Combining DataFrame

Concatenation

Use `pd.concat()` for Combines DataFrames along rows or columns.

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

df1

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

+

df2

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

Combining DataFrame

- Join Row wise with *axis=1*
- Empty values are filled with NaN

```
pd.concat([df1,df2,df3],axis=1)
```

	A	B	C	D	A	B	C	D
0	A0	B0	C0	D0	NaN	NaN	NaN	NaN
1	A1	B1	C1	D1	NaN	NaN	NaN	NaN
2	A2	B2	C2	D2	NaN	NaN	NaN	NaN
3	A3	B3	C3	D3	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	A4	B4	C4	D4
5	NaN	NaN	NaN	NaN	A5	B5	C5	D5
6	NaN	NaN	NaN	NaN	A6	B6	C6	D6
7	NaN	NaN	NaN	NaN	A7	B7	C7	D7

Combining DataFrame

Merging

The **merge** function allows you to merge DataFrames together using logic similar to merging SQL Tables

```
pd.merge(left,right,how='inner',on='key')
```

	A	B	key
0	A0	B0	K0
1	A1	B1	K1
2	A2	B2	K2
3	A3	B3	K3

+

	C	D	key
0	C0	D0	K0
1	C1	D1	K1
2	C2	D2	K2
3	C3	D3	K3

	A	B	key	C	D
0	A0	B0	K0	C0	D0
1	A1	B1	K1	C1	D1
2	A2	B2	K2	C2	D2
3	A3	B3	K3	C3	D3

Default merge type is `inner`, but *how* can be set to `left`, `right`, `outer`, etc.

Combining DataFrame

on can be list of keys as well

```
left = pd.DataFrame({'key1': ['K0', 'K0', 'K1', 'K2'],
                     'key2': ['K0', 'K1', 'K0', 'K1'],
                        'A': ['A0', 'A1', 'A2', 'A3'],
                        'B': ['B0', 'B1', 'B2', 'B3']})

right = pd.DataFrame({'key1': ['K0', 'K1', 'K1', 'K2'],
                     'key2': ['K0', 'K0', 'K0', 'K0'],
                        'C': ['C0', 'C1', 'C2', 'C3'],
                        'D': ['D0', 'D1', 'D2', 'D3']})
```

```
pd.merge(left, right, on=['key1', 'key2'])
```

	A	B	key1	key2	C	D
0	A0	B0	K0	K0	C0	D0
1	A2	B2	K1	K0	C1	D1
2	A2	B2	K1	K0	C2	D2

Combining DataFrame

Joining

Convenient when combining DataFrames on index rather than column keys.

```
left = pd.DataFrame({'A': ['A0', 'A1', 'A2'],
                     'B': ['B0', 'B1', 'B2']},
                     index=['K0', 'K1', 'K2'])
right = pd.DataFrame({'C': ['C0', 'C2', 'C3'],
                      'D': ['D0', 'D2', 'D3']},
                      index=['K0', 'K2', 'K3'])
```

.join uses same logic as merge. But join is a methos

```
left.join(right, how='inner')
```

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2

Combining DataFrame

Summary Table

Method	Function	Description	Best For
Concatenation	`pd.concat()`	Stacks DataFrames along rows or columns	Appending data
Merging	`pd.merge()`	SQL-like joins based on columns (keys)	Row alignment on common columns
Joining	`df1.join(df2)`	Joins based on DataFrame indexes	Index alignment without columns