

```
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
```

```
import pandas as pd
import numpy as np
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
```

```
import pandas as pd
import numpy as np
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
```

```
import pandas as pd
import numpy as np
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]) #Retrieve the first Element
```

```
1
```

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

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

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

```
1
```

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

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

```
a=[11.1,12.2,13.3,14.4,15.5]
s=pd.Series(a,index=[1,2,3,4,5])
print(s)
```

```
1    11.1
2    12.2
3    13.3
4    14.4
5    15.5
dtype: float64
```

Empty Data Frame

```
import pandas as pd
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 Frame using arrays

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

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

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

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

```
<ipython-input-16-ce59582bef39>:2: FutureWarning: Could not cast to float64, falling back to object. This behavior is deprecated. In a f
df=pd.DataFrame(data,columns=['Name','Age'],dtype=float)
```

Create using list of dict

```
import pandas as pd
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
```

```
import pandas as pd
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 2 column indices,values same as dictionary keys
df1=pd.DataFrame(data,index=['first','second'],columns=['a','b'])
#With 2 column indices with one index other name
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 Dataframe from 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)
#Adding a new column to an existing Dataframe object with label by passing on as Series
print("Adding a new column by Passing as Series: ")
df['three']=pd.Series([10,20,30],index=['a','b','c'])
print(df)
print("Adding a new column using exisiting columns in Dataframe: ")
df['four']=df['one']+df['three']
print(df)
```

Adding a new column 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 column using exisiting 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
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 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 DEL function: ")
del(df['one'])
print(df)
#Using pop function
print("Deleting the first column using POP function: ")
df.pop('two')
print(df)
```

```
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 the first 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'])
```

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

Slice Rows

```
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-37-d31ad2479008>:3: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future vers
df=df.append(df2)
```

Deletion of Rows

```
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)
# Using the drop function
df.drop('b',axis=0,inplace=True)
print(df)
```

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

Loading the data

```
df=pd.read_csv('fruit_dataset_SET A.csv')
```

```
df.head()
```

	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192	8.4	7.3	0.55
1	1	apple	granny_smith	180	8.0	6.8	0.59
2	1	apple	granny_smith	176	7.4	7.2	0.60
3	2	mandarin	mandarin	86	6.2	4.7	0.80
4	2	mandarin	mandarin	84	6.0	4.6	0.79

```
df.tail()
```

	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
54	4	lemon	unknown	116	6.1	8.5	0.71
55	4	lemon	unknown	116	6.3	7.7	0.72
56	4	lemon	unknown	116	5.9	8.1	0.73
57	4	lemon	unknown	152	6.5	8.5	0.72
58	4	lemon	unknown	118	6.1	8.1	0.70

```
df.tail()
```

	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
54	4	lemon	unknown	116	6.1	8.5	0.71
55	4	lemon	unknown	116	6.3	7.7	0.72
56	4	lemon	unknown	116	5.9	8.1	0.73
57	4	lemon	unknown	152	6.5	8.5	0.72
58	4	lemon	unknown	118	6.1	8.1	0.70

```
df.describe()
```

	fruit_label	mass	width	height	color_score
count	59.000000	59.000000	59.000000	59.000000	59.000000
mean	2.542373	163.118644	7.105085	7.693220	0.762881
std	1.208048	55.018832	0.816938	1.361017	0.076857
min	1.000000	76.000000	5.800000	4.000000	0.550000
25%	1.000000	140.000000	6.600000	7.200000	0.720000
50%	3.000000	158.000000	7.200000	7.600000	0.750000
75%	4.000000	177.000000	7.500000	8.200000	0.810000
max	4.000000	362.000000	9.600000	10.500000	0.930000

