■ Pandas Functions with Working Examples

1. Creating Objects

- pd.Series([1,2,3]) # Creates a 1D array-like object
- pd.DataFrame({'A':[1,2],'B':[3,4]}) # Creates a 2D table
- pd.read_csv('file.csv') # Reads CSV file into DataFrame
- df.to_csv('out.csv') # Exports DataFrame to CSV

2. Viewing & Inspecting Data

- df.head() # Shows first 5 rows
- df.tail() # Shows last 5 rows
- df.info() # Summary of DataFrame
- df.describe() # Statistical summary of numeric columns
- df.shape # Returns (rows, cols)
- df.columns # Lists column names
- df.dtypes # Shows data types of columns

3. Selection & Indexing

- df['col'] # Select single column
- df[['col1','col2']] # Select multiple columns
- df.loc[0] # Select row by label/index name
- df.iloc[0] # Select row by position
- df.at[0,'col'] # Fast access by label
- df.iat[0,1] # Fast access by integer position

4. Data Cleaning

- df.isnull() # Detect missing values
- df.dropna() # Drop missing values
- df.fillna(0) # Fill missing values with 0
- df.duplicated() # Check for duplicates
- df.drop_duplicates() # Remove duplicate rows
- df.replace(5,10) # Replace 5 with 10
- df.astype(float) # Convert column datatype

5. Filtering & Querying

- df[df['col']>10] # Filter rows where col > 10
- df.query('col > 10 and col2 < 5') # SQL-like query

6. Sorting

- df.sort_values('col') # Sort by values in a column
- df.sort_index() # Sort by index

7. Aggregation & Statistics

- df.sum() # Sum of all numeric values
- df.mean() # Mean of numeric columns
- df.median() # Median of numeric columns
- df.min(), df.max() # Min and Max values
- df.std(), df.var() # Standard deviation and variance
- df.corr() # Correlation between columns
- df.count() # Non-null counts
- df.nunique() # Number of unique values
- df['col'].value_counts() # Frequency of unique values

8. Group & Pivot

- df.groupby('col').mean() # Group by and take mean
- df.agg({'col':['min','max']}) # Multiple aggregations
- df.pivot(index='A', columns='B', values='C') # Reshape data
- df.pivot_table(values='val', index='A', columns='B', aggfunc='sum')
- df.melt() # Unpivot wide to long format

9. Merging & Joining

- pd.concat([df1, df2]) # Concatenate along rows
- df.merge(df2, on='id') # SQL-style join
- df.join(df2) # Join on index

10. String Operations

- df['col'].str.lower() # Convert to lowercase
- df['col'].str.upper() # Convert to uppercase
- df['col'].str.contains('abc') # Check substring
- df['col'].str.replace('a','b') # Replace text
- df['col'].str.split('-') # Split strings

11. Datetime Functions

- pd.to_datetime(df['date']) # Convert to datetime
- df['date'].dt.year # Extract year
- df['date'].dt.month # Extract month
- df['date'].dt.day # Extract day
- df['date'].dt.strftime('%Y-%m') # Format date
- df.resample('M').mean() # Resample monthly and calculate mean

12. Reshaping & Reindexing

- df.T # Transpose rows and columns
- df.stack() # Stack columns into rows
- df.unstack() # Unstack rows into columns
- df.reset_index() # Reset index
- df.set_index('col') # Set a column as index
- df.reindex([0,1,2]) # Reindex rows

13. Advanced Functions

- df.apply(lambda x:x*2) # Apply function to each column/row
- df['col'].map(lambda x:x*2) # Apply function to Series
- df.applymap(lambda x:x*2) # Apply function element-wise
- df.pipe(lambda df: df.dropna()) # Chain functions

14. Window Functions

- df.rolling(window=3).mean() # Moving average over 3 rows
- df.expanding().sum() # Cumulative sum
- df.ewm(span=3).mean() # Exponential weighted mean

15. Exporting Data

- df.to_csv('file.csv') # Save DataFrame to CSV
- df.to_excel('file.xlsx') # Save to Excel
- df.to_json('file.json') # Save to JSON
- df.to_sql('table', conn) # Save to SQL database