Arrays & Slices

Nigel Poulton
Author & Trainer

@nigelpoulton nigelpoulton.com

Agenda

Theory

Arrays vs Slices

Working with Slices

Getting Under the Hood

Expanding Slices

Miscellaneous

Recap

String Theory

Warning

Go might do things differently

Numbered lists containing elements of the **same type**

Numbered lists containing elements of the **same type**

Numbered lists of the **same type**

0 Go Fundamentals

1 Getting Started with Docker

2 Getting Started with Kubernetes

3 Docker Deep Dive

4 Kubernetes Deep Dive

5 Containers on AWS Wavelength

Numbered lists of the **same type**

0	“Go Fundamentals”
1	“Getting Started with Docker”
2	“Getting Started with Kubernetes”
3	“Docker Deep Dive”
4	“Kubernetes Deep Dive”
5	“Containers on AWS Wavelength”

Numbered lists of the **same type**

0	"Go Fundamentals"
1	"Getting Started with Docker"
2	"Getting Started with Kubernetes"
3	"Docker Deep Dive"
4	"Kubernetes Deep Dive"
5	"Containers on AWS Wavelength"

0	60
1	60
2	24
3	7
4	365
5	42

0	"seconds
1	"minutes"
2	24
3	7
4	365.24
5	"The answer to life, the universe, and everything"

