

Import Numpy and Pandas

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

from pandas import Series, DataFrame
```

Create some dataset and make some missing values in there

```
In [2]: missing = np.nan
```

```
In [3]: series_obj = Series (np.arange(5))
series_obj
```

```
Out[3]: 0    0
1    1
2    2
3    3
4    4
dtype: int32
```

```
In [4]: series_obj = Series(np.arange(5), index = ['pertama', missing, 'ketiga', missing,
series_obj
```

```
Out[4]: pertama    0
NaN              1
ketiga           2
NaN              3
keenan           4
dtype: int32
```

```
In [5]: series_obj.isnull()
```

```
Out[5]: pertama    False
NaN              False
ketiga           False
NaN              False
keenan           False
dtype: bool
```

```
In [13]: np.random.seed(25)
df_obj = DataFrame(np.random.rand(9).reshape(3,3))
df_obj
```

```
Out[13]:
```

	0	1	2
0	0.870124	0.582277	0.278839
1	0.185911	0.411100	0.117376
2	0.684969	0.437611	0.556229

```
In [15]: df_obj.loc[[1,2], [0,2]] = missing
df_obj
```

```
Out[15]:
```

	0	1	2
0	0.870124	0.582277	0.278839
1	NaN	0.411100	NaN
2	NaN	0.437611	NaN

Filling some missing values using 'fillna'

```
In [16]: filled_df = df_obj.fillna(0)
filled_df
```

```
Out[16]:
```

	0	1	2
0	0.870124	0.582277	0.278839
1	0.000000	0.411100	0.000000
2	0.000000	0.437611	0.000000

method ffill means first fill using previous same data. There another method, which is bfill means backfill

```
In [20]: filled_df = df_obj.fillna(method = 'ffill')
filled_df
```

```
Out[20]:
```

	0	1	2
0	0.870124	0.582277	0.278839
1	0.870124	0.411100	0.278839
2	0.870124	0.437611	0.278839

Drop missing values using 'dropna'

```
In [31]: drop_df_obj = df_obj.dropna(axis = 1)
drop_df_obj
```

Out[31]:

	1
0	0.582277
1	0.411100
2	0.437611

```
In [33]: df_obj = DataFrame(np.random.rand(36).reshape(6,6))
df_obj
```

Out[33]:

	0	1	2	3	4	5
0	0.883201	0.295432	0.512376	0.088702	0.641717	0.132421
1	0.766486	0.076742	0.331044	0.679852	0.509213	0.655146
2	0.602120	0.719055	0.415219	0.396542	0.825139	0.712552
3	0.097937	0.842154	0.440821	0.373989	0.913676	0.547778
4	0.251937	0.027474	0.206257	0.590885	0.163652	0.836928
5	0.775203	0.169041	0.766994	0.335366	0.472398	0.215064

```
In [46]: df_obj.loc[1:3,0] = missing
df_obj.loc[0:2,4] = missing
df_obj
```

Out[46]:

	0	1	2	3	4	5
0	0.883201	NaN	0.512376	0.088702	NaN	0.132421
1	NaN	NaN	0.331044	0.679852	NaN	0.655146
2	NaN	NaN	0.415219	0.396542	NaN	0.712552
3	NaN	0.842154	0.440821	0.373989	0.913676	0.547778
4	0.251937	0.027474	0.206257	0.590885	0.163652	0.836928
5	0.775203	0.169041	0.766994	0.335366	0.472398	0.215064

```
In [50]: drop = df_obj.dropna(axis = 0)
drop
```

Out[50]:

	0	1	2	3	4	5
4	0.251937	0.027474	0.206257	0.590885	0.163652	0.836928
5	0.775203	0.169041	0.766994	0.335366	0.472398	0.215064

```
In [51]: drop = df_obj.dropna(axis = 1)
drop
```

Out[51]:

	2	3	5
0	0.512376	0.088702	0.132421
1	0.331044	0.679852	0.655146
2	0.415219	0.396542	0.712552
3	0.440821	0.373989	0.547778
4	0.206257	0.590885	0.836928
5	0.766994	0.335366	0.215064