

pandas, the 23 lines you reuse forever.

D8A.ACADEMY/RESOURCES/PANDAS-SHEET

LOAD & LOOK

<code>pd.read_csv('f.csv')</code>	in
<code>df.head()</code> · <code>df.shape</code>	peek
<code>df.info()</code>	types + nulls
<code>df.describe()</code>	stats in one call

SELECT & FILTER

<code>df[['a', 'b']]</code>	columns
<code>df.loc[rows, cols]</code>	by label
<code>df[df.x > 0]</code>	boolean mask
<code>df.query('x > 0')</code>	readable mask

TRANSFORM

<code>df.assign(y=...)</code>	new column
<code>df['x'].astype(int)</code>	cast
<code>df.rename(columns=...)</code>	rename
<code>df.sort_values('x')</code>	sort

GROUP & JOIN

<code>df.groupby('k')</code>	← 80% of every analysis.
<code>.agg(rev=("r", "sum"))</code>	named result
<code>df.merge(d2, on='k')</code>	SQL join
<code>pd.concat([a,b])</code>	stack

CLEAN

<code>df.drop_duplicates()</code>	dedupe
<code>df.dropna()</code> · <code>fillna(0)</code>	nulls
<code>df['x'].str.strip()</code>	trim
<code>df['x'].str.lower()</code>	case

DATES

<code>pd.to_datetime(s)</code>	parse first
<code>s.dt.month</code> · <code>s.dt.year</code>	parts
<code>df.resample('MS').sum()</code>	monthly

WORTH MEMORIZING

01 The groupby you actually write

Named aggregations: readable results, no MultiIndex surprises.

```
summary = (df
    .groupby("country")
    .agg(orders=("id", "count"),
        revenue=("amount", "sum"),
        avg_basket=("amount", "mean"))
    .sort_values("revenue", ascending=False))
```

→ Each output column is named on the spot. Chain with parentheses, read it top to bottom.

02 Trap: the copy that was not one

The warning everyone ignores until a column silently never updates.

```
# SettingWithCopyWarning bait:
sub = df[df.x > 0]
sub["flag"] = 1 # may edit a copy

# do this instead:
sub = df[df.x > 0].copy()
sub["flag"] = 1 # yours, for sure
```

→ Filtering returns a view or a copy, pandas decides. An explicit `.copy()` ends the ambiguity.