Source task and conventions

Stack three 100×200 matrices so that the second slice of the third axis is B, and verify the result.

Conventional solution and proof

Choose the axis explicitly:

```
T = np.stack([A, B, C], axis=2)
assert T.shape == (100, 200, 3)
assert np.array_equal(T[:, :, 1], B)
```

By definition, $T_{ij0} = A_{ij}$, $T_{ij1} = B_{ij}$ and $T_{ij2} = C_{ij}$, proving the assertion for every entry. The tensor has three axes and $60,000$ scalar entries. The second coordinate is index one in zero-based indexing.

Structural or analytic derivation

The operation adjoins a three-valued label index to the matrix coordinates. Projection onto label one recovers B. A default stack on axis zero would instead have shape $(3, 100, 200)$ and require $T[1, :, :]$; the source's requested third-axis slice therefore determines the explicit axis choice.

Comparative cost and preferred method

Materializing the tensor copies $60,000$ entries, an unavoidable $\Theta(60,000)$ output cost for a new contiguous stack. The slice can be a constant-metadata view with no element copy. The index-level proof is the preferable method; complex analysis supplies no additional reduction.