In [1]:

```
# pandas
import numpy as np
import pandas as pd
```

In [4]:

```
# read data set
df = pd.read_csv("sets/records.csv")
df
```

Out[4]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
0	1001	A	10	2	3	50
1	1002	B	10	6	2	80
2	1003	A	10	3	2	60
3	1004	C	11	0	1	45
4	1005	C	12	5	2	75

In [ ]:

```

```

In [ ]:

```
# LOC  => It is used to access group of rows and columns by labels or boolean array
# syntax => Dataframe(df).loc[label]
```

In [5]:

```
df.loc[0]
```

Out[5]:

```
Student ID                1001
Section                      A
Class                       10
Study hrs                    2
Social Media usage hrs       3
Percentage                  50
Name: 0, dtype: object
```

In [6]:

```
df.loc[[2]]
```

Out[6]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
2	1003	A	10	3	2	60

In [7]:

```
df.loc[2]
```

Out[7]:

```
Student ID                1003
Section                      A
Class                       10
Study hrs                    3
Social Media usage hrs       2
Percentage                  60
Name: 2, dtype: object
```

In [ ]:

```

```

In [8]:

```
# two rows data
df.loc[[0,2]]
```

Out[8]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
0	1001	A	10	2	3	50
2	1003	A	10	3	2	60

In [9]:

```
df.loc[[1,3]]
```

Out[9]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
1	1002	B	10	6	2	80
3	1004	C	11	0	1	45

In [ ]:

```

```

In [10]:

```
df
```

Out[10]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
0	1001	A	10	2	3	50
1	1002	B	10	6	2	80
2	1003	A	10	3	2	60
3	1004	C	11	0	1	45
4	1005	C	12	5	2	75

In [11]:

```
df.loc[2,'Percentage']
```

Out[11]:

```
60
```

In [12]:

```
# df.loc[rows,column]

df.loc[1:3,'Percentage']
```

Out[12]:

```
1    80
2    60
3    45
Name: Percentage, dtype: int64
```

In [ ]:

```

```

In [13]:

```
df
```

Out[13]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
0	1001	A	10	2	3	50
1	1002	B	10	6	2	80
2	1003	A	10	3	2	60
3	1004	C	11	0	1	45
4	1005	C	12	5	2	75

In [14]:

```
# boolean Array
df.loc[[False]]
```

Out[14]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage

In [15]:

```
# boolean Array
df.loc[[False,True]]
```

Out[15]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
1	1002	B	10	6	2	80

In [16]:

```
# boolean Array
df.loc[[False,True,True]]
```

Out[16]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
1	1002	B	10	6	2	80
2	1003	A	10	3	2	60

In [ ]:

```

```

In [17]:

```
# LOC we use conditions
df
```

Out[17]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
0	1001	A	10	2	3	50
1	1002	B	10	6	2	80
2	1003	A	10	3	2	60
3	1004	C	11	0	1	45
4	1005	C	12	5	2	75

In [22]:

```
df.loc[df['Percentage']>59]
```

Out[22]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
1	1002	B	10	6	2	80
2	1003	A	10	3	2	60
4	1005	C	12	5	2	75

In [ ]:

```

```

In [23]:

```
# multiple conditions
df.loc[df['Percentage']>59,['Study hrs']]
```

Out[23]:

	Study hrs
1	6
2	3
4	5

In [24]:

```
# multiple conditions with multiple columns
df.loc[df['Percentage']>59,['Study hrs','Percentage']]
```

Out[24]:

	Study hrs	Percentage
1	6	80
2	3	60
4	5	75

In [ ]:

```

```

In [ ]:

```

```

In [25]:

```
data = pd.DataFrame({
                        'class':['A','B','C','B','A','B','A'],
                        'age':[20,21,22,23,24,25,26],
                        'gender':['M','F','M','F','M','F','M'],
                        'status':['Yes',np.NaN,'No','Yes','No','Yes','No']
                    })
data
```

Out[25]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	No
5	B	25	F	Yes
6	A	26	M	No

In [26]:

```
data.loc[0]
```

Out[26]:

```
class       A
age        20
gender      M
status    Yes
Name: 0, dtype: object
```

