

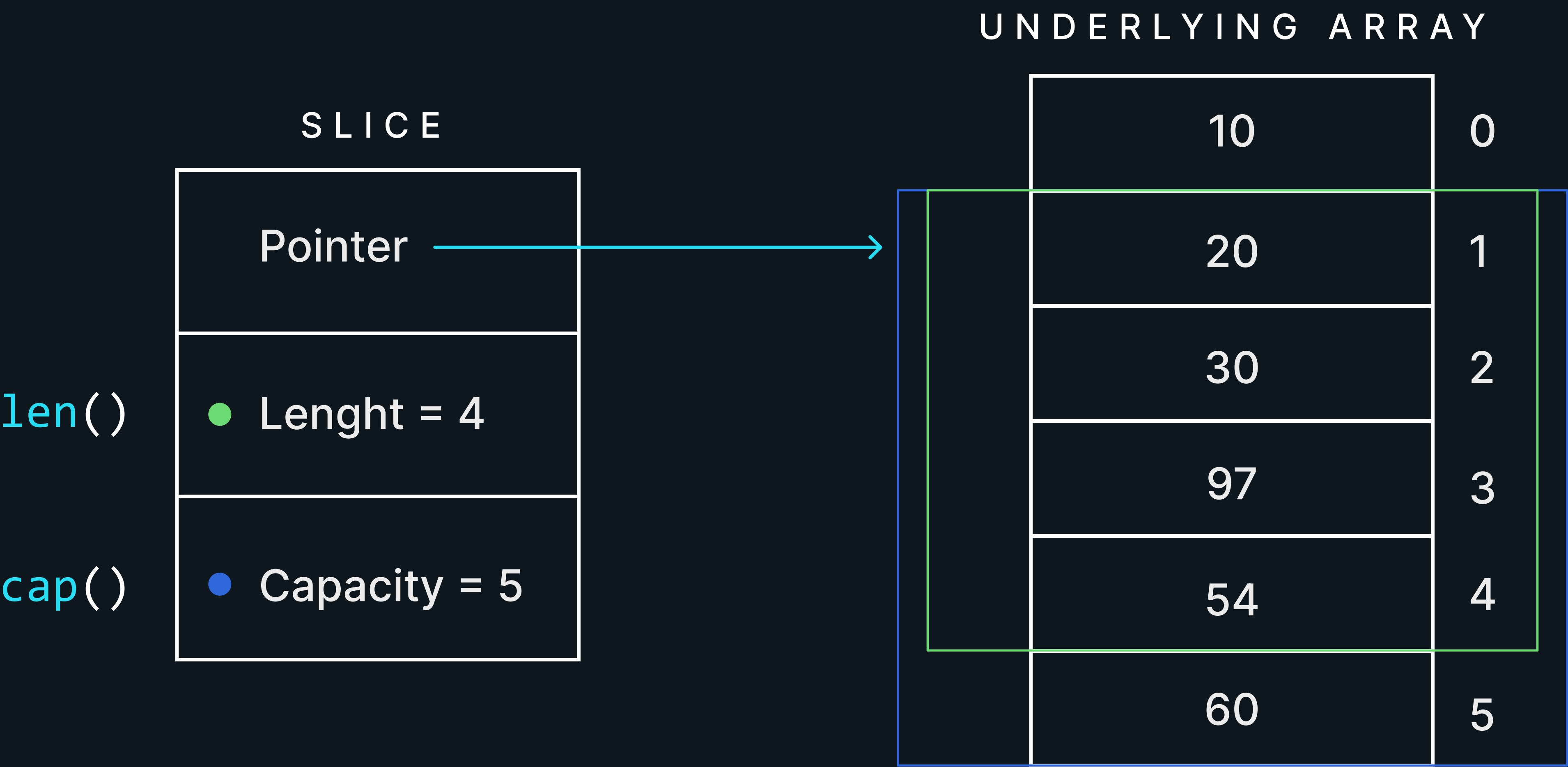


Slices

Are a dynamically-sized, flexible view of an underlying array. **Slices can grow or shrink as needed**, making them suitable for managing collections of variable size, such as lists, stacks, and queues.



Components Of A Slice



Slice Intialization

array[**start_index** : **end_index**]

↑ ↑
✓ Included ✗ Not Included

● array[0 : 3] ● array[1 : 6]

● array[: 4] ● array[:]

! Slices provide a more versatile way to work with collections of data compared to traditional arrays.



