

```
In [1]: from sklearn import datasets
from sklearn.datasets import load_iris
import pandas as pd
```

```
In [2]: iris_df = pd.DataFrame(load_iris().data, columns = load_iris().feature_names)
print("iris dataframe: ")
iris_df
```

iris dataframe:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

```
In [3]: print("iris target dataframe:")
specimen_class= pd.DataFrame(load_iris().target,columns=['specimen_class'])
specimen_class
```

iris target dataframe:

	specimen_class
0	0
1	0
2	0
3	0
4	0
...	...
145	2
146	2
147	2
148	2
149	2

150 rows × 1 columns

```
In [4]: print("Concatenating both the dataframes - columnwise: ")
df= pd.concat([iris_df, specimen_class],axis=1)
df
```

Concatenating both the dataframes - columnwise:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	specimen_class
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

```
In [5]: # CHECKING HEADERS OF CURRENT DATAFRAME
df.columns
```

```
Out[5]: Index(['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
             'petal width (cm)', 'specimen_class'],
             dtype='object')
```

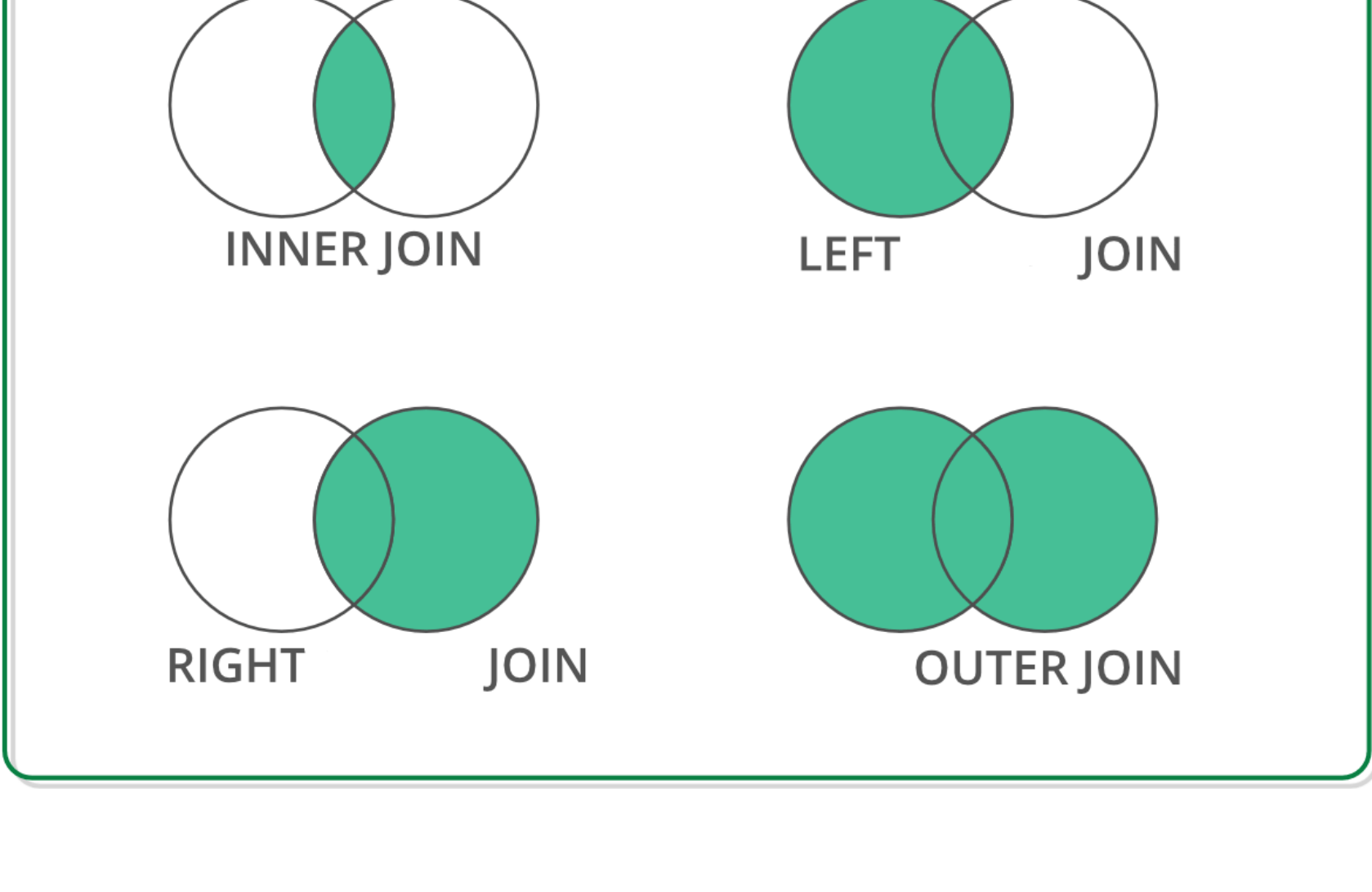
```
In [6]: # renaming the columns for ease of use
df.columns=['sepal_len','sepal_wid','petal_len','petal_wid','class']
```

```
In [7]: df
```

