```
d = {'one': pd.Series([10, 20, 30, 40],
                      index=['a', 'b', 'c', 'd']),
     'two': pd.Series([10, 20, 30, 40],
                      index=['a', 'b', 'c', 'd'])}

# creates Dataframe.
df = pd.DataFrame(d)

# print the data.
df
```

	one	two
a	10	10
b	20	20
c	30	30
d	40	40

Asad's Documentation

How to create dataFrame

```
# initital column and index

index=[1911022,1911023,1911024]
columns=['class_1','class_2','class_3','class_4']

# initialize data

# it insert data row wise , we need this
data = [
        {'class_1': 'P','class_2': 'P','class_3': 'P','class_4': 'P'},
        {'class_1': 'A','class_2': 'P','class_3': 'A','class_4': 'P'},
        {'class_1': 'P','class_2': 'P','class_3': 'A','class_4': 'P'},
       ]

# Creates pandas DataFrame.
df = pd.DataFrame(data,index,columns)

# print the data
df
```

	class_1	class_2	class_3	class_4
1911022	P	P	P	P
1911023	A	P	A	P
1911024	P	P	A	P

How to save dataframe

```
df.to_csv('name.csv')  # index=None    for ignore index
```

```
df.to_csv('encoded.csv', encoding='utf-8', index=False)
```

```
#try in local
#df.to_excel('xcel.xlsx', encoding='utf-8', index=False)
```

How to read excel files

```
'''
Two Types excel file
1. .xlsx file: Excel files which are created using excel software, these are .xlsx file
2. .csv file: In the above, which you exported, these are csv file

If you want to read these you need below code
'''
```

```
'\nTwo Types excel file\n1. .xlsx file: Excel files which are created using excel software, these are .xlsx files.\n2. .csv file: In the above, which you exported, these are csv file\n\nIf you want to read these you need below code\n'
```

```
#try in local
# For xlsx file, you need to install this by command promt
# pip install openpyxl
```

```
# read xlsx file
#xlsx_file=pd.read_excel("theorical.xlsx",nrows=3)
```

```
#csv file

df=pd.read_csv('name.csv',index_col=0)  #index_col=0 to avoid index # encoding='ISO-885
```

```
df
```

	class_1	class_2	class_3	class_4
1911022	P	P	P	P
1911023	A	P	A	P
1911024	P	P	A	P

How to skip rows during reading

```
# How to skip rows
df2 = pd.read_csv('name.csv', index_col=0, skiprows=1)   # skip first 1 rows
df2
```

	P	P.1	P.2	P.3
1911022				
1911023	A	P	A	P
1911024	P	P	A	P

First was column, thats why column eliminated

```
#import DataFrame and skip 2nd and 3rd row
df2 = pd.read_csv('name.csv', index_col=0, skiprows=[2, 3])
df2
```

	class_1	class_2	class_3	class_4
1911022	P	P	P	P

How to see index

```
df.index
```

```
Int64Index([1911022, 1911023, 1911024], dtype='int64')
```

```
# as a list
list(df.index)
```

```
[1911022, 1911023, 1911024]
```

```
#index quantity
len(df.index)
```

```
3
```

how to see columns

```
df.columns
```

```
Index(['class_1', 'class_2', 'class_3', 'class_4'], dtype='object')
```

```
# as a list
list(df.columns)
```

```
['class_1', 'class_2', 'class_3', 'class_4']
```

```
# column quantity
len(df.columns)
```

```
4
```

how to append new row

```
# Creating the new row
data = [                #row wise
        {'class_1': 'P','class_2': 'P','class_3': 'P','class_4': 'P','class_5':'A'}
       ]
index=[1911025]
new_row = pd.DataFrame(data,index)
new_row
```

	class_1	class_2	class_3	class_4	class_5
1911025	P	P	P	P	A

```
# for appending new_row at the end of df

df = df.append(new_row, ignore_index = False) # as we have specific index, not ignoring
```

```
df
```

	class_1	class_2	class_3	class_4	class_5
1911022	P	P	P	P	NaN
1911023	A	P	A	P	NaN
1911024	P	P	A	P	NaN
1911025	P	P	P	P	A