⚠ illegal ⚠

Arrays vs Slices

Arrays

Slices

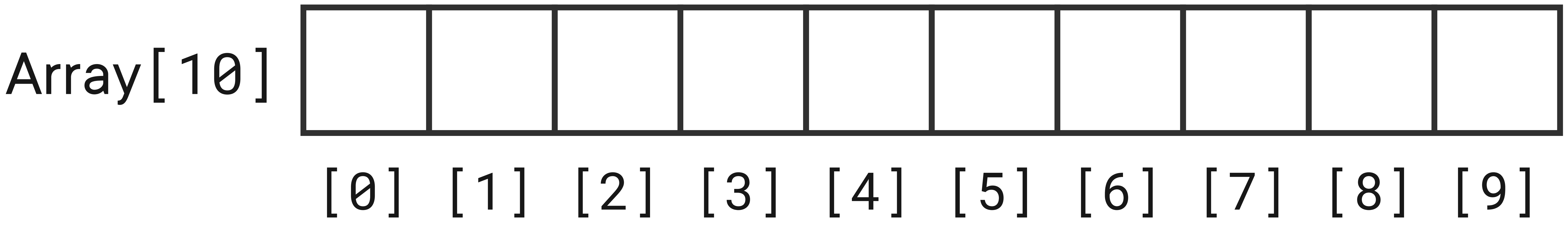
Arrays

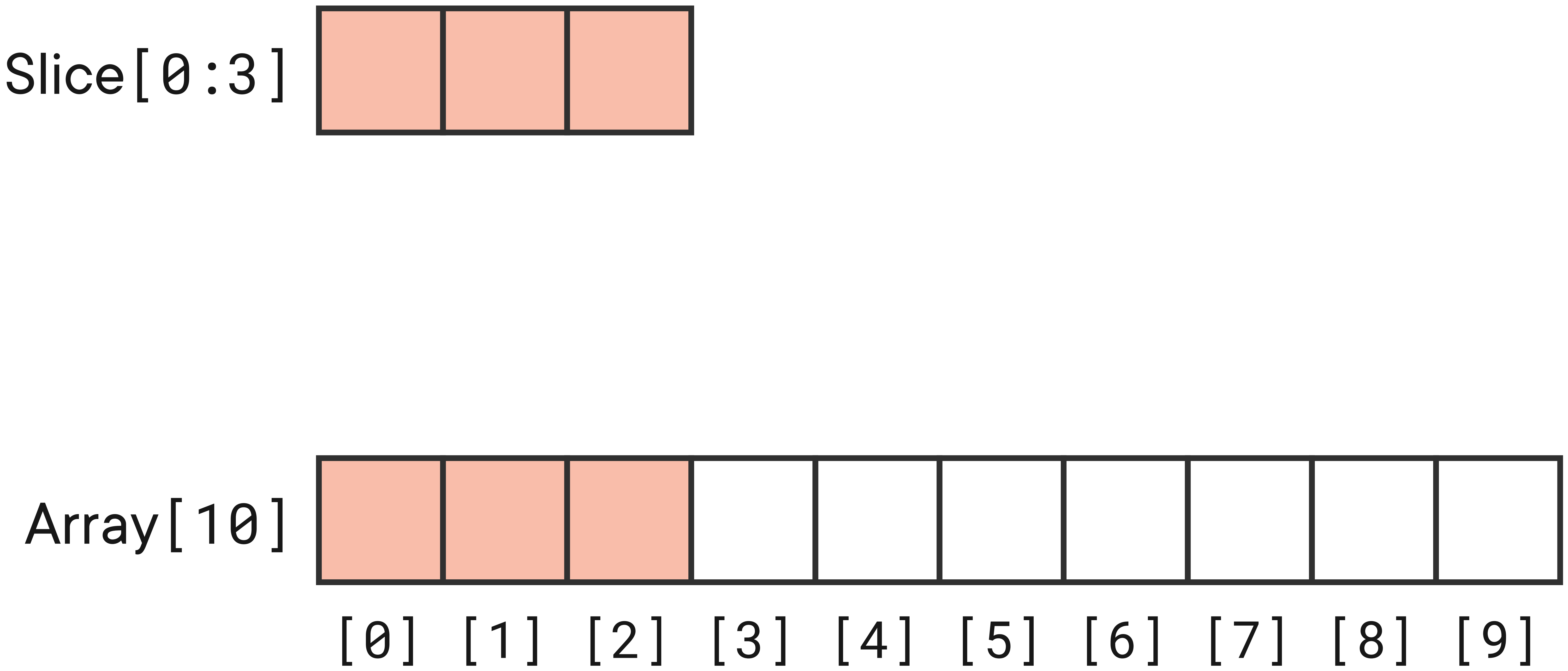
Have a fixed size

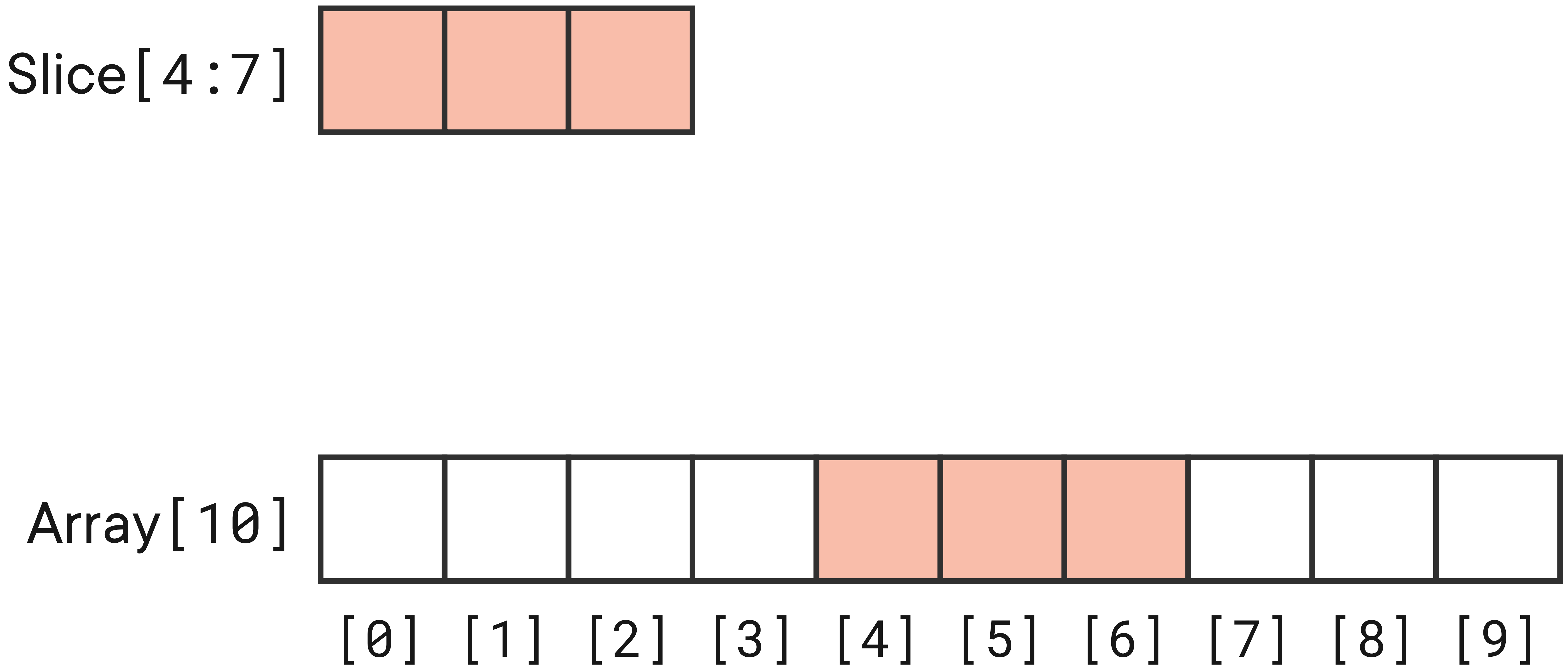
Slices

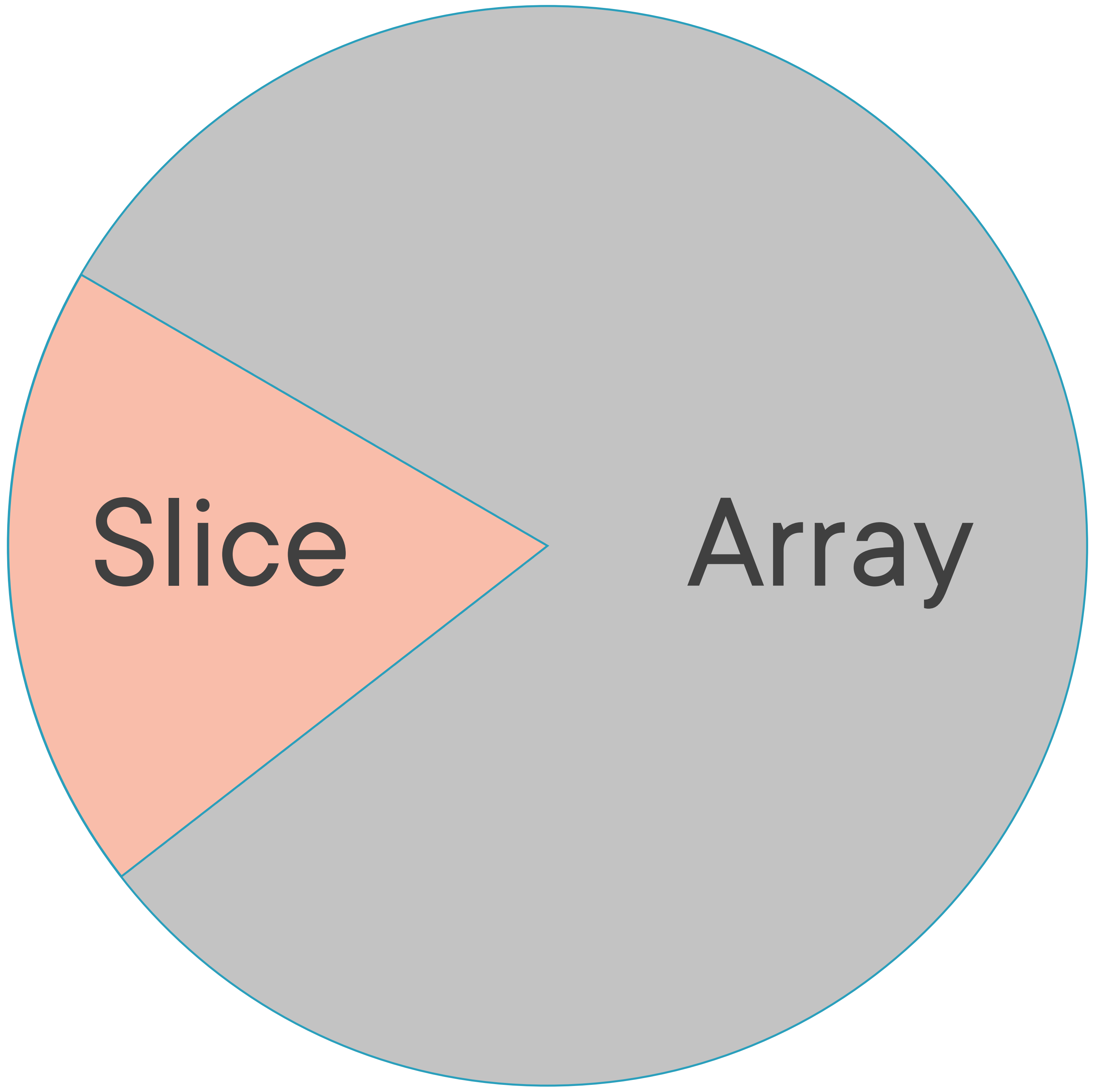
