Out[41]:

```
29    96
62    96
59    96
57    94
75    94
      ..
30     8
85     6
5      3
55     0
43     0
Length: 100, dtype: int32
```

In [42]:

```
marks=marks.sort_index()
marks
```

```
Out[42]: 1      88
         2      74
         3      78
         4      90
         5       3
                ..
         96     20
         97     19
         98     15
         99     86
         100    93
         Length: 100, dtype: int32
```

```
In [43]: #aggregate functions
         #sum
         marks.sum()
         std.sum()
```

```
Out[43]: np.float64(134.0)
```

```
In [45]: #mean
         marks.mean()
```

```
Out[45]: np.float64(51.89)
```

```
In [46]: #median
         marks.median()
```

```
Out[46]: 54.0
```

```
In [47]: #mode
         marks.mode()
```

```
Out[47]: 0    11
         1    19
         dtype: int32
```

```
In [49]: #value_counts()
         marks.value_counts().head()
```

```
Out[49]: 19    5
         11    5
         47    4
         62    4
         86    4
         Name: count, dtype: int64
```

```
In [50]: #Variance
         marks.var()
```

```
Out[50]: 874.4221212121213
```

In [52]:

```
#std
marks.std()
```

Out[52]:

```
29.57062936787314
```

In [53]:

```
#min/max
print(marks.min())
print(marks.max())
```

```
0
96
```

In [54]:

```
#count
marks.count()
```

Out[54]:

```
np.int64(100)
```

In [55]:

```
#quantile
print(marks.quantile(0.25))
print(marks.quantile(0.50))
print(marks.quantile(0.75))
```

```
22.0
54.0
79.25
```

In [57]:

```
#replace and clean
#replace
countries.replace('USA','SOUTH KORIA')
```

Out[57]:

```
0          India
1          China
2    SOUTH KORIA
3          Japan
4         Russia
dtype: object
```

In [56]:

```
#astype
marks.astype(float)
```

Out[56]:

```
1      88.0
2      74.0
3      78.0
4      90.0
5       3.0
       ... 
96     20.0
97     19.0
98     15.0
99     86.0
100    93.0
Length: 100, dtype: float64
```

In [58]:

```
#round
```

```
marks.round(2)
```

Out[58]:

```
1      88
2      74
3      78
4      90
5       3
       ..
96     20
97     19
98     15
99     86
100    93
Length: 100, dtype: int32
```

In [61]:

```
#clip
marks.clip(10,60).head(20) #--> 10 se niche 10 ho jaye ge values
```

Out[61]:

```
1     60
2     60
3     60
4     60
5     10
6     19
7     46
8     60
9     41
10    30
11    60
12    47
13    60
14    56
15    11
16    60
17    34
18    60
19    60
20    18
dtype: int32
```

In [62]:

```
#unique
#duplicated
#value_counts
#to_dict
```

In [63]:

```
marks.unique()
```

Out[63]:

```
array([88, 74, 78, 90,  3, 19, 46, 84, 41, 30, 86, 47, 62, 56, 11, 69, 34,
       63, 18, 33, 25, 87, 76, 96,  8, 66, 82, 15, 59, 44, 26,  0, 92, 32,
       36, 39, 21, 22, 89, 52, 12, 60, 94, 80, 70, 93, 37, 83, 40, 13, 64,
       79,  6, 67, 77, 23, 75, 20], dtype=int32)
```

In [68]:

```
marks[marks.duplicated()].head(15)
```

```
Out[68]: 19    90
         21    19
         22    11
         23    63
         27    63
         31    78
         32    63
         36    69
         39    47
         41    66
         42    19
         45    87
         48    47
         55     0
         58    11
         dtype: int32
```

```
In [70]: marks.drop_duplicates().head(15)
```

```
Out[70]: 1     88
         2     74
         3     78
         4     90
         5      3
         6     19
         7     46
         8     84
         9     41
         10    30
         11    86
         12    47
         13    62
         14    56
         15    11
         dtype: int32
```

```
In [71]: movies.value_counts()
```

```
Out[71]: SRK              1
         SK               1
         Ranbir kapoor    1
         AK               1
         Akshay Kumar     1
         Name: count, dtype: int64
```

```
In [72]: movies.value_counts().to_dict()
```

```
Out[72]: {'SRK': 1, 'SK': 1, 'Ranbir kapoor': 1, 'AK': 1, 'Akshay Kumar': 1}
```

```
In [73]: #filling
         #dropna
         #isnull
         std.isnull().sum()
```

```
Out[73]: np.int64(2)
```

```
In [75]: #dropna --> NA VALUE HATA DIYA
         std.dropna()
```

```
Out[75]: Englis    78.0
         SST       56.0
         Name: Vipul_Marks, dtype: float64
```

```
In [77]: #filling
         std.fillna(10)
```

```
Out[77]: Hindi      10.0
         Englis     78.0
         SST        56.0
         Science    10.0
         Name: Vipul_Marks, dtype: float64
```

```
In [78]: std[std.isnull()]
```

```
Out[78]: Hindi     NaN
         Science   NaN
         Name: Vipul_Marks, dtype: float64
```

```
 In [ ]:
```