```
import pandas as pd

#  pandas series, pandas dataframe

# pandas series
names = ['John','Steve','Roy']
s = pd.Series(names)
s
```

```
0     John
1    Steve
2      Roy
dtype: object
```

```
# pandas series
names = ['John','Steve','Roy']
s = pd.Series(names, index = [302, 520, 423])
s
```

```
302     John
520    Steve
423      Roy
dtype: object
```

```
# pandas series
names = ['John','Steve','Roy']
s = pd.Series(names, index = [302, 520])
s
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_13056/1535242997.py in <module>
      1 # pandas series
      2 names = ['John','Steve','Roy']
----> 3 s = pd.Series(names, index = [302, 520])
      4 s

~\anaconda3\lib\site-packages\pandas\core\series.py in __init__(self, data, index, dtype, name, copy, fastpath)
    428                 index = ibase.default_index(len(data))
    429             elif is_list_like(data):
--> 430                 com.require_length_match(data, index)
    431
    432             # create/copy the manager

~\anaconda3\lib\site-packages\pandas\core\common.py in require_length_match(data, index)
    529     """
    530     if len(data) != len(index):
```

```
--> 531         raise ValueError(
    532             "Length of values "
    533             f"({len(data)}) "

ValueError: Length of values (3) does not match length of index (2)
```

```
# pandas series
names = ['John','Steve','Roy']
s = pd.Series(names, index = [302, 520, 423, 550])
s
```

```
---------------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
~\AppData\Local\Temp/ipykernel_13056/2654073979.py in <module>
      1 # pandas series
      2 names = ['John','Steve','Roy']
----> 3 s = pd.Series(names, index = [302, 520, 423, 550])
      4 s

~\anaconda3\lib\site-packages\pandas\core\series.py in __init__(self,
data, index, dtype, name, copy, fastpath)
    428                 index = ibase.default_index(len(data))
    429             elif is_list_like(data):
--> 430                 com.require_length_match(data, index)
    431
    432             # create/copy the manager

~\anaconda3\lib\site-packages\pandas\core\common.py in
require_length_match(data, index)
    529     """
    530     if len(data) != len(index):
--> 531         raise ValueError(
    532             "Length of values "
    533             f"({len(data)}) "

ValueError: Length of values (3) does not match length of index (4)
```

```
# pandas series
names = ('John','Steve','Roy')
s = pd.Series(names, index = [302, 520, 423])
s
```

```
302     John
520    Steve
423      Roy
dtype: object
```

```
# create a series with values -
sub = ['python','java','ml','tableau']
s1 = pd.Series(sub, index = ['a','b','c','d'])
s1
```

```
a     python
b       java
c         ml
d    tableau
dtype: object
```

```
# we can create series using dict -
prod = {'Laptop' : 20, 'Chargers' : 40, 'notepads' : 25}
prod
```

```
{'Laptop': 20, 'Chargers': 40, 'notepads': 25}
```

```
s3 = pd.Series(prod)
s3
```

```
Laptop      20
Chargers    40
notepads    25
dtype: int64
```

```
s.index
```

```
Int64Index([302, 520, 423], dtype='int64')
```

```
s1.index
```

```
Index(['a', 'b', 'c', 'd'], dtype='object')
```

```
s3.index
```

```
Index(['Laptop', 'Chargers', 'notepads'], dtype='object')
```

```
s.values
```

```
array(['John', 'Steve', 'Roy'], dtype=object)
```

```
s1.values
```

```
array(['python', 'java', 'ml', 'tableau'], dtype=object)
```

```
s3.values
```

```
array([20, 40, 25], dtype=int64)
```

```
import numpy as np
a1 = np.array([10,20,6,8,9])
s4 = pd.Series(a1)
s4
```

```
0    10
1    20
2     6
3     8
4     9
dtype: int32
```

```
s4 + 5
```

```
0    15
1    25
2    11
3    13
4    14
dtype: int32
```

```
# filtering data
s4
```

```
0    10
1    20
2     6
3     8
4     9
dtype: int32
```

```
s4 > 8
```

```
0     True
1     True
2    False
3    False
4     True
dtype: bool
```

```
s4[ s4 > 8  ]
```

```
0    10
1    20
4     9
dtype: int32
```

```
s3[s3 > 30]
```

```
Chargers    40
dtype: int64
```

```
