

```
import numpy as np
import pandas as pd
list=[[1,'Smith',50000],[2,'Jones',60000]]
```

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

```
↗
```

	0	1	2
0	1	Smith	50000
1	2	Jones	60000

```
df.columns=['Empd','Name','Salary']
df
```

```
↗
```

	Empd	Name	Salary
0	1	Smith	50000
1	2	Jones	60000

```
df.info()
```

```
↗ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 2 entries, 0 to 1
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0   Empd    2 non-null        int64
1   Name    2 non-null        object
2   Salary  2 non-null        int64
dtypes: int64(2), object(1)
memory usage: 176.0+ bytes
```

```
df=pd.read_csv("/content/50_Startups.csv")
```

```
df.info()
```

```
↗ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  -
0   R&D Spend              50 non-null    float64
1   Administration         50 non-null    float64
2   Marketing Spend        50 non-null    float64
3   State                  50 non-null    object
4   Profit                 50 non-null    float64
dtypes: float64(4), object(1)
memory usage: 2.1+ KB
```

```
df.head()
```

```
↗
```

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91301.77	366168.42	Florida	166187.04

```
df.tail()
```

```
↗
```

	R&D Spend	Administration	Marketing Spend	State	Profit
45	1000.23	124153.04	1903.93	New York	64926.08
46	1315.46	115816.21	297114.46	Florida	49490.75
47	0.00	135426.92	0.00	California	42559.73
48	542.05	51743.15	0.00	New York	35673.41
49	0.00	116083.80	45173.06	California	14681.40

```
import numpy as np
import pandas as pd
df=pd.read_csv("/content/employee.csv")
```

```
df.head()
```

	emp id	name	salary
0	1	SREE VARSSINI K S	5000
1	2	SREEMATHI B	6000
2	3	SREYA G	7000
3	4	SREYASKARI MULLAPUDI	5000
4	5	SRI AKASH U G	8000

```
df.tail()
```

	emp id	name	salary
2	3	SREYA G	7000
3	4	SREYASKARI MULLAPUDI	5000
4	5	SRI AKASH U G	8000
5	6	SRI HARSHAVARDHANAN R	3000
6	7	SRI HARSHAVARDHANAN R	6000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0   emp id  7 non-null       int64
1   name    7 non-null       object
2   salary  7 non-null       int64
dtypes: int64(2), object(1)
memory usage: 296.0+ bytes
```

```
df.salary
```

	salary
0	5000
1	6000
2	7000
3	5000
4	8000
5	3000
6	6000

```
type(df.salary)
```

```
pandas.core.series.Series
def __init__(data=None, index=None, dtype: Dtype | None=None, name=None, copy: bool | None=None,
fastpath: bool=False) -> None

One-dimensional ndarray with axis labels (including time series).

Labels need not be unique but must be a hashable type. The object
supports both integer- and label-based indexing and provides a host of
methods for performing operations involving the index. Statistical
```

```
df.salary.mean()
```

```
5714.285714285715
```

```
df.salary.median()
```

```
6000.0
```

```
df.salary.mode()
```

```
salary
0    5000
1    6000
```

```
df.salary.var()
```

```
2571428.5714285714
```

```
df.salary.std()
```

```
1603.5674514745463
```

```
df.describe()
```

```
emp id    salary
count  7.000000    7.000000
mean    4.000000   5714.285714
std     2.160247   1603.567451
min     1.000000    3000.000000
25%     2.500000    5000.000000
50%     4.000000    6000.000000
75%     5.500000    6500.000000
max     7.000000    8000.000000
```

```
df.describe(include='all')
```

```
emp id    name    salary
count  7.000000      7    7.000000
unique    NaN      6      NaN
top    NaN  SRI HARSHAVARDHANAN R      NaN
freq    NaN      2      NaN
mean    4.000000      NaN  5714.285714
std     2.160247      NaN  1603.567451
min     1.000000      NaN    3000.000000
25%     2.500000      NaN    5000.000000
50%     4.000000      NaN    6000.000000
75%     5.500000      NaN    6500.000000
max     7.000000      NaN    8000.000000
```

```
empCol=df.columns
```

```
empCol
```

```
Index(['emp id', 'name ', 'salary'], dtype='object')
```

```
emparray=df.values
```

```
emparray
```

```
array([[1, 'SREE VARSSINI K S', 5000],
       [2, 'SREEMATHI B', 6000],
       [3, 'SREYA G', 7000],
       [4, 'SREYASKARI MULLAPUDI', 5000],
       [5, 'SRI AKASH U G', 8000],
```

```
[6, 'SRI HARSHAVARDHANAN R', 3000],  
[7, 'SRI HARSHAVARDHANAN R', 6000]], dtype=object)
```

```
employee_DF=pd.DataFrame(emparray,columns=empCol)
```

```
employee_DF
```



	emp id	name	salary
0	1	SREE VARSSINI K S	5000
1	2	SREEMATHI B	6000
2	3	SREYA G	7000
3	4	SREYASKARI MULLAPUDI	5000
4	5	SRI AKASH U G	8000
5	6	SRI HARSHAVARDHANAN R	3000
6	7	SRI HARSHAVARDHANAN R	6000

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