

▼ Basics

REF - https://www.tutorialspoint.com/python_pandas/python_pandas_dataframe.htm

▼ Package

```
1 import pandas as pd
```

```
1 df = pd.DataFrame()  
2  
3 print (df)
```

```
[> Empty DataFrame  
   Columns: []  
   Index: []
```

▼ Simple data frames

▼ DF from List

```
1 import pandas as pd  
2  
3 mylist = [1,2,3,4,5]  
4 df = pd.DataFrame(mylist)  
5  
6 print (df)
```

```
[>      0  
0      1  
1      2  
2      3  
3      4  
4      5
```

▼ Two column DF

```
1 import pandas as pd  
2 data = [['Alex',10],['Bob',12],['Clarke',13]]  
3 df = pd.DataFrame(data,columns=['Name', 'Age'])  
4 print (df)
```

```
[>
```

	Name	Age
0	Alex	10

```
1 import pandas as pd
2 data = [['Alex',10],['Bob',12],['Clarke',13]]
3 df = pd.DataFrame(data,columns=['Name','Age'],dtype=float)
4 print (df)
```

```
☞      Name  Age
0    Alex  10.0
1     Bob  12.0
2  Clarke  13.0
```

▼ DF from Dictionary

```
1 import pandas as pd
2 data = {'Name':['Tom', 'Jack', 'Steve', 'Ricky'],'Age':[28,34,29,42]}
3 df = pd.DataFrame(data)
4 print (df)
```

```
☞      Name  Age
0     Tom   28
1    Jack   34
2   Steve   29
3   Ricky   42
```

▼ Indexed rows

```
1 import pandas as pd
2 data = {'Name':['Tom', 'Jack', 'Steve', 'Ricky'],'Age':[28,34,29,42]}
3 df = pd.DataFrame(data, index=['rank1','rank2','rank3','rank4'])
4 print (df)
```

```
☞      Name  Age
rank1    Tom   28
rank2   Jack   34
rank3  Steve   29
rank4  Ricky   42
```

▼ DF from List of Dictionaries

```
1 import pandas as pd
2 data = [{'a': 1, 'b': 2},{'a': 5, 'b': 10, 'c': 20}]
3 df = pd.DataFrame(data)
4 print (df)
```

```
☞      a  b  c
0  1  2 NaN
1  5 10 20.0
```

```

1 import pandas as pd
2 data = [{'a': 1, 'b': 2}, {'a': 5, 'b': 10, 'c': 20}]
3 df = pd.DataFrame(data, index=['first', 'second'])
4 print (df)

```

```

[>
      a    b    c
first  1    2  NaN
second 5   10  20.0

```

```

1 import pandas as pd
2 data = [{'a': 1, 'b': 2}, {'a': 5, 'b': 10, 'c': 20}]
3
4 #With two column indices, values same as dictionary keys
5 df1 = pd.DataFrame(data, index=['first', 'second'], columns=['a', 'b'])
6
7 #With two column indices with one index with other name
8 df2 = pd.DataFrame(data, index=['first', 'second'], columns=['a', 'b1'])
9 print (df1)
10 print (df2)

```

```

[>
      a    b
first  1    2
second 5   10
      a  b1
first  1  NaN
second 5  NaN

```

▼ DF from Pandas-Series

```

1 import pandas as pd
2
3 d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']),
4      'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}
5
6 df = pd.DataFrame(d)
7 print (df)

```

```

[>
      one  two
a    1.0    1
b    2.0    2
c    3.0    3
d    NaN    4

```

▼ Advanced Data Modification Mechanisms

▼ Column Operations

▼ Selection by Column Name

```
1 import pandas as pd
2
3 d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']),
4      'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}
5
6 df = pd.DataFrame(d)
7 print (df ['one'])
```

```
☞ a    1.0
   b    2.0
   c    3.0
   d    NaN
   Name: one, dtype: float64
```

