FOUNDATION LESSON

WEEK 4 · LESSON 2

Read shape, dimensions and dtype

ndim counts axes; shape gives axis lengths; size counts elements

THE IDEA BEHIND THE OPERATION

MENTAL MODEL

The idea behind the operation

ndim counts axes; shape gives axis lengths; size counts elements

SAME VALUES, DIFFERENT STRUCTURES

CODE WALKTHROUGH W04-L02-S03

```
01  import numpy as np
02  vector = np.array([10, 20, 30], dtype=np.int64)
03  row = np.array([[10, 20, 30]], dtype=np.int64)
04  print(vector.ndim, vector.shape, vector.size)
05  print(row.ndim, row.shape, row.size)
06  print(vector.dtype)
```

WHAT TO TRACK

Compare (3,) with (1, 3).

PREDICT THE RESULT

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

SAME VALUES, DIFFERENT STRUCTURES — RESULT

CODE WALKTHROUGH W04-L02-S04

```
01  import numpy as np
02  vector = np.array([10, 20, 30], dtype=np.int64)
03  row = np.array([[10, 20, 30]], dtype=np.int64)
04  print(vector.ndim, vector.shape, vector.size)
05  print(row.ndim, row.shape, row.size)
06  print(vector.dtype)
```

WHAT TO TRACK

Compare (3,) with (1, 3).

VALIDATE THE RESULT

```
1 (3,) 3
2 (1, 3) 3
int64
```

A WAREHOUSE STOCK GRID

CODE WALKTHROUGH W04-L02-S05

```
01  import numpy as np
02  units = np.array([[12, 8, 0], [5, 9, 4]], dtype=np.int64)
03  print("Axes:", units.ndim)
04  print("Shape:", units.shape)
05  print("Elements:", units.size)
06  print("Rows:", len(units))
```

WHAT TO TRACK

Rows are two warehouses; columns are three products.

PREDICT THE RESULT

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

A WAREHOUSE STOCK GRID — RESULT

CODE WALKTHROUGH W04-L02-S06

```
01  import numpy as np
02  units = np.array([[12, 8, 0], [5, 9, 4]], dtype=np.int64)
03  print("Axes:", units.ndim)
04  print("Shape:", units.shape)
05  print("Elements:", units.size)
06  print("Rows:", len(units))
```

WHAT TO TRACK

Rows are two warehouses; columns are three products.

VALIDATE THE RESULT

```
Axes: 2
Shape: (2, 3)
Elements: 6
Rows: 2
```

A CONVERSION CAN LOSE DECIMALS

CODE WALKTHROUGH W04-L02-S07

```
01  import numpy as np
02  prices = np.array([12.75, 8.25])
03  whole = prices.astype(np.int64)
04  empty = np.array([])
05  print("Original:", prices)
06  print("Converted:", whole)
07  print("Empty:", empty.shape, empty.size)
```

WHAT TO TRACK

astype creates converted data;
truncation is not rounding.

PREDICT THE RESULT

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

A CONVERSION CAN LOSE DECIMALS — RESULT

CODE WALKTHROUGH W04-L02-S08

```
01  import numpy as np
02  prices = np.array([12.75, 8.25])
03  whole = prices.astype(np.int64)
04  empty = np.array([])
05  print("Original:", prices)
06  print("Converted:", whole)
07  print("Empty:", empty.shape, empty.size)
```

WHAT TO TRACK

astype creates converted data; truncation is not rounding.

VALIDATE THE RESULT

```
Original: [12.75  8.25]
Converted: [12  8]
Empty: (0,) 0
```

GUIDED MODIFICATION

LEARNER PRACTICE

Guided modification

Create a two-by-two integer stock array. Print ndim, shape, size and dtype. Confirm size is four and explain why len is two.

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

Compare np.array(7) with np.array([7]).
Print ndim, shape and size for each.
Explain why one value does not necessarily mean one dimension.

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

Read ndim, shape, size and dtype as separate properties. Business labels give axes their meaning. Conversion may discard information. Next, we select positions and slices, including the important difference between a view and a copy.

UNDERSTAND APPLY VALIDATE EXPLAIN