<--------------- :: DATA FRAME FULL ASSIGNMENT ::-------------------------->

1. ASSIGNMENT (CREATE A DATA FRAME)

In [8]:

```
import pandas as pd

date = {

      'name': ['ramiz', 'sahil ', 'kayara'],
      'age': [21, 22, 23],
      'course': ['ai', 'web', 'date scintes']
}

d =pd.DataFrame(date)

print(d)
```

```
     name  age        course
0   ramiz   21            ai
1  sahil    22           web
2  kayara   23  date scintes
```

2. ASSIGNMENT (ACCESSING ROWS)

In [9]:

```
import pandas as pd

date = {
        'name ': ['ramiz', 'aman', 'sahil'],
        'age': [21, 22, 23],
        'course': ['ai', 'python', 'robotic']
}

p =pd.DataFrame(date)

print(p[['age', 'course']])

print("The Showing the Student data ")

print(p.iloc[0])

df =df.set_index('name')

print(df.loc['ramiz']

print(df)
```

```
   age   course
0   21       ai
1   22   python
2   23  robotic
The Showing the Student data
name       ramiz
age           21
course        ai
Name: 0, dtype: object
```

3. ASSIGNMENT (MODIFY AND ANALYZE A DATAFRAME)

In [29]:

```
import pandas as pd

date = {
    'name': ['ramiz','sahil', 'neha', 'rohon'],
    'age': [21, 22, 23, 24],
    'course': ['python','ai', 'data science', 'web dev']  # also fixed spelling of 'data science'
}

df = pd.DataFrame(date)

df['marks'] = [80, 60, 50, 40]

df.loc[df['name'] == 'neha', "marks"] = 90
df.loc[len(df)] = ['arjun', 25, 'machine learning', 85]

df = df[df['name'] != 'rohon']

print(df)

print(">>>>>>>>")

print("\nSummary using describe():")
print(df.describe())

print("\nDataFrame info:")
print(df.info())
```

```
    name  age            course  marks
0  ramiz   21            python     80
1  sahil   22                ai     60
2   neha   23      data science     90
4  arjun   25  machine learning     85
>>>>>>>>

Summary using describe():
             age      marks
count   4.000000   4.000000
mean   22.750000  78.750000
std     1.707825  13.149778
min    21.000000  60.000000
25%    21.750000  75.000000
50%    22.500000  82.500000
75%    23.500000  86.250000
max    25.000000  90.000000

DataFrame info:
<class 'pandas.core.frame.DataFrame'>
Index: 4 entries, 0 to 4
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   name    4 non-null      object
 1   age     4 non-null      int64 
 2   course  4 non-null      object
 3   marks   4 non-null      int64 
dtypes: int64(2), object(2)
memory usage: 160.0+ bytes
None
```

4. ASSIGNMENT (ADD , DROP & EXPORT DATA)

In [40]:

```
import pandas as pd

date = {
    'name': ['ramiz', 'aman', 'neha'],
    'age': [21, 22, 23],
    'course': ["Ai", "Data Science", 'python'],
    'marks': [88, 76, 95]  # No space in 'marks'
}


df = pd.DataFrame(date)

new_row = {
    "name": 'zara',
    'age': 21,
    'course': 'web',
    'marks': 92
}
df = pd.concat([df, pd.DataFrame([new_row])], ignore_index=True)


df = df.drop('course', axis=1)


df.to_csv('Students.csv', index=False)


print(df)
```

```
    name  age  marks
0  ramiz   21     88
1   aman   22     76
2   neha   23     95
3   zara   21     92
```

5. ASSIGNMENT (DATAFRAME FILTERING USING CONDITIONS)

In [19]:

```
import pandas as pd

date = {
      "name": ['ramiz', 'Aman', 'neha', 'zara'],
      'age': [21, 22, 23, 20],
      'marks': [88, 76, 95, 92]
}

df = pd.DataFrame(date)

filtered_df = df[(df['marks'] > 90) & (df['age'] >= 21)]

print("Student with marks > 90 and Age > 21:")
print(filtered_df)
```

```
Student with marks > 90 and Age > 21:
   name  age  marks
2  neha   23     95
```