Can be resized

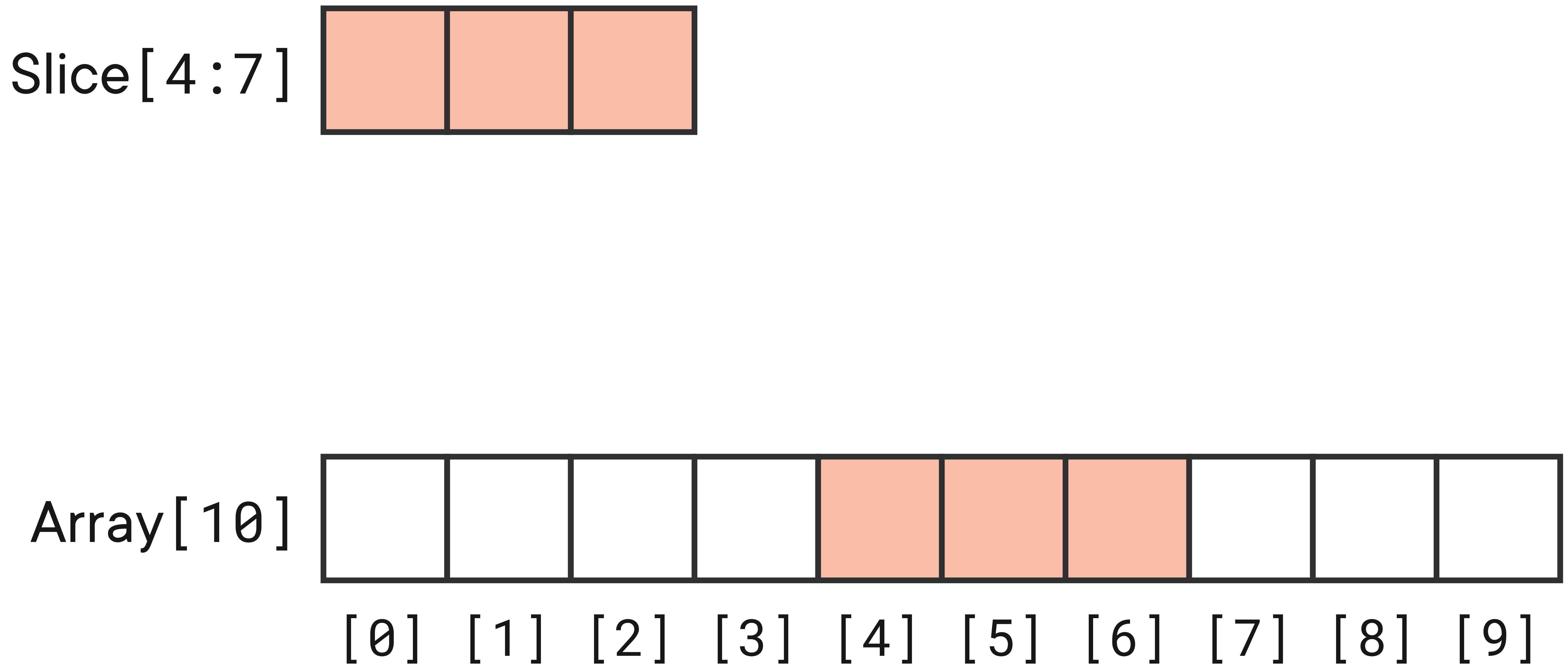
Slices are built on top of **arrays**

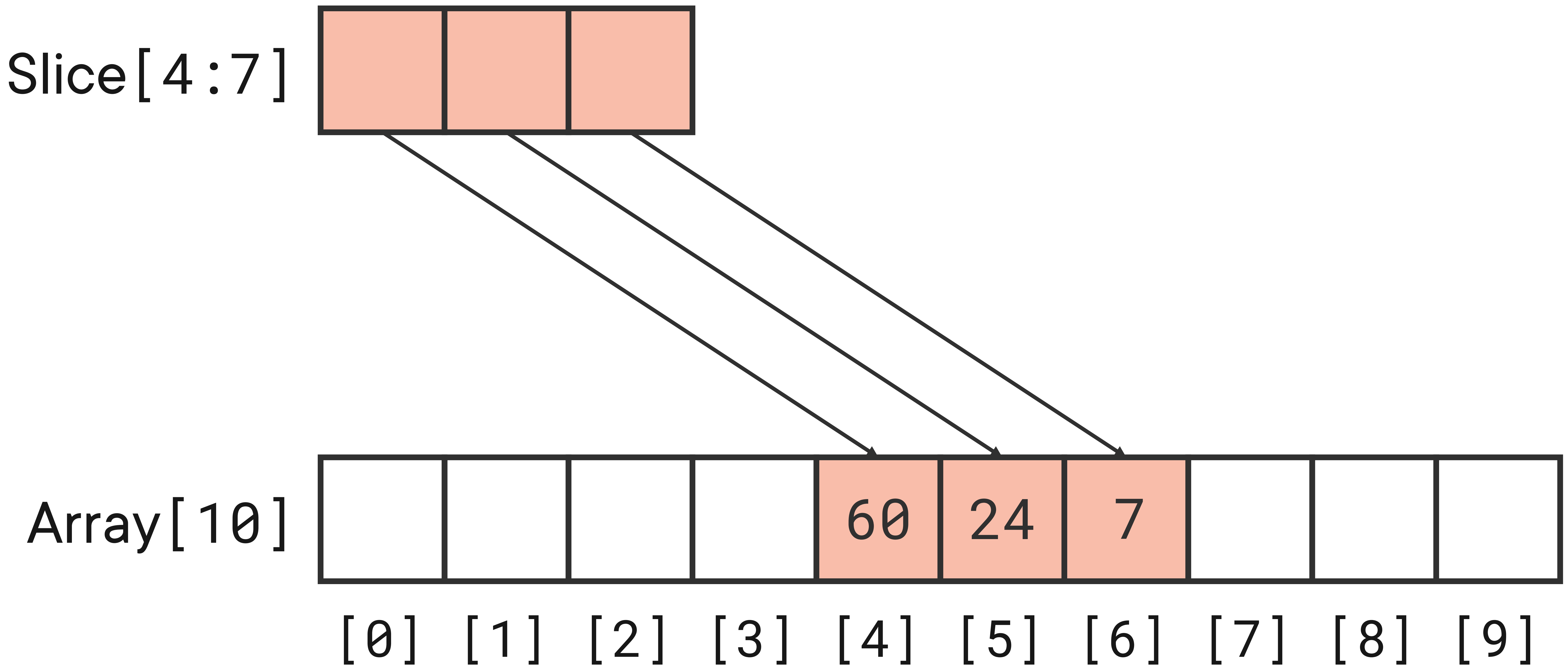

Array[10]
[0]
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]

