Concatenating, Merging and Joining DataFrames in Pandas

1. Concat
2. Append
3. Merge
4. Join

Let's undestand concatenating

```
# Let's create two dummy data frames to understand
import pandas as pd

df1 = pd.DataFrame({'item': ['A', 'B', 'C', 'D'],
                    'value': [1, 2, 3, 5]})

df2 = pd.DataFrame({'item': ['E', 'F', 'G', 'H'],
                   'value': [5, 6, 7, 8]})

print(df1)
print(df2)
```

```
  item  value
0    A      1
1    B      2
2    C      3
3    D      5
  item  value
0    E      5
1    F      6
2    G      7
3    H      8
```

```
df1
```

	item	value
0	A	1
1	B	2
2	C	3
3	D	5

```
# Let's stack them vertically by using pd.concat

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

	item	value
0	A	1
1	B	2
2	C	3
3	D	5
0	E	5
1	F	6
2	G	7

```
# What if they had different column names

df1 = pd.DataFrame({'item': ['A', 'B', 'C', 'D'],
                    'value': [1, 2, 3, 5]})

df2 = pd.DataFrame({'item': ['E', 'F', 'G', 'H'],
                   'quanity': [2, 2, 1, 5]})

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

	item	value	quanity
0	A	1.0	NaN
1	B	2.0	NaN
2	C	3.0	NaN
3	D	5.0	NaN
0	E	NaN	2.0
1	F	NaN	2.0
2	G	NaN	1.0
3	H	NaN	5.0

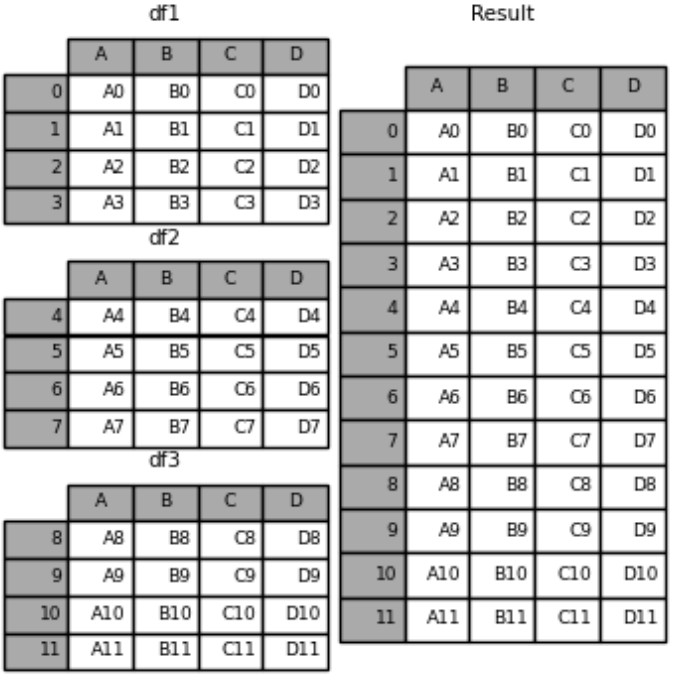
```
# What if they had duplicates in items

df1 = pd.DataFrame({'item': ['A', 'B', 'C', 'D'],
                    'value': [1, 2, 3, 5]})

df2 = pd.DataFrame({'item': ['D', 'F', 'G', 'H'],
                   'quanity': [2, 2, 1, 5]})

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

	index	item	value	quanity
0	0	A	1.0	NaN
1	1	B	2.0	NaN
2	2	C	3.0	NaN
3	3	D	5.0	NaN
4	0	D	NaN	2.0

```
# We can use axis = 1 to stack horizontally

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

	item	value	item	quanity
0	A	1	D	2
1	B	2	F	2
2	C	3	G	1
3	D	5	H	5

Vertical Concat



Horizontal Conat

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

df4

	B	D	F
2	B2	D2	F2
3	B3	D3	F3
6	B6	D6	F6
7	B7	D7	F7

Result

	A	B	C	D	B	D	F
0	A0	B0	C0	D0	NaN	NaN	NaN
1	A1	B1	C1	D1	NaN	NaN	NaN
2	A2	B2	C2	D2	B2	D2	F2
3	A3	B3	C3	D3	B3	D3	F3
6	NaN	NaN	NaN	NaN	B6	D6	F6
7	NaN	NaN	NaN	NaN	B7	D7	F7