# create a series from array - values [910, 750, 340, 765, 789]
# subtract 200 values
# display the values greater than 500
# display the index
# display only the vaues from the series
```

```
v = [910, 750, 340, 765, 789]

s1.values

# pandas dataframe -
# dataframe stores the data in form of rows and columns

d = {'Name' : ['John', 'Steve', 'Roy'],
    'Age' : [24, 27, 22],
    'City' : ['California', 'NY City', 'California']}
d
```

```
{'Name': ['John', 'Steve', 'Roy'],
 'Age': [24, 27, 22],
 'City': ['California', 'NY City', 'California']}
```

```
df = pd.DataFrame(d)
df
```

```
    Name  Age        City
0   John   24  California
1  Steve   27     NY City
2    Roy   22  California
```

```
df = pd.DataFrame(d, index = [320, 520, 423])
df
```

```
      Name  Age        City
320   John   24  California
520  Steve   27     NY City
423    Roy   22  California
```

```
# create a dataframe from the below dict
d2 = {'Subject' : ['Java', 'ML', 'Python', 'SQL'],
      'Passing marks' : [23, 30, 25, 22]}
df2 = pd.DataFrame(d2)
df2
```

```
  Subject  Passing marks
0    Java             23
1      ML             30
2  Python             25
3     SQL             22
```

```
df3 = pd.DataFrame(d2, index = ['sem1', 'sem1', 'sem2', 'sem2'] )
df3
```

```
     Subject  Passing marks
sem1    Java             23
sem1      ML             30
sem2  Python             25
sem2     SQL             22
```

```
# s.index, s.values
df.index
```

```
Int64Index([320, 520, 423], dtype='int64')
```

```
df2.index
```

```
RangeIndex(start=0, stop=4, step=1)
```

```
df3.index
```

```
Index(['sem1', 'sem1', 'sem2', 'sem2'], dtype='object')
```

```
df.values
```

```
array([['John', 24, 'California'],
       ['Steve', 27, 'NY City'],
       ['Roy', 22, 'California']], dtype=object)
```

```
# create a dataframe using numpy array
a2 = np.arange(0, 25).reshape(5,5)
a2
```

```
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19],
       [20, 21, 22, 23, 24]])
```

```
pd.DataFrame(a2)
```

	0	1	2	3	4
0	0	1	2	3	4
1	5	6	7	8	9
2	10	11	12	13	14
3	15	16	17	18	19
4	20	21	22	23	24

```
df5 = pd.DataFrame(a2,
                   columns = ['c1', 'c2','c3', 'c4', 'c5'],
                   index = ['r1', 'r2','r3', 'r4', 'r5'] )
df5
```

	c1	c2	c3	c4	c5
r1	0	1	2	3	4
r2	5	6	7	8	9
r3	10	11	12	13	14
r4	15	16	17	18	19
r5	20	21	22	23	24

```
# row selections and column selections
df
```

```
      Name  Age        City
320   John   24  California
520  Steve   27     NY City
423    Roy   22  California
```

```
df['Name']
```

```
320     John
520    Steve
423      Roy
Name: Name, dtype: object
```

```
df['Age']
```

```
320    24
520    27
423    22
Name: Age, dtype: int64
```

```
df['City']
```

```
320    California
520       NY City
423    California
Name: City, dtype: object
```

```
df5
```

```
    c1  c2   c3  c4  c5
r1   0   1    2   3   4
r2   5   6    7   8   9
r3  10  11   12  13  14
r4  15  16   17  18  19
r5  20  21   22  23  24
```

```
df5['c3']
```

```
r1     2
r2     7
r3    12
r4    17
r5    22
Name: c3, dtype: int32
```

```
df5['c4']
```

```
r1     3
r2     8
r3    13
```

```
r4    18
r5    23
Name: c4, dtype: int32
```

```
df5[  ['c3','c4']  ]
```

```
    c3  c4
r1   2   3
r2   7   8
r3  12  13
r4  17  18
r5  22  23
```

```
df5[  ['c5', 'c2', 'c5'] ]
```

```
    c5  c2  c5
r1   4   1   4
r2   9   6   9
r3  14  11  14
r4  19  16  19
r5  24  21  24
```

```
df5.index
```

```
Index(['r1', 'r2', 'r3', 'r4', 'r5'], dtype='object')
```

```
df5.columns
```

```
Index(['c1', 'c2', 'c3', 'c4', 'c5'], dtype='object')
```

```
# add a new column to the dataframe
df['Marks'] = 'NA'
```

```
df
```

```
      Name  Age        City Marks
320   John   24  California    NA
520  Steve   27     NY City    NA
423    Roy   22  California    NA
```

```
df['Marks'] = [42, 45, 46]
df
```

```
      Name  Age        City  Marks
320   John   24  California     42
520  Steve   27     NY City     45
423    Roy   22  California     46
```

```
df['updated_marks']   = df['Marks'] + 2
```

```
df
```

```
      Name  Age        City  Marks  updated_marks
320   John   24  California     42             44