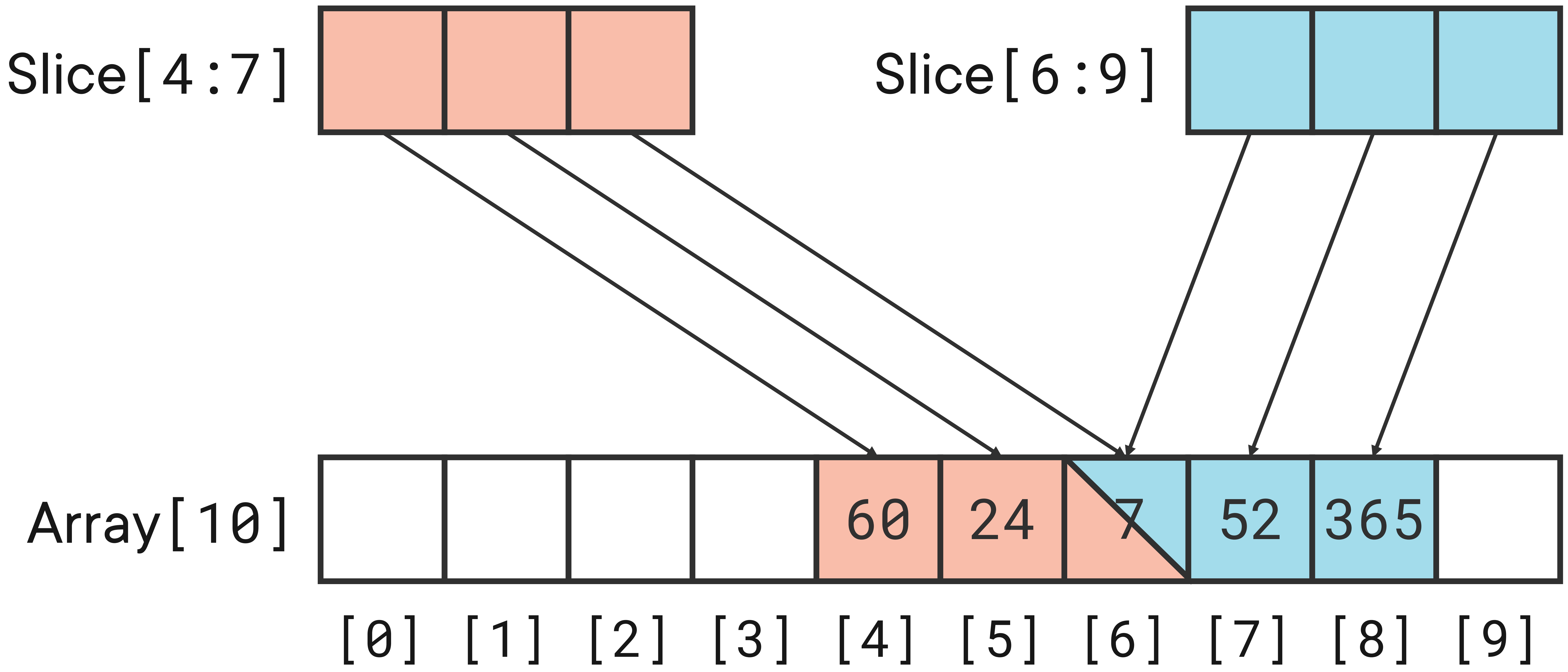

Slice[0:3]
Array[10]
[0]
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]


Slice[4:7]
Array[10]
[0]
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]

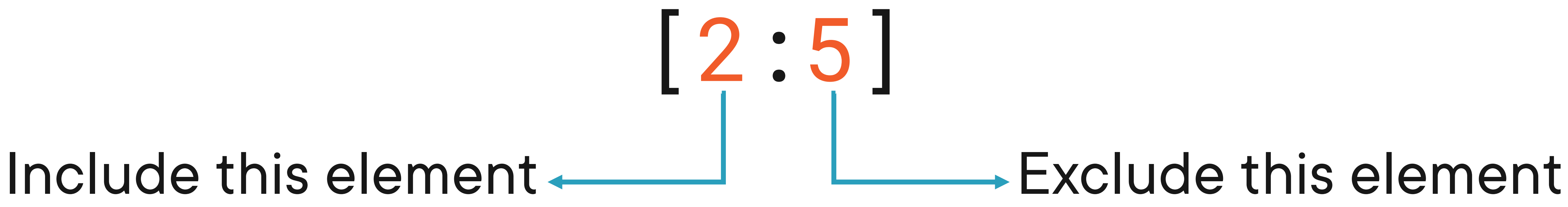

Slice
Array


Slice[4:7]
Array[10]
[0]
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]


Slice[4:7]
Array[10]
60
24
7
[0]
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]


Slice[4:7]
Slice[6:9]
Array[10]
60
24
7
52
365
[0]
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]

