In [2]:
```
import pandas as pd
import numpy as np
```

In [7]:
```
a = [1,2,3,4,5,6,'ankush','aman']
print(a)
print(type(a))
```

```
[1, 2, 3, 4, 5, 6, 'ankush', 'aman']
<class 'list'>
```

Series

In [8]:
```
a = pd.Series([1,2,3,4,5,6,'ankush','aman'])
print(a)
print(type(a))
```

```
0         1
1         2
2         3
3         4
4         5
5         6
6    ankush
7      aman
dtype: object
<class 'pandas.core.series.Series'>
```

In [11]:
```
a = pd.Series([1,2,3,4,5,6,'ankush','aman'],index=['a','b','c','d','e','f'
print(a)
print(type(a))
```

```
a         1
b         2
c         3
d         4
e         5
f         6
g    ankush
h      aman
dtype: object
<class 'pandas.core.series.Series'>
```

In [15]:
```
a = pd.Series([1,2,3,4,5,6],index=['a','b','c','d','e','f'],dtype='float')
print(a)
print(type(a))


# if charactor assign in list than it can not convert into float
```

```
a    1.0
b    2.0
c    3.0
d    4.0
e    5.0
f    6.0
dtype: float64
<class 'pandas.core.series.Series'>
```

In [16]:
```
a = pd.Series([1,2,3,4,5,6],index=['a','b','c','d','e','f'],dtype='float',
print(a)
print(type(a))
```

```
a    1.0
b    2.0
c    3.0
d    4.0
e    5.0
f    6.0
Name: xyz, dtype: float64
<class 'pandas.core.series.Series'>
```

In [18]:
```
a = pd.Series(index=['a','b','c','d','e','f'])
print(a)
print(type(a))
```

```
a   NaN
b   NaN
c   NaN
d   NaN
e   NaN
f   NaN
dtype: float64
<class 'pandas.core.series.Series'>

C:\Users\PC\AppData\Local\Temp\ipykernel_312\1309681546.py:1: FutureWarning:
The default dtype for empty Series will be 'object' instead of 'float64' in a
future version. Specify a dtype explicitly to silence this warning.
  a = pd.Series(index=['a','b','c','d','e','f'])
```

In [19]:
```
a = pd.Series(5.2,index=['a','b','c','d','e','f'])
print(a)
print(type(a))
```

```
a    5.2
b    5.2
c    5.2
d    5.2
e    5.2
f    5.2
dtype: float64
<class 'pandas.core.series.Series'>
```

In [21]:
```
s1 = pd.Series({'a':'ankush','b':'gupta','c':'aids'})
print(s1)
print(type(s1))
```

```
a    ankush
b     gupta
c      aids
dtype: object
<class 'pandas.core.series.Series'>
```

Slicing in Series

In [22]:
```
a = pd.Series([1,2,3,4,5,6],index=['a','b','c','d','e','f'],dtype='float',
print(a)
print(type(a))
```

```
a    1.0
b    2.0
c    3.0
d    4.0
e    5.0
f    6.0
Name: xyz, dtype: float64
<class 'pandas.core.series.Series'>
```

In [23]:
```
a['d']
```

Out[23]: 4.0

In [24]:
```
a['f']
```

Out[24]: 6.0

In [28]:
```
a['c':]
```

Out[28]:
```
c    3.0
d    4.0
e    5.0
f    6.0
Name: xyz, dtype: float64
```

In [29]:
```
1 a.max()
```

Out[29]:
```
6.0
```

In [30]:
```
1 a.min()
```

Out[30]:
```
1.0
```

In [35]:
```
1 s1 = pd.Series([1,2,3,4,5,6,7])
2 s2 = pd.Series([11,12,13,14,15,16,17])
3 print(s1)
4 print()
5 print(s2)
```

```
0    1
1    2
2    3
3    4
4    5
5    6
6    7
dtype: int64

0    11
1    12
2    13
3    14
4    15
5    16
6    17
dtype: int64
```

operation

In [37]:
```
1 print(s1 + s2)
```

```
0    12
1    14
2    16
3    18
4    20
5    22
6    24
dtype: int64
```