520  Steve   27     NY City     45             47
423    Roy   22  California     46             48
```

```
df5   # create a new column c6 - with all values as 0

# change values to [0,1,2,3,4]
df5['c6'] = 0
df5
```

```
    c1  c2  c3  c4  c5  c6
r1   0   1   2   3   4   0
r2   5   6   7   8   9   0
r3  10  11  12  13  14   0
r4  15  16  17  18  19   0
r5  20  21  22  23  24   0
```

```
df5['c6'] = [0,1,2,3,4]
df5
```

```
    c1  c2  c3  c4  c5  c6
r1   0   1   2   3   4   0
r2   5   6   7   8   9   1
r3  10  11  12  13  14   2
r4  15  16  17  18  19   3
r5  20  21  22  23  24   4
```

```
# add c2 and c3 values and store it in c7

df5['c7'] = df5['c2'] + df5['c3']
df5
```

```
    c1  c2  c3  c4  c5  c6  c7
r1   0   1   2   3   4   0   3
r2   5   6   7   8   9   1  13
r3  10  11  12  13  14   2  23
r4  15  16  17  18  19   3  33
r5  20  21  22  23  24   4  43
```

```
df  # find the records where age is less than 25
```

```
      Name  Age        City  Marks  updated_marks
320   John   24  California     42             44
520  Steve   27     NY City     45             47
423    Roy   22  California     46             48
```

```
df['Age'] < 25
```

```
320     True
520    False
```

```
423     True
Name: Age, dtype: bool
```

```
df[  df['Age'] < 25 ]
```

```
     Name  Age        City  Marks  updated_marks
320  John   24  California     42             44
423   Roy   22  California     46             48
```

```
df2 # find the records where passing marks is greater than 25
```

```
  Subject  Passing marks
0    Java             23
1      ML             30
2  Python             25
3     SQL             22
```

```
df
```

```
      Name  Age        City  Marks  updated_marks
320   John   24  California     42             44
520  Steve   27     NY City     45             47
423    Roy   22  California     46             48
```

```
df[df['City'] == 'California']
```

```
     Name  Age        City  Marks  updated_marks
320  John   24  California     42             44
423   Roy   22  California     46             48
```

```
df
```

```
  Subject  Passing marks
0    Java             23
1      ML             30
2  Python             25
3     SQL             22
```

```
df
```

```
      Name  Age        City  Marks  updated_marks
320   John   24  California     42             44
520  Steve   27     NY City     45             47
423    Roy   22  California     46             48
```

```
df[df['Name'] == 'John'][['Marks', 'Name']]
```

```
     Marks  Name
320     42  John
```

```
d3 = {"Name" : ['Rahul', 'Roy', 'Raman', 'Arun', 'Ramya'],
     'Age' : [22, 24, 21, 25, 26],
     'City' : ['blr', 'chn', 'blr', 'chn', 'chn'],
     'ml_marks' : [45, 42, 44, 23,22],
     'python_marks' : [34, 38, 37, 42, 22]}
df_1 = pd.DataFrame(d3)
df_1
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
1    Roy   24  chn        42            38
2  Raman   21  blr        44            37
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
# find the records where age > 23
df_1[df_1['Age'] > 23]
```

```
    Name  Age City  ml_marks  python_marks
1    Roy   24  chn        42            38
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
# find the people who got more than 40 in ml_marks
# find the records where city is chn
df_1[df_1['City'] == 'chn']
```

```
    Name  Age City  ml_marks  python_marks
1    Roy   24  chn        42            38
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
# filtering rows

# display the records where the city is chn and python marks is more
than 30
df_1
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
1    Roy   24  chn        42            38
2  Raman   21  blr        44            37
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
df_1['City'] == 'chn'
```

```
0    False
1     True
2    False
3     True
```

```
4     True
Name: City, dtype: bool
```

```
df_1['python_marks'] > 30
```

```
0     True
1     True
2     True
3     True
4    False
Name: python_marks, dtype: bool
```

```
df_1[(df_1['City'] == 'chn') & (df_1['python_marks'] > 30)]
```

```
