In [1]:
```
import pandas as pd
import numpy as np
from pandas import Series, DataFrame
```

In [2]:
```
#Create a simple series combining positive as well as negative random numbers and store the values in 'df' variable
#Print the above variable and show the output

df = pd.Series([1,2,3,4,-5,-6,-7,-8])
df
```

Out[2]:
```
0    1
1    2
2    3
3    4
4   -5
5   -6
6   -7
7   -8
dtype: int64
```

In [3]:
```
#Print out only the values in the variable

df.values
```

Out[3]:
```
array([ 1,  2,  3,  4, -5, -6, -7, -8], dtype=int64)
```

In [4]:
```
# Print out the index reference in the variable

df[[0,1,2,3,4,5,6,7,]]
```

Out[4]:
```
0    1
1    2
2    3
3    4
4   -5
5   -6
6   -7
7   -8
dtype: int64
```

In [5]:
```
#Create Series data structure with 5 values and give index labels as A,B,C,D,E

ser = pd.Series([1,2,3,4,5], index = ['A', 'B', 'C', 'D', 'E'])
ser
```

Out[5]:
```
A    1
B    2
C    3
D    4
E    5
dtype: int64
```

In [6]:
```
#Use following codes to create Series

sdata = {'Mumbai': 2000, 'Kolkata': 4000, 'Delhi': 10000, 'Chennai': 'NaN'}
obj3 = pd.Series(sdata)
obj3
```

Out[6]:
```
Mumbai      2000
Kolkata     4000
Delhi      10000
Chennai      NaN
dtype: object
```

In [7]:
```
#Use above create series and do the following:
#Add Gujrat index in the dataset with value as NaN

sdata = {'Mumbai': 2000, 'Kolkata': 4000, 'Delhi': 10000, 'Chennai': 'NaN', 'Gujrat': 'Nan'}
obj3 = pd.Series(sdata)
obj3
```

Out[7]:
```
Mumbai      2000
Kolkata     4000
Delhi      10000
Chennai      NaN
Gujrat       Nan
dtype: object
```

In [8]:
```
#Find out the missing values from the above data set and after incoporating point 1 above. Output to be given as boolean values

obj3.isnull()
```

Out[8]:
```
Mumbai     False
Kolkata    False
Delhi      False
Chennai    False
Gujrat     False
dtype: bool
```

In [9]:
```
#Replace the missing values with 2000 and 4000

sdata = {'Mumbai': 2000, 'Kolkata': 4000, 'Delhi': 10000, 'Chennai': 2000, 'Gujrat': 4000}
obj3 = pd.Series(sdata)
obj3
```

Out[9]:
```
Mumbai      2000
Kolkata     4000
Delhi      10000
Chennai     2000
Gujrat      4000
dtype: int64
```

In [10]:
```
#Create a DataFrame with following columns
#1. Country: India, China, Nepal, Bhutan, Srilanka
#2. Population: 1000, 2000, 500, 200, 50
#3. GDP: 5000, 10000, 200, 100,80
#4. Index: Use Population and GDP to be the row label indexes

df = {'Country' : ['India', 'China', 'Nepal', 'Bhutan', 'Srilanka'],
                  'Popultion' : [1000, 2000, 500, 200, 50],
                  'GDP' : [5000, 10000, 200, 100, 80]}
#df.index = ('Population', 'Population', 'Population', 'GDP', 'GDP')
df1 = pd.DataFrame(df)
```

In [11]:
```
df1
```

Out[11]:

| | Country | Popultion | GDP |
|---|---|---|---|
| **0** | India | 1000 | 5000 |
| **1** | China | 2000 | 10000 |
| **2** | Nepal | 500 | 200 |
| **3** | Bhutan | 200 | 100 |
| **4** | Srilanka | 50 | 80 |

In [12]:
```
#Use the above dataset to perform following tasks:
#1. Filter out all the values for China

df1.loc[1]
```

Out[12]:
```
Country      China
Popultion     2000
GDP          10000
Name: 1, dtype: object
```

In [13]:
```
#2. Filter out India's GDP

df1.iloc[0, [2]]
```

Out[13]:
```
GDP    5000
Name: 0, dtype: object
```

In [14]:
```
#3. Filter out GDP for all the countries

df1.iloc[[0, 1, 2, 3, 4], [2]]
```

Out[14]:

| | GDP |
|---|---|
| **0** | 5000 |
| **1** | 10000 |
| **2** | 200 |
| **3** | 100 |
| **4** | 80 |

In [15]:
```
#✔ Create random Series of 10 values and give index to be [a,b,c,d,e,f,g,h,i,j]

ran = pd.Series([1, 2, 3, 4, 5, 6, 7, 8, 9, 0],
                index = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'])
ran
```

Out[15]:
```
a    1
b    2
c    3
d    4
e    5
f    6
g    7
h    8
i    9
j    0
dtype: int64
```

In [16]:
```
#✔ Filter out value of index g

ran.loc['g']
```

Out[16]:
```
7
```

In [17]:
```
#✔ Filter out values of indexes e to j

ran[4:10]
```

Out[17]:
```
e    5
f    6
g    7
h    8
i    9
j    0
dtype: int64
```

In [18]:
```
#✔ Use filter to sub-set values less than 2

ran[ran<2]
```

Out[18]:
```
a    1
j    0
dtype: int64
```

In [19]:
```
#✔ Create following DataFrame
data1 = pd.DataFrame(np.arange(16).reshape((4, 4)), columns=list('bcde'),index=['Kolkata', 'Chennai', 'Mumbai', Delhi])
#List out the error if any and try to resolve the same
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[19], line 2
      1 #✔ Create following DataFrame
----> 2 data1 = pd.DataFrame(np.arange(16).reshape((4, 4)), columns=list('bcde'),index=['Kolkata', 'Chennai', 'Mumbai', Delhi])

NameError: name 'Delhi' is not defined
```

In [20]:
```
data1 = pd.DataFrame(np.arange(16).reshape((4, 4)), columns=list('bcde'),index=['Kolkata', 'Chennai', 'Mumbai', 'Delhi'])
data1
```

Out[20]:

| | b | c | d | e |
|---|---|---|---|---|
| **Kolkata** | 0 | 1 | 2 | 3 |
| **Chennai** | 4 | 5 | 6 | 7 |
| **Mumbai** | 8 | 9 | 10 | 11 |
| **Delhi** | 12 | 13 | 14 | 15 |

In [ ]:
```
```