Slices are passed to functions by **reference**


[2:5]
Include this element
Exclude this element

```
[ :5 ]
```

Implies index position 0

$$[\,4 : \,]$$

Implies end of slice

Appending to Slices

Arrays

Have a fixed size

Slices

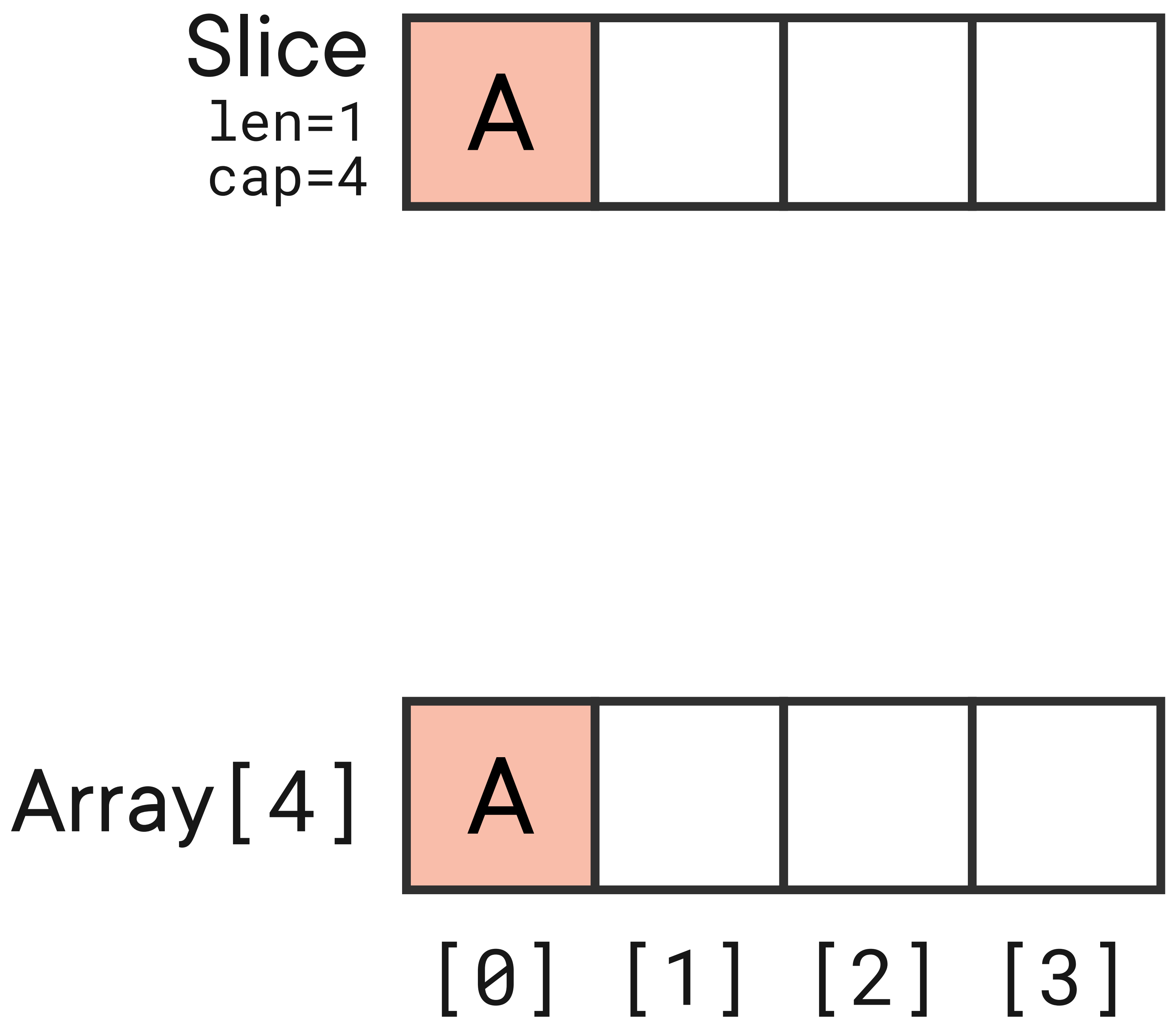
Can be resized

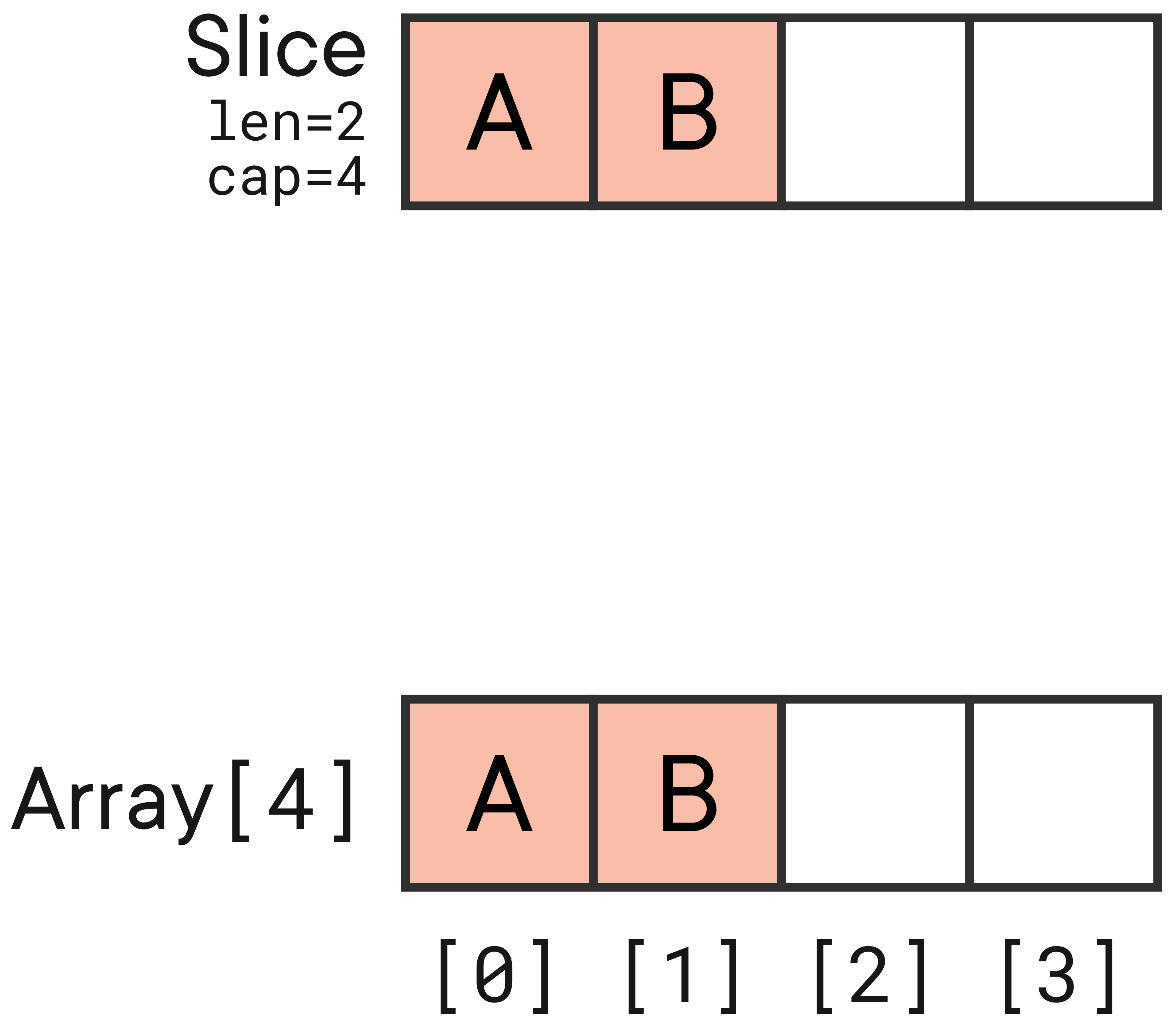
append( )

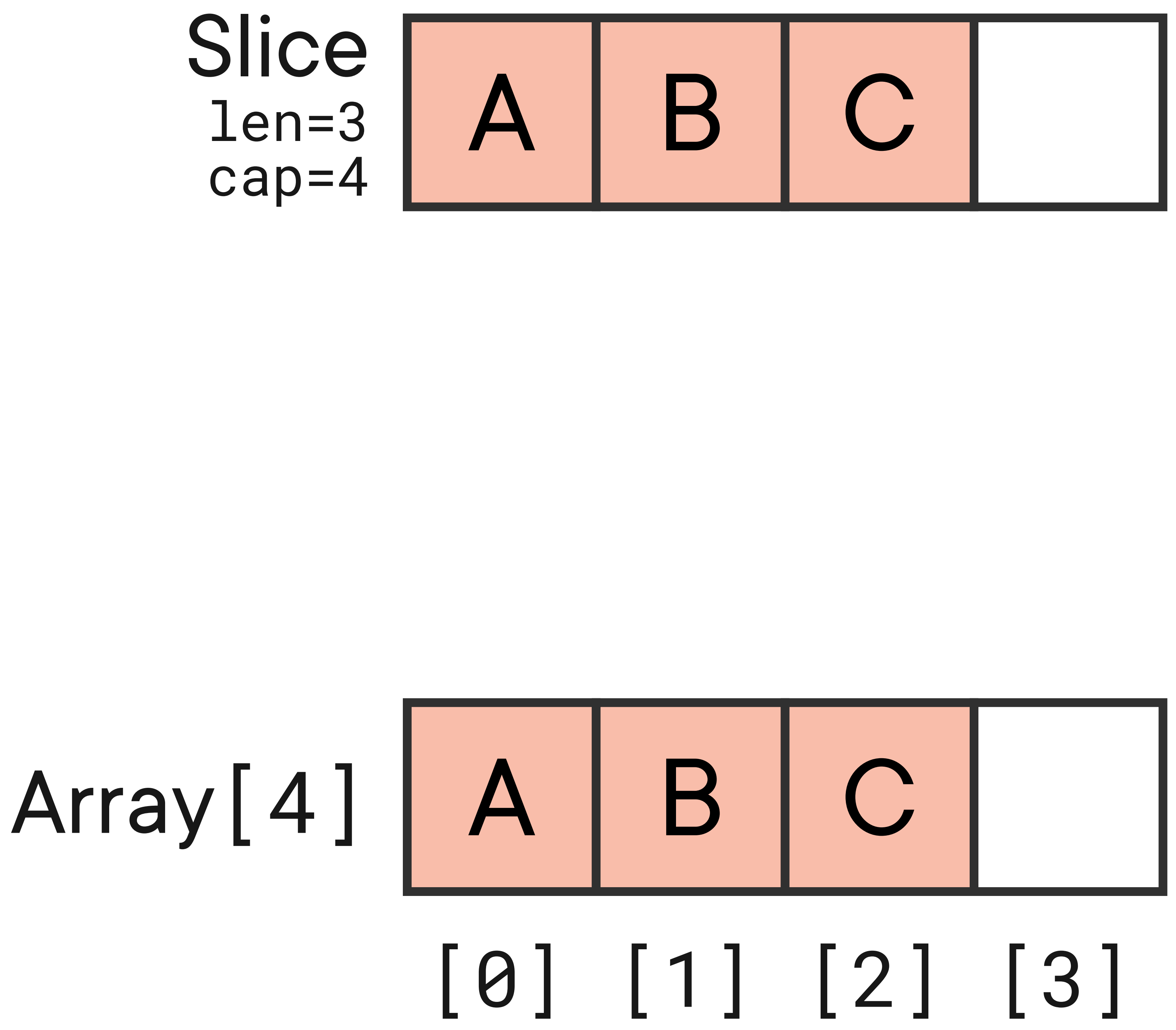
* append( ) is a built-in function, not a keyword

