Pandas - Interview - Questions - Answers

1. What is Pandas?

→ A Python library for data manipulation and analysis, built on NumPy.

2. Main data structures in Pandas?

→ `Series` (1D) and `DataFrame` (2D).

3. Difference between Series and DataFrame?

→ Series is 1D labeled array; DataFrame is a 2D labeled table.

4. How to read CSV files in Pandas?

→ Using `pd.read_csv('file.csv')`.

5. How to view top/bottom rows of a DataFrame?

→ `df.head()` and `df.tail()`.

6. How to check shape and info of a DataFrame?

→ `df.shape`, `df.info()`.

7. How to get basic statistics of numeric columns?

→ `df.describe()`.

8. How to select a single column?

→ `df['col_name']` or `df.col_name`.

9. How to select multiple columns?

→ `df[['col1', 'col2']]`.

10. How to filter rows by condition?

→ `df[df['col'] > value]`.

11. Difference between `.loc[]` and `.iloc[]`?

→ `.loc[]` → label-based; `.iloc[]` → integer position-based.

12. How to check for null values?

→ `df.isnull().sum()`.

13. How to fill missing values?

→ `df.fillna(value)`.

14. How to drop missing values?

→ `df.dropna()`.

15. How to sort data in Pandas?

→ `df.sort_values(by='col')`.

16. How to rename columns?

→ `df.rename(columns={'old': 'new'}, inplace=True)`.

17. How to add a new column?

→ `df['new_col'] = df['col1'] + df['col2']`.

18. How to delete a column?

→ `df.drop('col_name', axis=1, inplace=True)`.

19. How to group data by a column?

→ `df.groupby('col').mean()`.

20. How to get unique values in a column?

→ `df['col'].unique()`.

21. Difference between `unique()` and `nunique()`?

→ `unique()` returns values; `nunique()` counts them.

22. How to reset the index of a DataFrame?

→ `df.reset_index(drop=True, inplace=True)`.

23. How to merge two DataFrames?

→ `pd.merge(df1, df2, on='col')`.

24. How to concatenate DataFrames?

→ `pd.concat([df1, df2])`.

25. How to export a DataFrame to CSV?

→ `df.to_csv('file.csv', index=False)`.

26. Create a DataFrame from a dictionary

```
import pandas as pd
data = {'Name': ['A', 'B', 'C'], 'Age': [23, 25, 21]}
df = pd.DataFrame(data)
print(df)
```

27. Select rows where Age > 22

```
print(df[df['Age'] > 22])
```

28. Add a new column based on condition

```
df['Status'] = df['Age'].apply(lambda x: 'Adult' if x >= 22 else
'Teen')
print(df)
```