Similarly we can use Append to add multiple dataFrames onto another

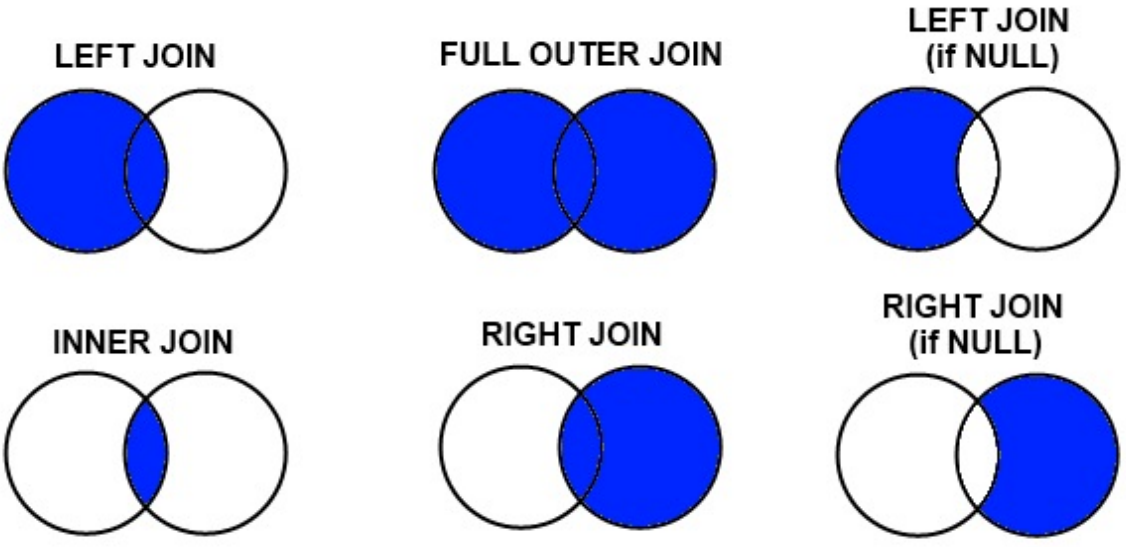
```
df1 = pd.DataFrame({'item': ['A', 'B', 'C', 'D'],
                    'value': [1, 2, 3, 5]})

df2 = pd.DataFrame({'item': ['D', 'F', 'G', 'H'],
                    'quanity': [2, 2, 1, 5]})

df3 = pd.DataFrame({'item': ['I', 'J', 'K', 'L'],
                    'quanity': [3, 4, 7, 25]})

df1.append([df2, df3])
```

	item	value	quanity
0	A	1.0	NaN
1	B	2.0	NaN
2	C	3.0	NaN
3	D	5.0	NaN
0	D	NaN	2.0
1	F	NaN	2.0
2	G	NaN	1.0
3	H	NaN	5.0
0	I	NaN	3.0
1	J	NaN	4.0
2	K	NaN	7.0
3	L	NaN	25.0

Pandas merge

Pandas has full-featured, high performance in-memory join operations idiomatically very similar to relational databases like SQL:

- **one-to-one joins:** for example when joining two DataFrame objects on their indexes (which must contain unique values).
- **many-to-one joins:** for example when joining an index (unique) to one or more columns in a different DataFrame.
- **many-to-many** joins: joining columns on columns.




```
dfA = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3'],
                    'A': ['A0', 'A1', 'A2', 'A3'],
                    'B': ['B0', 'B1', 'B2', 'B3']})

dfB = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3'],
                      'C': ['C0', 'C1', 'C2', 'C3'],
                      'D': ['D0', 'D1', 'D2', 'D3']})
dfA
```

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

```
dfB
```

	key	C	D
0	K0	C0	D0

```
# Merging on unique keys
# by default we're doing an inner join

pd.merge(dfA, dfB, on='key')
```

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

```
 # Understand the difference between left and right joins

dfA = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3'],
                    'A': ['A0', 'A1', 'A2', 'A3'],
                    'B': ['B0', 'B1', 'B2', 'B3']})

dfB = pd.DataFrame({'key': ['K4', 'K1', 'K2', 'K3'],
                      'C': ['C0', 'C1', 'C2', 'C3'],
                      'D': ['D0', 'D1', 'D2', 'D3']})

 pd.merge(dfA, dfB, how='left', on='key')
```

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

```
# Inner Joins

pd.merge(dfA, dfB, how='inner', on='key')
```

	key	A	B	C	D
0	K1	A1	B1	C1	D1
1	K2	A2	B2	C2	D2
2	K3	A3	B3	C3	D3

```
# Outer Joins

pd.merge(dfA, dfB, how='outer', on='key')
```

	key	A	B	C	D
0	K0	A0	B0	NaN	NaN
1	K1	A1	B1	C1	D1
2	K2	A2	B2	C2	D2
3	K3	A3	B3	C3	D3
4	K4	NaN	NaN	C0	D0

Joining

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

left
```

	A	B
K0	A0	B0
K1	A1	B1
K2	A2	B2

```
right = pd.DataFrame({'C': ['C0', 'C2', 'C3'],
                      'D': ['D0', 'D2', 'D3']},
                     index=['K0', 'K2', 'K3'])

right
```

	C	D
K0	C0	D0
K2	C2	D2
K3	C3	D3

```
left.join(right)
```

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

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

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

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2
K3	NaN	NaN	C3	D3