In [27]:

```
data.loc[[0]]
```

Out[27]:

	class	age	gender	status
0	A	20	M	Yes

In [28]:

```
data.loc[data.age>22]
```

Out[28]:

	class	age	gender	status
3	B	23	F	Yes
4	A	24	M	No
5	B	25	F	Yes
6	A	26	M	No

In [30]:

```
data.loc[(data.age>22) & (data.age<25)]
```

Out[30]:

	class	age	gender	status
3	B	23	F	Yes
4	A	24	M	No

In [ ]:

```

```

In [31]:

```
data
```

Out[31]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	No
5	B	25	F	Yes
6	A	26	M	No

In [33]:

```
data.loc[(data.age>22),['class','status']]
```

Out[33]:

	class	status
3	B	Yes
4	A	No
5	B	Yes
6	A	No

In [ ]:

```

```

In [34]:

```
data
```

Out[34]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	No
5	B	25	F	Yes
6	A	26	M	No

In [36]:

```
# loc to update pericular column value
data.loc[(data.age>23),['status']] = 'Yes'
data
```

Out[36]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	Yes
5	B	25	F	Yes
6	A	26	M	Yes

In [ ]:

```

```

In [ ]:

```

```

In [37]:

```
data
```

Out[37]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	Yes
5	B	25	F	Yes
6	A	26	M	Yes

In [39]:

```
# updating multiple column with multiple rows
data.loc[(data.age>23),['status','gender']] = ['No','F']
data
```

Out[39]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	F	No
5	B	25	F	No
6	A	26	F	No

In [ ]:

```

```

In [40]:

```
# logic
data2 = pd.DataFrame({
                      'class':['A','B','C','B','A','B','A'],
                      'age':[20,21,22,23,24,25,26],
                      'gender':['M','F','M','F','M','F','M'],
                      'status':['Yes',np.NaN,'No','Yes','No','Yes','No']
                     })
data2
```

Out[40]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	No
5	B	25	F	Yes
6	A	26	M	No

In [44]:

```
data2.loc[lambda data2:data2['age']==24]
```

Out[44]:

	class	age	gender	status
4	A	24	M	No

In [45]:

```
data2.loc[lambda data2:data2['age']>=24]
```

Out[45]:

	class	age	gender	status
4	A	24	M	No
5	B	25	F	Yes
6	A	26	M	No

In [46]:

```
data2
```

Out[46]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	No
5	B	25	F	Yes
6	A	26	M	No

In [47]:

```
# change particular column all value
# slice (:)
data2.loc[:]
```

Out[47]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	No
5	B	25	F	Yes
6	A	26	M	No

In [48]:

```
data2.loc[0:]
```

Out[48]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	No
5	B	25	F	Yes
6	A	26	M	No

In [50]:

```
data2.loc[0:5]
```

Out[50]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	No
5	B	25	F	Yes

In [ ]:

```
```

In [54]:

```
# all rows with first 2 column
data2.loc[:,::2]
```

Out[54]:

	class	gender
0	A	M
1	B	F
2	C	M
3	B	F
4	A	M
5	B	F
6	A	M

In [55]:

```
data2.loc[:,::1]
```

Out[55]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	No
5	B	25	F	Yes
6	A	26	M	No

In [56]:

```
data2.loc[:,::3]
```

Out[56]:

	class	status
0	A	Yes
1	B	NaN
2	C	No
3	B	Yes
4	A	No
5	B	Yes
6	A	No

In [ ]:

```

```

In [57]:

```
# change particular column with all value
data2
```

Out[57]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	No
5	B	25	F	Yes
6	A	26	M	No

In [58]:

```
data2.loc[:,'age']=50
```

In [59]:

```
data2
```

Out[59]:

	class	age	gender	status
0	A	50	M	Yes
1	B	50	F	NaN
2	C	50	M	No
3	B	50	F	Yes
4	A	50	M	No
5	B	50	F	Yes
6	A	50	M	No

In [ ]:

```

```

In [ ]:

```

```

In [60]:

```
# ILOC => Integer location - Based Index  or boolean Array
data2
```

Out[60]:

	class	age	gender	status
0	A	50	M	Yes
1	B	50	F	NaN
2	C	50	M	No
3	B	50	F	Yes
4	A	50	M	No
5	B	50	F	Yes
6	A	50	M	No

In [61]:

```
data2.iloc[0]
```

Out[61]:

```
class       A
age        50
gender      M
status    Yes
Name: 0, dtype: object
```

