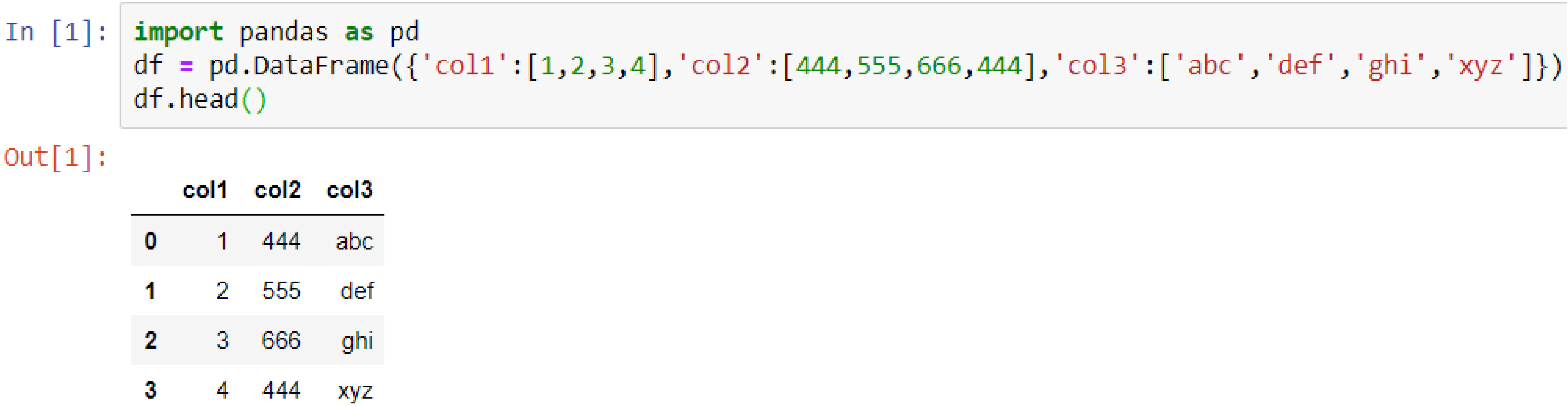
Module 02

Operations

Data Science Developer

Purwadhika
Startup and Coding School

Create DataFrame



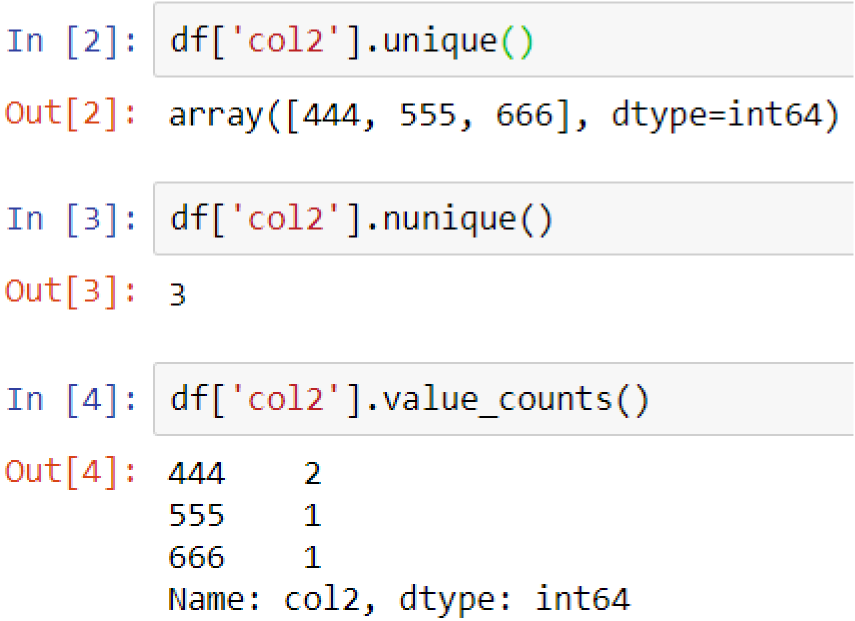
```
In [1]: import pandas as pd
        df = pd.DataFrame({'col1':[1,2,3,4],'col2':[444,555,666,444],'col3':['abc','def','ghi','xyz']})
        df.head()
```

