FOUNDATION LESSON

WEEK 4 · LESSON 8

Load and inspect a CSV table

Specify important types and verify required columns before analysis

THE IDEA BEHIND THE OPERATION

MENTAL MODEL

The idea behind the operation

Specify important types and verify required columns before analysis

PRESERVE INVOICE IDENTIFIERS

CODE WALKTHROUGH W04-L08-S03

```
01  import pandas as pd
02  df = pd.read_csv("data/invoices.csv", encoding="utf-8",
03                   dtype={"invoice_id": "string", "units": "int64
↪   "})
04  print(df.head(2).to_string(index=False))
05  print("Shape:", df.shape)
06  print("IDs:", df["invoice_id"].tolist())
```

WHAT TO TRACK

Invoice identifiers are text, including leading zeros.

PREDICT THE RESULT

Predict the output.
Explain each operation.
Pause before continuing.

PRESERVE INVOICE IDENTIFIERS — RESULT

CODE WALKTHROUGH W04-L08-S04

```
01  import pandas as pd
02  df = pd.read_csv("data/invoices.csv", encoding="utf-8",
03                   dtype={"invoice_id": "string", "units": "int64
↪   "})
04  print(df.head(2).to_string(index=False))
05  print("Shape:", df.shape)
06  print("IDs:", df["invoice_id"].tolist())
```

WHAT TO TRACK

Invoice identifiers are text, including leading zeros.

VALIDATE THE RESULT

```
invoice_id product  units
price       date
       001       A      2
12.0 2026-01-01
       002       B      5
3.0 2026-01-02
Shape: (3, 5)
IDs: ['001', '002', '003']
```

INSPECT TYPES AND NUMERICAL SUMMARIES

CODE WALKTHROUGH W04-L08-S05

```
01  import pandas as pd
02  df = pd.read_csv("data/invoices.csv", encoding="utf-8",
03                   dtype={"invoice_id": "string"})
04  print(df.dtypes.to_string())
05  print(df["units"].describe().to_string())
06  # Run df.info() in the notebook for the full structural report.
```

WHAT TO TRACK

Describe summarises; it does not validate the business rules.

PREDICT THE RESULT

Predict the output.
Explain each operation.
Pause before continuing.

INSPECT TYPES AND NUMERICAL SUMMARIES — RESULT

CODE WALKTHROUGH W04-L08-S06 1/2

```
01  import pandas as pd
02  df = pd.read_csv("data/invoices.csv", encoding="utf-8",
03                   dtype={"invoice_id": "string"})
04  print(df.dtypes.to_string())
05  print(df["units"].describe().to_string())
06  # Run df.info() in the notebook for the full structural report.
```

WHAT TO TRACK

Describe summarises; it does not validate the business rules.

VALIDATE THE RESULT

```
invoice_id    string[python]
product               object
units                  int64
price                float64
date                  object
count    3.000000
mean     2.333333
std      2.516611
min      0.000000
25%      1.000000
50%      2.000000
75%      3.500000
```

INSPECT TYPES AND NUMERICAL SUMMARIES — RESULT

CODE WALKTHROUGH W04-L08-S06 2/2

WHAT TO TRACK

Describe summarises; it does not validate the business rules.

VALIDATE THE RESULT

```
max       5.000000
```

A REQUIRED COLUMN IS ABSENT

CODE WALKTHROUGH W04-L08-S07

```
01  import pandas as pd
02  df = pd.read_csv("data/missing_column.csv", encoding="utf-8")
03  required = {"invoice_id", "units", "price"}
04  missing = required - set(df.columns)
05  if missing:
06      raise ValueError("missing required columns: " + ", ".join(s
↪   orted(missing)))
07  print("Schema accepted")
```

WHAT TO TRACK

Require invoice_id, units and price before calculations.

PREDICT THE RESULT

Predict the output.
Explain each operation.
Pause before continuing.

A REQUIRED COLUMN IS ABSENT — RESULT

CODE WALKTHROUGH W04-L08-S08

```
01  import pandas as pd
02  df = pd.read_csv("data/missing_column.csv", encoding="utf-8")
03  required = {"invoice_id", "units", "price"}
04  missing = required - set(df.columns)
05  if missing:
06      raise ValueError("missing required columns: " + ", ".join(s
↪   orted(missing)))
07  print("Schema accepted")
```

WHAT TO TRACK

Require invoice_id, units and price before calculations.

VALIDATE THE RESULT

```
ValueError: missing required
columns: price
```

GUIDED MODIFICATION

LEARNER PRACTICE

Guided modification

Load invoices.csv with invoice_id as string. Print the first two rows, the shape and column names. Check that units and price are present.

PREDICT · MODIFY · RUN

01 Restate the supplied rules.

02 Predict the result.

03 Modify and run the code.

04 Compare actual output.

05 Explain any difference.

INDEPENDENT APPLICATION

LEARNER PRACTICE

Independent application

Run info on the supplied invoices and explain why it returns None. Then load missing_column.csv and report the missing price column explicitly without fabricating values.

PREDICT · MODIFY · RUN

01 Restate the supplied rules.

02 Predict the result.

03 Modify and run the code.

04 Compare actual output.

05 Explain any difference.

RECAP AND NEXT STEP

Recap and next step

Protect identifier meaning, inspect structure and apply explicit schema checks. A preview and summary support understanding but do not certify data quality. Next, we select labelled rows and positional rows deliberately.

UNDERSTAND APPLY VALIDATE EXPLAIN