

```
In [1]: import pandas as pd
```

```
In [2]: dataframe=pd.read_csv("F:\ADC LAB\employee.csv")
```

```
In [3]: dataframe
```

```
Out[3]:
```

	Name	Age	salary
0	developer	27	70000
1	developer	29	90000
2	manager	29	61000
3	manager	28	60000
4	tester	42	150000
5	tester	39	155000
6	tester	41	160000
7	developer	38	162000
8	manager	36	154000
9	manager	35	130000
10	developer	37	137000
11	tester	26	45000
12	manager	27	48000
13	manager	28	51000
14	developer	29	49500
15	developer	32	53000
16	manager	40	65000
17	developer	41	63000
18	developer	43	64000
19	developer	39	80000
20	developer	41	82000
21	developer	39	58000

```
In [4]: df=pd.read_excel("F:\ADC LAB\emp.xlsx")
```

In [5]: `df`

Out[5]:

	Name	Age	salary
0	developer	27	70000
1	developer	29	90000
2	manager	29	61000
3	manager	28	60000
4	tester	42	150000
5	tester	39	155000
6	tester	41	160000
7	developer	38	162000
8	manager	36	154000
9	manager	35	130000
10	developer	37	137000
11	tester	26	45000
12	manager	27	48000
13	manager	28	51000
14	developer	29	49500
15	developer	32	53000
16	manager	40	65000
17	developer	41	63000
18	developer	43	64000
19	developer	39	80000
20	developer	41	82000
21	developer	39	58000

In [6]: `dict={'name':['A','B','C'],'Roll No':[4,5,6],'Marks':[18,19,17]}`

In [7]: `dict`

Out[7]: {'name': ['A', 'B', 'C'], 'Roll No': [4, 5, 6], 'Marks': [18, 19, 17]}

In [8]: `dataframe=pd.DataFrame(dict)`

In [9]: `dataframe`

Out[9]:

	name	Roll No	Marks
0	A	4	18
1	B	5	19
2	C	6	17

```
In [10]: tupleData=[('A',4,18),('B',6,17),('D',7,20)]
```

```
In [11]: DF=pd.DataFrame(tupleData,columns=["name","Roll No","Marks"])
```

```
In [12]: DF
```

```
Out[12]:
```

	name	Roll No	Marks
0	A	4	18
1	B	6	17
2	D	7	20

```
In [13]: dataframe.head()
```

```
Out[13]:
```

	name	Roll No	Marks
0	A	4	18
1	B	5	19
2	C	6	17

```
In [14]: df=pd.read_excel("F:\ADC LAB\emp.xlsx")
```

In [15]:

```
df
```

Out[15]:

	Name	Age	salary
0	developer	27	70000
1	developer	29	90000
2	manager	29	61000
3	manager	28	60000
4	tester	42	150000
5	tester	39	155000
6	tester	41	160000
7	developer	38	162000
8	manager	36	154000
9	manager	35	130000
10	developer	37	137000
11	tester	26	45000
12	manager	27	48000
13	manager	28	51000
14	developer	29	49500
15	developer	32	53000
16	manager	40	65000
17	developer	41	63000
18	developer	43	64000
19	developer	39	80000
20	developer	41	82000
21	developer	39	58000

In [16]:

```
df.head()
```

Out[16]:

	Name	Age	salary
0	developer	27	70000
1	developer	29	90000
2	manager	29	61000
3	manager	28	60000
4	tester	42	150000

In [17]: `df.tail()`

Out[17]:

	Name	Age	salary
17	developer	41	63000
18	developer	43	64000
19	developer	39	80000
20	developer	41	82000
21	developer	39	58000

In [18]: `df.head(7)`

Out[18]:

	Name	Age	salary
0	developer	27	70000
1	developer	29	90000
2	manager	29	61000
3	manager	28	60000
4	tester	42	150000
5	tester	39	155000
6	tester	41	160000

In [19]: `df.tail(10)`

Out[19]:

	Name	Age	salary
12	manager	27	48000
13	manager	28	51000
14	developer	29	49500
15	developer	32	53000
16	manager	40	65000
17	developer	41	63000
18	developer	43	64000
19	developer	39	80000
20	developer	41	82000
21	developer	39	58000

In [27]: `import numpy as np
df=pd.DataFrame(np.arange(0,20).reshape(5,4),index=["Row1","Row2","Row3","Row4",
columns=["col1","col2","col3","col4"])`

In [28]: `df`

Out[28]:

	col1	col2	col3	col4
Row1	0	1	2	3
Row2	4	5	6	7
Row3	8	9	10	11
Row4	12	13	14	15
Row5	16	17	18	19

In [29]: `df[['col1','col2']]`

Out[29]:

	col1	col2
Row1	0	1
Row2	4	5
Row3	8	9
Row4	12	13
Row5	16	17

In [30]: `df[0:2]`

Out[30]:

	col1	col2	col3	col4
Row1	0	1	2	3
Row2	4	5	6	7

creating DataFrame by reading csv file

In [31]: `import pandas as pd
import numpy as np`

In [32]: `df=pd.read_csv("F:\ADC LAB\employee.csv")`

In [33]: `df`

Out[33]:

