FOUNDATION LESSON

WEEK 6 · LESSON 9

Configuration, secrets and reusable functions

Configuration controls reproducible choices; secrets grant access and must stay outside published files.

THE MENTAL MODEL

MENTAL MODEL

The mental model

Configuration controls reproducible choices; secrets grant access and must stay outside published files.

LOAD AND VALIDATE A PARAMETER

CODE WALKTHROUGH W06-L09-S03

```
01 import json,math
02 config=json.loads('{"cost_factor":1.1,"period":"2025-Q1"}')
03 factor=config["cost_factor"]
04 if isinstance(factor,bool) or not isinstance(factor,(int,float)
↪ ):
05     raise ValueError("cost_factor must be numeric")
06 if not math.isfinite(factor) or factor<0:
07     raise ValueError("cost_factor must be finite and nonnegativ
↪ e")
08 print("Period:",config["period"],"Cost factor:",factor)
```

WHAT TO TRACK

Load and validate a parameter

PREDICT THE RESULT

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

LOAD AND VALIDATE A PARAMETER · RESULT

CODE WALKTHROUGH W06-L09-S04

```
01 import json,math
02 config=json.loads('{"cost_factor":1.1,"period":"2025-Q1"}')
03 factor=config["cost_factor"]
04 if isinstance(factor,bool) or not isinstance(factor,(int,float)
↪ ):
05     raise ValueError("cost_factor must be numeric")
06 if not math.isfinite(factor) or factor<0:
07     raise ValueError("cost_factor must be finite and nonnegativ
↪ e")
08 print("Period:",config["period"],"Cost factor:",factor)
```

WHAT TO TRACK

Load and validate a parameter

VALIDATE THE RESULT

```
Period: 2025-Q1 Cost factor: 1.1
```

APPLY THE SUPPLIED REUSABLE PIPELINE

CODE WALKTHROUGH W06-L09-S05

```
from pathlib import Path
import sys
sys.path.insert(0,str(Path("src").resolve()))
from pipeline_tools import compute_metrics, summarise
import pandas as pd
d=pd.DataFrame({"entity":["A","A"],"month":["2025-01","2025-02"],"revenue":[100,120],"cost":[70,80],"cfo":[25,35],"capex":[5,10]})
config={"cost_factor":1.1}
r=compute_metrics(d,cost_factor=config["cost_factor"])
print(summarise(r)["profit"].to_dict())
```

WHAT TO TRACK

Dependency: supplied learner/src/pipeline_tools.py

PREDICT THE RESULT

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

APPLY THE SUPPLIED REUSABLE PIPELINE · RESULT

CODE WALKTHROUGH W06-L09-S06

```
from pathlib import Path
import sys
sys.path.insert(0,str(Path("src").resolve()))
from pipeline_tools import compute_metrics, summarise
import pandas as pd
d=pd.DataFrame({"entity":["A","A"],"month":["2025-01","2025-02"],"revenue":[100,120],"cost":[70,80],"cfo":[25,35],"capex":[5,10]})
config={"cost_factor":1.1}
r=compute_metrics(d,cost_factor=config["cost_factor"])
print(summarise(r)["profit"].to_dict())
```

WHAT TO TRACK

Dependency: supplied learner/src/pipeline_tools.py

VALIDATE THE RESULT

```
{'A': 55.0}
```

AN ABSENT OPTIONAL CREDENTIAL

CODE WALKTHROUGH W06-L09-S07

```
01  import os
02  token=os.environ.get("AISYNAPSE_OPTIONAL_API_KEY")
03  if token:
04      print("Optional credential present; value is not displayed"
↪   )
05  else:
06      print("Offline mode: no credential required")
```

WHAT TO TRACK

An absent optional credential

PREDICT THE RESULT

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

AN ABSENT OPTIONAL CREDENTIAL · RESULT

CODE WALKTHROUGH W06-L09-S08

```
01  import os
02  token=os.environ.get("AISYNAPSE_OPTIONAL_API_KEY")
03  if token:
04      print("Optional credential present; value is not displayed"
↪   )
05  else:
06      print("Offline mode: no credential required")
```

WHAT TO TRACK

An absent optional credential

VALIDATE THE RESULT

```
Offline mode: no credential required
```

GUIDED PRACTICE

LEARNER PRACTICE

Guided practice

Validate that a supplied cost factor of -1 is invalid and print a clear message without running the calculation.

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 PRACTICE

LEARNER PRACTICE

Independent practice

Create a config JSON with cost_factor 0.9 and a separate .env.example containing only a placeholder variable name.

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.

CHECK YOUR UNDERSTANDING

MENTAL MODEL

Check your understanding

Diagnostic: Does connecting a service justify copying its real key into an exercise file? No. Required exercises remain offline and use no credentials.

RECAP AND NEXT STEP

Recap and next step

Validate configuration before using it, reuse the same functions, and keep optional credentials outside notebooks and public archives.

UNDERSTAND APPLY VALIDATE EXPLAIN