   Name  Age City  ml_marks  python_marks
1   Roy   24  chn        42            38
3  Arun   25  chn        23            42
```

```
df_1
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
1    Roy   24  chn        42            38
2  Raman   21  blr        44            37
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
# loc and iloc
```

```
df_1[0 : 3]
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
1    Roy   24  chn        42            38
2  Raman   21  blr        44            37
```

```
df_1[0 : 3 : 2]
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
2  Raman   21  blr        44            37
```

```
df[320]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
   3360             try:
-> 3361                 return self._engine.get_loc(casted_key)
```

```
   3362             except KeyError as err:

~\anaconda3\lib\site-packages\pandas\_libs\index.pyx in
pandas._libs.index.IndexEngine.get_loc()

~\anaconda3\lib\site-packages\pandas\_libs\index.pyx in
pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 320

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call
last)
~\AppData\Local\Temp/ipykernel_13056/1171506268.py in <module>
----> 1 df[320]

~\anaconda3\lib\site-packages\pandas\core\frame.py in
__getitem__(self, key)
   3456             if self.columns.nlevels > 1:
   3457                 return self._getitem_multilevel(key)
-> 3458             indexer = self.columns.get_loc(key)
   3459             if is_integer(indexer):
   3460                 indexer = [indexer]

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in
get_loc(self, key, method, tolerance)
   3361                 return self._engine.get_loc(casted_key)
   3362             except KeyError as err:
-> 3363                 raise KeyError(key) from err
   3364
   3365         if is_scalar(key) and isna(key) and not self.hasnans:

KeyError: 320

# loc and iloc - for indexing and slicing of rows as well as cols
df.iloc[0]

Name                   John
Age                      24
City             California
Marks                    42
updated_marks            44
Name: 320, dtype: object
```

```
df_1.iloc[0]
```

```
Name            Rahul
Age                22
City              blr
ml_marks           45
python_marks       34
Name: 0, dtype: object
```

```
df_1.iloc[0 : 3 : 2]
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
2  Raman   21  blr        44            37
```

```
df_1  # select only index 4
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
1    Roy   24  chn        42            38
2  Raman   21  blr        44            37
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
df_1.iloc[4]
```

```
Name            Ramya
Age                26
City              chn
ml_marks           22
python_marks       22
Name: 4, dtype: object
```

```
# select row 1 to 3
```

```
df_1.iloc[1 : 4 , 0  ]
```

```
1      Roy
2    Raman
3     Arun
Name: Name, dtype: object
```

```
df_1.iloc[1 : 4 , [0 , 2]  ]
```

```
    Name City
1    Roy  chn
2  Raman  blr
3   Arun  chn
```

```
df_1.iloc[1 : 4 , 0 : 4  ]
```

```
    Name  Age City  ml_marks
1    Roy   24  chn        42
2  Raman   21  blr        44
3   Arun   25  chn        23
```

```
df_1
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
1    Roy   24  chn        42            38
2  Raman   21  blr        44            37
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
# display the rows 0,2,3 and columns city, python_marks
df_1.iloc[ [0,2,3] , [2,4] ]
```

```
  City  python_marks
0  blr            34
2  blr            37
3  chn            42
```

```
df_1.iloc[  [2,4] ,  [1,3]  ]
```

```
   Age  ml_marks
2   21        44
4   26        22
```

```
# loc
df_1
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
1    Roy   24  chn        42            38
2  Raman   21  blr        44            37
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
df_1.loc[0]
```

```
Name            Rahul
Age                22
City              blr
ml_marks           45
python_marks       34
Name: 0, dtype: object
```

```
df_1.loc[0 : 3 ]
