

1. Create a DataFrame from the following data: data = { 'Name': ['Alice', 'Bob', 'Charlie', 'David'], 'Age': [24, 27, 22, 32], 'City': ['New York', 'Los Angeles', 'Chicago', 'Houston'] }
- Write code to:
- a) Display the first two rows
 - b) Print the column names
 - c) Show the shape of the DataFrame
 - d) Display the summary info of the DataFrame

```
In [7]: import pandas as pd
import numpy as np
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David'],
    'Age': [24, 27, 22, 32],
    'City': ['New York', 'Los Angeles', 'Chicago', 'Houston']
}
df=pd.DataFrame(data);
print(df.head(2),"\n")
print(list(df.columns))
print(df.shape,"\n")
df.info()
```

	Name	Age	City
0	Alice	24	New York
1	Bob	27	Los Angeles

['Name', 'Age', 'City']
(4, 3)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype  
 ---  -- 
 0   Name      4 non-null       object 
 1   Age       4 non-null       int64  
 2   City      4 non-null       object 
dtypes: int64(1), object(2)
memory usage: 228.0+ bytes
```

2. Using the DataFrame above, write code to:
- a) Select only the Name column
 - b) Select both Name and City columns
 - c) Select the second row using .iloc
 - d) Select the row where Name is 'Charlie' using .loc

```
In [10]: import pandas as pd
import numpy as np
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David'],
    'Age': [24, 27, 22, 32],
    'City': ['New York', 'Los Angeles', 'Chicago', 'Houston']
}
df=pd.DataFrame(data);
print(df['Name'], "\n")
print(df[['Name', 'City']], "\n")
print(df.iloc[2], "\n")
print(df.loc[df['Name']=='Charlie'], "\n")
```

```
0      Alice
1      Bob
2    Charlie
3    David
Name: Name, dtype: object
```

```
      Name        City
0   Alice    New York
1     Bob  Los Angeles
2  Charlie   Chicago
3   David    Houston
```

```
Name    Charlie
Age      22
City   Chicago
Name: 2, dtype: object
```

```
      Name  Age        City
2  Charlie   22   Chicago
```

3. Filter the DataFrame to show:
- a) People older than 25
 - b) People living in 'Chicago' or 'Houston'
 - c) People whose age is between 23 and 30

```
In [16]: import pandas as pd
import numpy as np
data = {
'Name': ['Alice', 'Bob', 'Charlie', 'David'],
'Age': [24, 27, 22, 32],
'City': ['New York', 'Los Angeles', 'Chicago', 'Houston']
}
df=pd.DataFrame(data);
a=df[df['Age']>25]
print(a, "\n")
b=df[(df['City']=='Chicago')|(df['City']=='Houston')]
print(b, "\n")
c=df[(df['Age']>23)&(df['Age']<30)]
print(c, "\n")
```

```
      Name  Age        City
1     Bob   27  Los Angeles
3   David   32    Houston
```

```
      Name  Age        City
2  Charlie   22   Chicago
3   David   32    Houston
```

```
      Name  Age        City
0   Alice   24    New York
1     Bob   27  Los Angeles
```

4. Modify the DataFrame:
- a) Add a new column Score with values [85, 90, 88, 95]
 - b) Change Bob's age to 28
 - c) Remove the City column
 - d) Drop the row of David

```
In [22]: import pandas as pd
import numpy as np
```

```

data = {
'Name': ['Alice', 'Bob', 'Charlie', 'David'],
'Age': [24, 27, 22, 32],
'City': ['New York', 'Los Angeles', 'Chicago', 'Houston']
}
df=pd.DataFrame(data);
df['Score']=[85, 90, 88, 95]
print(df,"\n")
df.loc[1,'Age']=28
print(df,"\n")
df=df.drop('City',axis = 1)
print(df,"\n")
df=df.drop(3)
print(df,"\n")

```

	Name	Age	City	Score
0	Alice	24	New York	85
1	Bob	27	Los Angeles	90
2	Charlie	22	Chicago	88
3	David	32	Houston	95

	Name	Age	City	Score
0	Alice	24	New York	85
1	Bob	28	Los Angeles	90
2	Charlie	22	Chicago	88
3	David	32	Houston	95

	Name	Age	Score
0	Alice	24	85
1	Bob	28	90
2	Charlie	22	88
3	David	32	95

	Name	Age	Score
0	Alice	24	85
1	Bob	28	90
2	Charlie	22	88

5. Create a new DataFrame: data = { 'Department': ['HR', 'IT', 'HR', 'IT'], 'Salary': [30000, 50000, 35000, 55000], 'Experience': [2, 5, 3, 6] } Write code to:
- a) Group by Department and find average Salary
 - b) Find maximum Experience in each Department
 - c) Calculate total Salary paid

