

# Merging, Joining, and Concatenating







# Merging, Joining, and Concatenating

There are 3 main ways of combining DataFrames together: Merging, Joining and Concatenating. In this lecture we will discuss these 3 methods with examples.

## **Example DataFrames**

Python-Data-Science-and X

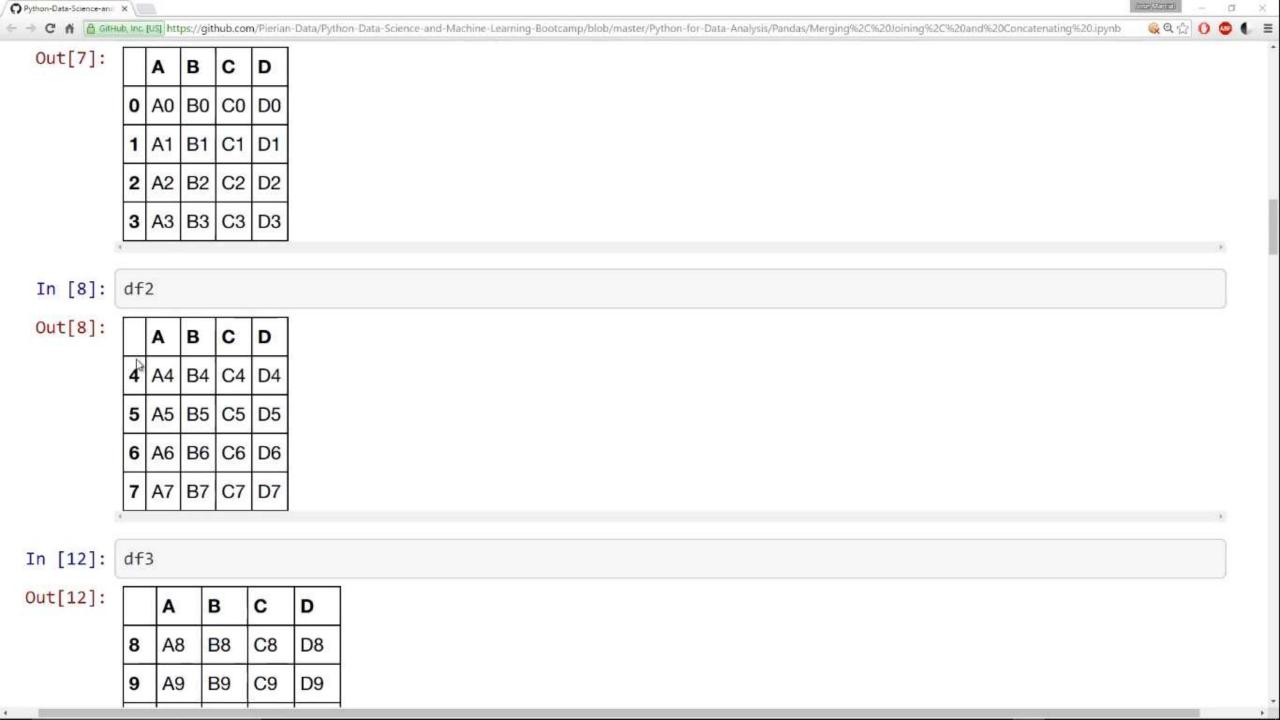
Python-Data-Science-and X





### Example DataFrames

```
In [3]: import pandas as pd
In [4]: df1 = pd.DataFrame({'A': ['A0', 'A1', 'A2', 'A3'],
                                'B': ['B0', 'B1', 'B2', 'B3'],
                                 'C': ['C0', 'C1', 'C2', 'C3'],
                                 'D': ['D0', 'D1', 'D2', 'D3']},
                                index=[0, 1, 2, 3])
In [5]: df2 = pd.DataFrame({'A': ['A4', 'A5', 'A6', 'A7'],
                                 'B': ['B4', 'B5', 'B6', 'B7'],
                                 'C': ['C4', 'C5', 'C6', 'C7'],
                                 'D': ['D4', 'D5', 'D6', 'D7']},
                                 index=[4, 5, 6, 7])
In [6]: df3 = pd.DataFrame({'A': ['A8', 'A9', 'A10', 'A11'],
                                 'B': ['B8', 'B9', 'B10', 'B11'],
                                 'C': ['C8', 'C9', 'C10', 'C11'],
                                 'D': ['D8', 'D9', 'D10', 'D11']},
                                index=[8, 9, 10, 11])
In [7]:
        df1
```





