Pandas & NumPy – Interview Questions & Answers (Clean One-Column Format)

SECTION 1: PANDAS INTERVIEW QUESTIONS

1. What is Pandas? Pandas is a Python library used for data manipulation and analysis. It provides Series and DataFrame objects.

2. Difference between Series and DataFrame. Series is 1D labeled data; DataFrame is 2D table with labeled axes.

3. How to read files in Pandas? pd.read_csv(), pd.read_excel(), pd.read_json(), pd.read_sql() etc.

4. Handling missing values: df.isnull(), df.dropna(), df.fillna(), ffill(), bfill()

5. loc vs iloc: loc = label-based; iloc = integer-based.

6. How to filter rows? df[df["Age"] > 30], df.query("Age > 30")

7. Apply, map, applymap: map() → Series element-wise apply() → row/column wise applymap() → DataFrame element-wise

8. Merge vs Join vs Concat: merge = SQL-like join on columns join = join on index concat = append rows/cols

9. GroupBy: df.groupby("col")["salary"].agg(["sum","mean","count"])

10. Improving performance: Use vectorization, categoricals, chunking, avoid Python loops.

SECTION 2: NUMPY INTERVIEW QUESTIONS

1. What is NumPy? A Python library for numerical computing using fast N-dimensional arrays.

2. ndarray? NumPy's main array object.

3. Why NumPy is faster? Uses C backend, contiguous memory, vectorization, SIMD.

4. Broadcasting: Automatic expansion of arrays of different shapes during operations.

5. Create arrays: np.zeros, np.ones, arange, linspace, random

6. arange vs linspace: arange uses step; linspace uses number of samples.

7. Reshaping: arr.reshape(), arr.flatten()

8. Slicing: arr[0:5], arr[:,1], arr[1:3, 2:4]

9. Handle NaN: np.isnan, np.nanmean, np.nan_to_num

10. Copy vs View: copy() independent; view() shares memory.

11. Matrix multiplication: A @ B or np.dot(A,B)

12. Performance tips: Vectorize, proper dtypes, inplace operations.

SECTION 3: CODING TASKS

1. Replace negative values: df[df < 0] = 0

2. Remove duplicates: df.drop_duplicates(subset=["id"])

3. Top 5 salaries: df.nlargest(5, "salary")

4. Convert to datetime: pd.to_datetime(df["date"])

5. Reverse NumPy array: arr[::-1]

6. Unique values: np.unique(arr)

END OF DOCUMENT.