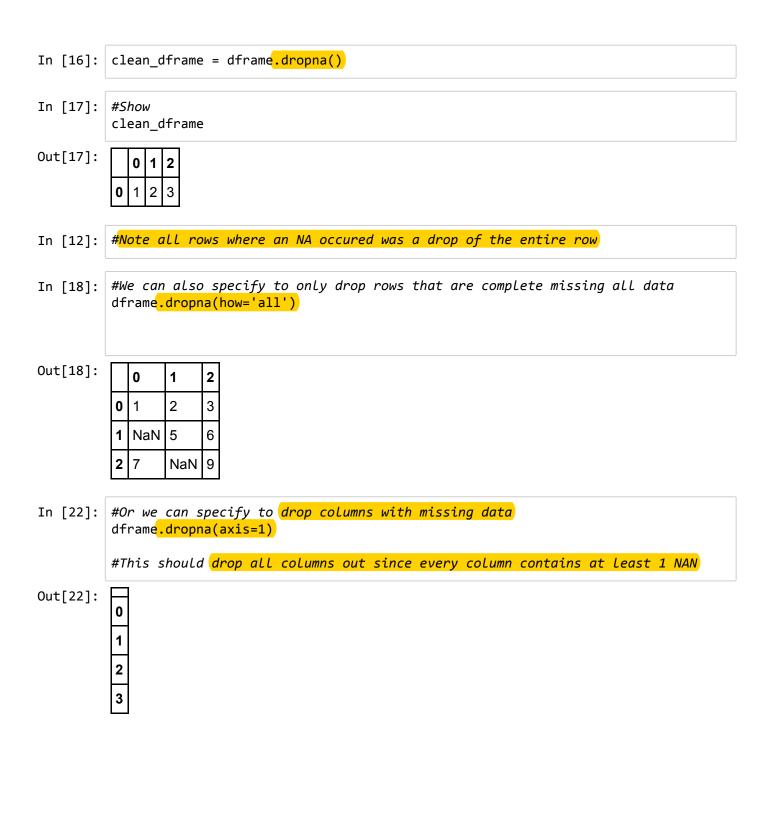
```
In [1]: import numpy as np
         from pandas import Series,DataFrame
         import pandas as pd
In [2]: #Now we'll learn how to deal with missing data, a very common task when analyzing
         data = Series(['one','two', np.nan, 'four'])
 In [3]: #Show data
         data
Out[3]: 0
               one
         1
               two
         2
               NaN
              four
         dtype: object
In [5]: #Find the missing values
         data.isnull()
 Out[5]: 0
              False
              False
         2
               True
              False
         dtype: bool
 In [7]: #We can simply drop the NAN
         data.dropna()
Out[7]: 0
               one
         1
               two
              four
         dtype: object
In [14]: # In a DataFrame we need to be a little more careful!
         dframe = DataFrame([[1,2,3],[np.nan,5,6],[7,np.nan,9],[np.nan,np.nan,np.nan]])
         #Show
In [15]:
         dframe
Out[15]:
                      2
          0 1
                 2
                      3
          1 | NaN | 5
                      6
            7
                      9
          2
                 NaN
                 NaN NaN
            NaN
```



In [26]: #We can also threshold teh missing data as well

#For example if we only want rows with at least 3 data points dframe2 = DataFrame([[1,2,3,np.nan],[2,np.nan,5,6],[np.nan,7,np.nan,9],[1,np.nan,

#Show dframe2

Out[26]:

		0	1	2	3
	0	1	2	3	NaN
	1	2	NaN	5	6
	2	NaN	7	NaN	9
	3	1	NaN	NaN	NaN

In [28]: #Droppin any rows tht dont have at Least 2 data points dframe2.dropna(thresh=2)

Out[28]:

		0	1	2	3
	0	1	2	3	NaN
	1	2	NaN	5	6

In [29]: #Dropiing rows without at Least 3 data points dframe2.dropna(thresh=3)

Out[29]:

		0	1	2	3
	0	1	2	ვ	NaN
	1	2	NaN	5	6

In [30]: #We can also fill any NAN dframe2.fillna(1)

Out[30]:

	0	1	2	3
0	1	2	3	1
1	2	1	5	6
2	1	7	1	9
3	1	1	1	1

In [33]: #Can also fill in diff values for diff columns dframe2.fillna({0:0,1:1,2:2,3:3}) Out[33]: 1 2 3 0 1 2 3 2 5 7 2 0 3 1 1 2 In [34]: #Note that we still have access to the original dframe dframe2 Out[34]: 0 2 3 0 1 3 NaN 1 2 NaN 5 6 NaN 9 2 NaN 7 3 1 NaN NaN NaN In [35]: #If we want to modify the exsisting object, use inplace dframe2.fillna(0,inplace=True) In [36]: #Now let's see the dframe dframe2 Out[36]: 0 1 2 0 1 2 3 1 2 0 5 **2** 0 7 0 0

In []: #Awesome! Next we'll learn about Index Hierarchy