In [62]:

```
data2.iloc[[0]]
```

Out[62]:

	class	age	gender	status
0	A	50	M	Yes

In [63]:

```
# all rows
data2.iloc[:]
```

Out[63]:

	class	age	gender	status
0	A	50	M	Yes
1	B	50	F	NaN
2	C	50	M	No
3	B	50	F	Yes
4	A	50	M	No
5	B	50	F	Yes
6	A	50	M	No

In [64]:

```
# particular rows range value

data2.iloc[2:6]
```

Out[64]:

	class	age	gender	status
2	C	50	M	No
3	B	50	F	Yes
4	A	50	M	No
5	B	50	F	Yes

In [ ]:

```

```

In [65]:

```
# particular rows

data2.iloc[[2,5]]
```

Out[65]:

	class	age	gender	status
2	C	50	M	No
5	B	50	F	Yes

In [ ]:

```

```

In [66]:

```
# particular rows with particular column
# sec row with sec column value
data.iloc[2,2]
```

Out[66]:

```
'M'
```

In [ ]:

```

```

In [70]:

```
# diff b/w loc and iloc
data2.loc[data2.age>0]
```

Out[70]:

	class	age	gender	status
0	A	50	M	Yes
1	B	50	F	NaN
2	C	50	M	No
3	B	50	F	Yes
4	A	50	M	No
5	B	50	F	Yes
6	A	50	M	No

In [71]:

```
data2.iloc[data2.age>0]
```

```
---------------------------------------------------------------------------
NotImplementedError                       Traceback (most recent call last)
<ipython-input-71-8cab1dc58123> in <module>
----> 1 data2.iloc[data2.age>0]

~\Anaconda3\lib\site-packages\pandas\core\indexing.py in __getitem__(self, k
ey)
   1498
   1499             maybe_callable = com.apply_if_callable(key, self.obj)
-> 1500             return self._getitem_axis(maybe_callable, axis=axis)
   1501
   1502     def _is_scalar_access(self, key):

~\Anaconda3\lib\site-packages\pandas\core\indexing.py in _getitem_axis(self,
key, axis)
   2214
   2215         if com.is_bool_indexer(key):
-> 2216             self._validate_key(key, axis)
   2217             return self._getbool_axis(key, axis=axis)
   2218

~\Anaconda3\lib\site-packages\pandas\core\indexing.py in _validate_key(self,
key, axis)
   2058             if hasattr(key, 'index') and isinstance(key.index, Index
):
   2059                 if key.index.inferred_type == 'integer':
-> 2060                     raise NotImplementedError("iLocation based boole
an "
   2061                                               "indexing on an intege
r type "
   2062                                               "is not available")

NotImplementedError: iLocation based boolean indexing on an integer type is
 not available
```

In [ ]:

```

```

In [73]:

```
# Logic
data3 = pd.DataFrame({
                        'class':['A','B','C','B','A','B','A'],
                        'age':[20,21,22,23,24,25,26],
                        'gender':['M','F','M','F','M','F','M'],
                        'status':['Yes',np.NaN,'No','Yes','No','Yes','No']
                      })
data3
```

Out[73]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	No
5	B	25	F	Yes
6	A	26	M	No

In [75]:

```
data3.iloc[5,1]
```

Out[75]:

```
25
```

In [76]:

```
data3.iloc[:,1]
```

Out[76]:

```
0    20
1    21
2    22
3    23
4    24
5    25
6    26
Name: age, dtype: int64
```

In [77]:

```
data3.iloc[:,3]
```

Out[77]:

```
0    Yes
1    NaN
2     No
3    Yes
4     No
5    Yes
6     No
Name: status, dtype: object
```

In [ ]:

```

```

In [78]:

```
# partcular rows
data3.iloc[[2,5]]
```

Out[78]:

	class	age	gender	status
2	C	22	M	No
5	B	25	F	Yes

In [ ]:

```

```

In [79]:

```
# rows
data3.iloc[[False,True,True,False,True,False]]
```

Out[79]:

	class	age	gender	status
1	B	21	F	NaN
2	C	22	M	No
4	A	24	M	No

In [ ]:

```

```

In [81]:

```
data3
```

Out[81]:

	class	age	gender	status
0	A	20	M	Yes
1	B	21	F	NaN
2	C	22	M	No
3	B	23	F	Yes
4	A	24	M	No
5	B	25	F	Yes
6	A	26	M	No

In [82]:

```
# columns
data3.iloc[[False,True,True,False,True,False],[False,True,True,False]]
```

Out[82]:

	age	gender
1	21	F
2	22	M
4	24	M

In [ ]:

```

```