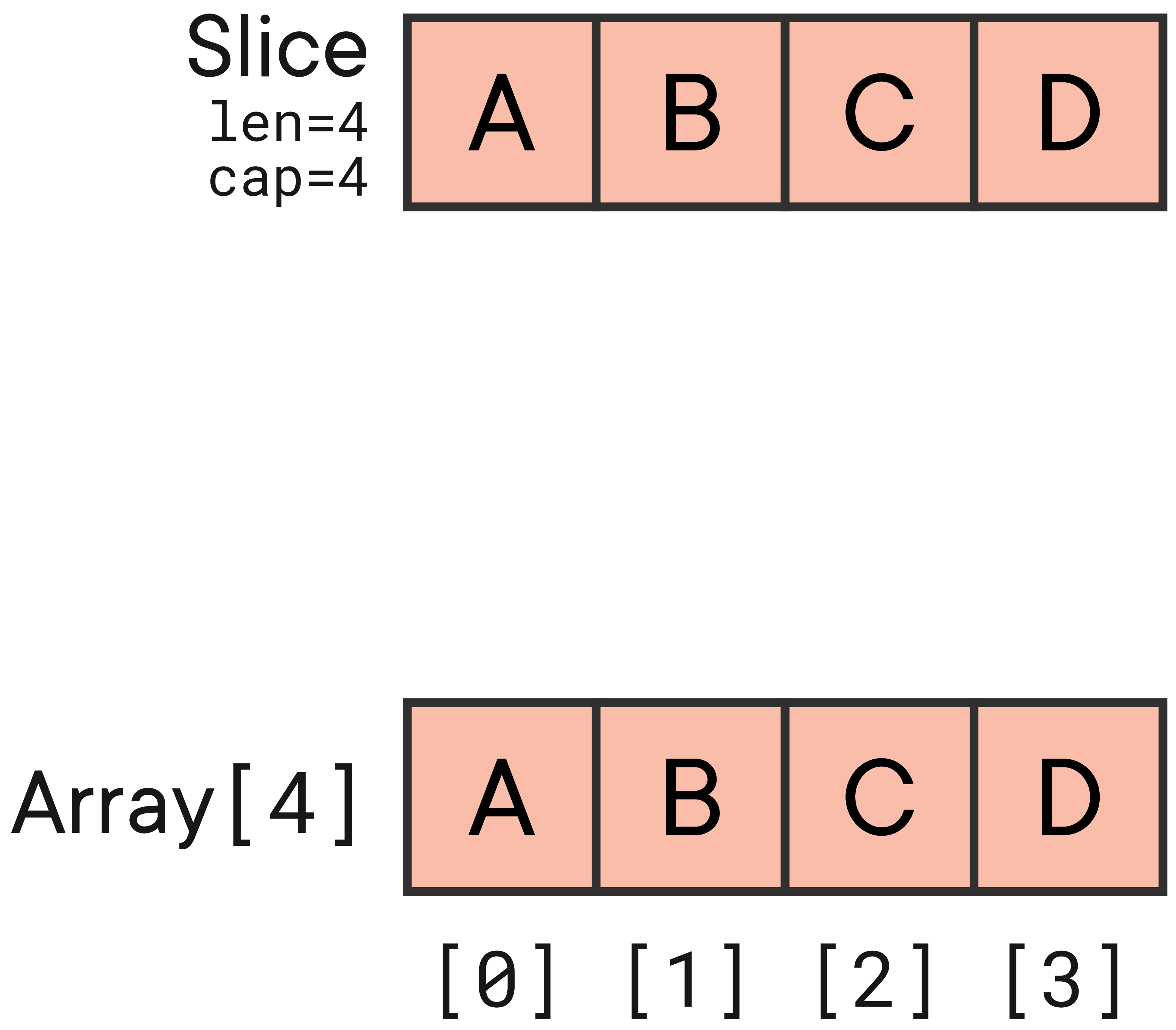
```
slice = append(slice, 5)
```

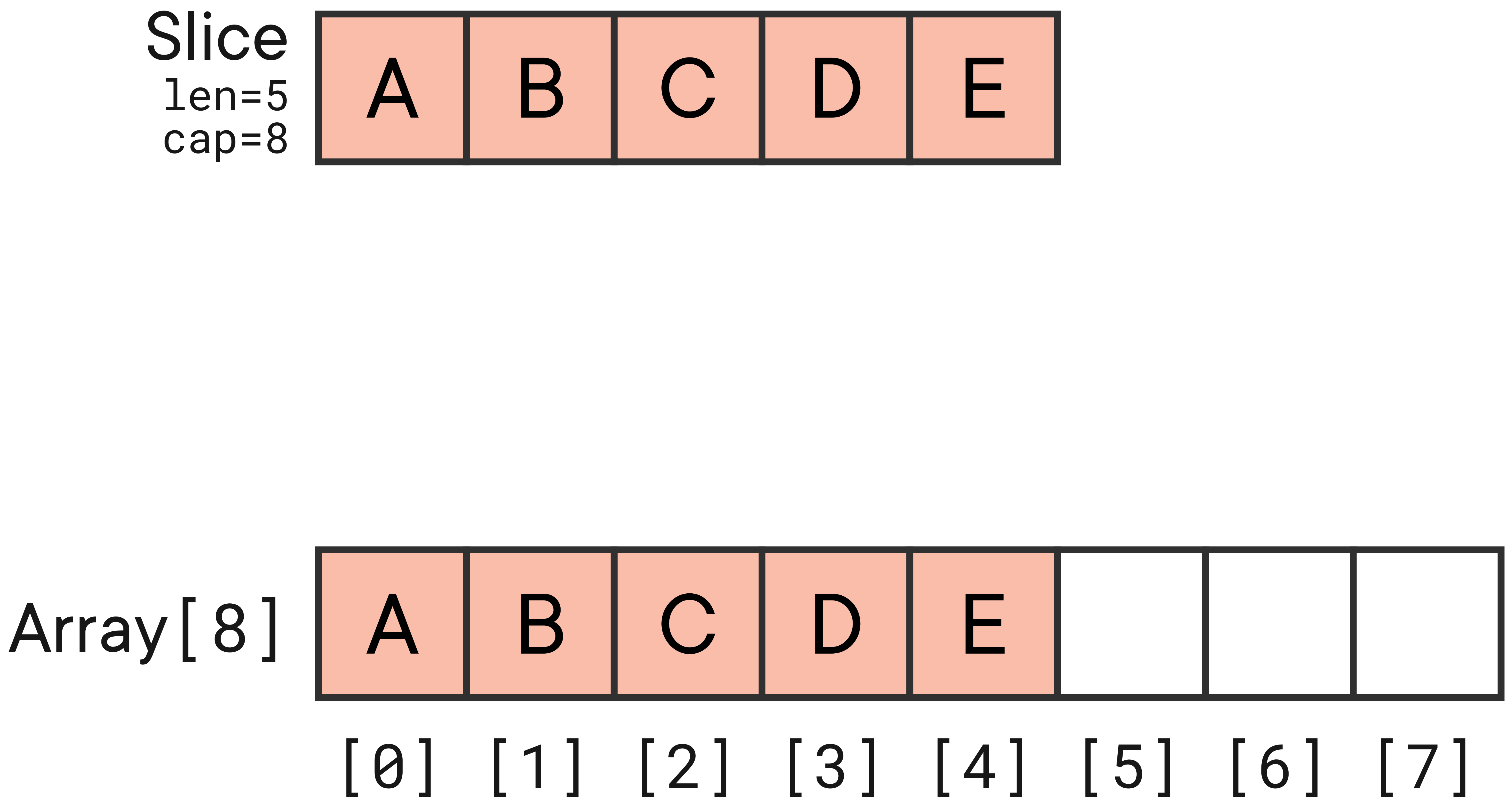
* append( ) is a built-in function, not a keyword