Out[1]:

	col1	col2	col3
0	1	444	abc
1	2	555	def
2	3	666	ghi
3	4	444	xyz

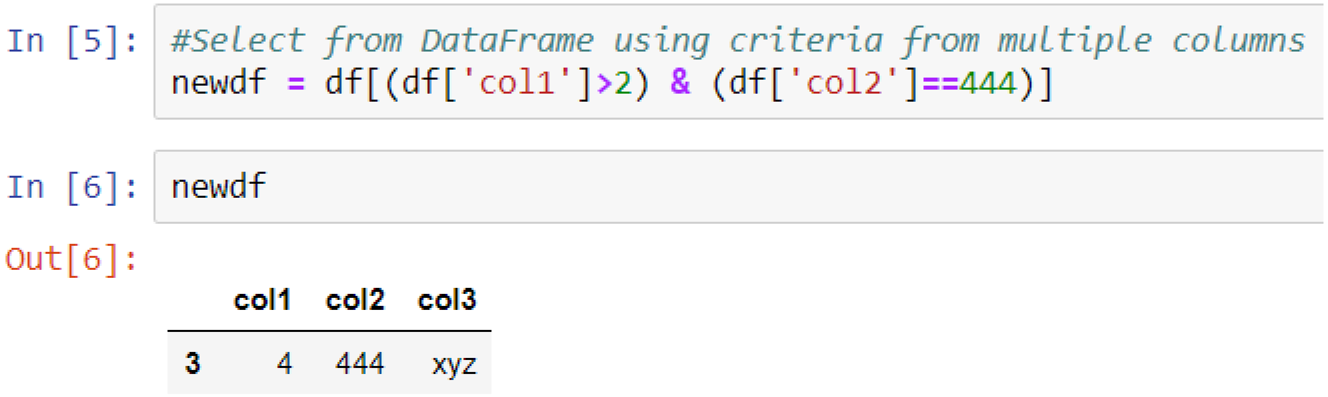
Purwadhika
Startup and Coding School

Info on Unique Values



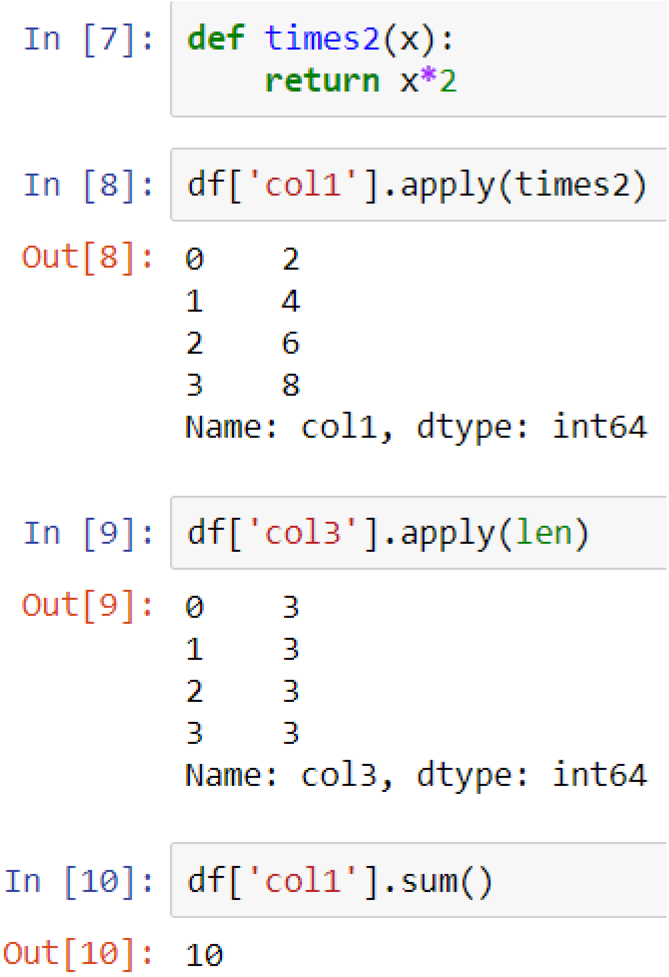
```
In [2]: df['col2'].unique()

Out[2]: array([444, 555, 666], dtype=int64)

In [3]: df['col2'].nunique()

Out[3]: 3

In [4]: df['col2'].value_counts()

Out[4]: 444    2
        555    1
        666    1
        Name: col2, dtype: int64
```

