

```
import pandas as pd
import numpy as np
data=np.array(['a','b','c','d'])
s=pd.Series(data)
print(s)
```

```
0    a
1    b
2    c
3    d
dtype: object
```

```
data=np.array(['a','b','c','d'])
s=pd.Series(data,index=[100,101,102,103])
print(s)
```

```
100    a
101    b
102    c
103    d
dtype: object
```

+ Code

+ Text

```
data={'a':0.,'b':1.,'c':2.}
s=pd.Series(data)
print(s)
```

```
a    0.0
b    1.0
c    2.0
dtype: float64
```

```
data={'a':0.,'b':1.,'c':2.}
s=pd.Series(data,index=['b','c','d','a'])
print(s)
```

```
b    1.0
c    2.0
d    NaN
a    0.0
dtype: float64
```

```
s=pd.Series(5, index=[0,1,2,3])
print(s)
```

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

```
s=pd.Series([1,2,3,4,5],index=['a','b','c','d','e'])
print(s[0])
```

```
1
```

```
print(s[:3])
```

```
a    1
b    2
c    3
dtype: int64
```

```
print(s['a'])
```

```
1
```

```
print(s[['a','c','d']])
```

```
a    1
c    3
d    4
dtype: int64
```

```
import pandas as pd
df=pd.DataFrame()
print(df)
```

```
Empty DataFrame
Columns: []
Index: []
```

```
date=[1,2,3,4,5]
df=pd.DataFrame(data)
print(df)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-4-2fc407104040> in <cell line: 2>()
      1 date=[1,2,3,4,5]
----> 2 df=pd.DataFrame(data)
      3 print(df)

NameError: name 'data' is not defined
```

```
data=[['Alex',10],['Bob',12],['Clarke',13]]
df=pd.DataFrame(data,columns=['Name','Age'],dtype=float)
print(df)
```

```
      Name  Age
0   Alex  10.0
1    Bob  12.0
2  Clarke  13.0
```

```
<ipython-input-20-3dca126ae7af>:2: FutureWarning: Could not cast to float64, falling back to object. This behavior is deprecated
df=pd.DataFrame(data,columns=['Name','Age'],dtype=float)
```

```
# creating a dataframe
data=[{'a':1,'b':2},{'a':5,'b':10,'c':20}]
df=pd.DataFrame(data)
print(df)
```

```
      a  b  c
0   1  2 NaN
1   5 10 20.0
```

```
#list of dictionaries and row indices
data=[{'a':1,'b':2},{'a':5,'b':10,'c':20}]
df=pd.DataFrame(data,index=['first','second'])
print(df)
```

```
      a  b  c
first  1  2 NaN
second 5 10 20.0
```

```
data=[{'a':1,'b':2},{'a':5,'b':10,'c':20}]
df1=pd.DataFrame(data,index=['first','second'], columns=['a','b'])
df2=pd.DataFrame(data,index=['first','second'], columns=['a','b1'])
print(df1)
print(df2)
```

```
      a  b
first  1  2
second 5 10
      a  b1
first  1 NaN
second 5 NaN
```

```
#creating a data frame from the list of series
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)
print(df)
```

```
      one  two
a   1.0    1
b   2.0    2
c   3.0    3
d   NaN    4
```

```
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)
print(df['one'])
```

```
a    1.0
b    2.0
c    3.0
d    NaN
Name: one, dtype: float64
```

```
#column addition
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)
print("Adding a new column by passing the Series:")
df['three']=pd.Series([10,20,30],index=['a','b','c'])
print(df)
print("Adding new column using the existing columns in DataFrame:")
df['four']=df['one']+df['three']
print(df)
```

```
Adding a new column by passing the Series:
   one  two  three
a  1.0   1  10.0
b  2.0   2  20.0
c  3.0   3  30.0
d  NaN   4   NaN
Adding new column using the existing columns in DataFrame:
   one  two  three  four
a  1.0   1  10.0  11.0
b  2.0   2  20.0  22.0
c  3.0   3  30.0  33.0
d  NaN   4   NaN   NaN
```

```
d={'one':pd.Series([1,2,3],index=['a','b','c']),
  'two':pd.Series([1,2,3,4],index=['a','b','c','d']),
  'three':pd.Series([10,20,30],index=['a','b','c'])}
df=pd.DataFrame(d)
print("Our Data Frame is:")
print(df)
print("Deleting the first column using DEL Function:")
del(df['one'])
print(df)
print("deleting another column using pop function")
df.pop('two')
print(df)
```

```
Our Data Frame is:
   one  two  three
a  1.0   1  10.0
b  2.0   2  20.0
c  3.0   3  30.0
d  NaN   4   NaN
Deleting the first column using DEL Function:
   two  three
a    1  10.0
b    2  20.0
c    3  30.0
d    4   NaN
deleting another column using pop function
   three
a  10.0
b  20.0
c  30.0
d   NaN
```

```
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)
print(df.loc['b']) #to select the row using location
```

```
one    2.0
two    2.0
```

Name: b, dtype: float64

```
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)
print(df.iloc[2])
```

```
one    3.0
two    3.0
Name: c, dtype: float64
```

```
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)
print(df[2:4])
```

```
   one  two
c  3.0    3
d  NaN    4
```

```
#ADDITION OF ROWS
df=pd.DataFrame([[1,2],[3,4]],columns=['a','b'])
df2=pd.DataFrame([[5,6],[7,8]],columns=['a','b'])
df=df.append(df2)
print(df)
```

```
   a  b
0  1  2
1  3  4
0  5  6
1  7  8
```

<ipython-input-26-ea945746da26>:4: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

```
df=df.append(df2)
```

```
#deletion of row
df=pd.DataFrame([[1,2],[3,4]],columns=['a','b'])
df2=pd.DataFrame([[5,6],[7,8]],columns=['a','b'])
df.drop(0)
print(df)
```

```
   a  b
0  1  2
1  3  4
```

```
# loading the data
df=pd.read_csv("/content/lab.csv")
df.head()
```

	Empid	Ename	Salary	DOJ
0	1001	Ganesh	1000.00	10-10-00
1	1002	Anil	23000.50	3/20/2002
2	1003	Gaurav	NaN	05-11-03
3	1004	Hema chandra	16500.50	02-11-04
4	1005	Laxmi Prasanna	12000.75	03-02-02

```
df.tail()
```

	Empid	Ename	Salary	DOJ
1	1002	Anil	23000.50	3/20/2002
2	1003	Gaurav	NaN	05-11-03
3	1004	Hema chandra	16500.50	02-11-04
4	1005	Laxmi Prasanna	12000.75	03-02-02
5	1006	Anant	9999.99	05-05-03

df.shape

(6, 4)

df.info

```
<bound method DataFrame.info of
0  1001      Ganesh  1000.00  10-10-00
1  1002      Anil   23000.50  3/20/2002
2  1003      Gaurav    NaN    05-11-03
3  1004  Hema chandra  16500.50  02-11-04
4  1005  Laxmi Prasanna 12000.75  03-02-02
5  1006      Anant   9999.99  05-05-03>
```

df.describe()

	Empid	Salary
count	6.000000	5.000000
mean	1003.500000	12500.348000
std	1.870829	8139.622234
min	1001.000000	1000.000000
25%	1002.250000	9999.990000
50%	1003.500000	12000.750000
75%	1004.750000	16500.500000
max	1006.000000	23000.500000

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.