6. ASSIGNMENT (GROUPBY & AGGREGATION IN PANDAS)

In [1]:

```
import pandas as pd

date = {
    'name': ['ramiz','sahil', 'neha', 'rohon'],
    'marks': [90, 100, 50,  80],
    'course': ['python','ai', 'data science', 'web dev']  # also fixed spelling of 'data science'
}
df = pd.DataFrame(date)

grouppad = df.groupby('course')

average_marks = grouppad['marks'].mean()

total_marks = grouppad['marks'].sum()



print('Average marks is the per course:\n', average_marks)

print("\n Total marks per course is: \n", total_marks)
```

```
Average marks is the per course:
 course
ai              100.0
data science     50.0
python           90.0
web dev          80.0
Name: marks, dtype: float64

 Total marks per course is:
 course
ai              100
data science     50
python           90
web dev          80
Name: marks, dtype: int64
```

7. ASSIGNMENT(SORTING & RENAMING)

In [2]:

```
import pandas as pd

date = {
    'name': ['ramiz', 'sahil', 'neha', 'rohon'],
    'marks': [90, 100, 50, 80],
    'course': ['python', 'ai', 'data science', 'web dev']
}

df =pd.DataFrame(date)


df_sourse = df.sort_values(by='marks', ascending=False)

df_rename =df_sourse.rename(columns={
```

```
          'name': 'Student: Name',
          'marks': 'Score'
})
print(df_rename)
```

```
  Student: Name  Score        course
1         sahil    100            ai
0         ramiz     90        python
3         rohon     80       web dev
2          neha     50  data science
```

8. ASSIGNMENT(HANDLING MISSING DATA (NAN)

In [9]:

```
import pandas as pd
import numpy as np

date = {
    'name': ['ramiz','sahil', 'neha', 'rohon'],
    'marks': [90, None, 50, None],
    'course': ['python','ai', 'data science', None]
}

df =pd.DataFrame(date)
print("\nShowin the Data\n")

print("The Total Data is : \n",df)

print('Chicking the Missing value  in the data libary ', df.isnull())

date_filled =df.fillna({

       'marks': 0,
        'course': 'unknone'
})

print("\n After the Showing the date is : \n:", date_filled)

date_dropped=df.dropna()
print('\n After the data is dropped \n',date_dropped)
```

```
Showin the Data

The Total Data is :
      name  marks        course
0  ramiz   90.0        python
1  sahil    NaN            ai
2   neha   50.0  data science
3  rohon    NaN          None
Chicking the Missing value  in the data libary      name  marks  course
0  False  False   False
1  False   True   False
2  False  False   False
3  False   True    True

 After the Showing the date is :
:     name  marks        course
0  ramiz   90.0        python
1  sahil    0.0            ai
2   neha   50.0  data science
3  rohon    0.0       unknone

 After the data is dropped
      name  marks        course
0  ramiz   90.0        python
2   neha   50.0  data science
```

9. ASSIGNMENT(APPLY LAMBDA FUNCTIONS IN PANDAS)

In [17]:

```
import pandas as pd

date = {
         'name': ['ramiz','sahil', 'neha', 'rohon'],
         'marks': [90, 20, 50, 70],
}
f =pd.DataFrame(date)

df ['Result'] = df['marks'].apply (

lambda x : 'excellent'
if  x >=90
else ('good' if x >=70 else 'Needs Improvement '))

print(df)
```

```
    name  marks        course              Result
0  ramiz   90.0        python           excellent
1  sahil    NaN            ai  Needs Improvement
2   neha   50.0  data science  Needs Improvement
3  rohon    NaN          None  Needs Improvement
```

10. [THE MINI PROJECT]

In [ ]:

```
import pandas as pd

df =pd.read_csv('Book2.csv')

df.columns =df.columns.str.strip()

print("First show the  5 rows :")

print(df.head())
df ['Marks'] =df['Marks'].fillna(df['Marks'].mean())

course_avg =df.groupby("Course")['Marks'].mean()
print("\n Average marks p;er course:")
print(course_avg)
print('\n Top Scorers:')
top_students =df[df["Marks"] >= 90]
print("\n Top Scorers:")
print(top_students[['Name', "Marks"]])
df.to_csv("Student.csv", index=False)
```