Selecting Data



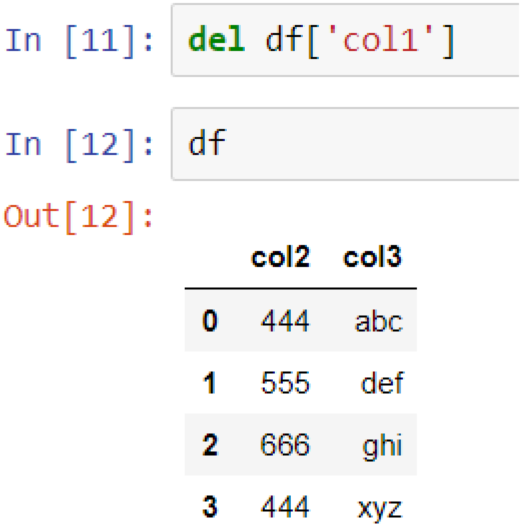
```
In [5]: #Select from DataFrame using criteria from multiple columns
        newdf = df[(df['col1']>2) & (df['col2']==444)]

In [6]: newdf

Out[6]:
           col1  col2  col3
        3     4   444   xyz
```

Applying Functions



```
In [7]: def times2(x):
            return x*2

In [8]: df['col1'].apply(times2)

Out[8]: 0    2
        1    4
        2    6
        3    8
        Name: col1, dtype: int64

In [9]: df['col3'].apply(len)

Out[9]: 0    3
        1    3
        2    3
        3    3
        Name: col3, dtype: int64

In [10]: df['col1'].sum()

Out[10]: 10
```

Permanently Removing a Column



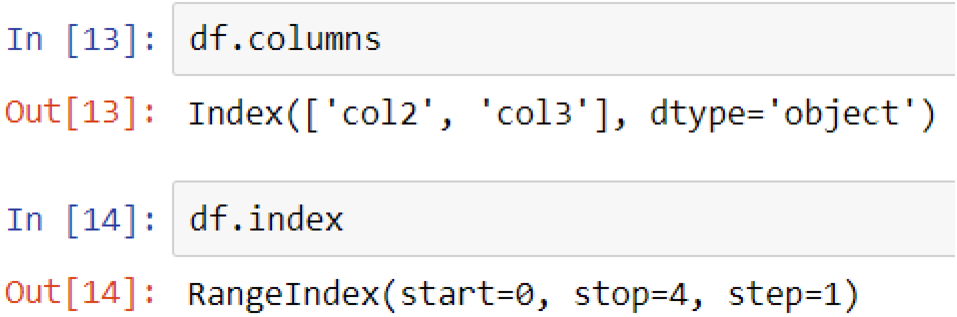
```
In [11]: del df['col1']
```

```
In [12]: df
```

Out[12]:

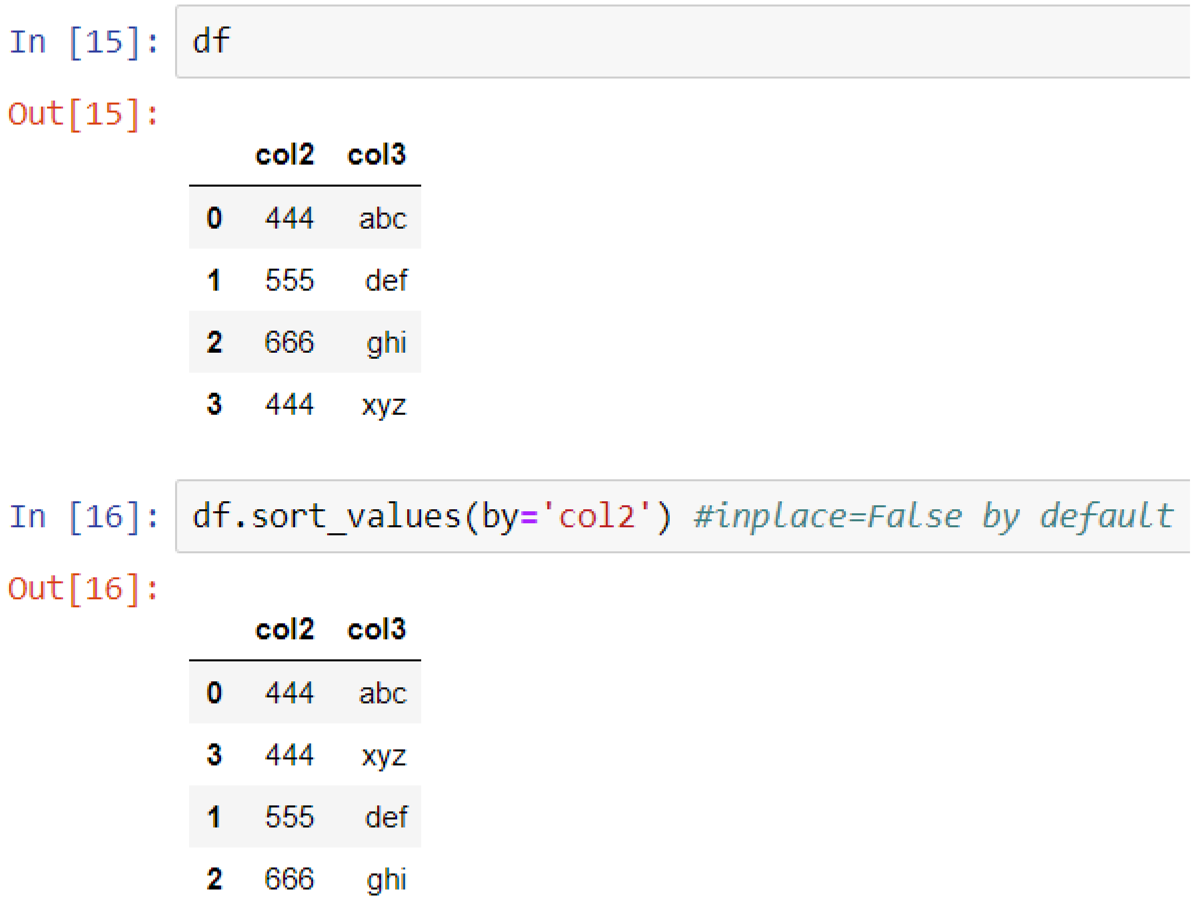
	col2	col3
0	444	abc
1	555	def
2	666	ghi
3	444	xyz

Get Column and Index Names



```
In [13]: df.columns
Out[13]: Index(['col2', 'col3'], dtype='object')

In [14]: df.index
Out[14]: RangeIndex(start=0, stop=4, step=1)
```

