```
import pandas as pd
```

The pd.merge() function implements a number of types of joins: the one-to-one,many-to-one, and many-to-many joins.

All three types of joins are accessed via an identical call to the pd.merge()

## One-to-one joins

simplest type of merge expression is the one-to-one join, which is very similar to the column-wise concatenation

```
df1 = pd.DataFrame({'employee': ['Bob', 'Jake', 'Lisa', 'Sue'],
'group': ['Accounting', 'Engineering', 'Engineering', 'HR']})
df2 = pd.DataFrame({'employee': ['Lisa', 'Bob', 'Jake', 'Sue'],
'hire_date': [2004, 2008, 2012, 2014]})

df3 = pd.merge(df1, df2)

df3
```

```
  employee        group  hire_date
0      Bob   Accounting       2008
1     Jake  Engineering       2012
2     Lisa  Engineering       2004
3      Sue           HR       2014
```

## Many-to-one joins

Many-to-one joins are joins in which one of the two key columns contains duplicate entries. For the many-to-one case, the resulting DataFrame will preserve those duplicate entries as appropriate

```
df4 = pd.DataFrame({'group': ['Accounting', 'Engineering', 'HR'],
'supervisor': ['Carly', 'Guido', 'Steve']})
print(df3);
```

```
  employee        group  hire_date
0      Bob   Accounting       2008
1     Jake  Engineering       2012
2     Lisa  Engineering       2004
3      Sue           HR       2014
```

```
print(df4);
```

```
         group supervisor
0   Accounting      Carly
1  Engineering      Guido
2           HR      Steve
```

```
print(pd.merge(df3, df4))
```

```
  employee        group  hire_date supervisor
0      Bob   Accounting       2008      Carly
1     Jake  Engineering       2012      Guido
2     Lisa  Engineering       2004      Guido
3      Sue           HR       2014      Steve
```

## Many-to-many joins

If the key column in both the left and right array contains duplicates, then the result is a many-to-many merge.

Consider the following, where we have a DataFrame showing one or more skills associated with a particular group.

```
df5 = pd.DataFrame({'group': ['Accounting', 'Accounting',
'Engineering', 'Engineering', 'HR', 'HR'],'skills': ['math',
'spreadsheets', 'coding', 'linux',
'spreadsheets', 'organization']})
print(df1);
```

```
  employee        group
0      Bob   Accounting
1     Jake  Engineering
2     Lisa  Engineering
3      Sue           HR
```

```
print(df5);
```

```
         group        skills
0   Accounting          math
1   Accounting  spreadsheets
2  Engineering        coding
3  Engineering         linux
4           HR  spreadsheets
5           HR  organization
```

```
print(pd.merge(df1, df5))
```

```
  employee        group        skills
0      Bob   Accounting          math
1      Bob   Accounting  spreadsheets
2     Jake  Engineering        coding
3     Jake  Engineering         linux
4     Lisa  Engineering        coding
5     Lisa  Engineering         linux
6      Sue           HR  spreadsheets
7      Sue           HR  organization
```

## Basic Merge Using a Dataframe Column

```
import pandas as pd
df1 = pd.DataFrame({
    "city": ["new york","chicago","orlando"],
    "temperature": [21,14,35],
})
df1
```

```
       city  temperature
0  new york           21
1   chicago           14
2   orlando           35
```

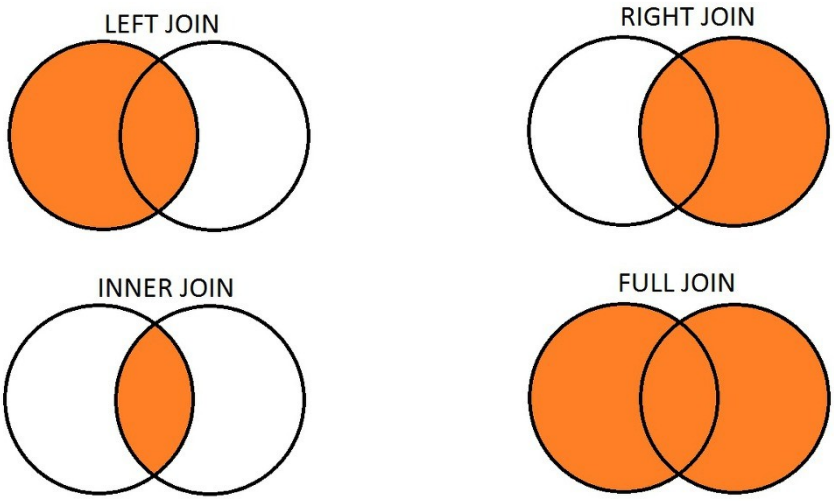
```
df2 = pd.DataFrame({
    "city": ["chicago","new york","orlando"],
    "humidity": [65,68,75],
})
df2
```

```
       city  humidity
0   chicago        65
1  new york        68
2   orlando        75
```

```
df3 = pd.merge(df1, df2, on="city")
df3
```

```
       city  temperature  humidity
0  new york           21        68
1   chicago           14        65
2   orlando           35        75
```