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
1    Roy   24  chn        42            38
```

```
2  Raman   21  blr        44            37
3   Arun   25  chn        23            42
```

```
df_1.loc[0 : 3 : 2 ]
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
2  Raman   21  blr        44            37
```

```
df_1.loc[1 : 3]
```

```
    Name  Age City  ml_marks  python_marks
1    Roy   24  chn        42            38
2  Raman   21  blr        44            37
3   Arun   25  chn        23            42
```

```
df_1.loc[  [0,2,3]  ]
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
2  Raman   21  blr        44            37
3   Arun   25  chn        23            42
```

```
df_1.loc[  0 : 3  , 'Age' : 'ml_marks'  ]
```

```
   Age City  ml_marks
0   22  blr        45
1   24  chn        42
2   21  blr        44
3   25  chn        23
```

```
df_1.loc[  [0,2,3] ,  ['Age', 'python_marks'] ]
```

```
   Age  python_marks
0   22            34
2   21            37
3   25            42
```

```
# iloc, iloc in integer based rows and columns
# loc is name based rows and cols
# in iloc end is exclusive
# in loc end is inclusive
```

```
df_1.shape
```

```
(5, 5)
```

```
df_1.dtypes
```

```
Name            object
Age              int64
City            object
ml_marks         int64
```

```
python_marks     int64
dtype: object
```

```
df_1.head(2)
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
1    Roy   24  chn        42            38
```

```
df_1.tail(2)
```

```
    Name  Age City  ml_marks  python_marks
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
df_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Name          5 non-null      object
 1   Age           5 non-null      int64 
 2   City          5 non-null      object
 3   ml_marks      5 non-null      int64 
 4   python_marks  5 non-null      int64 
dtypes: int64(3), object(2)
memory usage: 328.0+ bytes
```

```
df_1['Age'].min()
```

```
21
```

```
df_1['ml_marks'].max()
```

```
45
```

```
df_1['ml_marks'].sum()
```

```
176
```

```
df_1['ml_marks'].mean()
```

```
35.2
```

```
df_1['ml_marks'].var()
```

```
135.7
```

```
df_1['ml_marks'].std()
```

```
11.64903429473877
```

```
# find the min python marks, max python marks, mean python marks, var, std, median
```

```
df_1['ml_marks'].median()
```

```
42.0
```

```
df_1
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
1    Roy   24  chn        42            38
2  Raman   21  blr        44            37
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
df_1['City'].unique()
```

```
array(['blr', 'chn'], dtype=object)
```

```
df_1['City'].nunique()
```

```
2
```

```
df_1['City'].value_counts()
```

```
chn    3
blr    2
Name: City, dtype: int64
```

```
df_1['City'].value_counts(normalize = True) * 100
```

```
chn    60.0
blr    40.0
Name: City, dtype: float64
```

```
# sort_values
df_1
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
1    Roy   24  chn        42            38
2  Raman   21  blr        44            37
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
df_1.sort_values(by = 'python_marks' )
```

```
    Name  Age City  ml_marks  python_marks
4  Ramya   26  chn        22            22
0  Rahul   22  blr        45            34
2  Raman   21  blr        44            37
```

```
1    Roy   24  chn        42            38
3   Arun   25  chn        23            42
```

```
df_1.sort_values(by = 'python_marks' , ascending = False )
```

```
    Name  Age City  ml_marks  python_marks
3   Arun   25  chn        23            42
1    Roy   24  chn        42            38
2  Raman   21  blr        44            37
0  Rahul   22  blr        45            34
4  Ramya   26  chn        22            22
```

```
df_1.sort_values(by = 'City')
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
2  Raman   21  blr        44            37
1    Roy   24  chn        42            38
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
df_1.sort_values(by = ['City' , 'python_marks']  )
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