29. Read only specific columns from a CSV

```
df = pd.read_csv('data.csv', usecols=['Name', 'Salary'])
```

30. Drop duplicate rows

```
df = df.drop_duplicates()
```

31. Replace specific values in a column

```
df['Gender'] = df['Gender'].replace({'M': 'Male', 'F': 'Female'})
```

32. Find max value in a column

```
print(df['Salary'].max())
```

33. Sort by multiple columns

```
df.sort_values(by=['Department', 'Salary'], ascending=[True, Fals
e])
```

34. Get correlation between columns

```
print(df.corr())
```

35. Filter rows with multiple conditions

```
df[(df['Age'] > 25) & (df['Salary'] > 40000)]
```

36. Group by and count

```
df.groupby('Department')['EmployeeID'].count()
```

37. Get column datatypes

```
print(df.dtypes)
```

38. Convert datatype of a column

```
df['Age'] = df['Age'].astype(float)
```

39. Create pivot table

```
pd.pivot_table(df, values='Salary', index='Department', aggfunc='m
ean')
```

40. Rename index of a DataFrame

```
df.index = ['A', 'B', 'C']
```

41. Reset and set new index

```
df.reset_index(inplace=True)
df.set_index('Name', inplace=True)
```

42. Apply custom function on column

```
df['Bonus'] = df['Salary'].apply(lambda x: x * 0.10)
```

43. Find rows with null values in specific column

```
df[df['Salary'].isnull()]
```

44. Fill missing with mean

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

45. Drop rows with any missing value

```
df.dropna(inplace=True)
```

46. Convert column to datetime

```
df['JoinDate'] = pd.to_datetime(df['JoinDate'])
```

47. Extract year and month from date

```
df['Year'] = df['JoinDate'].dt.year
df['Month'] = df['JoinDate'].dt.month
```

48. Value counts of a column

```
print(df['Department'].value_counts())
```

49. Apply string operations on column

```
df['Name'] = df['Name'].str.upper()
```

50. Merge and join examples

```
merged = pd.merge(df1, df2, how='inner', on='EmployeeID')
joined = df1.join(df2.set_index('EmployeeID'), on='EmployeeID')
```

51. Basic groupby with aggregation

```
import pandas as pd
df = pd.DataFrame({
    'Dept': ['HR','IT','HR','IT','Finance'],
    'Salary': [50000,60000,55000,65000,70000]
})
print(df.groupby('Dept')['Salary'].mean())
```

52. Groupby multiple columns

```
df['Level'] = ['L1','L2','L1','L2','L1']
print(df.groupby(['Dept','Level'])['Salary'].sum())
```

53. Count number of employees per group

```
print(df.groupby('Dept')['Salary'].count())
```

54. Aggregate multiple functions

```
print(df.groupby('Dept')['Salary'].agg(['sum','mean','max']))
```

55. Apply custom function on groups

```
def range_func(x):
    return x.max() - x.min()
print(df.groupby('Dept')['Salary'].apply(range_func))
```

56. Groupby and get first/last item

```
print(df.groupby('Dept')['Salary'].first())
print(df.groupby('Dept')['Salary'].last())
```

57. Filter groups based on condition

```
high_avg = df.groupby('Dept').filter(lambda x: x['Salary'].mean()
> 60000)
print(high_avg)
```

58. Transform groups and broadcast

```
df['Salary_norm'] = df.groupby('Dept')['Salary'].transform(lambda
x: x/x.mean())
print(df)
```

59. Get group sizes

```
print(df.groupby('Dept').size())
```

In [1]:

```
import pandas as pd

# Sample DataFrame
df = pd.DataFrame({
    'Dept': ['HR','IT','HR','IT','Finance'],
    'Salary': [50000,60000,55000,65000,70000],
    'Bonus': [5000,6000,5500,6500,7000]
})

# 1️⃣ Using a list of functions on a single column
print(df.groupby('Dept')['Salary'].agg(['sum', 'mean', 'max']))

# 2️⃣ Using a dict to apply different functions to different columns
print(df.groupby('Dept').agg({
    'Salary': ['sum', 'mean'],
    'Bonus': 'max'
}))

```

```
            sum     mean    max
Dept
Finance   70000  70000.0  70000
HR       105000  52500.0  55000
IT       125000  62500.0  65000
         Salary          Bonus
            sum     mean   max
Dept
Finance   70000  70000.0  7000
HR       105000  52500.0  5500
IT       125000  62500.0  6500
```

In [2]:

```python
import pandas as pd

# Sample DataFrame
df = pd.DataFrame({
    'Dept': ['HR','IT','HR','IT','Finance'],
    'Salary': [50000,60000,55000,65000,70000],
    'Bonus': [5000,6000,5500,6500,7000]
})

# Groupby with custom column names
agg_df = df.groupby('Dept').agg(
    Total_Salary=('Salary', 'sum'),
    Average_Salary=('Salary', 'mean'),
    Max_Bonus=('Bonus', 'max')
)

print(agg_df)

```

```
         Total_Salary  Average_Salary  Max_Bonus
Dept
Finance         70000         70000.0       7000
HR             105000         52500.0       5500
IT             125000         62500.0       6500
```

In [ ]:

```python

```