```
Start coding or generate with AI.
```

import the pandas library and aliasing as pd

```
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
```

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

```
1
```

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

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

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

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

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

```
   0
0  1
1  2
2  3
3  4
4  5
```

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

```
                         Name  Age
0                        Alex   10
1                         Bob   12
2  Shamithaaaaaaaaaaaaaaaaa   13
```

```
import pandas as pd
```

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

```
                         Name   Age
0                        Alex  10.0
1                         Bob  12.0
2  Shamithaaaaaaaaaaaaaaaaa  13.0
<ipython-input-4-7ceeb70c6172>:2: FutureWarning: Could not cast to float64, falling back to object. This behavior is deprecated. In
  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}]
#with two column indices,values same as dictionary keys
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
```

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

```
Adding a new coulumn by passing as 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 a new coulumn 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
```

```
#Using the previous DataFrame , we will delete a column , using del function
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=[
df=pd.DataFrame(d)
print("Our dataframe is: ")
print(df)
```

```
Our dataframe 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
```

```
#using del function
print("Deleting the first column using the DEL function: ")
del(df['one'])
print(df)
```

```
Deleting the first column using the DEL function:
   two  three
a    1   10.0
b    2   20.0
c    3   30.0
d    4    NaN
```

```
#using pop function
print("Deleting another column using POP function: ")
df.pop('two')
print(df)
```

```
Deleting another column using POP function:
   three
a   10.0
b   20.0
c   30.0
d    NaN
```

Rows can be selectd by passing row label to a loc function

```
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'])
```

```
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
```

Multiple rows can be selected using ':' operator

```
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-21-d31ad2479008>:3: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future
  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=df.append(df2)
#Drop row with label 0
df=df.drop(0)
print(df)
```

```
   a  b
1  3  4
1  7  8
<ipython-input-23-6cdd276a5b23>:3: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future
  df=df.append(df2)
```

```
d=pd.read_csv("/content/advertising (1).csv")
```

```
d.head()
```

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9

```
d.tail()
```

	TV	Radio	Newspaper	Sales
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	14.0
197	177.0	9.3	6.4	14.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	18.4

```
d.shape
```

```
(200, 4)
```

```
d.describe()
```

	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000	36.525000	45.100000	19.050000
max	296.400000	49.600000	114.000000	27.000000

```
d.info
```

```
<bound method DataFrame.info of         TV  Radio  Newspaper  Sales
0    230.1   37.8       69.2   22.1
1     44.5   39.3       45.1   10.4
2     17.2   45.9       69.3   12.0
3    151.5   41.3       58.5   16.5
4    180.8   10.8       58.4   17.9
..     ...    ...        ...    ...
195   38.2    3.7       13.8    7.6
196   94.2    4.9        8.1   14.0
197  177.0    9.3        6.4   14.8
198  283.6   42.0       66.2   25.5
199  232.1    8.6        8.7   18.4

[200 rows x 4 columns]>
```