PYTHON

Master Pandas Like a Pro! 🐼

A Pandas Cheat Sheet for Data Manupulation

TUSHAR WAGH

@tusharwagh

Swipe > >

Pandas

Pandas is a powerful Python library for data manipulation and analysis. It provides numerous methods for handling **Series**, **DataFrames**, and **Index** objects. Below is a categorized list of the most important Pandas methods.

---

1. DataFrame Creation & Inspection

- `pd.DataFrame(data, columns, index, dtype)` – Create a DataFrame
- `pd.Series(data, index, dtype)` – Create a Series
- `df.head(n)` – First `n` rows
- `df.tail(n)` – Last `n` rows
- `df.info()` – Summary of DataFrame
- `df.describe()` – Statistics summary
- `df.shape` – Dimensions (rows, columns)
- `df.dtypes` – Data types of each column
- `df.columns` – Column labels
- `df.index` – Row labels
- `df.memory_usage()` – Memory usage

---

2. Data Selection & Filtering

- `df['col']` – Select a column
- `df[['col1', 'col2']]` – Select multiple columns
- `df.iloc[row_idx, col_idx]` – Select by position
- `df.loc[row_label, col_label]` – Select by label
- `df[df['col'] > value]` – Filter rows
- `df.query('col > value')` – Query data
- `df.at[row_label, col_label]` – Access single value
- `df.iat[row_idx, col_idx]` – Fast access to single value

Swipe > >

Pandas

3. Data Cleaning & Handling Missing Values

- `df.isna()` /`df.isnull()` – Check for missing values
- `df.notna()` /`df.notnull()` – Check for non-missing values
- `df.fillna(value)` – Fill missing values
- `df.dropna()` – Drop missing values
- `df.dropna(axis=1)` – Drop columns with missing values
- `df.replace(old, new)` – Replace values
- `df.duplicated()` – Check for duplicate rows
- `df.drop_duplicates()` – Remove duplicate rows

---

4. Data Transformation

- `df.apply(func, axis=0/1)` – Apply function to columns/rows
- `df.map(func)` – Apply function element-wise (Series)
- `df.astype(dtype)` – Change data type
- `df.rename(columns={'old': 'new'})` – Rename columns
- `df.rename(index={'old': 'new'})` – Rename index
- `df.sort_values(by='col', ascending=True)` – Sort by column
- `df.sort_index()` – Sort by index
- `df.clip(lower, upper)` – Limit values
- `df.interpolate()` – Interpolate missing values

---

5. Aggregation & Grouping

- `df.count()` – Count non-null values
- `df.sum()` – Sum values
- `df.mean()` – Mean value
- `df.median()` – Median value
- `df.min()`/`df.max()` – Min/Max value
- `df.std()`/`df.var()` – Standard deviation & variance
- `df.cumsum()` – Cumulative sum

Swipe > >

Pandas

- `df.cumprod()` – Cumulative product
- `df.agg(['sum', 'mean'])` – Apply multiple aggregation functions
- `df.groupby('col').agg({'col2': 'sum'})` – Group & aggregate
- `df.pivot_table(values, index, columns, aggfunc)` – Pivot table

---

6. Joining & Merging

- df1.append(df2) – Append rows
- `df1.merge(df2, on='key', how='inner')` – Merge DataFrames
- `pd.concat([df1, df2], axis=0)`– Concatenate rows
- `pd.concat([df1, df2], axis=1)` – Concatenate columns
- `df.set_index('col')` – Set column as index
- `df.reset_index()` – Reset index

---

7. Pivoting & Reshaping

- `df.pivot(index, columns, values)` – Reshape data
- `df.melt(id_vars, value_vars)` – Unpivot data
- `df.stack()` – Stack columns to rows
- `df.unstack()` – Unstack rows to columns
- `df.T` – Transpose DataFrame
- `df.explode('list_column')` – Expand lists in a column

---

8. Time Series Methods

- `df['date_col'] = pd.to_datetime(df['date_col'])` – Convert to datetime
- `df.set_index('date_col')` – Set datetime index
- `df.resample('M').sum()` – Resample data
- df.shift(1) – Shift rows
- df.diff() – Compute difference

Swipe > >

Pandas

9. Input/Output (I/O) Operations

- `pd.read_csv('file.csv')` – Read CSV file
- `df.to_csv('file.csv', index=False)` – Write CSV file
- `pd.read_excel('file.xlsx')` – Read Excel file
- `df.to_excel('file.xlsx', index=False)` – Write Excel file
- `pd.read_json('file.json')` – Read JSON file
- `df.to_json('file.json')` – Write JSON file
- `pd.read_sql(query, connection)` – Read SQL data
- `df.to_sql(name, connection, if_exists='replace')` – Write to SQL

10. Statistical & Mathematical Operations

- `df.corr()` – Correlation matrix
- `df.cov()` – Covariance matrix
- `df.skew()` – Skewness
- `df.kurt()` – Kurtosis
- `df.rank()` – Rank values
- `df.mode()` – Most frequent values

11. Working with Categories

- `df['col'] = df['col'].astype('category')` – Convert to category
- `df['col'].cat.categories` – Get category labels
- `df['col'].cat.codes` – Get category codes
- `df['col'].cat.add_categories(['new'])` – Add category

Swipe > >

Pandas

12. Working with Strings

- `df['col'].str.lower()` / `.upper()` – Convert case
- `df['col'].str.strip()` – Trim spaces
- `df['col'].str.replace('old', 'new')` – Replace text
- `df['col'].str.contains('text')` – Check for substring
- `df['col'].str.extract(r'(pattern)')` – Extract regex pattern
- `df['col'].str.split('delimiter')` – Split strings

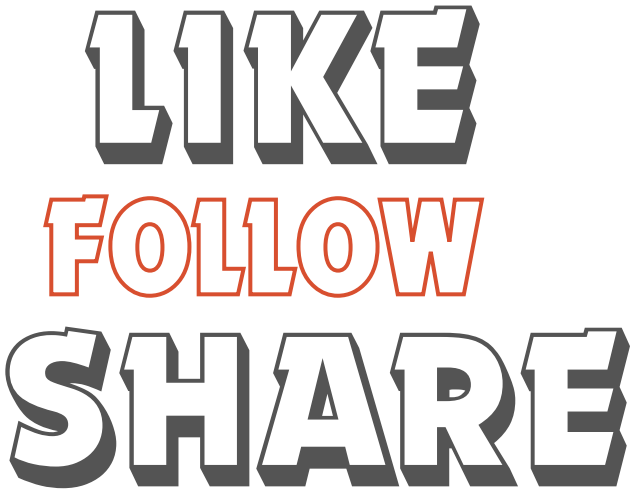
---

13. Working with MultiIndex

- `df.set_index(['col1', 'col2'])` – Set multiple indexes
- `df.reset_index()` – Reset multi-index
- `df.swaplevel()` – Swap index levels
- `df.sort_index(level=1)` – Sort by index level

TUSHAR WAGH

@tusharwagh



End