```

In [37]: import pandas as pd
import numpy as np
data = {
'Department': ['HR', 'IT', 'HR', 'IT'],
'Salary': [30000, 50000, 35000, 55000],
'Experience': [2, 5, 3, 6]
}
df=pd.DataFrame(data);
a=df.groupby('Department')['Salary'].mean()
print(a,"\n")
c=df.groupby('Department')['Experience'].max()
print(c,"\n")
b=df['Salary'].sum()
print(b,"\n")

```

```
Department
HR      32500.0
IT      52500.0
Name: Salary, dtype: float64
```

```
Department
HR      3
IT      6
Name: Experience, dtype: int64
```

170000

6. Given a DataFrame with missing values: data = { 'Student': ['John', 'Emma', 'Sam', 'Olivia'], 'Marks': [80, None, 75, 90] } Write code to:
- a) Fill missing marks with 0
 - b) Drop rows with missing values
 - c) Sort the DataFrame by Marks in descending order

```
In [40]: import pandas as pd
import numpy as np
data = {
    'Student': ['John', 'Emma', 'Sam', 'Olivia'],
    'Marks': [80, np.nan, 75, 90]
}
df=pd.DataFrame(data);
b=df.dropna()
print(b, "\n")
df=df.fillna(0)
print(df, "\n")
df=df.sort_values(by=['Marks'], ascending=[False])
print(df, "\n")
```

```
Student  Marks
0   John    80.0
2   Sam     75.0
3 Olivia   90.0
```

```
Student  Marks
0   John    80.0
1   Emma     0.0
2   Sam     75.0
3 Olivia   90.0
```

```
Student  Marks
3 Olivia   90.0
0   John    80.0
2   Sam     75.0
1   Emma     0.0
```

8. You are given the following DataFrame: import pandas as pd data = { 'Product': ['Laptop', 'Tablet', 'Smartphone', 'Monitor', 'Keyboard'], 'Price': [70000, 30000, 25000, 15000, 2000], 'Stock': [10, 25, 50, 15, 100] } df = pd.DataFrame(data) Task: Write a line of code using .loc[] to display the details of the first and third products in the DataFrame. Expected Output: Product Price Stock 0 Laptop 70000 10 2 Smartphone 25000 50

```
In [45]: import pandas as pd
data = {
    'Product': ['Laptop', 'Tablet', 'Smartphone', 'Monitor', 'Keyboard'],
    'Price': [70000, 30000, 25000, 15000, 2000],
    'Stock': [10, 25, 50, 15, 100]
}
df = pd.DataFrame(data)
print(df.loc[[0, 2]])
```

	Product	Price	Stock
0	Laptop	70000	10
2	Smartphone	25000	50

9) You are given the following data: import pandas as pd data = { 'Subject': ['Math', 'Science', 'English'], 'Marks': [88, 92, 85] } Task: Create a DataFrame from the above data and assign custom row labels: 'Student1', 'Student2', and 'Student3' using the index argument. Then, using .loc[], print the marks obtained by 'Student2'. Expected Output: Subject Science Marks 92 Name: Student2, dtype: object

```
In [48]: import pandas as pd
data = {
    'Subject': ['Math', 'Science', 'English'],
    'Marks': [88, 92, 85]
}
df = pd.DataFrame(data, index=['Student1', 'Student2', 'Student3'])
print(df, "\n")
df=df.loc['Student2']
print(df, "\n")
```

	Subject	Marks
Student1	Math	88
Student2	Science	92
Student3	English	85

Subject	Science
Marks	92

Name: Student2, dtype: object

7. For any DataFrame:

- a) Save it as a CSV file named students.csv
- b) Load a CSV file named employees.csv
- c) Set Name as the index
- d) Reset the index

```
In [50]: import pandas as pd
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David'],
    'Age': [24, 27, 22, 32],
    'City': ['New York', 'Los Angeles', 'Chicago', 'Houston']
}
df=pd.DataFrame(data);
df.to_csv('students.csv', index=False)
print("csv files'output.csv'created successful.\n")
df = pd.read_csv('students.csv')
print(df, "\n")
```

```
csv files'output.csv' created successful.
```

	Name	Age	City
0	Alice	24	New York
1	Bob	27	Los Angeles
2	Charlie	22	Chicago
3	David	32	Houston

```
In [ ]:
```