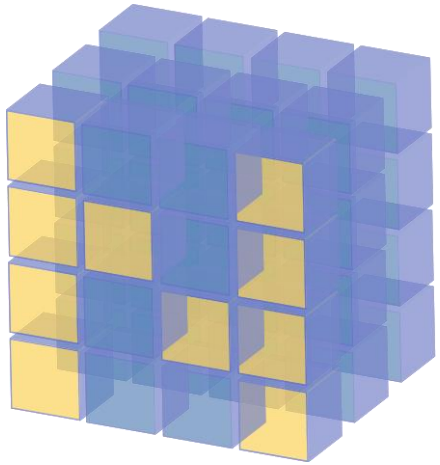
PRIME INTUIT
Finishing School

NumPy



NumPy

What are we interested in:

- ✓ What are **n-d arrays**
- ✓ What is **broadcasting**
- ✓ How to **load and save n-d arrays**
- ✓ How to **use statistical functions**

NumPy

```
# Comparing performance with lists….

N = 1000000000

%%time
List1 = list(range(N))
For I in range(N)
List1[i] = List1[i] * List1[i]

%%time
List1 = list(range(N))
List1 = [item * item for item in List1]
```

NumPy

```
%%time
List1 = list(range(N))
List1 = map(lambda x: x * x, List1)

%%time
List1 = list(range(N))
List_sum = 0
For item in List1
List_Sum + =  item

%%time
List1 = list(range(N))
List1_sum = sum(
```

PRIME INTUIT
Finishing School

NumPy

```
Imprt numpy as np
%%time
Arr =np.arrange(N)
Arr = Arr * Arr

%%time
Arr =np.arrange(N)
Arr_sum = np.sum(Arr)
```