▼ Adding a new column

```
1 import pandas as pd
2
3 d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']),
4      'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}
5
6 df = pd.DataFrame(d)
7
8 # Adding a new column to an existing DataFrame object with column label by passing new
9
10 print ("Adding a new column by passing as Series:")
11 df['three']=pd.Series([10,20,30],index=['a','b','c'])
12 print (df)
13
14 print ("Adding a new column using the existing columns in DataFrame:")
15 df['four']=df['one']+df['three']
16
17 print (df)
```

```
☞ Adding a new column by passing as Series:
   one  two  three
a  1.0   1   10.0
b  2.0   2   20.0
c  3.0   3   30.0
d  NaN   4    NaN
Adding a new column using the existing columns in DataFrame:
   one  two  three  four
a  1.0   1   10.0  11.0
b  2.0   2   20.0  22.0
c  3.0   3   30.0  33.0
d  NaN   4    NaN   NaN
```

▼ Deletion of a column

```

1 # Using the previous DataFrame, we will delete a column
2 # using del function
3 import pandas as pd
4
5 d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']),
6      'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd']),
7      'three' : pd.Series([10,20,30], index=['a','b','c'])}
8
9 df = pd.DataFrame(d)
10 print ("Our dataframe is:")
11 print (df)
12
13 # using del function
14 print ("Deleting the first column using DEL function:")
15 del df['one']
16 print (df)
17
18 # using pop function
19 print ("Deleting another column using POP function:")
20 df.pop('two')
21 print (df)

```

```

[> Our dataframe is:
      one  two  three
a  1.0    1   10.0
b  2.0    2   20.0
c  3.0    3   30.0
d  NaN    4    NaN
Deleting the first column using DEL function:
      two  three
a     1   10.0
b     2   20.0
c     3   30.0
d     4    NaN
Deleting another column using POP function:
      three
a    10.0
b    20.0
c    30.0
d     NaN

```

▼ Column selection by label

```

1 import pandas as pd
2
3 d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']),
4      'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}
5
6 df = pd.DataFrame(d)
7 print (df['one'])

```

```
[>
```

```
a    1.0
b    2.0
c    3.0
```

▼ Column selection by position (integer)

```
1 import pandas as pd
2
3 d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']),
4      'two' : pd.Series([10, 20, 30, 40], index=['a', 'b', 'c', 'd'])}
5
6 df = pd.DataFrame(d)
7 print (df.iloc[:,0])
8 print (df.iloc[:,1])
```

```
☞ a    1.0
   b    2.0
   c    3.0
   d    NaN
   Name: one, dtype: float64
   a    10
   b    20
   c    30
   d    40
   Name: two, dtype: int64
```

▼ Row Operations

▼ Select row by index label

```
1 import pandas as pd
2
3 d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']),
4      'two' : pd.Series([10, 20, 30, 40], index=['a', 'b', 'c', 'd'])}
5
6 df = pd.DataFrame(d)
7 print (df.loc['b'])
```

```
☞ one    2.0
   two   20.0
   Name: b, dtype: float64
```

▼ Select row by index position

```
1 import pandas as pd
2
3 d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']),
4      'two' : pd.Series([10, 20, 30, 40], index=['a', 'b', 'c', 'd'])}
5
```

```
6 df = pd.DataFrame(d)
7 print (df.iloc[2])
```

```
↳ one      3.0
   two     30.0
   Name: c, dtype: float64
```

▼ Slicing Rows

```
1 import pandas as pd
2
3 d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']),
4      'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}
5
6 df = pd.DataFrame(d)
7 print (df[2:4])
```

```
↳      one  two
   c  3.0    3
   d  NaN    4
```

▼ Adding a New Row

```
1 import pandas as pd
2
3 df = pd.DataFrame([[1, 2], [3, 4]], columns = ['a','b'])
4 df2 = pd.DataFrame([[5, 6], [7, 8]], columns = ['a','b'])
5
6 df = df.append(df2)
7 print (df)
```

```
↳      a  b
   0  1  2
   1  3  4
   0  5  6
   1  7  8
```

▼ Deletion of a Row

```
1 import pandas as pd
2
3 df = pd.DataFrame([[1, 2], [3, 4]], columns = ['a','b'],index=['row1','row2'])
4 df2 = pd.DataFrame([[5, 6], [7, 8]], columns = ['a','b'], index=['row3','row4'])
5
6 df = df.append(df2)
7
8 print ('before dropping\n',df)
9
10 # Drop rows with label 'row1'
11 df1 = df.drop('row1')
```

```

11 df = df.drop('row1',
12
13 print ('after dropping row1 \n',df1)
14
15 df2 = df.drop('row3')
16
17 print ('after dropping row3 \n',df2)

