Selecting a Row from a DataFrame

Pandas makes use of two unique wrapper attributes for indexing rows from a **DataFrame** or a cell from a **Series** data structure. These attributes are the **iloc** and **loc** – they are also known as indexers. The **iloc** attribute allows you to select or slice row(s) of a DataFrame using the intrinsic Python index format, whereas the **loc** attribute uses the explicit indices assigned to the DataFrame. If no explicit index is found, **loc** returns the same value as **iloc**.

Remember that the data type of a DataFrame row is a **Series** because it is a vector or 1-D array.

Let's select the first row from the DataFrame.

```
# using explicit indexing
my_DF.loc[0]
'Output':
age                  15
state_of_origin    Lagos
Name: 0, dtype: object
# using implicit indexing
my_DF.iloc[0]
'Output':
age                  15
state_of_origin    Lagos
Name: 0, dtype: object
# let's see the data type
type(my_DF.loc[0])
'Output':  pandas.core.series.Series
```

Now let's create a DataFrame with explicit indexing and test out the **iloc** and **loc** methods. Pandas will return an error if **iloc** is used for explicit indexing or if **loc** is used for implicit Python indexing.

```
my_DF = pd.DataFrame({'age': [15,17,21,29,25], \
            'state_of_origin':['Lagos', 'Cross River', 'Kano', 'Abia',
            'Benue']},\
            index=['a','a','b','b','c'])
# observe the string indices
```

```
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']
```