# Pandas Essentials Cheatsheet

**For Filtering Data**

## 1. Basics & I/O

```
pd.Series(data)
```

Create a Series

```
pd.DataFrame(data, columns=...)
```

Create a DataFrame

**Basic Attributes & Info:**

```
.shape
```

(rows, columns)

```
.columns
```

Column names

```
.dtypes
```

Data types of each column

```
.info()
```

Summary of DataFrame

```
.describe()
```

Statistical summary

**Preview Data:**

```
.head(n)
```

First n rows

```
.tail(n)
```

Last n rows

```
.sample(n)
```

Random sample

**Input / Output:**

```
pd.read_csv()` / `df.to_csv()
```

```
pd.read_excel()` / `df.to_excel()
```

```
pd.read_parquet()
```

```
pd.read_json()
```

```
pd.read_sql()
```

---

## 2. Indexing & Selection

```
df['col']
```

Select single column

```
df[['col1','col2']]
```

Select multiple columns

**Label-based (loc):**

```
df.loc[label]
```

```
df.loc[:, 'col']
```

```
df.loc[rows, cols]
```

**Position-based (iloc):**

```
df.iloc[position]
```

**Conditional Selection:**

```
df[df['age'] > 25]
```

```
df.query('age > 25')`
```

---

## 3. Data Cleaning (Most Important for ML)

**Missing Values:**

```
isna() / isnull
```

```
dropna()
```

```
fillna(value)
```

```
fillna(method='ffill')
```

```
interpolate()
```

**Duplicates & Renaming:**

```
drop_duplicates(subset=..., keep='first')
```

```
df.rename(columns={'old':'new'})
```

**String Operations (`.str`):**

```
.str.lower()`, `.str.upper()`, `.str.strip()
```

```
.str.replace()`, `.str.contains()
```

```
.str.split()
```

```
.str.extract(r'(\d+)')
```

```
.str.cat()
```

**Type Conversion:**

```
df['col'].astype('float32')
```

**Value Replacement:**

```
df.replace({old: new})
```

---

## 4. Operations & Aggregations

**Sorting:**

```
df.sort_values(by='col', ascending=False)
```

**Unique & Counts:**

```
value_counts()
```

```
unique()
```

```
nunique()
```

**GroupBy:**

```
groupby('col').agg({'col1': ['mean','sum'], 'col2':'count'})
```

**Apply Functions:**

```
df.apply(lambda x: ...)
```

```
df['col'].map(func)
```

```
df.map(func)` *(applymap is deprecated)*
```

**Other Utilities:**

```
pd.crosstab()
```

```
pd.cut()
```

```
pd.qcut()
```

---

## 5. Merging & Combining

```
pd.concat([df1, df2], axis=0/1, ignore_index=True)
```

```
pd.merge(df1, df2, on='key', how='inner/left/right/outer')
```

```
df1.join(df2, on='key', how='left')
```

**Pivot Tables:**

```
pivot_table(values, index, columns, aggfunc='mean')
```

---

## 6. Reshaping

```
df.melt(id_vars=..., value_vars=...)
```

```
df.stack()
```

```
df.unstack()
```

```
df.pivot(index, columns, values)
```

**MultiIndex:**

```
df.set_index(['col1','col2'])
```

```
df.reset_index()
```

---

## 7. Time Series (Very Useful in ML)

```
pd.to_datetime(df['date'])v
```

**Datetime Accessor:**

```
.dt.year
```

```
.dt.month
```

```
.dt.day_name()
```

**Resampling:**

```
df.set_index('date').resample('D').mean()
```

**Rolling Window:**

```
df['col'].rolling(window=7).mean()
```

**Shifting & Changes:**

```
df['col'].shift(1)
```

```
df['col'].pct_change()
```

---

## 💡 Bonus Tips

- Memory optimization:

```
df['cat_col'] = df['cat_col'].astype('category')
```

- Prefer vectorized operations over loops
- Show all columns:

```
pd.options.display.max_columns = None
```

---