```
import pandas as pd
df1 = pd.DataFrame({
   'id':[1,2,3,4,5],
   'Name': ['Ram', 'Sham', 'Sita', 'Geeta', 'Meena'],
   'subject_id':['DBMS','TOC','SPOS','MATHS','WT']})
df2 = pd.DataFrame(
   {'id':[1,2,3,4,5],
   'Name': ['X', 'Y', 'Z', 'A', 'B'],
   'subject_id':['DS','TOC','IOT','MATHS','WT']})
print(df1)
print(df2)
```

```
   id   Name subject_id
0   1    Ram       DBMS
1   2   Sham        TOC
2   3   Sita       SPOS
3   4  Geeta      MATHS
4   5  Meena         WT
   id Name subject_id
0   1    X         DS
1   2    Y        TOC
2   3    Z        IOT
3   4    A      MATHS
4   5    B         WT
```

```
print(pd.merge(df1,df2,on='id'))              #Merge Two DataFrames on a Key
```

```
   id Name_x subject_id_x Name_y subject_id_y
0   1    Ram         DBMS      X           DS
1   2   Sham          TOC      Y         DELD
2   3   Sita         SPOS      Z          IOT
3   4  Geeta        MATHS      A        MATHS
4   5  Meena           WT      B           WT
```

```
#Merge Two DataFrames on Multiple Keys
print(pd.merge(df1,df2,on=['id','subject_id']))
```

```
   id Name_x subject_id Name_y
0   2   Sham        TOC      Y
1   4  Geeta      MATHS      A
2   5  Meena         WT      B
```

```
print(pd.merge(df1,df2,on='subject_id',how='left'))               #left join
```

```
   id_x Name_x subject_id  id_y Name_y
0     1    Ram       DBMS   NaN    NaN
1     2   Sham        TOC   2.0      Y
2     3   Sita       SPOS   NaN    NaN
3     4  Geeta      MATHS   4.0      A
4     5  Meena         WT   5.0      B
```

```
print(pd.merge(df1,df2,on='subject_id',how='right'))            #right join
```

```
   id_x Name_x subject_id  id_y Name_y
0   NaN    NaN         DS     1      X
1   2.0   Sham        TOC     2      Y
2   NaN    NaN        IOT     3      Z
3   4.0  Geeta      MATHS     4      A
4   5.0  Meena         WT     5      B
```

```
print(pd.merge(df1,df2,on='subject_id',how='inner'))
```

```
   id_x Name_x subject_id  id_y Name_y
0     2   Sham        TOC     2      Y
1     4  Geeta      MATHS     4      A
2     5  Meena         WT     5      B
```

```
print(pd.merge(df1,df2,on='subject_id',how='outer'))
```

```
   id_x Name_x subject_id  id_y Name_y
0   1.0    Ram       DBMS   NaN    NaN
1   2.0   Sham        TOC   2.0      Y
2   3.0   Sita       SPOS   NaN    NaN
3   4.0  Geeta      MATHS   4.0      A
4   5.0  Meena         WT   5.0      B
```

```
      5   NaN    NaN         DS   1.0      X
      6   NaN    NaN        IOT   3.0      Z
```

```
import pandas as pd
df1 = pd.DataFrame({
   'id':[1,2,3,4,5],
   'Name': ['Ram', 'Sham', 'Sita', 'Geeta', 'Meena'],
   'subject_id':['DBMS','TOC','SPOS','MATHS','WT']})
df2 = pd.DataFrame(
   {'id':[1,2,3,4,5],
   'Name': ['X', 'Y', 'Z', 'A', 'B'],
   'subject_id':['DS','DELD','IOT','MATHS','WT']})
```

```
print(pd.concat([df1,df2]))
```

```
        id   Name subject_id
     0   1    Ram       DBMS
     1   2   Sham        TOC
     2   3   Sita       SPOS
     3   4  Geeta      MATHS
     4   5  Meena         WT
     0   1      X         DS
     1   2      Y       DELD
     2   3      Z        IOT
     3   4      A      MATHS
     4   5      B         WT
```

```
print(pd.concat([df1,df2],keys=['x','y'])) #to associate specific keys
```

```
          id   Name subject_id
     x 0   1    Ram       DBMS
       1   2   Sham        TOC
       2   3   Sita       SPOS
       3   4  Geeta      MATHS
       4   5  Meena         WT
     y 0   1      X         DS
       1   2      Y       DELD
       2   3      Z        IOT
       3   4      A      MATHS
       4   5      B         WT
```

```
print(pd.concat([df1,df2],keys=['x','y'], ignore_index=True)) #ignore_index to True.
```

```
        id   Name subject_id
     0   1    Ram       DBMS
     1   2   Sham        TOC
     2   3   Sita       SPOS
     3   4  Geeta      MATHS
     4   5  Meena         WT
     5   1      X         DS
     6   2      Y       DELD
     7   3      Z        IOT
     8   4      A      MATHS
     9   5      B         WT
```

```
#If two objects need to be added along axis=1, then the new columns will be appended
```

```
print(pd.concat([df1,df2],axis=1))
```

```
        id   Name subject_id  id Name subject_id
     0   1    Ram       DBMS   1    X         DS
     1   2   Sham        TOC   2    Y       DELD
     2   3   Sita       SPOS   3    Z        IOT
     3   4  Geeta      MATHS   4    A      MATHS
     4   5  Meena         WT   5    B         WT
```

```
import pandas as pd
df = pd.DataFrame([[10, 20, 30, 40], [7, 14, 21, 28], [55, 15, 8, 12],
                   [15, 14, 1, 8], [7, 1, 1, 8], [5, 4, 9, 2]],
                  columns=['Apple', 'Orange', 'Banana', 'Pear'],
                  index=['Basket1', 'Basket2', 'Basket3', 'Basket4',
                         'Basket5', 'Basket6'])
print(df)
```

```
            Apple  Orange  Banana  Pear
     Basket1     10      20      30    40
     Basket2      7      14      21    28
```

```
Basket3     55      15       8    12
Basket4     15      14       1     8
Basket5      7       1       1     8
Basket6      5       4       9     2
```

```
print(df.Apple.mid_range())
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-3-e5aff4aaffac> in <module>()
----> 1 print(df.Apple.mid_range())

/usr/local/lib/python3.7/dist-packages/pandas/core/generic.py in __getattr__(self,
name)
   5485         ):
   5486             return self[name]
-> 5487         return object.__getattribute__(self, name)
   5488
   5489     def __setattr__(self, name: str, value) -> None:

AttributeError: 'Series' object has no attribute 'mid_range'
```

SEARCH STACK OVERFLOW

```
print("\n----------- Calculate Mean -----------\n")
print(df.mean())

print("\n----------- Calculate Median -----------\n")
print(df.median())

print("\n----------- Calculate Mode -----------\n")
print(df.mode())
```

```
----------- Calculate Mean -----------

Apple     16.500000
Orange    11.333333
Banana    11.666667
Pear      16.333333
dtype: float64

----------- Calculate Median -----------

Apple      8.5
Orange    14.0
Banana     8.5
Pear      10.0
dtype: float64

----------- Calculate Mode -----------

   Apple  Orange  Banana  Pear
0      7      14       1     8
```