

ARRAY e SLICE

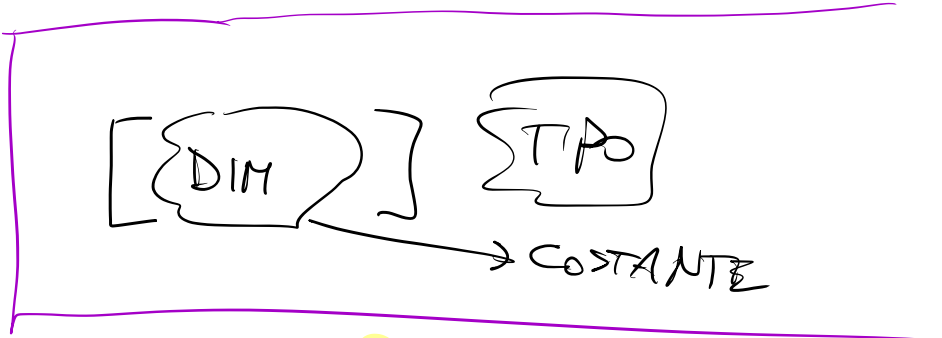
- x_1, x_2, \dots, x_n

$$\mu = \frac{x_1 + \dots + x_n}{n} = \frac{\sum_{i=1}^n x_i}{n}$$

$\mu = 12.31 \text{ €}$

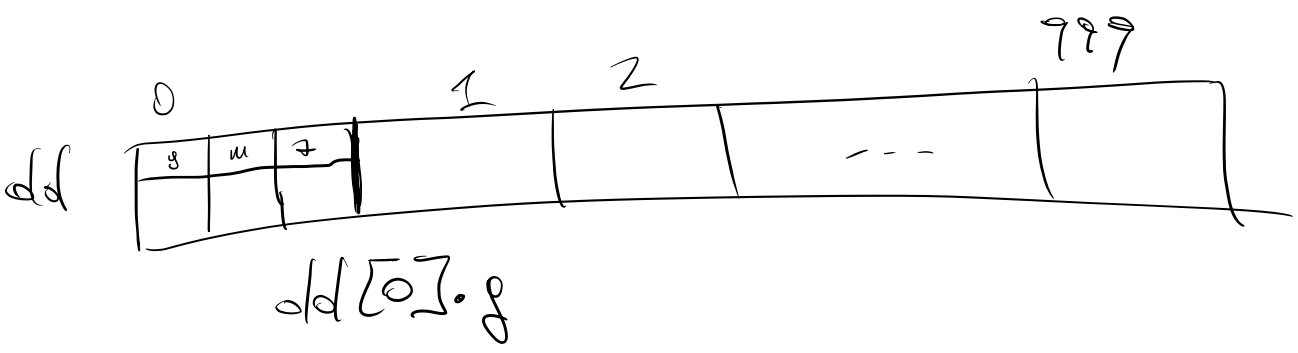
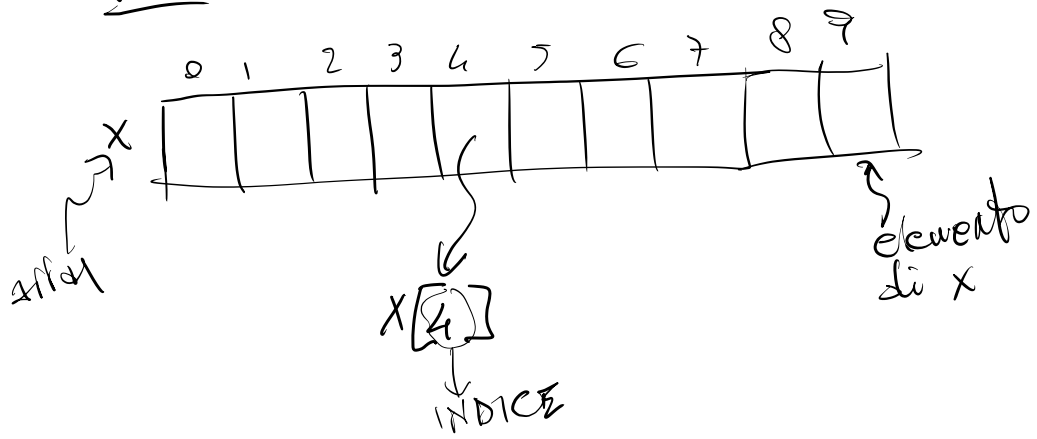
$$\sigma = \sqrt{\frac{(x_1 - \mu)^2 + \dots + (x_n - \mu)^2}{n}} = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n}}$$

