```
my_DF
'Output':
   age state_of_origin
a   15           Lagos
a   17     Cross River
b   21            Kano
b   29            Abia
c   25           Benue
# select using explicit indexing
my_DF.loc['a']
Out[196]:
   age state_of_origin
a   15           Lagos
a   17     Cross River
# let's try to use loc for implicit indexing
my_DF.loc[0]
'Output':
    Traceback (most recent call last):
    TypeError: cannot do label indexing on <class 'pandas.core.indexes.
    base.Index'>
        with these indexers [0] of <class 'int'>
```

Selecting Multiple Rows and Columns from a DataFrame

Let's use the **loc** method to select multiple rows and columns from a Pandas DataFrame.

```
# select rows with age greater than 20
my_DF.loc[my_DF.age > 20]
'Output':
   age state_of_origin
2   21            Kano
3   29            Abia
4   25           Benue
# find states of origin with age greater than or equal to 25
my_DF.loc[my_DF.age >= 25, 'state_of_origin']
```

```
'Output':
Out[29]:
3     Abia
4    Benue
```

Slice Cells by Row and Column from a DataFrame

First let's create a DataFrame. Remember, we use **iloc** when no explicit index or row labels are assigned.

```
my_DF = pd.DataFrame({'age': [15,17,21,29,25], \
            'state_of_origin':['Lagos', 'Cross River', 'Kano', 'Abia',
            'Benue']})
my_DF
'Output':
   age state_of_origin
0   15           Lagos
1   17     Cross River
2   21            Kano
3   29            Abia
4   25           Benue
# select the third row and second column
my_DF.iloc[2,1]
'Output': 'Kano'
# slice the first 2 rows - indexed from zero, excluding the final index
my_DF.iloc[:2,]
'Output':
   age state_of_origin
0   15           Lagos
1   17     Cross River
# slice the last three rows from the last column
my_DF.iloc[-3:,-1]
'Output':
2     Kano
3     Abia
4    Benue
Name: state_of_origin, dtype: object
```