Initializing A Slice

- Name of our Slice
- Empty Square Brackets
- Data Type
- Values



Remember you don't need to specify the size of the slice at the moment of declaration

```
main.go

package main
import "fmt"

func main {
    slice := []int{10, 20, 30}
    fmt.Println(slice)
}
```

Slice and sub_slice

```
main.go

package main
import "fmt"

func main {
    slice := []int{10, 20, 30, 40, 50, 60, 70, 80, 90, 100}

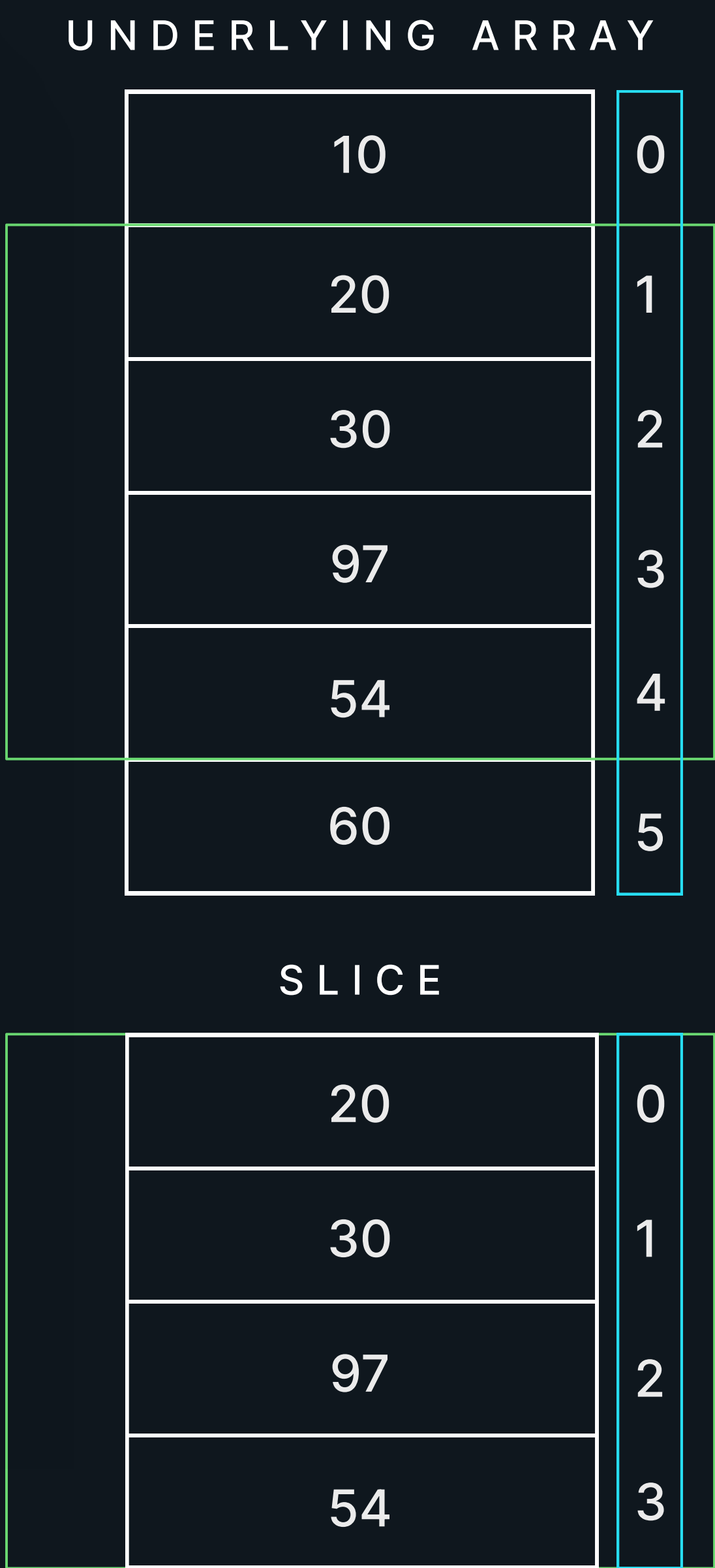
    slice := arr[1:8]
    fmt.Println(slice)

    sub_slice := slice[0:3]
    fmt.Println(sub_slice)
}

>>>go run main.go
[20 30 40 50 60 70 80]
[20 30 40]
```

- Name of our Slice
- Name of sub slice
- Syntax
- Output

Slice and index numbers



- Index numbers are affected in slice

Appending to a slice

```
>_
func append($ []T, vs ...T) []T

>_
slice = append(slice, element-1, element-2)

>_
slice = append(slice, 10, 20, 30)
```

- Built-in append function
- Slice name
- Values of data type
- Result = slice containing all the elements of the original slice + provided values

Deleting from a slice

```
main.go

package main
import "fmt"

func main {
    arr := [5]int{10, 20, 30, 40, 50}
    i := 2
    fmt.Println(arr)

    slice_1 := arr[:i]
    slice_2 := arr[i+1:]

    new_slice := append(slice_1,
        slice_2...)

    fmt.Println(new_slice)
}

>>>go run main.go
[10 20 30 40 50]
[10 20 40 50]
```

- We would delete the element at index 2
- Slicing the array
- Appending the 2 slices together
- Output



The process involves slicing two arrays and then appending them together!

Looping through a slice

```
main.go

package main
import "fmt"

func main {
    arr := []int{10, 20, 30, 40, 50}

    for _, value := range arr {
        fmt.Println()value
    }

    >>>go run main.go
10
20
30
40
50
```

- Code
- Output



Ready to embark on your journey into the world of Go (Golang)?

Join our introductory course to Go (Golang) and unlock the power of this versatile programming language!



TAUGHT BY:

Priyanka Yadav

★★★★☆ 4.8

<https://kode.wiki/48m805N>

