Indexing

- To access a row or column of the DataFrame
- Types

  1. Label Based Indexing
  2. Location(Position) Based Indexing

  Note that all Indexing operations returns selected row or column as a Series.

1. Label Based Indexing

- Differenet Label Based Indexing Operations

  1. Select a row by label: **Series = DataFrame.loc[row_label]**
  2. Select a column by label

  a) Select a column by label: **Series = DataFrame[col_label]**
  b) Select a column by label with loc method: **Series = DataFrame.loc[: , col_label]**

In [1]:
```
# Import pandas
import pandas as pd
```

In [2]:
```
# Create a dataFrame from a dictionary of lists with some row labels (We can use dataframe's index argument to set labels)
df = pd.DataFrame({"Name":["Ravi", "Prem", "Satya"], "Department":["Sales", "Admin", "Finance"], "Salary":[6000, 5500, 7000]}, ir
print(df)
```

```
        Name Department  Salary
ID001   Ravi      Sales    6000
ID002   Prem      Admin    5500
ID003  Satya    Finance    7000
```

In [3]:
```
# Select a row by label

series_ID001 = df.loc["ID001"]
print(series_ID001)
print(type(series_ID001))

print("-----------------")

series_ID002 = df.loc["ID002"]
print(series_ID002)
```

```
Name           Ravi
Department    Sales
Salary         6000
Name: ID001, dtype: object
<class 'pandas.core.series.Series'>
-----------------
Name           Prem
Department    Admin
Salary         5500
Name: ID002, dtype: object
```

In [4]:
```
# Select a column by a label

Name_series = df["Name"]
print(Name_series)
```

```
ID001     Ravi
ID002     Prem
ID003    Satya
Name: Name, dtype: object
```

In [5]:
```
# Selecting a column by label with loc mehod
dept_series = df.loc[:,"Department"]
print(dept_series)
```

```
ID001      Sales
ID002      Admin
ID003    Finance
Name: Department, dtype: object
```

In [10]:
```
# Select multiple columns as a new dataframe
dept_df = df[["Name","Salary"]]
dept_df
```

Out[10]:

	Name	Salary
ID001	Ravi	6000
ID002	Prem	5500
ID003	Satya	7000

In [13]:
```
# Select the rows where salary is greater then 5500
dept_df1 = df[df["Salary"] > 5500]
dept_df1
```

Out[13]:

	Name	Department	Salary
ID001	Ravi	Sales	6000
ID003	Satya	Finance	7000

2. Location(Position) Based Indexing

- Location Based Indexing Operations

  1. Select a row by integer location: **Series = DataFrame.iloc[row_location]**
  2. Select a column by integer location **Series = DataFrame.iloc[: , col_location]**

In [6]:
```
print(df)
```

```
        Name Department  Salary
ID001   Ravi      Sales    6000
ID002   Prem      Admin    5500
ID003  Satya    Finance    7000
```

In [7]:
```
# Select a row by position(integer location)

s1 = df.iloc[0]
print(s1)
print(type(s1))

print("----------")

s2 = df.iloc[1]
print(s2)
```

```
Name           Ravi
Department    Sales
Salary         6000
Name: ID001, dtype: object
<class 'pandas.core.series.Series'>
----------
Name           Prem
Department    Admin
Salary         5500
Name: ID002, dtype: object
```

In [8]:
```
# Select a column by position(integer location)
name_series1 = df.iloc[:,0]
print(name_series1)
```

```
ID001     Ravi
ID002     Prem
ID003    Satya
Name: Name, dtype: object
```

In [14]:
```
# Select multiple columns
dept_df3 = df.iloc[:, 0:3:2]
dept_df3
```

Out[14]:

	Name	Salary
ID001	Ravi	6000
ID002	Prem	5500
ID003	Satya	7000

In [ ]: