

**Pandas Exam Paper 1 (Total 30 Questions - 2 Marks Each)**



**Section A: Data Creation and Importing (7 Questions)**

1. **Creating a DataFrame**   
Create a DataFrame using pd.DataFrame() with columns: 'Name', 'Age', and 'City' and values for 3 individuals.

**Answer:**import pandas as pd

data = {

'Name': [Sachin, 'Ramesh', 'Suresh'],

'Age': [15, 30, 35],

'City': ['India', 'London', 'Kolkata']

}

df = pd.DataFrame(data)

print(df)

2. **Reading CSV File**   
 Write the command to read a CSV file named data.csv into a DataFrame.

**Answer:**import pandas as pd

df = pd.read\_csv('data.csv')

3. **Reading Excel File**   
 How would you load data from an Excel file called data.xlsx into a DataFrame?

**Answer:**

import pandas as pd

df = pd.read\_excel('data.xlsx')

4. **Reading JSON File**   
 Load a JSON file named data.json into a DataFrame.

**Answer:**

import pandas as pd

df= pd.read\_json('data.json')

5. **Reading HTML Table**   
 Parse an HTML file containing a table and return it as a DataFrame.

**Answer:**

import pandas as pd

df = pd.read\_html('file.html')[0]

6. **Creating DataFrame from a Dictionary**   
Create a DataFrame using a dictionary with two columns: 'Product' and 'Price', containing 3 items.

**Answer:**

import pandas as pd

data = {

'Product': ['Kiwi', 'Banana', 'Apple'],

'Price': [120.3, 22.5, 32.5]

}

dfr = pd.DataFrame(data)

print(dfr)

7. **Exploring DataFrame from CSV**   
After loading a CSV into a DataFrame, what command would you use to see the first 5 rows?

**Answer:**



**Section B: Data Inspection (7 Questions)**

8. **Viewing First Few Rows**   
 Use the appropriate command to display the first 10 rows of a DataFrame df .

**Answer:**

df.head(10)

9. **Viewing Last Few Rows**   
 Show the last 3 rows of the DataFrame df .

**Answer:** df.tail(3)

10. **Checking DataFrame Information**   
Which command provides concise information about the DataFrame, such as data types and memory usage?

**Answer:  
df.info()**

11. **Descriptive Statistics**   
 How do you generate descriptive statistics like mean, median, and standard deviation for

numeric columns in a DataFrame?   
**Answer: df.describe()**

12. **Checking Data Types**   
 What command returns the data types of each column in the DataFrame?

**Answer:**

**df.dtypes**

13. **Checking DataFrame Shape**   
 How do you find the number of rows and columns in the DataFrame?

**Answer: df.shape()**

14. **DataFrame Summary**   
 Explain what df.info() does and what kind of information it provides.

**Answer:**The df.info() method in pandas is used to provide a concise summary of a DataFrame. It gives you an overview of the structure of the DataFrame, which can help you quickly understand its contents. Here’s what df.info() does and the kind of information it provides:

df.info() is a useful method to quickly inspect the structure and integrity of a DataFrame. It helps you to:

Identify missing values,

Check the data types of each column,

Understand the size of the DataFrame, and

Assess memory usage.



**Section C: Indexing and Selecting Data (8 Questions)**

15. **Setting an Index**   
 Set the 'ID' column as the index for the DataFrame df .

**Answer:**df = df.set\_index('ID')

16. **Resetting an Index**   
 How do you reset the index of the DataFrame and return it to the default integer index?

**Answer:**

df = df.reset\_index()

17. **Selecting Data by Position**   
 Retrieve the third row of the DataFrame using iloc[] .

**Answer: xyz\_1 = df.iloc[2]**

18. **Selecting Data by Label**   
 Use loc[] to access all rows where the 'Age' column is greater than 30.

**Answer: fltdf = df.loc[df['Age'] > 30]**

19. **Querying the DataFrame**   
 Use query() to select rows where the 'Salary' is greater than 50000.

**Answer:**

**filtered\_df = df.query('Salary > 50000')**

20. **Sorting Values**   
 Sort the DataFrame df by the 'Price' column in ascending order.

**Answer:**

**df\_sorted = df.sort\_values(by='Price', ascending=True)**

21. **Selecting Top N Rows by Value**   
 Select the top 3 rows with the highest values in the 'Marks' column using nlargest() .

**Answer: top\_3\_marks = df.nlargest(3, 'Marks')**

22. **Selecting Smallest N Rows by Value**   
 Use nsmallest() to return the bottom 2 rows based on the 'Age' column.

**Answer: bottom\_2\_age = df.nsmallest(2, 'Age')**



**Section D: Data Cleaning (8 Questions)**

23. **Detecting Missing Values**   
 Write the command to detect missing values in the DataFrame df .

**Answer:**

**missing\_values = df.isnull()**

24. **Removing Missing Values**   
 Remove rows with missing values in the DataFrame df .

**Answer:**df\_cleaned = df.dropna()

25. **Filling Missing Values**   
 Fill missing values in the 'Salary' column with the mean salary value.

**Answer:**

df['Salary'] = df['Salary'].fillna(df['Salary'].mean())

26. **Dropping Duplicate Rows**   
 How do you remove duplicate rows from the DataFrame?

**Answer:**

df\_cleaned = df.drop\_duplicates()

27. **Replacing Values**   
 Replace all occurrences of the value 'M' in the 'Gender' column with 'Male'.

**Answer:**df['Gender'] = df['Gender'].replace('M', 'Male')

28. **Converting Data Types**   
 Convert the 'Age' column to integers using astype() .

**Answer:**

df['Age'] = df['Age'].astype(int)

29. **Handling Missing Values in Specific Column**   
 Remove rows where the 'Age' column contains missing values.

**Answer:**

df\_cleaned = df.dropna(subset=['Age'])

30. **Filling Missing Values Using Forward Fill**   
 Use the forward fill method to fill missing values in the DataFrame df .

**Answer:**df\_filled = df.fillna(method='ffill')