ARRAY



```

var x [10] float64
var y, z [150] int
var ss [100] string
type data struct {
    g, m, z int
}
var dd [1000] data
    
```



```

func main () {
  var port [200] float64
  var n int
  var x, s, media, sq, sqm float64

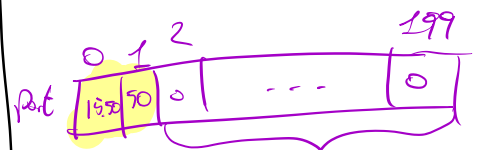
```

```

  for n < len(port) {
    fmt.Scan(&x)
    if x == -1 {
      break
    }
    port[n] = x
    n++
  }

```

CICLO DI LETTURA



```

  for i := 0; i < n; i++ {
    s += port[i]
  }

```

```

  media = s / float64(n)

```

```

  for i := 0; i < n; i++ {
    sq += (port[i] - media) * (port[i] - media)
  }

```

```

  sqm = math.Sqrt(sq / float64(n))

```

```

  fmt.Println(media, sqm)

```

}

ARRAY E CICLI

I° $\left[\begin{array}{l} \text{for } i := 0 ; i < \text{len}(a) ; i++ \{ \\ \dots a[i] \} \dots \\ \} \end{array} \right.$

II° $\left[\begin{array}{l} \text{for } i := \text{range } a \{ \\ \dots a[i] \} \dots \\ \} \end{array} \right.$ $\left. \begin{array}{l} \text{come I°} \\ \text{fino a} \\ \text{len}(a) \end{array} \right]$

III° $\left[\begin{array}{l} \text{for } i, x := \text{range } a \{ \\ \dots i \dots x \dots \\ \} \end{array} \right.$ $\left. \begin{array}{l} \text{come I°} \\ \text{fino a} \\ \text{len}(a) \end{array} \right]$

NON MODIFICA IL CONTENUTO

```

I0 {
  for i := 0; i < n; i++ {
    s += port[i]
  }
}

```



```

III0 {
  for i, x := range port {
    if i == n {
      break
    }
    s += x
  }
}

```

ARRAY E FUNZIONI

```

func somma (x [100] float64) float64 {
  ...
}

```

```

func main () {
  var a [100] float64
  s := somma(a)
}

```

}

ARRAY: ASSEGNAMENTO E CONFRONTO

var a, b [100] int

a = b
↑
copia

a[3] = b[7]

if a == b

if a != b

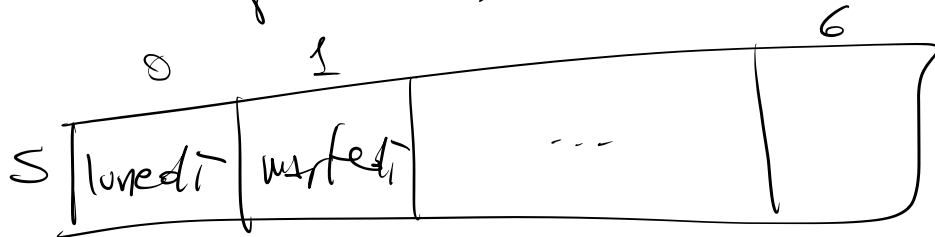
LETTERALI ARRAY

x := [8] int {1, 2, 3, 4, 5, 15, 30, 123}
 x := [8] int {1, 2, 3}
 x := [8] int {1: 5, 7: 700}
 x := [...] int {50, 60, 70}

l^---

var s [7] string

s = [...] { "lunedì", "martedì", "mercoledì",
"giovedì", "venerdì" }



s[0][0]

for _, r := range s[0] {

}

& s[5]