## Type Of DataBase Joins




```
df1 = pd.DataFrame({
    "city": ["new york","chicago","orlando", "baltimore"],
    "temperature": [21,14,35, 38],
})
df1
```

```
        city  temperature
0   new york           21
1    chicago           14
2    orlando           35
3  baltimore           38
```

```
df2 = pd.DataFrame({
    "city": ["chicago","new york","san diego"],
    "humidity": [65,68,71],
})
df2
```

```
        city  humidity
0    chicago        65
1   new york        68
2  san diego        71
```

```
df3=pd.merge(df1,df2,on="city",how="inner")
df3
```

```
       city  temperature  humidity
0  new york           21        68
1   chicago           14        65
```

```
df3=pd.merge(df1,df2,on="city",how="outer")
df3
```

```
        city  temperature  humidity
0   new york         21.0      68.0
1    chicago         14.0      65.0
2    orlando         35.0       NaN
3  baltimore         38.0       NaN
4  san diego          NaN      71.0
```

```
df3=pd.merge(df1,df2,on="city",how="left")
df3
```

```
        city  temperature  humidity
0   new york           21      68.0
1    chicago           14      65.0
2    orlando           35       NaN
3  baltimore           38       NaN
```

```
df3=pd.merge(df1,df2,on="city",how="right")
df3
```

```
        city  temperature  humidity
0    chicago         14.0        65
1   new york         21.0        68
2  san diego          NaN        71
```

## indicator flag

```
df3=pd.merge(df1,df2,on="city",how="outer",indicator=True)
df3
```

```
        city  temperature  humidity      _merge
0   new york         21.0      68.0        both
1    chicago         14.0      65.0        both
2    orlando         35.0       NaN   left_only
3  baltimore         38.0       NaN   left_only
4  san diego          NaN      71.0  right_only
```

## suffixes

```
df1 = pd.DataFrame({
    "city": ["new york","chicago","orlando", "baltimore"],
    "temperature": [21,14,35,38],
    "humidity": [65,68,71, 75]
})
df1
```

```
        city  temperature  humidity
0   new york           21        65
1    chicago           14        68
```

```
2    orlando           35        71
3  baltimore           38        75
```

```
df2 = pd.DataFrame({
    "city": ["chicago","new york","san diego"],
    "temperature": [21,14,35],
    "humidity": [65,68,71]
})
df2
```

```
        city  temperature  humidity
0    chicago           21        65
1   new york           14        68
2  san diego           35        71
```

```
df3= pd.merge(df1,df2,on="city",how="outer",
suffixes=('_first','_second'))
df3
```

```
        city  temperature_first  ...  temperature_second
humidity_second
0   new york               21.0  ...                14.0
68.0
1    chicago               14.0  ...                21.0
65.0
2    orlando               35.0  ...                 NaN
NaN
3  baltimore               38.0  ...                 NaN
NaN
4  san diego                NaN  ...                35.0
71.0

[5 rows x 5 columns]
```

## join

```
df1 = pd.DataFrame({
    "city": ["new york","chicago","orlando"],
    "temperature": [21,14,35],
})
df1.set_index('city',inplace=True)
df1
```

```
          temperature
city
new york           21
chicago            14
orlando            35
```

```
df2 = pd.DataFrame({
    "city": ["chicago","new york","orlando"],
```

```
    "humidity": [65,68,75],
})
df2.set_index('city',inplace=True)
df2
```

| city | humidity |
|---|---|
| chicago | 65 |
| new york | 68 |
| orlando | 75 |

```
df1.join(df2,lsuffix='_l', rsuffix='_r')
```

| city | temperature | humidity |
|---|---|---|
| new york | 21 | 68 |
| chicago | 14 | 65 |
| orlando | 35 | 75 |