

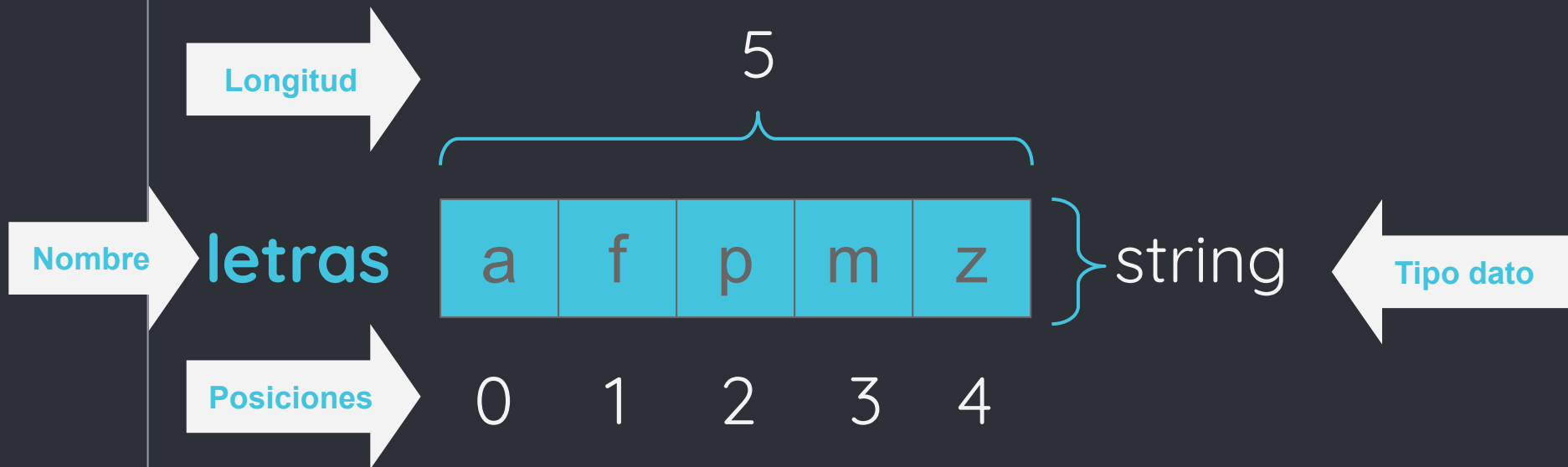
# ● Estructuras de datos

Golang

# • Arrays

Colección ordenada de valores el mismo tipo de dato de longitud fija.

Son copiados cuando se asignan o se pasan en funciones.



planetas[9]

Mercurio	0
Venus	1
Tierra	2
Marte	3
Jupiter	4
Saturno	5
Urano	6
Neptuno	7
Pluton	8

Declaracion

```
var planetas [9]string
```

Asignacion

```
planetas[0] := "Mercurio"
```

```
planetas := [9]string{"Mercurio", "Venus", "Tierra", "Marte",  
"Jupiter", "Saturno", "Urano", "Neptuno", "Pluton"}
```

Composite  
literal

Acceso

```
pluton := planetas[8] //"Pluton"
```

Loop

```
for i := 0; i < len(planetas); i++ {  
    fmt.Println(i, planetas[i])  
}
```

```
for i, planeta := range planetas {  
    fmt.Println(i, planeta)  
}
```

# • Slices

- Considerado como un segmento de un array.
- Consta de 3 componentes
  - Puntero
  - Longitud
  - Capacidad
- Comparten la misma data subyacente cuando se asignan o se pasan en funciones.
- Se puede adjuntar métodos a los slices.

terrestres :=  
`planetas[0:4]`  
`planetas[:4]`

Mercurio

0

Venus

1

Tierra

2

Marte

3

gaseosos :=  
`planetas[4:6]`

Jupiter

4

Saturno

5

gelidos :=  
`planetas[6:9]`  
`planetas[6:]`

Urano

6

Neptuno

7

Pluton

8

`planetas[9]`

`todos := planetas[:]`