2  Raman   21  blr        44            37
4  Ramya   26  chn        22            22
1    Roy   24  chn        42            38
3   Arun   25  chn        23            42
```

```
df_1
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
1    Roy   24  chn        42            38
2  Raman   21  blr        44            37
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
# rename, reset_index, set_index
df_1.rename(columns = {'ml_marks' : 'machine_learning_marks',
'Name'  :  'stud_name'}  )
```

```
  stud_name  Age City  machine_learning_marks  python_marks
0     Rahul   22  blr                      45            34
1       Roy   24  chn                      42            38
2     Raman   21  blr                      44            37
3      Arun   25  chn                      23            42
4     Ramya   26  chn                      22            22
```

```
# set_index
df_1.set_index('Name')
```

```
       Age City  ml_marks  python_marks
Name
Rahul   22  blr        45            34
Roy     24  chn        42            38
Raman   21  blr        44            37
Arun    25  chn        23            42
Ramya   26  chn        22            22
```

```
# set_index
df_1.set_index('Name', inplace=True)
```

```
df_1
```

```
       Age City  ml_marks  python_marks
Name
Rahul   22  blr        45            34
Roy     24  chn        42            38
Raman   21  blr        44            37
Arun    25  chn        23            42
Ramya   26  chn        22            22
```

```
df_1.reset_index(inplace=True)
```

```
df_1
```

```
    Name  Age City  ml_marks  python_marks
0  Rahul   22  blr        45            34
1    Roy   24  chn        42            38
2  Raman   21  blr        44            37
3   Arun   25  chn        23            42
4  Ramya   26  chn        22            22
```

```
# drop
df_1.drop(columns = ['City', 'python_marks'])
```

```
    Name  Age  ml_marks
0  Rahul   22        45
1    Roy   24        42
2  Raman   21        44
3   Arun   25        23
4  Ramya   26        22
```

```
df_1.drop(columns = ['City', 'python_marks']  , inplace=True)
```

```
df_1
```

```
    Name  Age  ml_marks
0  Rahul   22        45
1    Roy   24        42
2  Raman   21        44
3   Arun   25        23
4  Ramya   26        22
```

```
df_1.drop(index = [0,2] )
```

```
    Name  Age  ml_marks
1    Roy   24        42
3   Arun   25        23
4  Ramya   26        22
```

```
# pandas - Series, DataFrame
# pd.Series(data, index = []) - but just a single column
# on Series we can perform any arithmatic, we can do comparison
# .index - gives the indexes, .values - values from the series
# pd.DataFrame(dict, columns =  , index = )
# df[col] -.> fives the specified col,
# df[ [col1, col2...coln] ]  -> when we want to slect more than 1
column
# df[ df[col]  > value ] -> filter the rows
# how to add new column to dataframe, df[new_col] = value
# loc and iloc
# .min, .max, .var, .sum, .std, .var, .median, .shape, dtypes - type
of data in each col
# .info(), .head(), tail(), .rename(columns =
{}), .set_index(), .reset_index()
# .unique(), nunique(), .value_counts(), drop(column), sort_values
```

```
d3 = {"Account_type" : ['current', 'saving', 'saving', 'current',
'current'],
     'Loan_Amount' : [20000, 15000, 16000, 25000, 22000],
     'interest_rate' : [10, 11.2, 9.2, 10.1, 12],
     'customer_type' : ['New', 'New', 'Existing', 'New', 'Existing']}
df_2 = pd.DataFrame(d3)
df_2
```

```
  Account_type  Loan_Amount  interest_rate customer_type
0      current        20000           10.0           New
1       saving        15000           11.2           New
2       saving        16000            9.2      Existing
3      current        25000           10.1           New
4      current        22000           12.0      Existing
```

```
# How many people have got loan more than 20000
# What is the count of new and existing customers
# What is the minimum int rate?
# What is the maximum loan given to a customer?
# rename the column customer_type to cust_type
# add a new column , age and assign the values [32, 31, 22, 25, 28]
```