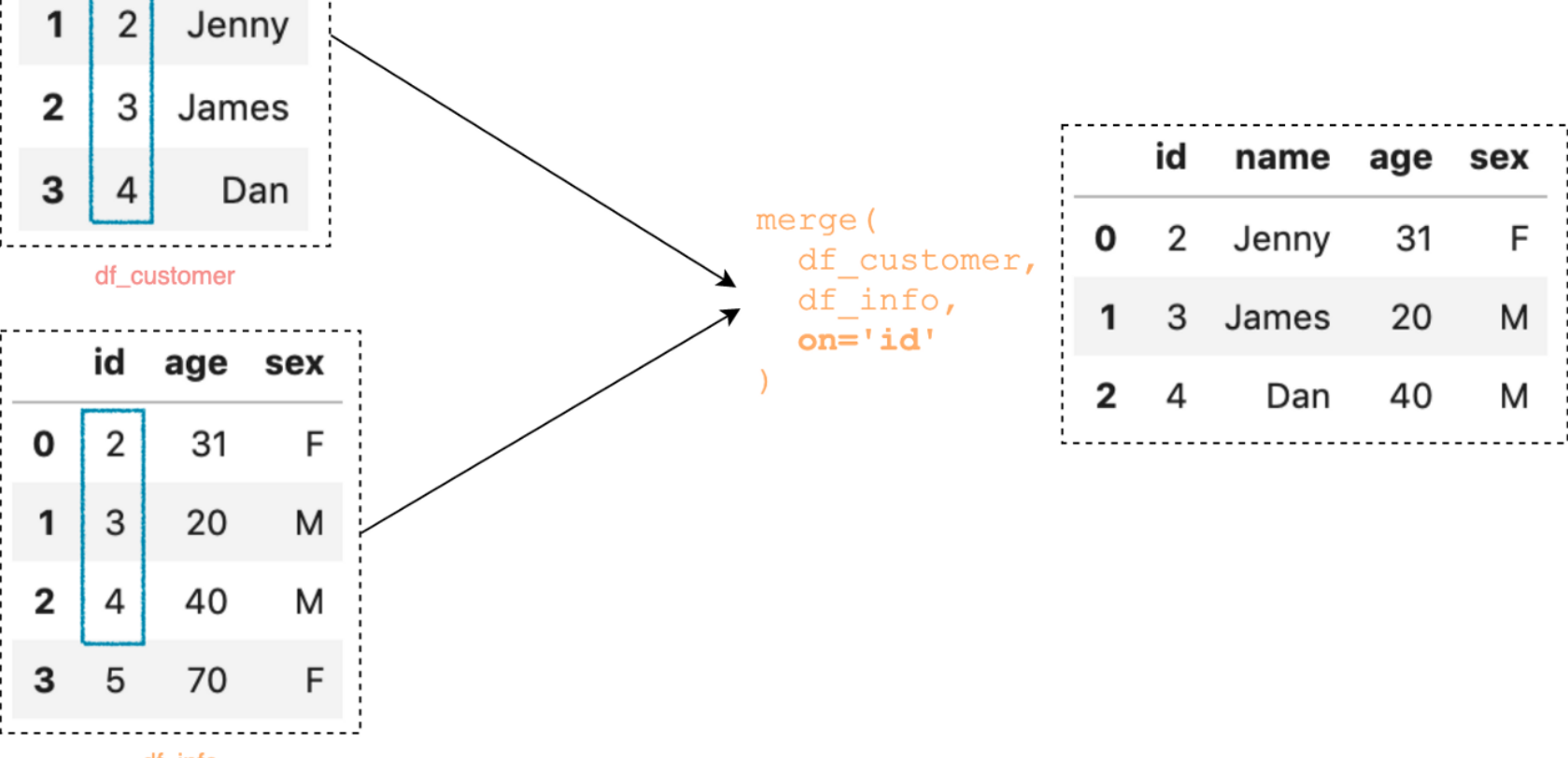
	sepal_len	sepal_wid	petal_len	petal_wid	class
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

MERGING DATASETS WITH DIFFERENT TYPES OF JOINS: INNER JOIN(DEFAULT), LEFT JOIN, RIGHT JOIN & OUTER JOIN



INNER JOIN (DEFAULT) --> HOW DOES IT WORK?



```
In [8]: # Let's split the dataframe in the middle (75-75) and merge it on 'class' by 'inner join'(default)
df1 = df[:75]
df1
```

	sepal_len	sepal_wid	petal_len	petal_wid	class
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...
70	5.9	3.2	4.8	1.8	1
71	6.1	2.8	4.0	1.3	1
72	6.3	2.5	4.9	1.5	1
73	6.1	2.8	4.7	1.2	1
74	6.4	2.9	4.3	1.3	1

75 rows × 5 columns

```
In [9]: df2=df[75:]
df2
```

	sepal_len	sepal_wid	petal_len	petal_wid	class
75	6.6	3.0	4.4	1.4	1
76	6.8	2.8	4.8	1.4	1
77	6.7	3.0	5.0	1.7	1
78	6.0	2.9	4.5	1.5	1
79	5.7	2.6	3.5	1.0	1
...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