PRIME INTUIT
Finishing School

High Dimensional Array & Creating NumPy Array

1	2	3	4

One Dimension Array

1	2	3	4
5	6	7	8
9	10	11	12

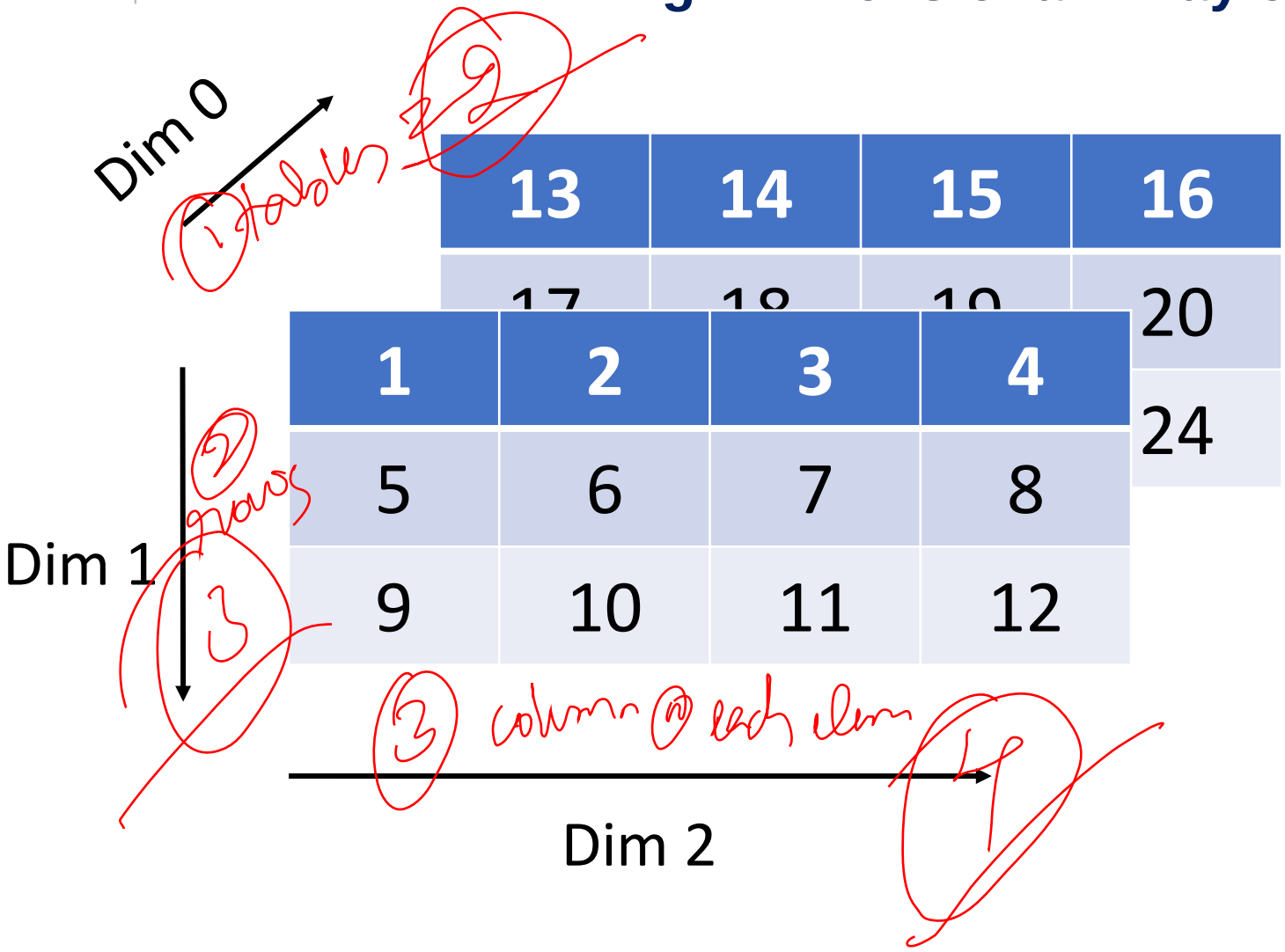
Two Dimension Array

Three Dimension Array

1	2	3	4	13	14	15	16
5	6	7	8	17	18	19	20
