```
my_DF
'Output':
   age state_of_origin
0   15           Lagos
1   17     Cross River
2   21            Kano
3   29            Abia
4   25           Benue
```

Selecting a Column from a DataFrame

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

```
my_DF['age']
'Output':
0    15
1    17
2    21
3    29
4    25
Name: age, dtype: int64
# check data type
type(my_DF['age'])
'Output':  pandas.core.series.Series
```

To select multiple columns, enclose the column names as **strings** with the double square brackets **[[ ]]**. The following code is an example:

```
my_DF[['age','state_of_origin']]
'Output':
   age state_of_origin
0   15           Lagos
1   17     Cross River
2   21            Kano
3   29            Abia
4   25           Benue
```

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
```