Sorting and Ordering a DataFrame

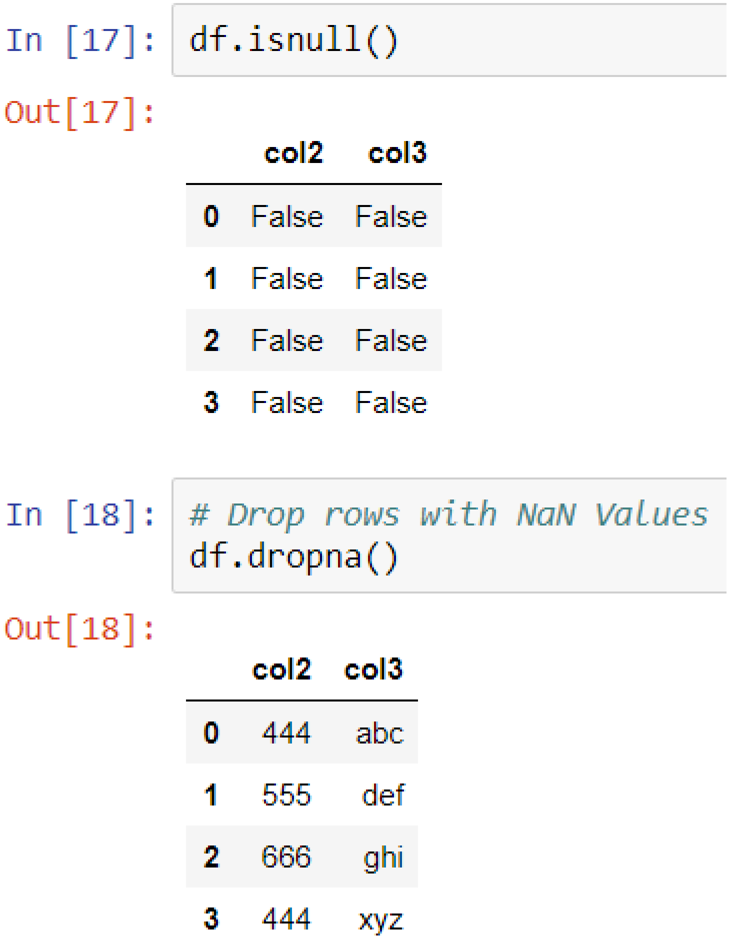


```
In [15]: df

Out[15]:
     col2  col3
  0   444   abc
  1   555   def
  2   666   ghi
  3   444   xyz
```

```
In [16]: df.sort_values(by='col2') #inplace=False by default

Out[16]:
     col2  col3
  0   444   abc
  3   444   xyz
  1   555   def
  2   666   ghi
```

Purwadhika
Startup and Coding School

Check for Null Values



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

Out[17]:

	col2	col3
0	False	False
1	False	False
2	False	False
3	False	False

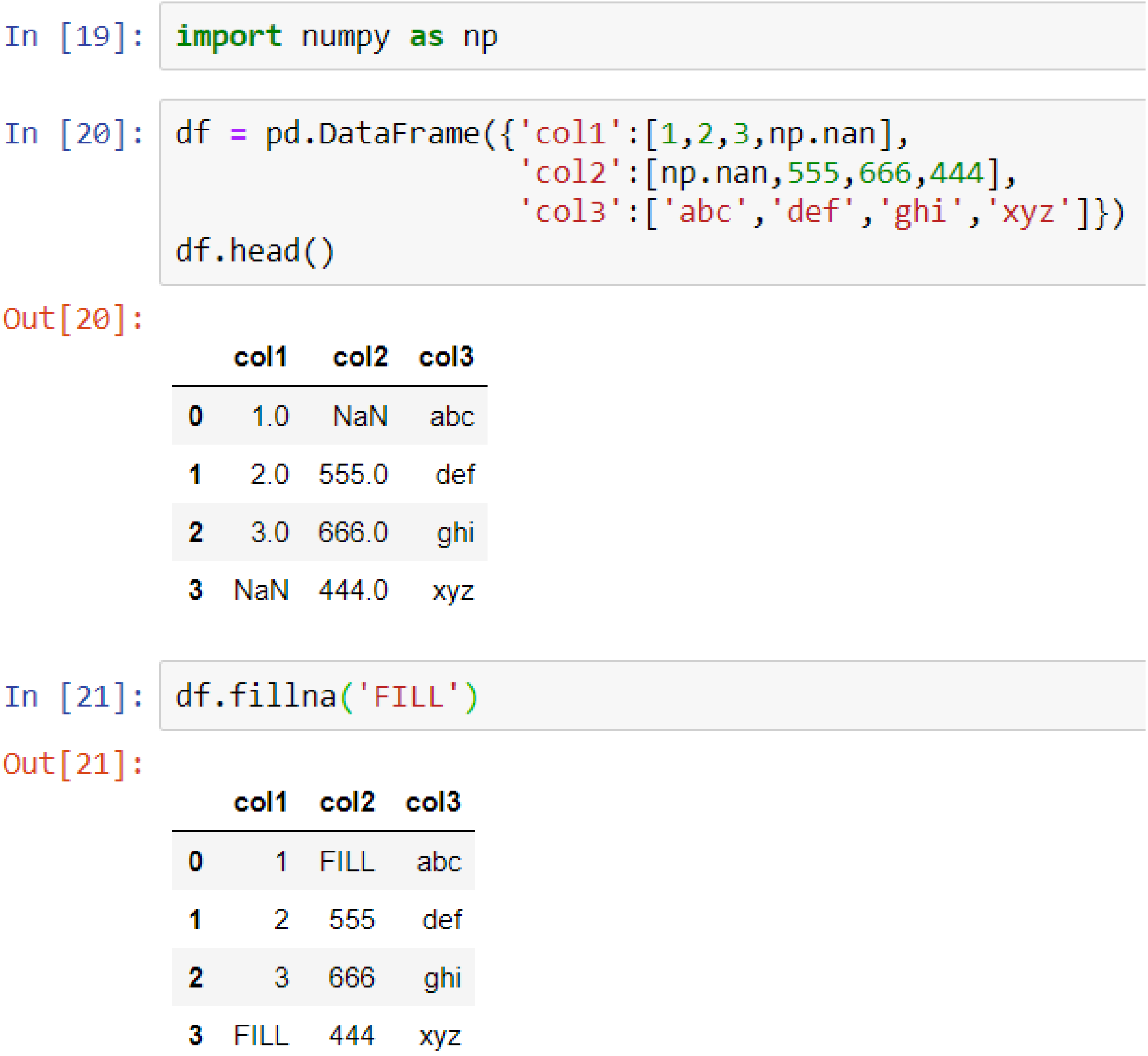
```
In [18]: # Drop rows with NaN Values
         df.dropna()
```

Out[18]:

	col2	col3
0	444	abc
1	555	def
2	666	ghi
3	444	xyz

Purwadhika
Startup and Coding School

Filling in NaN with Something Else



In [19]:
```
import numpy as np
```

In [20]:
```
df = pd.DataFrame({'col1':[1,2,3,np.nan],
                   'col2':[np.nan,555,666,444],
                   'col3':['abc','def','ghi','xyz']})
df.head()
```

Out[20]:

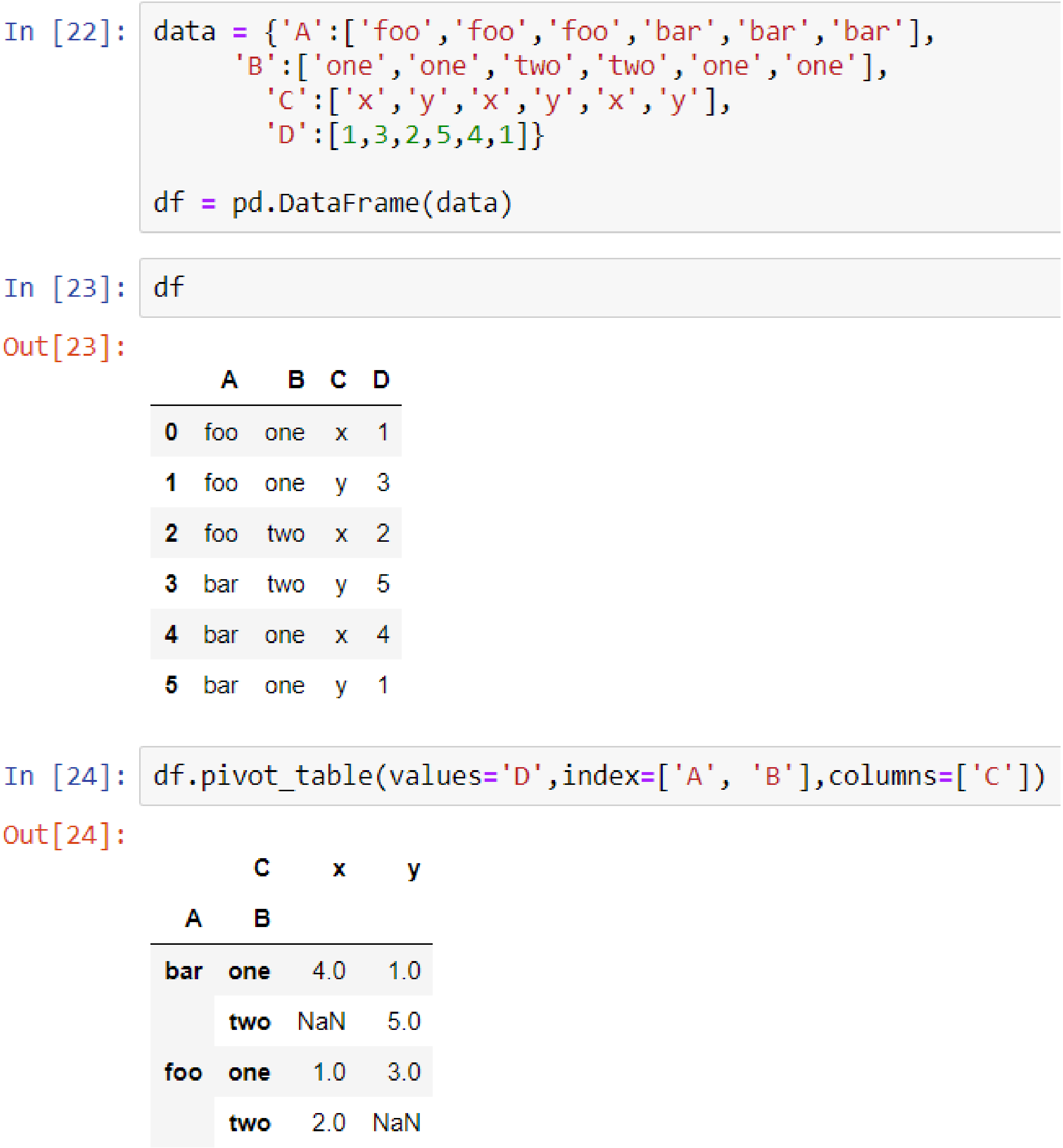
	col1	col2	col3
0	1.0	NaN	abc
1	2.0	555.0	def
2	3.0	666.0	ghi
3	NaN	444.0	xyz

In [21]:
```
df.fillna('FILL')
```

Out[21]:

	col1	col2	col3
0	1	FILL	abc
1	2	555	def
2	3	666	ghi
3	FILL	444	xyz

Pivot Table



```
In [22]: data = {'A':['foo','foo','foo','bar','bar','bar'],
               'B':['one','one','two','two','one','one'],
                 'C':['x','y','x','y','x','y'],
                 'D':[1,3,2,5,4,1]}

         df = pd.DataFrame(data)
```

```
In [23]: df
```

```
Out[23]:
        A    B  C  D
   0  foo  one  x  1
   1  foo  one  y  3
   2  foo  two  x  2
   3  bar  two  y  5
   4  bar  one  x  4
   5  bar  one  y  1
```

```
In [24]: df.pivot_table(values='D',index=['A', 'B'],columns=['C'])
```

```
Out[24]:
             C    x    y
      A    B
    bar  one   4.0  1.0
         two   NaN  5.0
    foo  one   1.0  3.0
         two   2.0  NaN
```

Purwadhika
Startup and Coding School