9	10	11	12	21	22	23	24

PRIME INTUIT
Finishing School

High Dimensional Array & Creating NumPy Array

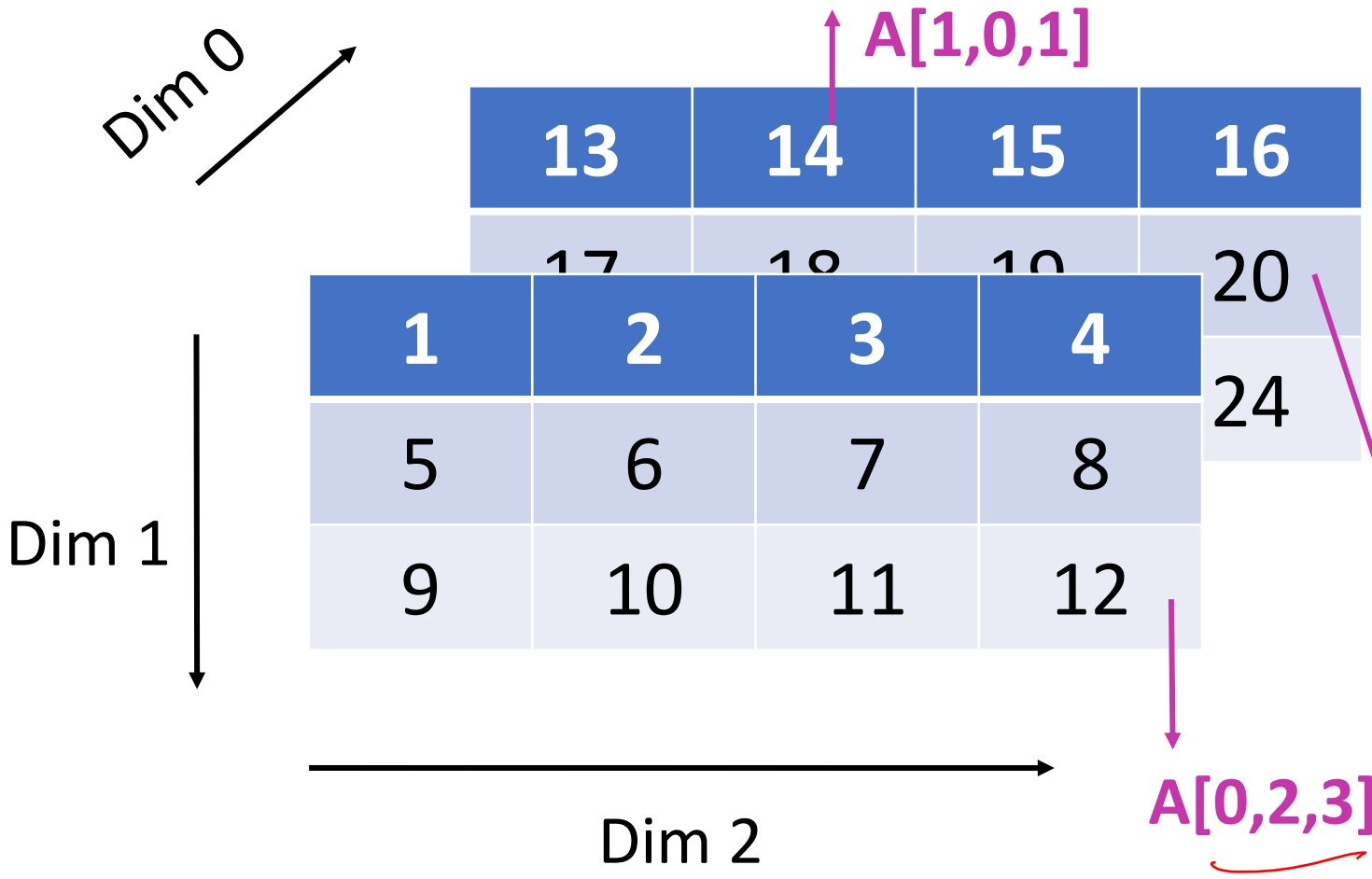


We index dimensions backwards in order we added them

So we have a 3 Dim array of size 2 x 3 x 4 (Shape of array)