df.info

```

<bound method DataFrame.info of
0      1      apple      granny_smith      192      8.4      7.3      0.55
1      1      apple      granny_smith      180      8.0      6.8      0.59
2      1      apple      granny_smith      176      7.4      7.2      0.60
3      2      mandarin      mandarin      86      6.2      4.7      0.80
4      2      mandarin      mandarin      84      6.0      4.6      0.79
5      2      mandarin      mandarin      80      5.8      4.3      0.77
6      2      mandarin      mandarin      80      5.9      4.3      0.81
7      2      mandarin      mandarin      76      5.8      4.0      0.81
8      1      apple      braeburn      178      7.1      7.8      0.92
9      1      apple      braeburn      172      7.4      7.0      0.89
10     1      apple      braeburn      166      6.9      7.3      0.93
11     1      apple      braeburn      172      7.1      7.6      0.92
12     1      apple      braeburn      154      7.0      7.1      0.88
13     1      apple      golden_delicious      164      7.3      7.7      0.70
14     1      apple      golden_delicious      152      7.6      7.3      0.69
15     1      apple      golden_delicious      156      7.7      7.1      0.69
16     1      apple      golden_delicious      156      7.6      7.5      0.67
17     1      apple      golden_delicious      168      7.5      7.6      0.73
18     1      apple      cripps_pink      162      7.5      7.1      0.83
19     1      apple      cripps_pink      162      7.4      7.2      0.85
20     1      apple      cripps_pink      160      7.5      7.5      0.86
21     1      apple      cripps_pink      156      7.4      7.4      0.84
22     1      apple      cripps_pink      140      7.3      7.1      0.87
23     1      apple      cripps_pink      170      7.6      7.9      0.88
24     3      orange      spanish_jumbo      342      9.0      9.4      0.75
25     3      orange      spanish_jumbo      356      9.2      9.2      0.75
26     3      orange      spanish_jumbo      362      9.6      9.2      0.74
27     3      orange      selected_seconds      204      7.5      9.2      0.77
28     3      orange      selected_seconds      140      6.7      7.1      0.72
29     3      orange      selected_seconds      160      7.0      7.4      0.81
30     3      orange      selected_seconds      158      7.1      7.5      0.79
31     3      orange      selected_seconds      210      7.8      8.0      0.82
32     3      orange      selected_seconds      164      7.2      7.0      0.80
33     3      orange      turkey_navel      190      7.5      8.1      0.74
34     3      orange      turkey_navel      142      7.6      7.8      0.75
35     3      orange      turkey_navel      150      7.1      7.9      0.75
36     3      orange      turkey_navel      160      7.1      7.6      0.76
37     3      orange      turkey_navel      154      7.3      7.3      0.79
38     3      orange      turkey_navel      158      7.2      7.8      0.77
39     3      orange      turkey_navel      144      6.8      7.4      0.75
40     3      orange      turkey_navel      154      7.1      7.5      0.78
41     3      orange      turkey_navel      180      7.6      8.2      0.79
42     3      orange      turkey_navel      154      7.2      7.2      0.82
43     4      lemon      spanish_belsan      194      7.2      10.3      0.70
44     4      lemon      spanish_belsan      200      7.3      10.5      0.72
45     4      lemon      spanish_belsan      186      7.2      9.2      0.72
46     4      lemon      spanish_belsan      216      7.3      10.2      0.71
47     4      lemon      spanish_belsan      196      7.3      9.7      0.72
48     4      lemon      spanish_belsan      174      7.3      10.1      0.72
49     4      lemon      unknown      132      5.8      8.7      0.73
50     4      lemon      unknown      130      6.0      8.2      0.71
51     4      lemon      unknown      116      6.0      7.5      0.72
52     4      lemon      unknown      118      5.9      8.0      0.72
53     4      lemon      unknown      120      6.0      8.4      0.74
54     4      lemon      unknown      116      6.1      8.5      0.71
55     4      lemon      unknown      116      6.3      7.7      0.72
56     4      lemon      unknown      116      5.9      8.1      0.73

```

df.shape

(59, 7)