75 rows × 5 columns

```
In [11]: merged_df_inner_join=pd.merge(df1,df2,on='class')
merged_df_inner_join
```

	sepal_len_x	sepal_wid_x	petal_len_x	petal_wid_x	class	sepal_len_y	sepal_wid_y	petal_len_y	petal_wid_y
0	7.0	3.2	4.7	1.4	1	6.6	3.0	4.4	1.4
1	7.0	3.2	4.7	1.4	1	6.8	2.8	4.8	1.4
2	7.0	3.2	4.7	1.4	1	6.7	3.0	5.0	1.7
3	7.0	3.2	4.7	1.4	1	6.0	2.9	4.5	1.5
4	7.0	3.2	4.7	1.4	1	5.7	2.6	3.5	1.0
...
620	6.4	2.9	4.3	1.3	1	5.7	3.0	4.2	1.2
621	6.4	2.9	4.3	1.3	1	5.7	2.9	4.2	1.3
622	6.4	2.9	4.3	1.3	1	6.2	2.9	4.3	1.3
623	6.4	2.9	4.3	1.3	1	5.1	2.5	3.0	1.1
624	6.4	2.9	4.3	1.3	1	5.7	2.8	4.1	1.3

625 rows × 9 columns

```
In [12]: merged_df_leftjoin=pd.merge(df1,df2,on='class',how='left')
merged_df_leftjoin
```

	sepal_len_x	sepal_wid_x	petal_len_x	petal_wid_x	class	sepal_len_y	sepal_wid_y	petal_len_y	petal_wid_y
0	5.1	3.5	1.4	0.2	0	NaN	NaN	NaN	NaN
1	4.9	3.0	1.4	0.2	0	NaN	NaN	NaN	NaN
2	4.7	3.2	1.3	0.2	0	NaN	NaN	NaN	NaN
3	4.6	3.1	1.5	0.2	0	NaN	NaN	NaN	NaN
4	5.0	3.6	1.4	0.2	0	NaN	NaN	NaN	NaN
...
670	6.4	2.9	4.3	1.3	1	5.7	3.0	4.2	1.2
671	6.4	2.9	4.3	1.3	1	5.7	2.9	4.2	1.3
672	6.4	2.9	4.3	1.3	1	6.2	2.9	4.3	1.3
673	6.4	2.9	4.3	1.3	1	5.1	2.5	3.0	1.1
674	6.4	2.9	4.3	1.3	1	5.7	2.8	4.1	1.3

675 rows × 9 columns

```
In [13]: merged_df_rightjoin=pd.merge(df1,df2,on='class',how='right')
merged_df_rightjoin
```

	sepal_len_x	sepal_wid_x	petal_len_x	petal_wid_x	class	sepal_len_y	sepal_wid_y	petal_len_y	petal_wid_y
0	7.0	3.2	4.7	1.4	1	6.6	3.0	4.4	1.4
1	6.4	3.2	4.5	1.5	1	6.6	3.0	4.4	1.4
2	6.9	3.1	4.9	1.5	1	6.6	3.0	4.4	1.4
3	5.5	2.3	4.0	1.3	1	6.6	3.0	4.4	1.4
4	6.5	2.8	4.6	1.5	1	6.6	3.0	4.4	1.4
...
670	NaN	NaN	NaN	NaN	2	6.7	3.0	5.2	2.3
671	NaN	NaN	NaN	NaN	2	6.3	2.5	5.0	1.9
672	NaN	NaN	NaN	NaN	2	6.5	3.0	5.2	2.0
673	NaN	NaN	NaN	NaN	2	6.2	3.4	5.4	2.3
674	NaN	NaN	NaN	NaN	2	5.9	3.0	5.1	1.8

675 rows × 9 columns

```
In [14]: merged_df_outerjoin=pd.merge(df1,df2,on='class',how='outer')
merged_df_outerjoin
```

	sepal_len_x	sepal_wid_x	petal_len_x	petal_wid_x	class	sepal_len_y	sepal_wid_y	petal_len_y	petal_wid_y
0	5.1	3.5	1.4	0.2	0	NaN	NaN	NaN	NaN
1	4.9	3.0	1.4	0.2	0	NaN	NaN	NaN	NaN
2	4.7	3.2	1.3	0.2	0	NaN	NaN	NaN	NaN
3	4.6	3.1	1.5	0.2	0	NaN	NaN	NaN	NaN
4	5.0	3.6	1.4	0.2	0	NaN	NaN	NaN	NaN
...
720	NaN	NaN	NaN	NaN	2	6.7	3.0	5.2	2.3
721	NaN	NaN	NaN	NaN	2	6.3	2.5	5.0	1.9
722	NaN	NaN	NaN	NaN	2	6.5	3.0	5.2	2.0
723	NaN	NaN	NaN	NaN	2	6.2	3.4	5.4	2.3
724	NaN	NaN	NaN	NaN	2	5.9	3.0	5.1	1.8

725 rows × 9 columns