FOUNDATION LESSON

WEEK 3 · LESSON 7

Organise reusable code into modules

Imports and useful standard-library tools

UNDERSTAND THE IDEA FIRST

MENTAL MODEL

Understand the idea first

Think of a module as a labelled toolbox. import math makes the toolbox available under the name math. math.ceil selects one tool from it. Importing does execute a module’s top-level statements when it is first loaded in a process, so keep demonstration prints out of a reusable module. Use module-qualified names to make the origin of a function clear.

USE A STANDARD-LIBRARY MODULE

CODE WALKTHROUGH W03-L07-S03

```
01  import math
02
03  print(math.ceil(4.2))
04  print(math.floor(4.2))
```

WHAT TO TRACK

Ceil rounds toward positive infinity; floor toward negative infinity.

PREDICT THE RESULT

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

USE A STANDARD-LIBRARY MODULE — RESULT

CODE WALKTHROUGH W03-L07-S04

```
01  import math
02
03  print(math.ceil(4.2))
04  print(math.floor(4.2))
```

WHAT TO TRACK

Ceil rounds toward positive infinity; floor toward negative infinity.

VALIDATE THE RESULT

```
5
4
```

KEEP A LOCAL KPI MODULE SMALL

CODE WALKTHROUGH W03-L07-S05

```
01  import kpi_tools
02
03  print(kpi_tools.profit(1200, 800))
04  print(kpi_tools.profit(0, 0))
```

WHAT TO TRACK

Supplied dependency kpi_tools.py defines profit(revenue, cost).

PREDICT THE RESULT

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

KEEP A LOCAL KPI MODULE SMALL — RESULT

CODE WALKTHROUGH W03-L07-S06

```
01  import kpi_tools
02
03  print(kpi_tools.profit(1200, 800))
04  print(kpi_tools.profit(0, 0))
```

WHAT TO TRACK

Supplied dependency kpi_tools.py defines profit(revenue, cost).

VALIDATE THE RESULT

```
400
0
```

DIAGNOSE AN UNAVAILABLE MODULE

CODE WALKTHROUGH W03-L07-S07

```
01  import aisynapse_missing_demo_module
```

WHAT TO TRACK

Deliberate missing dependency; this name is fictional.

PREDICT THE RESULT

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

DIAGNOSE AN UNAVAILABLE MODULE — RESULT

CODE WALKTHROUGH W03-L07-S08

```
01  import aisynapse_missing_demo_module
```

WHAT TO TRACK

Deliberate missing dependency; this name is fictional.

VALIDATE THE RESULT

```
ModuleNotFoundError: No module named 'aisynapse_missing_demo_module'
```

PAUSE AND APPLY

LEARNER PRACTICE

Pause and apply

Open the supplied kpi_tools.py file. Add a revenue(quantity, price) function returning the product. In a separate runner, import kpi_tools and print revenue for three items priced at twenty. Keep demonstration prints in the runner.

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

Modules organise reusable definitions. Imports make their names available, and module-qualified calls show where a tool comes from. Keep local dependency files with the runner and avoid standard-library name collisions. Next, we use pathlib to work with portable file paths.

UNDERSTAND APPLY VALIDATE EXPLAIN