FOUNDATION LESSON

WEEK 4 · LESSON 9

Select with loc and iloc

loc selects labels; iloc selects integer positions

THE IDEA BEHIND THE OPERATION

MENTAL MODEL

The idea behind the operation

loc selects labels; iloc selects integer positions

LABEL THIRTY AND POSITION ONE

CODE WALKTHROUGH W04-L09-S03

```
01  import pandas as pd
02  df = pd.DataFrame({"units": [5, 8, 2]}, index=[10, 30, 50])
03  print("Label:", df.loc[30, "units"])
04  print("Position:", df.iloc[1, 0])
05  print("Label slice:", df.loc[10:30, "units"].tolist())
06  print("Position slice:", df.iloc[0:1, 0].tolist())
```

WHAT TO TRACK

Index labels are 10, 30 and 50.

PREDICT THE RESULT

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

LABEL THIRTY AND POSITION ONE — RESULT

CODE WALKTHROUGH W04-L09-S04

```
01  import pandas as pd
02  df = pd.DataFrame({"units": [5, 8, 2]}, index=[10, 30, 50])
03  print("Label:", df.loc[30, "units"])
04  print("Position:", df.iloc[1, 0])
05  print("Label slice:", df.loc[10:30, "units"].tolist())
06  print("Position slice:", df.iloc[0:1, 0].tolist())
```

WHAT TO TRACK

Index labels are 10, 30 and 50.

VALIDATE THE RESULT

```
Label: 8
Position: 8
Label slice: [5, 8]
Position slice: [5]
```

ONE COLUMN OR A ONE-COLUMN TABLE

CODE WALKTHROUGH W04-L09-S05

```
01  import pandas as pd
02  df = pd.DataFrame({"product": ["A", "B"], "units": [5, 8]})
03  column = df["units"]
04  table = df[["units"]]
05  print(type(column).__name__, column.shape)
06  print(type(table).__name__, table.shape)
07  print(df.loc[0:1, ["product", "units"]].to_string(index=False))
```

WHAT TO TRACK

A column-name list preserves a DataFrame result.

PREDICT THE RESULT

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

ONE COLUMN OR A ONE-COLUMN TABLE — RESULT

CODE WALKTHROUGH W04-L09-S06

```
01  import pandas as pd
02  df = pd.DataFrame({"product": ["A", "B"], "units": [5, 8]})
03  column = df["units"]
04  table = df[["units"]]
05  print(type(column).__name__, column.shape)
06  print(type(table).__name__, table.shape)
07  print(df.loc[0:1, ["product", "units"]].to_string(index=False))
```

WHAT TO TRACK

A column-name list preserves a DataFrame result.

VALIDATE THE RESULT

```
Series (2,)
DataFrame (2, 1)
product  units
      A      5
      B      8
```

A MISSING LABEL DOES NOT MEAN FIRST POSITION

CODE WALKTHROUGH W04-L09-S07

```
01  import pandas as pd
02  df = pd.DataFrame({"units": [5, 8]}, index=[10, 30])
03  print(df.loc[0, "units"])
```

WHAT TO TRACK

Zero is not an index label in this table.

PREDICT THE RESULT

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

A MISSING LABEL DOES NOT MEAN FIRST POSITION — RESULT

CODE WALKTHROUGH W04-L09-S08

```
01  import pandas as pd
02  df = pd.DataFrame({"units": [5, 8]}, index=[10, 30])
03  print(df.loc[0, "units"])
```

WHAT TO TRACK

Zero is not an index label in this table.

VALIDATE THE RESULT

```
KeyError: 0
```

GUIDED MODIFICATION

LEARNER PRACTICE

Guided modification

Create values [7, 9, 11] with labels [100, 200, 300]. Select label 200 and position one. Compare loc[100:200] with iloc[0:1].

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

Load invoices.csv, then select invoice_id and units as a DataFrame. Select the final row by position and explain why its default label should not be treated as the invoice identifier.

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

Separate labels from positions and predict result type and shape. A missing label needs an intentional repair. Next, filtering and sorting will change row order while often retaining old index labels.

UNDERSTAND APPLY VALIDATE EXPLAIN