```
In [2]: #Q1

        import pandas as pd
        import numpy as np
```

```
In [5]: data=([4,8,15,16,23,42])
        ser=pd.Series(data)
        print(ser)
```

```
0     4
1     8
2    15
3    16
4    23
5    42
dtype: int64
```

```
In [4]: #Q2

        import pandas as pd
        import numpy as np
        data=[1,2,3,4,5,6,7,8,11]
        ser=pd.Series(data)
        print(ser)
```

```
0     1
1     2
2     3
3     4
4     5
5     6
6     7
7     8
8    11
dtype: int64
```

```
In [13]: #Q3
         import pandas as pd
         data={"name":['Alice','Bob','Claire'],
               "age":[25,30,27],
               "Gender":['female','male','female']
              }
         df=pd.DataFrame(data)
         df.set_index('name',inplace=True)
```

```
In [14]: df
```

Out[14]:

	age	Gender
name		
Alice	25	female
Bob	30	male
Claire	27	female

In [ ]:

```
#Q4
#A Pandas DataFrame is a 2 dimensional data structure, like a 2 dimensional array,
#DataFrames are one of the most common data structures used in modern data analytic


#A Python one-dimensional labelled array called a Pandas Series may hold any form o
#Each component of a series has a unique identification thanks to an index. It is p
#For actions that only involve one column of data, a Series performs more quickly t

#As noted in the table, a Pandas Series is a 1D array of data, but a single-column
```

In [19]:

```
#EX Series

import pandas as pd

# Create a Pandas Series from a list
data = [1000, 2000, 3000, 4000, 5000]
s = pd.Series(data)

# Print the Series
print(s)
```

```
0    1000
1    2000
2    3000
3    4000
4    5000
dtype: int64
```

In [20]:

```
#EX DataFrame

import pandas as pd

# Create a DataFrame with a single column using a Python list
data = [1000, 2000, 3000, 4000, 5000]
df = pd.DataFrame(data, columns=['Column1'])

# Print the DataFrame
print(df)
```

```
   Column1
0     1000
1     2000
2     3000
3     4000
4     5000
```

Q5

1.Read data:-We can read data in pandas data frame as read_csv(). 2.Head and Tail:- Head returns the first rows, if no input is given it will always show above 5 rows. In contrast to see below rows, we can use df.tail(). 3.Shape size and info:-We can use df.shape, it gives a total number of rows and then columns. df.size() returns the number of rows times number of columns in the data frame. We can also use df.info(), from that we get different information such as rows from RangeIndex, Data columns and then data type of each column.

4.isna():-if one needs to get the total number of null values in a data, we can use df.isna().

5.Describe():-understand basic statistics of variables we can use df.describe(). 6.Nunique():-To get the total unique values of variables, we can use df.nunique(). 7.Columns:-To know the names of all the variables in a data frame, we can use df.columns. 8.

In [21]:
```
#Q6

#DataFrames are both value and size-mutable
#A Series, by contrast, is only value-mutable,not size-mutable. The length of a Ser
# In Panel Data and size are mutable
```

In [28]:
```
#Q7

# Importing Pandas library
import pandas as pd

# Creating two lists
author = ['Jitender', 'Purnima',
          'Arpit', 'Jyoti']
article = [210, 211, 114, 178]

# Creating two Series by passing lists
auth_series = pd.Series(author)
article_series = pd.Series(article)

# Creating a dictionary by passing Series objects as values
frame = {'Author': auth_series,
         'Article': article_series}

# Creating DataFrame by passing Dictionary
result = pd.DataFrame(frame)

# Printing elements of Dataframe
print(result)
```

```
     Author  Article
0  Jitender      210
1   Purnima      211
2     Arpit      114
3     Jyoti      178
```

```
In [ ]:
```