Slice
len=1
cap=4
A
Array[4]
A
[0]
[1]
[2]
[3]


Slice
len=2
cap=4
A
B
Array[4]
A
B
[0]
[1]
[2]
[3]


Slice
len=3
cap=4
A
B
C
Array[4]
A
B
C
[0]
[1]
[2]
[3]


Slice
len=4
cap=4
A
B
C
D
Array[4]
A
B
C
D
[0]
[1]
[2]
[3]


Slice
len=5
cap=8
A
B
C
D
E
Array[8]
A
B
C
D
E
[0]
[1]
[2]
[3]
[4]
[5]
[6]
[7]

Length starts out as 1 with a capacity of 4

```
Length starts out as 1 with a capacity of 4
Slice length is 2 but capacity is 4
```

```
Length starts out as 1 with a capacity of 4
Slice length is 2 but capacity is 4
Slice length is 3 but capacity is 4
```

```
Length starts out as 1 with a capacity of 4
Slice length is 2 but capacity is 4
Slice length is 3 but capacity is 4
Slice length is 4 but capacity is 4
```

```
Length starts out as 1 with a capacity of 4
Slice length is 2 but capacity is 4
Slice length is 3 but capacity is 4
Slice length is 4 but capacity is 4
Slice length is 5 but capacity is 8
```

```
Length starts out as 1 with a capacity of 4
Slice length is 2 but capacity is 4
Slice length is 3 but capacity is 4
Slice length is 4 but capacity is 4
Slice length is 5 but capacity is 8
Slice length is 6 but capacity is 8
Slice length is 7 but capacity is 8
Slice length is 8 but capacity is 8
Slice length is 9 but capacity is 16
```

```
Length starts out as 1 with a capacity of 4
Slice length is 2 but capacity is 4
Slice length is 3 but capacity is 4
Slice length is 4 but capacity is 4
Slice length is 5 but capacity is 8
Slice length is 6 but capacity is 8
Slice length is 7 but capacity is 8
Slice length is 8 but capacity is 8
Slice length is 9 but capacity is 16
Slice length is 10 but capacity is 16
Slice length is 11 but capacity is 16
Slice length is 12 but capacity is 16
Slice length is 13 but capacity is 16
Slice length is 14 but capacity is 16
Slice length is 15 but capacity is 16
Slice length is 16 but capacity is 16
Slice length is 17 but capacity is 32
```

[1 2 3 4 5]

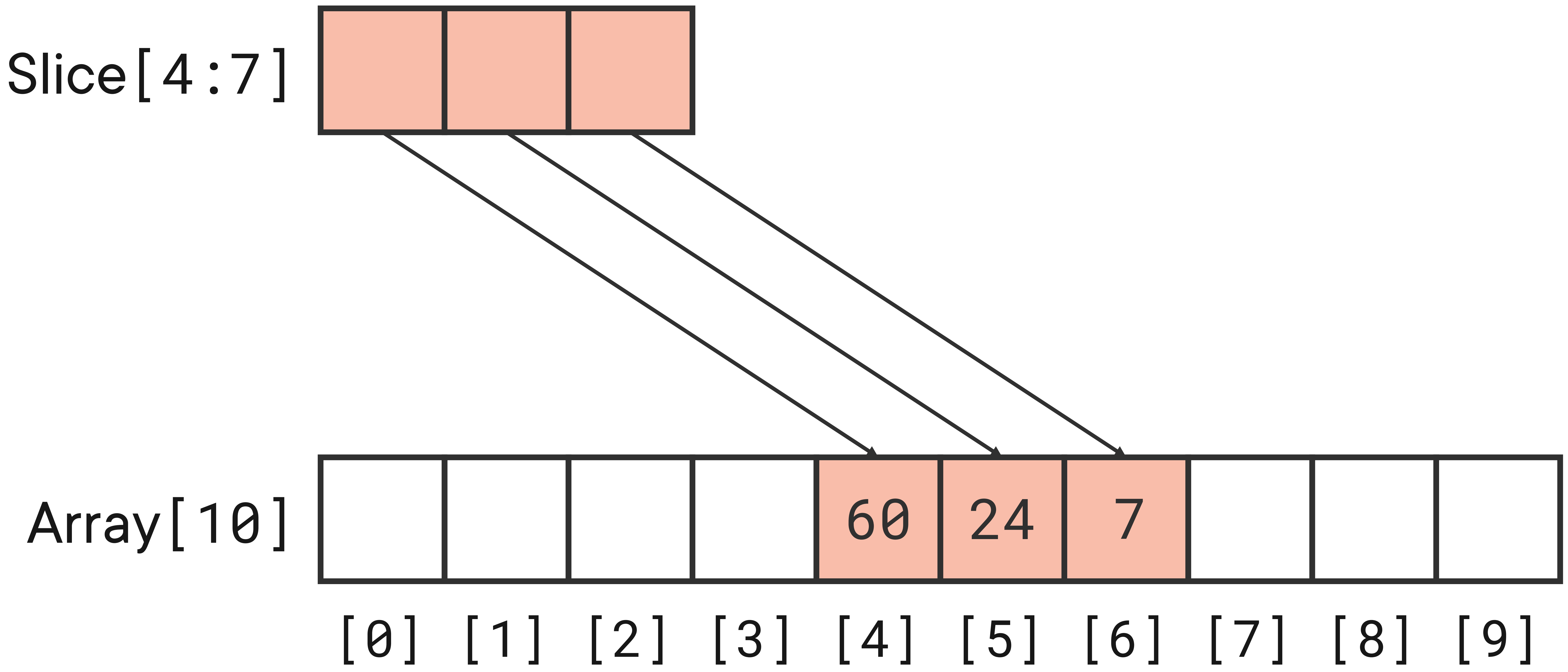
$$\begin{array}{l} [\,1 \quad 2 \quad 3 \quad 4 \quad 5\,] \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}$$

