FOUNDATION LESSON

WEEK 2 · LESSON 16

Terminate and comprehend

Avoiding infinite loops / List comprehensions

TERMINATE AND COMPREHEND

MENTAL MODEL

Terminate and comprehend

A bounded while loop needs a changing condition and a clear stopping explanation.
List comprehensions express a transformation and optional filter.
Explain the equivalent explicit loop first. A short comprehension is not necessarily better if it obscures validation.

BOUNDED REPETITION

CODE WALKTHROUGH W02-L16-S03

```
01  attempts = 0
02  value = 1
03  while value < 10 and attempts < 20:
04      value *= 2
05      attempts += 1
06  if value >= 10:
07      print("target reached")
08  else:
09      print("attempt limit reached")
10  print(value, attempts)
```

WHAT TO TRACK

Bounded repetition

PREDICT THE RESULT

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

BOUNDED REPETITION

CODE WALKTHROUGH W02-L16-S04

```
01 attempts = 0
02 value = 1
03 while value < 10 and attempts < 20:
04     value *= 2
05     attempts += 1
06 if value >= 10:
07     print("target reached")
08 else:
09     print("attempt limit reached")
10 print(value, attempts)
```

WHAT TO TRACK

Bounded repetition

VALIDATE THE RESULT

```
target reached
16 4
```

FILTER AND TRANSFORM

CODE WALKTHROUGH W02-L16-S05

```
01  amounts = [100, -20, 300]
02  positive_doubled = [amount * 2 for amount in amounts if amount
↪   > 0]
03  print(positive_doubled)
```

WHAT TO TRACK

Filter and transform

PREDICT THE RESULT

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

FILTER AND TRANSFORM

CODE WALKTHROUGH W02-L16-S06

```
01  amounts = [100, -20, 300]
02  positive_doubled = [amount * 2 for amount in amounts if amount
↪   > 0]
03  print(positive_doubled)
```

WHAT TO TRACK

Filter and transform

VALIDATE THE RESULT

[200, 600]

EQUIVALENT LOOP

CODE WALKTHROUGH W02-L16-S07

```
01  amounts = [100, -20, 300]
02  positive_doubled = []
03  for amount in amounts:
04      if amount > 0:
05          positive_doubled.append(amount * 2)
06  print(positive_doubled)
```

WHAT TO TRACK

Equivalent loop

PREDICT THE RESULT

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

EQUIVALENT LOOP

CODE WALKTHROUGH W02-L16-S08

```
01  amounts = [100, -20, 300]
02  positive_doubled = []
03  for amount in amounts:
04      if amount > 0:
05          positive_doubled.append(amount * 2)
06  print(positive_doubled)
```

WHAT TO TRACK

Equivalent loop

VALIDATE THE RESULT

[200, 600]

INDEPENDENT PREDICTION

LEARNER PRACTICE

Independent prediction

Change one input in terminate and comprehend and test a boundary or edge case.

01 Predict the exact output.
02 Change only one input.
03 Run in a fresh Python process.
04 Explain any difference.

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.

TERMINATE AND COMPREHEND

Recap and next step

A bounded while loop needs a changing condition and a clear stopping explanation. List comprehensions express a transformation and optional filter. Explain the equivalent explicit loop first. A short comprehension is not necessarily better if it obscures validation.

UNDERSTAND APPLY VALIDATE EXPLAIN