How to append new column

```
data = ['P','A','P','A']
data
```

```
['P', 'A', 'P', 'A']
```

```
df['class_5']=data
df
```

	class_1	class_2	class_3	class_4	class_5
1911022	P	P	P	P	P
1911023	A	P	A	P	A
1911024	P	P	A	P	P
1911025	P	P	P	P	A

```
# if all data keep same

data='P'
df['class_6']=data
df
```

	class_1	class_2	class_3	class_4	class_5	class_6
1911022	P	P	P	P	P	P
1911023	A	P	A	P	A	P
1911024	P	P	A	P	P	P
1911025	P	P	P	P	A	P

how to see specific data

```
#accessing row by index

df.loc[1911025]
```

```
class_1    P
class_2    P
class_3    P
class_4    P
class_5    A
class_6    P
Name: 1911025, dtype: object
```

```
#accessing in a range of index

df.loc[1911023:1911025]
```

	class_1	class_2	class_3	class_4	class_5	class_6
1911023	A	P	A	P	A	P
1911024	P	P	A	P	P	P
1911025	P	P	P	P	A	P

```
# top row
df.head(2)
```

	class_1	class_2	class_3	class_4	class_5	class_6
1911022	P	P	P	P	P	P
1911023	A	P	A	P	A	P

```
# last row
df.tail(2)
```

	class_1	class_2	class_3	class_4	class_5	class_6
1911024	P	P	A	P	P	P
1911025	P	P	P	P	A	P

```
#accessing a specific column
```

```
df['class_5']
```

```
1911022    P
1911023    A
1911024    P
1911025    A
Name: class_5, dtype: object
```

```
#accessing a specific cell by column and index

df['class_5'][1911025]
```

```
'A'
```

```
# taking 2 sample row randomly
df.sample(2)
```

	class_1	class_2	class_3	class_4	class_5	class_6
1911023	A	P	A	P	A	P
1911022	P	P	P	P	P	P

How to update a row

```
df.loc[1911022]
```

```
class_1    P
class_2    P
class_3    P
class_4    P
class_5    P
class_6    P
Name: 1911022, dtype: object
```

```
#just a value
df.at[1911022,'class_1']='A'
```

```
df
```

	class_1	class_2	class_3	class_4	class_5	class_6
1911022	A	P	P	P	P	P
1911023	A	P	A	P	A	P
1911024	P	P	A	P	P	P
1911025	P	P	P	P	A	P

```
#whole row

column=['class_1','class_2','class_3','class_4','class_5','class_6']
```

```
# or
column= list(df.columns)  # this output is same as abve

data=['A','A','A','A','A','A']

df.at[1911022,column]=data
df
```

	class_1	class_2	class_3	class_4	class_5	class_6
1911022	A	A	A	A	A	A
1911023	A	P	A	P	A	P
1911024	P	P	A	P	P	P
1911025	P	P	P	P	A	P

how to update column

```
# df['column name'] = df['column name'].replace(['old value'], 'new value')

df['class_6']=df['class_6'].replace(['A'],'P')
df
```

	class_1	class_2	class_3	class_4	class_5	class_6
1911022	A	A	A	A	A	P
1911023	A	P	A	P	A	P
1911024	P	P	A	P	P	P
1911025	P	P	P	P	A	P

```
list(df['class_6'])
```

```
['P', 'P', 'P', 'P']
```

```
new_value=['A','A','A','A']
df['class_6']=df['class_6'].replace(list(df['class_6']),new_value)
df
```

	class_1	class_2	class_3	class_4	class_5	class_6
1911022	A	A	A	A	A	A
1911023	A	P	A	P	A	A
1911024	P	P	A	P	P	A
1911025	P	P	P	P	A	A

Query

```
# df[ query here ]

df[ df.class_1=='P' ]
```

	class_1	class_2	class_3	class_4	class_5	class_6
1911024	P	P	A	P	P	A
1911025	P	P	P	P	A	A

```
df[ df.class_1=='P'] [['class_1']]
```

	class_1
1911024	P
1911025	P

```
df[ df.class_1=='A'] [['class_1','class_2']]
```

	class_1	class_2
1911022	A	A
1911023	A	P

```
df[ df.class_1=='A'].index
```

```
Int64Index([1911022, 1911023], dtype='int64')
```

```
list(df[ df.class_1=='A'].index)
```

```
[1911022, 1911023]
```

```
df[ (df.class_1=='P') &  (df.class_2=='P') ]
```

	class_1	class_2	class_3	class_4	class_5	class_6
1911024	P	P	A	P	P	A
1911025	P	P	P	P	A	A

```
list( df[ (df.class_1=='P') &  (df.class_2=='P') ] . index )
```

```
[1911024, 1911025]
```