PRIME INTUIT
Finishing School

High Dimensional Array & Creating NumPy Array

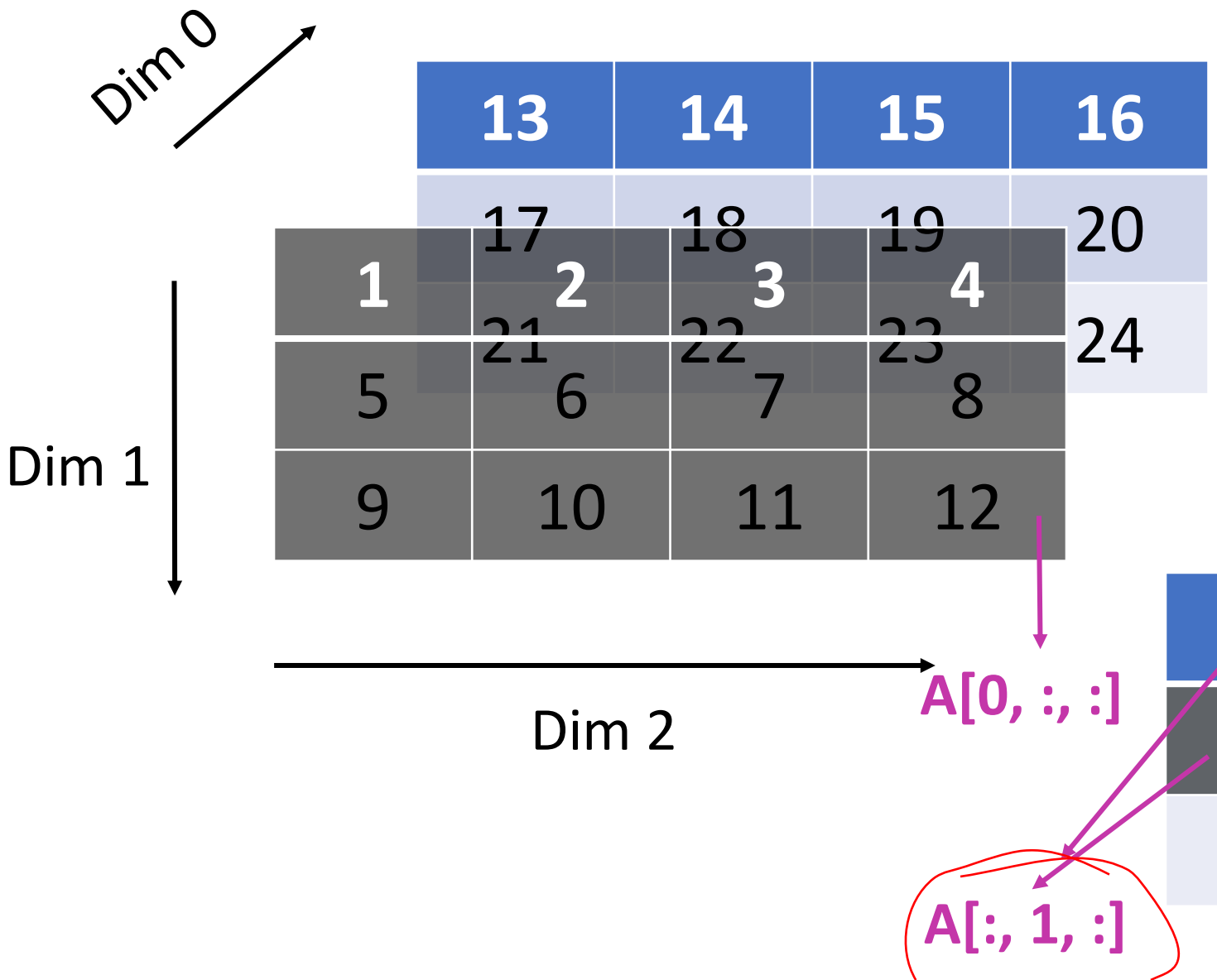


We can index each item of the 3Dim array by specifying 3 co-ordinates

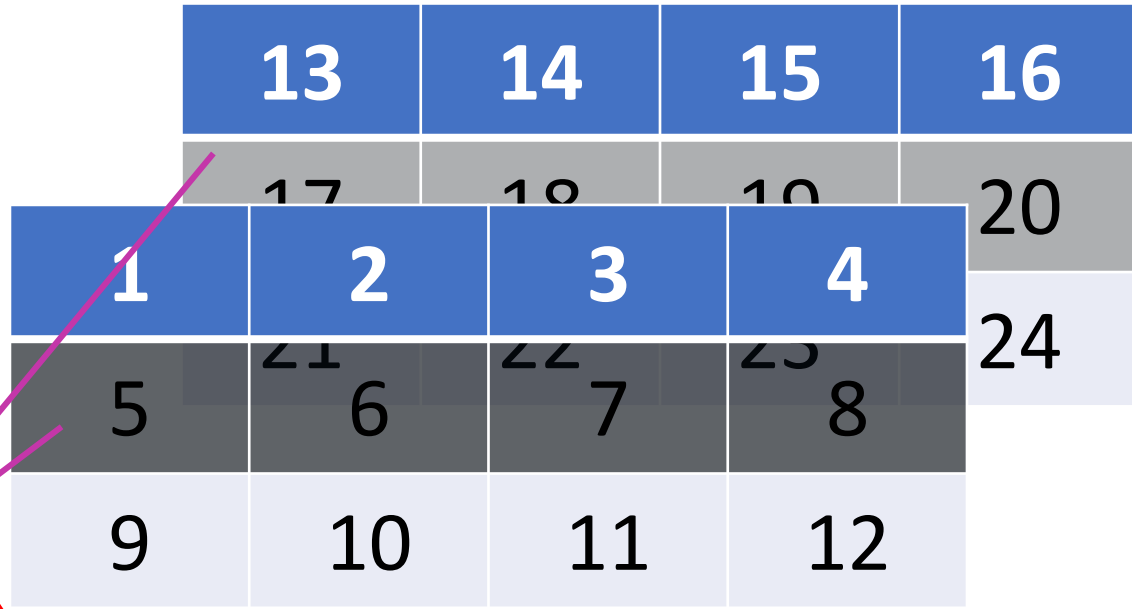
Remember we index from 0

PRIME INTUIT
Finishing School

High Dimensional Array & Creating NumPy Array

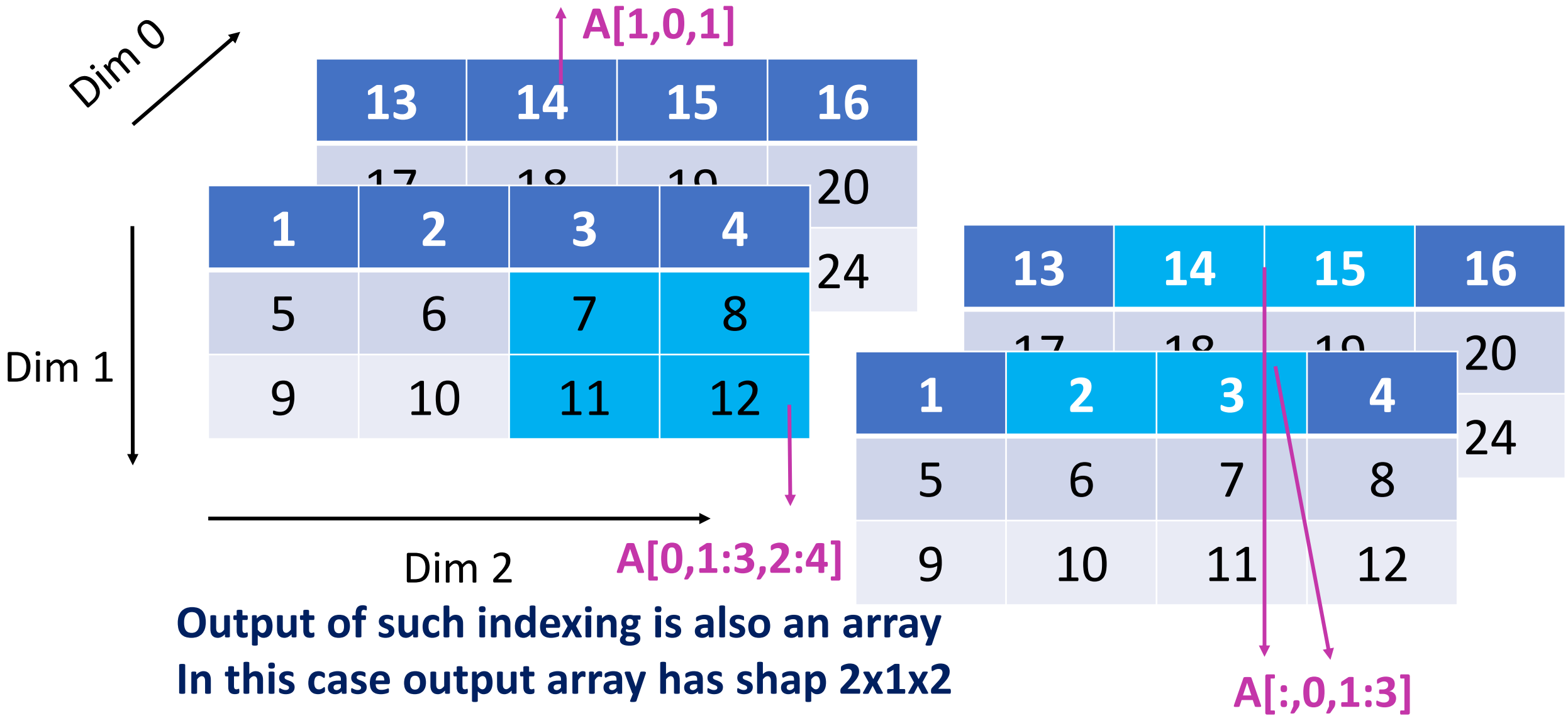


We can refer to slices of the data with partial indices



PRIME INTUIT
Finishing School

High Dimensional Array & Creating NumPy Array



Output of such indexing is also an array

In this case output array has shap 2x1x2

PRIME INTUIT
Finishing School

High Dimensional Array & Creating NumPy Array

```
Arr =np.arrange(5)
Print(Arr, type(Arr))


Arr = np.Arr([0, 2, 4, 6, 8])
Print(Arr, type(Arr))

Arr  # this would print an array

Array([0, 2, 4, 6, 8])
Print(Arr, type(Arr))
```

PRIME INTUIT
Finishing School

High Dimensional Array & Creating NumPy Array

```
Arr =np.arrange(5)
Print(Arr, type(Arr))

Arr = np.Arr([0, 2, 4, 6, 8])
Print(Arr, type(Arr))

Arr  # this would print an array
Arr.dtype

Arr.ndim
Arr.shape
Arr.size
```

PRIME INTUIT
Finishing School

High Dimensional Array & Creating NumPy Array

Arr.itemsize

Size (in bytes) of each element of NP array

#2 dimensional Array

```
Arr2d = np.array([
                    [1, 2, 3],
                    [4, 5, 6]
])

Arr2d

Arr2d.ndim
```