10	A10	B10	C10	D10
11	A11	B11	C11	D11

#### Concatenation

Concatenation basically glues together DataFrames. Keep in mind that dimensions should match along the axis you are concatenating on. You can use **pd.concat** and pass in a list of DataFrames to concatenate together:

In [10]: pd.concat([df1,df2,df3])

#### Out[10]:

	Α	В	С	D
0	A0	В0	CO	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	АЗ	ВЗ	СЗ	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8

concatenating on. You can use pd.concat and pass in a list of DataFrames to concatenate together:







pd.concat([df1,df2,df3]) In [10]:

Out[10]:

	Α	В	С	D
0	A0	В0	CO	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	АЗ	ВЗ	СЗ	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6 D7
7	A7	B7	C7	
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

pd.concat([df1,df2,df3],axis=1) In [18]:



10	AIV	טוט	010	טוטן
11	A11	B11	C11	D11

In [18]: pd.cohcat([df1,df2,df3],axis=1)

Out[18]:

	Α	В	С	D	A	В	С	D	A	В	С	D
0	A0	ВО	C0	D0	NaN							
1	A1	B1	C1	D1	NaN							
2	A2	B2	C2	D2	NaN							
3	АЗ	ВЗ	СЗ	D3	NaN							
4	NaN	NaN	NaN	NaN	A4	B4	C4	D4	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	A5	B5	C5	D5	NaN	NaN	NaN	NaN
6	NaN	NaN	NaN	NaN	A6	B6	C6	D6	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	A7	B7	C7	D7	NaN	NaN	NaN	NaN
8	NaN	A8	B8	C8	D8							
9	NaN	A9	B9	C9	D9							
10	NaN	A10	B10	C10	D10							
11	NaN	A11	B11	C11	D11							

Make sure that information lining up correctly when you are joining along the axis.



# **Example DataFrames**

```
In [28]: left = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3'],
                              'A': ['A0', 'A1', 'A2', 'A3'],
                              'B': ['B0', 'B1', 'B2', 'B3']})
         right = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3'],
                                   'C': ['C0', 'C1', 'C2', 'C3'],
                                   'D': ['D0', 'D1', 'D2', 'D3']})
```

```
In [29]: left
```

