```
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
```

```
s = pd.Series(data, index=[100,101,102,103])
print(s)
```

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

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

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

```
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], index=[0,1,2,3])
print(s[0])
```

```
1
```

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

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

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

```
5
```

```
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],['Clarke',13]]
df = pd.DataFrame(data, columns=['Name','Age'])
print(df)
```

```
      Name  Age
0    Alex   10
1     Bob   12
2  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-24-f98f1f6b732a>:1: FutureWarning: Could not cast to float64, falling back to object. This behavior is deprecated. In a future version, when a dtype is passed to 'DataFrame', either all column
  df = pd.DataFrame(data, columns=['Name','Age'],dtype=float)
```

```
data = {'Name':['Tom','Jacky','Steve','Ricky'], 'Age':[28,34,29,42]}
df=pd.DataFrame(data)
print(df)
```

```
    Name  Age
0    Tom   28
1  Jacky   34
2  Steve   29
3  Ricky   42
```

```
data = {'Name':['Tom','Jacky','Steve','Ricky'], 'Age':[28,34,29,42]}
df=pd.DataFrame(data, index=['rank1','rank2','rank3','rank4'])
print(df)
```

```
        Name  Age
rank1    Tom   28
rank2  Jacky   34
rank3  Steve   29
rank4  Ricky   42
```

```
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
```

```
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
```

```
import pandas as pd
```

```
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 3rd col: ")
df['three']=pd.Series([10,20,30], index=['a','b','c'])
print(df)
```

```
adding 3rd col:
   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
```

```
df['four'] = df['one']+df['three']
print(df)
```

```
   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
```

```
print(df)
print("Deleting one col: ")
del df['one'] #pop
print(df)
```

```
   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
Deleting one col:
   two  three  four
a    1   10.0  11.0
b    2   20.0  22.0
c    3   30.0  33.0
d    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']) }
df=pd.DataFrame(d)
print(df.loc['b'])
```

```
one    2.0
two    2.0
Name: b, dtype: float64
```

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

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

```
df=pd.DataFrame(d)
print(df[2:4])
```

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

```
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-17-f7aead6d93ee>:3: 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)
```

```
df = df.drop(0)
print(df)
```

```
   a  b
1  3  4
1  7  8
```

```
sales=[100,130,111,92,35]
cust_acc = ['B100','3101','X102','P103','R104']
city = ['A','B','C','D','E']
```