```

```

[> before dropping
      a  b
row1  1  2
row2  3  4
row3  5  6
row4  7  8
after dropping row1
      a  b
row2  3  4
row3  5  6
row4  7  8
after dropping row3
      a  b
row1  1  2
row2  3  4
row4  7  8

```

▼ Data Frames - File I/O

▼ Saving to a file

```

1 import pandas as pd
2
3 df = pd.DataFrame([[1,2],[10,20]],index=['row1','row2'],columns=['a','b'])
4
5 print (df)
6
7 df.to_csv('test.csv',header=True)

```

```

[>      a  b
row1   1  2
row2  10 20

```

```

1 ! cat test.csv

```

```

[> ,a,b
row1,1,2
row2,10,20

```

▼ Loading from File

```

1 import pandas as pd

```



```

2
3 df1 = pd.read_csv('test.csv',header=0,index_col=0)
4
5 print (df1)

```

```

[>]
      a  b
row1  1  2
row2 10 20

```

▼ Merging Data Frames

REF - <https://datacarpentry.org/python-socialsci/11-joins/index.html>

▼ Concatenation

```

1 df2 = df1.copy()
2
3 print (df2)

```

```

[>]
      a  b
row1  1  2
row2 10 20

```

```

1 df3 = pd.concat([df1,df2],axis=0)
2
3 print (df3)

```

```

[>]
      a  b
row1  1  2
row2 10 20
row1  1  2
row2 10 20

```

```

1 df3 = pd.concat([df1,df2],axis=1)
2
3 print (df3)

```

```

[>]
      a  b  a  b
row1  1  2  1  2
row2 10 20 10 20

```

▼ Merge

```

1 df3 = pd.merge(df1,df2,on='a')
2
3 print (df3)

```

```

↳      a  b_x  b_y
   0    1    2    2
   1   10   20   20

```

▼ Read/Write JSON

```

1 import json
2
3 # a Python object (dict):
4 x = [
5     {
6         "wfid": "1", "status": "pending", "node": "1", "data": "none"
7     },
8     {
9         "wfid": "1", "status": "delivered", "node": 1, "data": "none"
10    }
11 ]
12
13 import pandas as pd
14
15 df = pd.DataFrame(x)
16
17 print (df)
18
19 print ('another')
20
21 print (df[df['status']=='pending']['node'])

```

```

↳      wfid      status node  data
   0     1    pending     1  none
   1     1  delivered     1  none
another
   0     1
Name: node, dtype: object

```

```

1 print (df[df['status']=='pending']['node'])
2
3 df.loc[df['status']=='pending', 'node']='1000'
4
5 print (df)

```

```

↳      0     1
Name: node, dtype: object
      wfid      status node  data
   0     1    pending 1000  none
   1     1  delivered     1  none

```

```

1 x = df.iloc[0]
2
3 print (x['wfid'])

```

```

↳      1

```

```
1 mycode = 'lambda x : 100 if x[\'wfid\']==\'1\' else 0'
2
3 print (mycode)
```

```
☞ lambda x : 100 if x['wfid']== '1' else 0
```

```
1 fun = eval(mycode)
2
3 print (type(fun))
4
5 print (df.iloc[0])
6 print (fun(df.iloc[0]))
```

```
☞
```

```
1 print (df.index)
```

```
☞
```

```
1 fun = lambda x : 1 if x==1 else 0
2
3 fun(0)
```

```
☞
```

```
1 df[df['status']=='pending'].index
```

```
☞
```

```
1 print (df.iloc[0]['status'])
2
3 df.iloc[0]['status'] = 'x'
4
5 print (df.iloc[0]['status'])
```

```
☞
```

```
1 for x in df[df['status']=='pending'].iterrows() :
2     print (type(x),x)
3
```

```
☞
```

```
1 header = df.columns
2
3 print (header)
```



```
1 print ([x for x in df.columns])
2 for index,row in df.iterrows() :
3     for x in row :
4         print (x)
```



```
1 new_row = {'wfid':'2','status':'pending'}
2
3 df.append(new_row,ignore_index=True)
```



```
1 new_row = {'status':'pending', 'wfid':'3'}
2
3 df.append(new_row,ignore_index=True)
```



```
1 df2 = pd.DataFrame()
2
3 df2.append(new_row,ignore_index=True)
4
5 print (df2.to_string())
```



