

**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**

**import numpy as np**

**Data= { “Name” : [“Harshi”, “Sam”, “Sony”],**

**“Age”: [ 32, 33, 32],**

**“City”: [“New York”, “Dallas”, “Miami”]}**

**df= pd.Dataframe(data=Data)**

**display(df)**

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

**Answer:**

**file= pd.read\_csv(r"data.csv")**

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

**Answer:**

**file= pd.read\_xlsx(r”data.xlsx”)**

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

**Answer:**

**json\_file= r“data.json”**

**df= pd.read\_json(json\_file)**

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

**Answer:**

**html\_file\_path= r"https://worldpopulationreview.com/countries"**

**df\_html= pd.read\_html(html\_file\_path)**

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**

**import numpy as np**

**data1= { “Product” : [“Television”, “Mobiles”, “Laptop”],**

**“Price”: [ 110000, 35000, 65000] }**

**df= pd.Dataframe(data=data1)**

**display(df)**

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

**Answer:**

**file= pd.read\_csv(r"data.csv")**

**file.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:**

**file= pd.read\_csv(r"data.csv")**

**file.head(10)**

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

**Answer:**

**file= pd.read\_csv(r"data.csv")**

**file.tail(3)**

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

**Answer:**

**file= pd.read\_csv(r"data.csv")**

**file.info()**

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

numeric columns in a DataFrame?   
**Answer:**

**file= pd.read\_csv(r"data.csv")**

**file.describe()**

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

**Answer:**

**file= pd.read\_csv(r"data.csv")**

**file.dtypes**

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

**Answer:**

**file= pd.read\_csv(r"data.csv")**

**file.shape**

**file.rows**

**file.columns**

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

**Answer: It provides data types of each column in the data frame including memory usage of the data frame.**



**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.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.reset\_index()**

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

**Answer:**

**df.iloc[[2]]**

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

**Answer:**

**df.loc[df['Age'] > 30]**

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.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:**

**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.dropna(inplace=True)**

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

**Answer:**

**df['Salary'].fillna(df['Salary'].mean(), inplace=True)**

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

**Answer:**

**df.drop\_duplicates(inplace=True)**

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

**Answer:**

**df['Gender'].replace('M', 'Male', inplace=True)**

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.dropna(subset=['Age'], inplace=True)**

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

**Answer:**

**df.fillna(method='ffill', inplace=True)**

