

**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**

**# Creating the DataFrame**

**data = {**

**'Name': ['Alice', 'Bob', 'Charlie'],**

**'Age': [25, 30, 35],**

**'City': ['New York', 'Los Angeles', 'Chicago']**

**}**

**df = pd. DataFrame(data)**

**# Displaying the DataFrame**

**print(df)**

**Output:**

**Name Age City**

**0 Alice 25 New York**

**1 Bob 30 Los Angeles**

**2 Charlie 35 Chicago**

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

**Answer:** **import pandas as pd**

**# Reading the CSV file into a DataFrame**

**df = pd. read \_ csv('data.csv')**

**# Display the first few rows of the DataFrame**

**print(df. head())**

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

**Answer:** **import pandas as pd**

**# Reading the Excel file into a DataFrame**

**df = pd. read \_ excel('data.xlsx')**

**# Display the first few rows of the DataFrame**

**print(df .head())**

**df = pd.read\_excel("data.xlsx", sheet\_name="Sheet1")**

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

**Answer:** **import pandas as pd**

**# Reading the JSON file into a DataFrame**

**df = pd. read \_ json ('data . json')**

**# Display the first few rows of the DataFrame**

**print(df.head())**

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

**Answer:** **import pandas as pd**

**# Reading the table from an HTML file**

**dfs = pd. read \_html('data.html') # Returns a list of DataFrames**

**# Display the first table found in the HTML file**

**df = dfs[0] # Selecting the first table**

**print(df.head())**

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**

**# Creating the dictionary**

**data = {**

**'Product': ['Laptop', 'Phone', 'Tablet'],**

**'Price': [1000, 500, 300]**

**}**

**# Creating the DataFrame**

**df = pd. DataFrame(data)**

**# Displaying the DataFrame**

**print(df)**

**Output:**

**Product Price**

**0 Laptop 1000**

**1 Phone 500**

**2 Tablet 300**

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

**Answer:** **import pandas as pd**

**# Load the CSV file into a DataFrame**

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

**# Display the first 5 rows**

**print(df .head())**



**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:**

**import pandas as pd**

**# Display the first 10 rows**

**df.head(10)**

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

**Answer:** **import pandas as pd**

**# Show the last 3 rows**

**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:** **import pandas as pd**

**# Sample DataFrame**

**data = {**

**'Age': [25, 30, 35, 40, 45],**

**'Salary': [50000, 60000, 70000, 80000, 90000]**

**}**

**df = pd.DataFrame(data)**

**# Generate descriptive statistics**

**print(df.describe()) # Summary statistics including mean and std**

**# Individual statistics**

**print("Mean:\n", df.mean()) # Mean**

**print("Median:\n", df.median()) # Median**

**print("Standard Deviation:\n", df.std()) # Standard Deviation**

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

**Answer: df.shape**

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:** **Displays the number of non-null values in each column.**

* **Shows the data types of each column.**
* **Provides the memory usage of the DataFrame.**
* **Lists the total number of rows and columns.**



**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:**

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

**Answer:** **df.iloc[2]**

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

**Answer:** **df.query("Salary > 50000")**

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

**Answer: df = df.sort\_values(by='Price')**

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

**Answer:** **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:** **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:** **df.isnull()**

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

**Answer:** **df = 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 = 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 = 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:**

