FOUNDATION LESSON

WEEK 4 · LESSON 1

Create numerical arrays

One dtype describes an array; values still need business meaning

THE IDEA BEHIND THE OPERATION

MENTAL MODEL

The idea behind the operation

One dtype describes an array; values still need business meaning

LIST VALUES AND ARRAY VALUES

CODE WALKTHROUGH W04-L01-S03

```
01  import numpy as np
02  counts = [10, 20, 30]
03  array = np.array(counts)
04  print("List:", counts)
05  print("Array:", array)
```

WHAT TO TRACK

Three supplied counts, in units.

PREDICT THE RESULT

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

LIST VALUES AND ARRAY VALUES — RESULT

CODE WALKTHROUGH W04-L01-S04

```
01  import numpy as np
02  counts = [10, 20, 30]
03  array = np.array(counts)
04  print("List:", counts)
05  print("Array:", array)
```

WHAT TO TRACK

Three supplied counts, in units.

VALIDATE THE RESULT

```
List: [10, 20, 30]
Array: [10 20 30]
```

AN INITIAL STOCK PLAN

CODE WALKTHROUGH W04-L01-S05

```
01  import numpy as np
02  current = np.array([12, 0, 7])
03  planned = np.zeros(3)
04  print("Current:", current)
05  print("Planned:", planned)
06  print("Baseline:", np.ones(3))
07  print("Fixed fee:", np.full(3, 5.0))
08  print("Month labels:", np.arange(1, 4))
```

WHAT TO TRACK

Zero planned units means no extra units currently planned.

PREDICT THE RESULT

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

AN INITIAL STOCK PLAN — RESULT

CODE WALKTHROUGH W04-L01-S06

```
01  import numpy as np
02  current = np.array([12, 0, 7])
03  planned = np.zeros(3)
04  print("Current:", current)
05  print("Planned:", planned)
06  print("Baseline:", np.ones(3))
07  print("Fixed fee:", np.full(3, 5.0))
08  print("Month labels:", np.arange(1, 4))
```

WHAT TO TRACK

Zero planned units means no extra units currently planned.

VALIDATE THE RESULT

```
Current: [12  0  7]
Planned: [0. 0. 0.]
Baseline: [1. 1. 1.]
Fixed fee: [5. 5. 5.]
Month labels: [1 2 3]
```

EMPTY DATA AND A RAGGED TABLE

CODE WALKTHROUGH W04-L01-S07

```
01  import numpy as np
02  empty = np.array([])
03  print("Empty:", empty)
04  try:
05      bad = np.array([[10, 20], [30]])
06  except ValueError as error:
07      print(type(error).__name__)
08      print("Rows must have matching lengths")
```

WHAT TO TRACK

An empty array is permitted; this numeric table must be rectangular.

PREDICT THE RESULT

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

EMPTY DATA AND A RAGGED TABLE — RESULT

CODE WALKTHROUGH W04-L01-S08

```
01  import numpy as np
02  empty = np.array([])
03  print("Empty:", empty)
04  try:
05      bad = np.array([[10, 20], [30]])
06  except ValueError as error:
07      print(type(error).__name__)
08      print("Rows must have matching lengths")
```

WHAT TO TRACK

An empty array is permitted; this numeric table must be rectangular.

VALIDATE THE RESULT

```
Empty: []
ValueError
Rows must have matching
lengths
```

GUIDED MODIFICATION

LEARNER PRACTICE

Guided modification

Create current stock [5, 0, 9, 2] and a four-entry zero plan. Print both. Confirm that zero stock and zero planned additions are two different supplied business facts.

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

Create three month labels with arange and five evenly spaced planning rates from 0.0 to 0.2 with linspace. Print them. Explain the excluded arange stop and included linspace endpoint.

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

An array gives values a consistent representation. Create it from a collection or an initialisation function, and keep business meaning explicit. Empty input and ragged rows are different cases. Next, we describe array dimensions, shape, size and dtype.

UNDERSTAND APPLY VALIDATE EXPLAIN