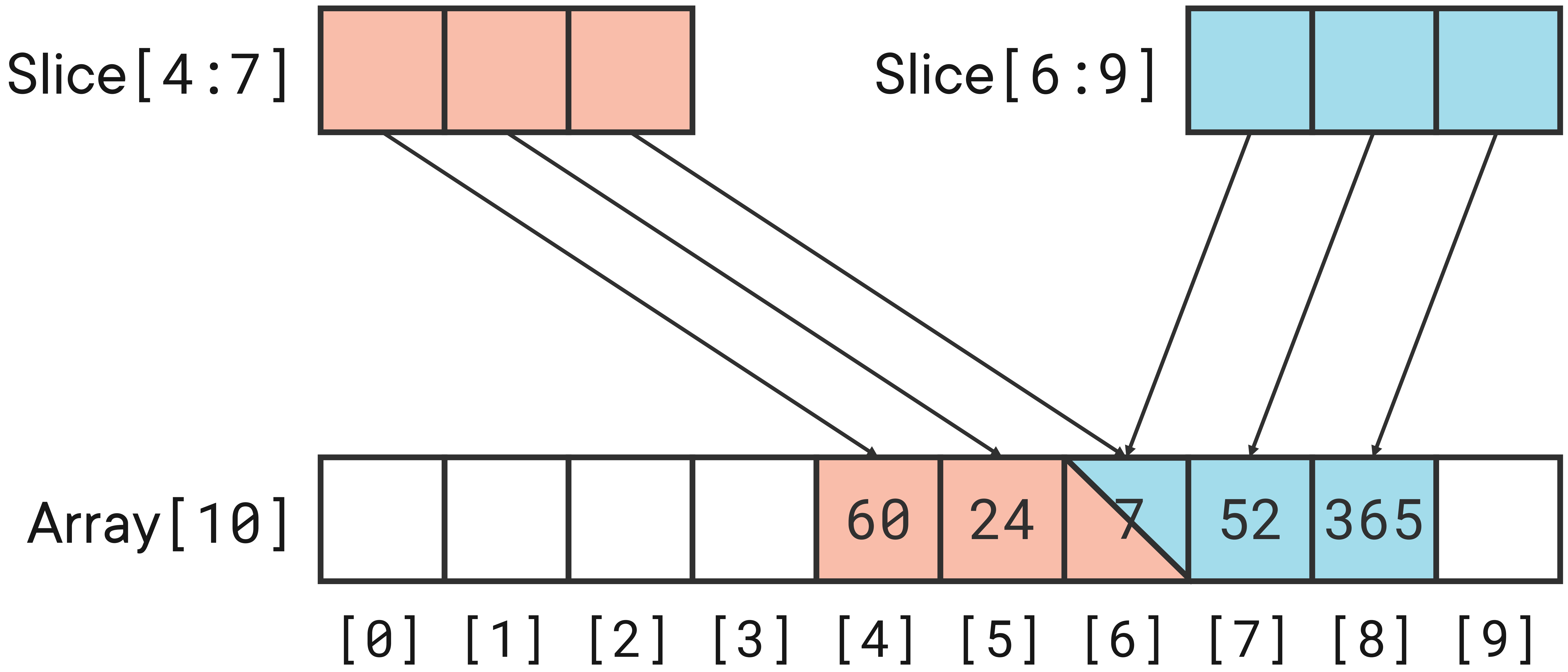
```
[1 2 3 4 5]
1
2
3
4
5
mySlice NOW contains [1 2 3 4 5 10 20 30]
and has a new length of x and capacity of y
```

```
[1 2 3 4 5]
1
2
3
4
5
mySlice NOW contains [1 2 3 4 5 10 20 30]
and has a new length of 8 and capacity of 10
```

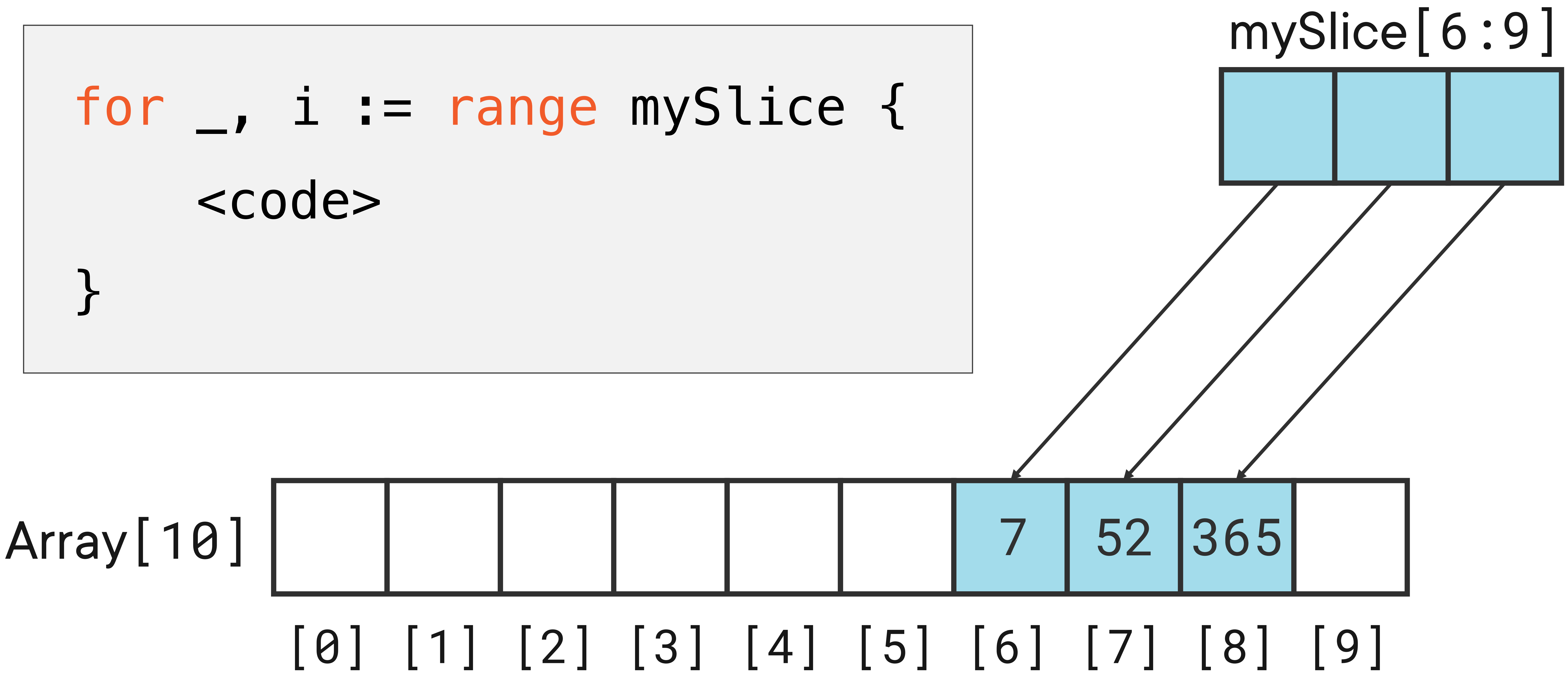
Recap

Arrays	Slices
Numbered lists of single type	Numbered lists of single type
Static	Dynamic


Slice[4:7]
Array[10]
60
24
7
[0]
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]


Slice[4:7]
Slice[6:9]
Array[10]
60
24
7
52
365
[0]
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]

```
for _, i := range mySlice {
    <code>
}
```



Up Next:
Working with Maps