DataFrame

In [43]:
```
data1 = pd.DataFrame([1,2,3,4,5,6])
data2 = pd.Series([1,2,3,4,5,6])
print('This is dataframe ')
print(data1)
print('\n\nThis is series ')
print(data2)
```

```
This is dataframe
   0
0  1
1  2
2  3
3  4
4  5
5  6


This is series
0    1
1    2
2    3
3    4
4    5
5    6
dtype: int64
```

In [46]:
```
data3 = pd.DataFrame([[1,2,3],[4,5,6],[7,8,9]])
data3
```

Out[46]:

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

In [47]:
```
data3 = pd.DataFrame([[1,2,3],[4,5,6],[7,8,9]],columns=['a','b','c'])
data3
```

Out[47]:

	a	b	c
0	1	2	3
1	4	5	6
2	7	8	9

In [49]:
```
data3 = pd.DataFrame([{'id':[1,2,3],'id2':[4,5,6],'id3':[7,8,9]}])
data3
```

Out[49]:

	id	id2	id3
0	[1, 2, 3]	[4, 5, 6]	[7, 8, 9]

In [51]:
```
data3 = pd.DataFrame({'id':[1,2,3,4],'sn':[11,12,13,14],'mn':[21,22,23,24]}
data3
```

Out[51]:

	id	sn	mn
0	1	11	21
1	2	12	22
2	3	13	23
3	4	14	24

In [58]:
```
arr1 = np.linspace(1,26,25).reshape(5,5)
print(arr1)

df=pd.DataFrame(arr1,dtype=int)
df
```

```
[[ 1.          2.04166667  3.08333333  4.125       5.16666667]
 [ 6.20833333  7.25        8.29166667  9.33333333 10.375     ]
 [11.41666667 12.45833333 13.5        14.54166667 15.58333333]
 [16.625      17.66666667 18.70833333 19.75       20.79166667]
 [21.83333333 22.875      23.91666667 24.95833333 26.        ]]

C:\Users\PC\AppData\Local\Temp\ipykernel_312\1831896845.py:4: FutureWarning:
In a future version, passing float-dtype values and an integer dtype to DataF
rame will retain floating dtype if they cannot be cast losslessly (matching S
eries behavior). To retain the old behavior, use DataFrame(data).astype(dtyp
e)
  df=pd.DataFrame(arr1,dtype=int)
```

Out[58]:

	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	11	12	13	14	15
3	16	17	18	19	20
4	21	22	23	24	26

In [62]:
```
df=pd.DataFrame(arr1,dtype=int,columns=[f'{v}{i}' for i,v in enumerate('ff
df
```

```
C:\Users\PC\AppData\Local\Temp\ipykernel_312\3719523278.py:1: FutureWarning:
In a future version, passing float-dtype values and an integer dtype to DataF
rame will retain floating dtype if they cannot be cast losslessly (matching S
eries behavior). To retain the old behavior, use DataFrame(data).astype(dtyp
e)
  df=pd.DataFrame(arr1,dtype=int,columns=[f'{v}{i}' for i,v in enumerate('fff
ff')])
```

Out[62]:

	f0	f1	f2	f3	f4
0	1	2	3	4	5
1	6	7	8	9	10
2	11	12	13	14	15
3	16	17	18	19	20
4	21	22	23	24	26

In [ ]:
```

```

In [65]:
```
df1 = pd.DataFrame({'id':[101,102,103,104,105,106,107],'name':['ram','kapi
df1
```

Out[65]:

	id	name	per
0	101	ram	45
1	102	kapil	67
2	103	riya	89
3	104	megha	34
4	105	saket	56
5	106	aman	78
6	107	pranay	99

In [67]:
```
df2 = pd.DataFrame({'Grade1':['a+','b+','b+','a++','c','d','a++']})
df2
```

Out[67]:

	Grade1
0	a+
1	b+
2	b+
3	a++
4	c
5	d
6	a++

In [ ]:
```
1 df1['Grade2'] = df2
```

In [71]:
```
1 df1[5] = df2
```

In [72]:
```
1 df1
```

Out[72]:

	id	name	per	Grade2	5
0	101	ram	45	a+	a+
1	102	kapil	67	b+	b+
2	103	riya	89	b+	b+
3	104	megha	34	a++	a++
4	105	saket	56	c	c
5	106	aman	78	d	d
6	107	pranay	99	a++	a++

In [ ]:
```
1 df1
```