SLICE

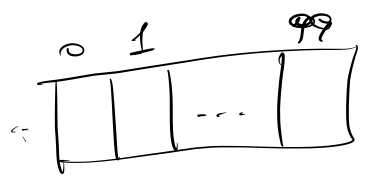
- LE SLICE NON HANNO UNA DIM. FISSA

[] TIPO

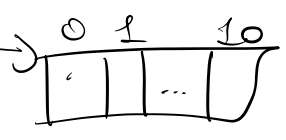
Decl.	<u>var</u>	x	[] int
	<u>var</u>	ss	[] string
	<u>var</u>	a, b	[] float64
	<u>var</u>	dd	[] data

↓
VALORE INIZIALE nil
~~info~~ non nec. costante

→ x = make([] int, 100)

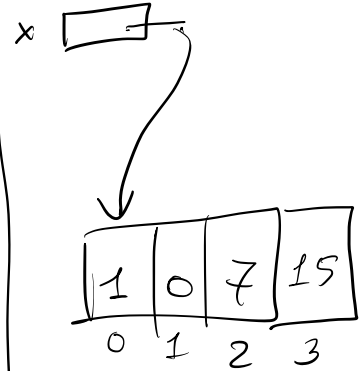


x = make([] int, 10)

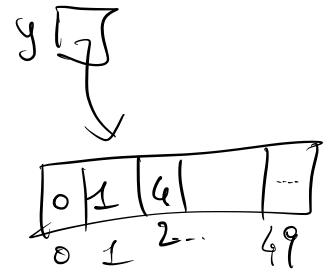


PROLUNGAMEN 7

```
→ var x []int
   x = make([]int, 3)
   x[0] = 1
   x[2] = 7
   x = append(x, 15)
```



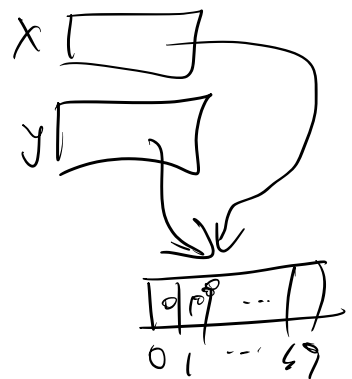
```
var y []int
for i := 0; i < 50; i++
    y = append(y, i*i)
}
```



ASSESSAMEN 7

```
var x, y []int
for i := 0; i < 50; i++
    y = append(y, i*i)
}

x = y
x[0] = 1000
```



CONFRONTI

NOI
2

LETTERAZI

$x := \underbrace{[]}_{\text{iat}} \{1, 2, 4, 12, 9\}$


```

func main () {
  var port [ ] float64
  var n int
  var x, s, media, sq, sqm float64

```

```

  for {
    fmt.Scan (&x)
    if x == -1 {
      break
    }
    port = append(port, x)

```

→ CICLO DI LETTURA

```

  }
  n := len(port)
  for i := 0; i < n; i++ {
    s += port[i]

```

```

  }
  media = s / float64(n)
  for i := 0; i < n; i++ {
    sq += (port[i] - media) * (port[i] - media)

```

```

  }
  sqm = math.Sqrt(sq / float64(n))
  fmt.Println(media, sqm)

```

}

FUNÇÃO

```
func soma (x [] float64) float64 {  
    s := 0.0  
    for _, v := range x {  
        s += v  
    }  
    return s  
}  
  
var part [] float64  
s := soma (part)
```

arr x [int]