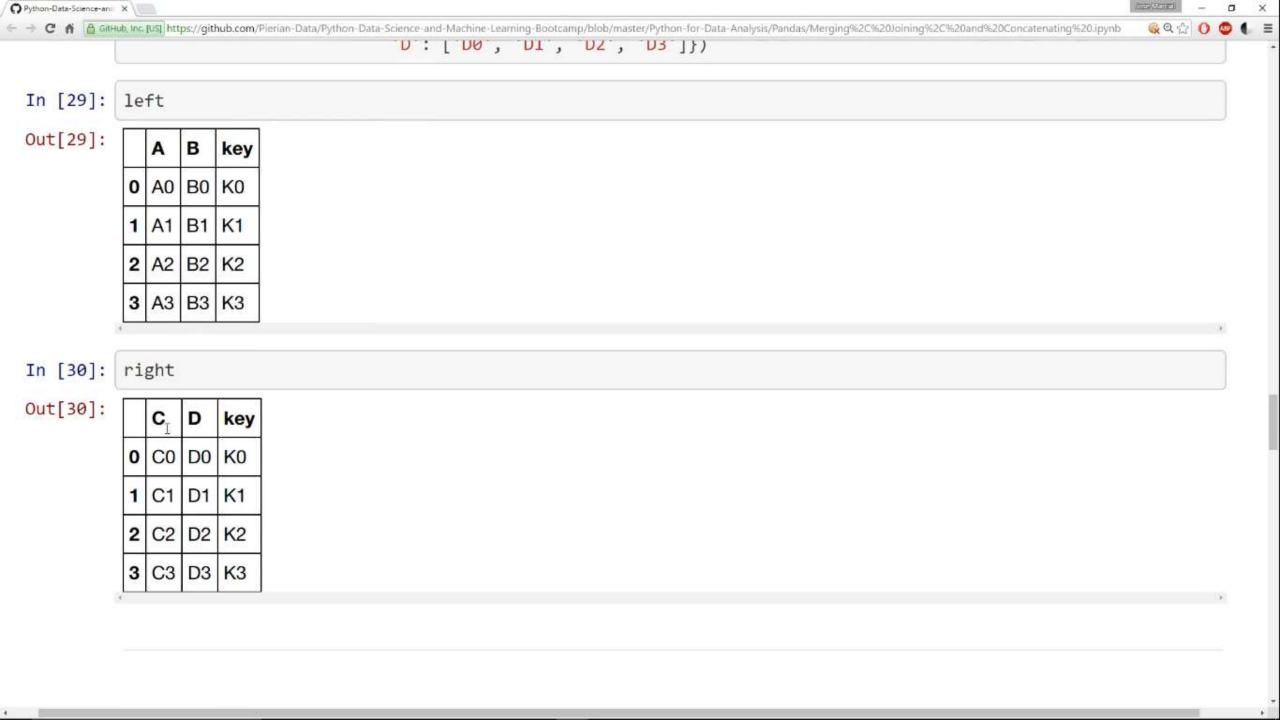
Out[29]: A B key

	~	В	Key
0	A0	B0	K0
1	A1	B1	K1
2	A2	B2	K2
3	АЗ	ВЗ	КЗ

In [30]: right

Out[30]: C D key

	C	D	ke
0	C0	D0	K0
1	C1	D1	K1



# Merging

The **merge** function allows you to merge DataFrames together using a similar logic as merging SQL Tables together. For example:

```
In [35]:
         pd.merge(left,right,how='inner',on='key')
Out[35]:
                  key C
                         D
              В
         0 A0 B0 K0
                      C0 D0
         1 A1 B1 K1
                      C1 D1
         2 A2 B2 K2
                      C2 D2
                                          D
         3 A3 B3 K3 C3 D3
```

Or to show a more complicated example:

```
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'],
                                   101. [ 1001 1011 1021 1021]
```

#### Or to show a more complicated example:

In [39]: pd.merge(left, right, on=['key1', 'key2'])

Out[39]:

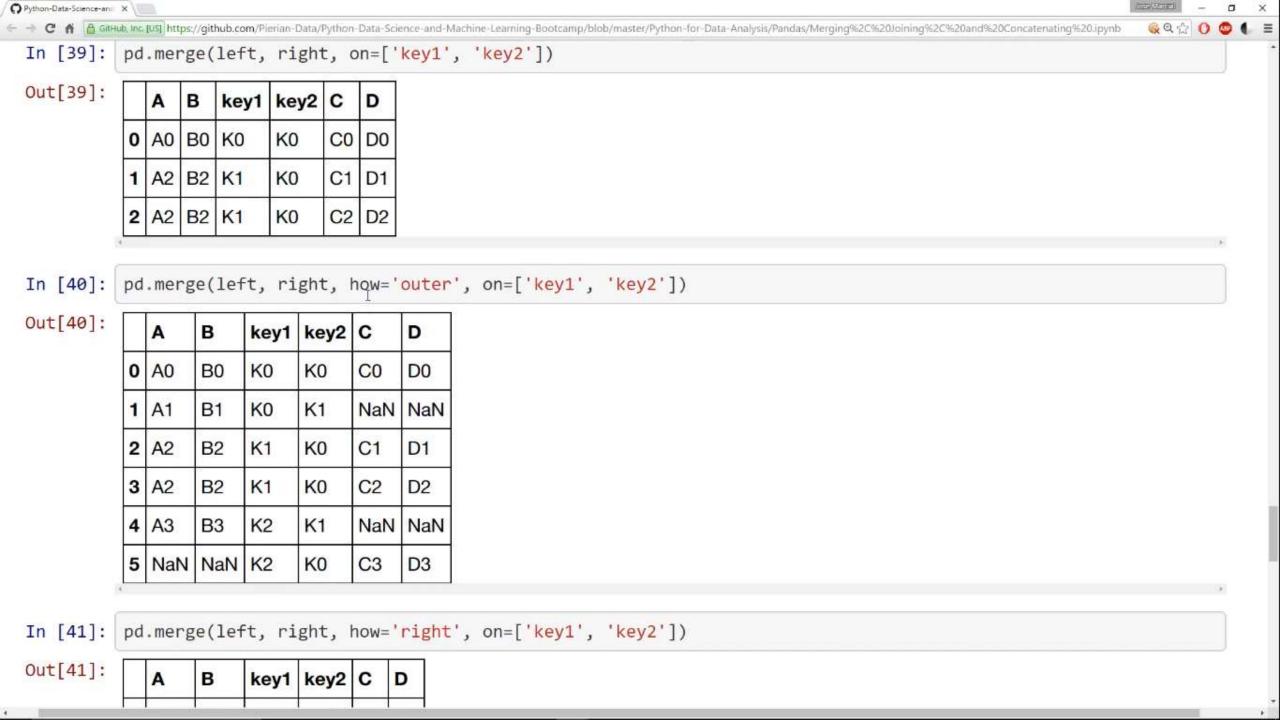
Python-Data-Science-and X

	Α	В	key1	key2	С	D
0	A0	ВО	K0	K0	C0	DO
1	A2	B2	K1	K0	C1	D1
2	A2	B2	K1	K0	C2	D2

In [40]: pd.merge(left, right, how='outer', on=['key1', 'key2'])

Out[40]:

	Α	В	key1	key2	С	D
0	ΔΩ	BO	KN	κn	CO	חח





#### In [40]: pd.merge(left, right, how='outer', on=['key1', 'key2'])

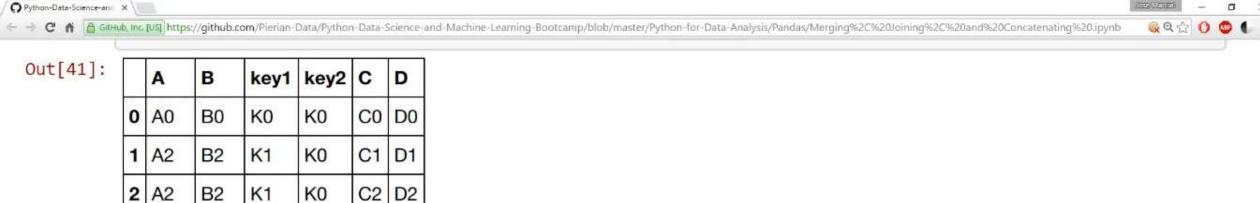
#### Out[40]:

	A	В	key1	key2	С	D
0	A0	B0	K0	K0	C0	D0
1	A1	B1	K0	K1	NaN	NaN
2	A2	B2	K1	K0	C1	D1
3	A2	B2	K1	K0	C2	D2
4	АЗ	ВЗ	K2	K1	NaN	NaN
5	NaN	NaN	K2	K0	СЗ	D3

#### Out[41]:

	A	В	key1	key2	С	D
0	A0	В0	K0	K0	C0	D0
1	A2	B2	K1	K0	C1	D1
2	A2	B2	K1	K0	C2	D2
3	NaN	NaN	K2	K0	СЗ	D3

```
In [42]: pd.merge(left, right, how='left', on=['key1', 'key2'])
```



In [42]: pd.merge(left, right, how='left', I on=['key1', 'key2'])

#### Out[42]:

	Α	В	key1	key2	С	D
0	A0	ВО	K0	K0	C0	D0
1	A1	В1	K0	K1	NaN	NaN
2	A2	B2	K1	K0	C1	D1
3	A2	B2	K1	K0	C2	D2
4	АЗ	ВЗ	K2	K1	NaN	NaN

D2

C3 D3

K<sub>0</sub>

# **Joining**

3 NaN NaN K2

Joining is a convenient method for combining the columns of two potentially differently-indexed DataFrames into a single result DataFrame.





#### @Q 0

# Joining

Joining is a convenient method for combining the columns of two potentially differently-indexed DataFrames into a single result DataFrame.

```
In [46]:
         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'])
In [47]:
         left.join(right)
Out[47]:
                 В
                    C
                         D
```

**KO** AO BO CO D<sub>0</sub> **K1** | A1 | B1 | NaN | NaN **K2** A2 B2 C2 D2

In [48]: left.join(right, how='outer')

Out[48]:

В D Joining is a convenient method for combining the columns of two potentially differently-indexed DataFrames into a single result DataFrame.

```
In [47]: left.join(right)
```

Out[47]:

O Python-Data-Science-and X

	A	В	С	D
K0	A0	ВО	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2

```
In [48]: left.join(right, how='outer')
```

Out[48]:

		A	В	С	D
7.0	K0	A0	В0	C0	D0
	K1	A1	B1	NaN	NaN