	Name	Age	salary
0	developer	27	70000
1	developer	29	90000
2	manager	29	61000
3	manager	28	60000
4	tester	42	150000
5	tester	39	155000
6	tester	41	160000
7	developer	38	162000
8	manager	36	154000
9	manager	35	130000
10	developer	37	137000
11	tester	26	45000
12	manager	27	48000
13	manager	28	51000
14	developer	29	49500
15	developer	32	53000
16	manager	40	65000
17	developer	41	63000
18	developer	43	64000
19	developer	39	80000
20	developer	41	82000
21	developer	39	58000

In [34]: `df=pd.read_excel("F:\ADC LAB\employee.xls")`

In [35]: `# create dataframe with dictionary`
`dict={'name':['Mohit','Mayuri','Avanti'],'Roll No':[1,2,3],`
`'Marks':[18,17,19]}`

In [36]: `dict`

Out[36]: `{'name': ['Mohit', 'Mayuri', 'Avanti'],`
`'Roll No': [1, 2, 3],`
`'Marks': [18, 17, 19]}`

In [37]: `#create dataframe by using DataFrame() of pandas`
`dataFrame=pd.DataFrame(dict)`

```
In [38]: dataframe
```

```
Out[38]:
```

	name	Roll No	Marks
0	Mohit	1	18
1	Mayuri	2	17
2	Avanti	3	19

```
In [39]: #create dataframe with tuple
tupleData=[('Mohit',1,18),('Mayuri',2,17),('Avanti',3,19)]
```

```
In [40]: tupleData
```

```
Out[40]: [('Mohit', 1, 18), ('Mayuri', 2, 17), ('Avanti', 3, 19)]
```

```
In [41]: df=pd.DataFrame(tupleData.columns=["name","Roll No","Marks"])
```

```
In [42]: df
```

```
Out[42]:
```

	name	Roll No	Marks
0	Mohit	1	18
1	Mayuri	2	17
2	Avanti	3	19

methods of DataFrame

```
In [43]: #head() is used to read and display head section of the dataframe
# by default it will top 5 rows
df=pd.read_csv("F:\ADC LAB\employee.csv")
```

```
In [44]: df.head()
```

```
Out[44]:
```

	Name	Age	salary
0	developer	27	70000
1	developer	29	90000
2	manager	29	61000
3	manager	28	60000
4	tester	42	150000


```
In [45]: #tail() is used to display bottom rows of dataframe
df.tail()
```

```
Out[45]:
```

	Name	Age	salary
17	developer	41	63000
18	developer	43	64000
19	developer	39	80000
20	developer	41	82000
21	developer	39	58000

```
In [ ]: df=pd.DataFrame(np.arange(0,20).reshape(5,4),index=["Row1","Row2","Row3","Row4",
                    columns=['col1','col2','col3','col4'])
```

```
In [48]: df
```

```
Out[48]:
```

	col1	col2	col3	col4
Row1	0	1	2	3
Row2	4	5	6	7
Row3	8	9	10	11
Row4	12	13	14	15
Row5	16	17	18	19

indexing : selecting particular set of dta such as rows, columns and individual cells

```
In [49]: # Display col1 and col2 elements of the above dataframe
# we can display the elements with the help of column names
df[['col1','col2']]
```

```
Out[49]:
```

	col1	col2
Row1	0	1
Row2	4	5
Row3	8	9
Row4	12	13
Row5	16	17

```
In [50]: #display row2 and row3 elements
df[1:3]
```

```
Out[50]:
```

	col1	col2	col3	col4
Row2	4	5	6	7
Row3	8	9	10	11

```
In [51]: #use loc to display row data
#display 3rd row elements
df.loc["Row2"]
```

```
Out[51]: col1    4
         col2    5
         col3    6
         col4    7
         Name: Row2, dtype: int32
```

```
In [52]: #display the datatype of second row elements
type(df.loc["Row1"])
```

```
Out[52]: pandas.core.series.Series
```

```
In [53]: # display intersect row and column data
# for this we will use iloc method
df.iloc[2:4,0:2]
```

```
Out[53]:
```

	col1	col2
Row3	8	9
Row4	12	13

DataFrame basic operations

```
In [54]: # not a number - nan()
df1=pd.DataFrame(data=[[1,np.nan,2],[4,5,6]],index=["row1","row2"],
                  columns=["col1","col2","col3"])
```

```
In [55]: df1
```

```
Out[55]:
```

	col1	col2	col3
row1	1	NaN	2
row2	4	5.0	6

```
In [56]: df1.isnull() # isnull() returns true for NaN cell and false for other cells
```

```
Out[56]:
```

	col1	col2	col3
row1	False	True	False
row2	False	False	False

```
In [57]: # replace NaN cell with a specified value
df1.replace(to_replace='NaN',value=4)#it will not replace
```

```
Out[57]:
```

	col1	col2	col3
row1	1	NaN	2
row2	4	5.0	6

```
In [58]: df1.replace(to_replace=np.nan,value=4)
```

```
Out[58]:
```

	col1	col2	col3
row1	1	4.0	2
row2	4	5.0	6

```
In [59]: #count number of NaN cells  
df1.isnull().sum()#it will count the NaN cells column wise
```

```
Out[59]: col1    0  
col2    1  
col3    0  
dtype: int64
```

task - create a student.csv file. the file contains Roll No, Name, phone No as columns. keep Some fields empty

1. create a dataframe by reading student.csv file
2. find how many NaN cells are there
3. Replace the NaN cells with a default number