

ANCORA SULLE STRUCT

```
type persona struct {  
    nome, cognome string  
    cfisc string  
    dataDiNascita data  
}
```

```
var p1, p2, p3 persona
```

```
p1 = persona { "Paolo",  
    "Boldi", "BLDPLA...",  
    data { 29, 11, 68 } }
```

```
p2 = p1
```

↳ copia campo per campo

```
p3 = persona { nome: "Paolo",  
    dataDiNascita:  
    data { 29, 11, 1968 } }
```

ESEMPIO

```
type   data   struct {  
        g      int  
        m      int  
        a      int  
}
```

```
func   giorniDaEpoa (d data) int  
        :   d.g,   d.m,   d.a
```

↳

```
func   toString (d data) string {  
        return   strconv.Itoa(d.g)  
                + "/" + strconv.Itoa(d.m)  
                + "/" + strconv.Itoa(d.a)  
}
```

main

```
x = toString(date{29, 11, 1968})  
fmt.Println(x)  
29/11/1968
```

func

```
func fromString(s string) (d date, ok bool) {  
    if strings.Count(s, '/') != 2 {  
        return  
    }  
}
```

METHOD 1

```
index1 := strings.IndexRune  
        (s, '/')
```

```
index2 := strings.IndexRune  
        (s[index1+1:], '/')  
+ index + 1
```

gs := s[: index1]

ms := s[index1+1: index2]

as := s[index2+1:]

d.g, err = strconv.Atoi(gs)

if err != nil {
 return

}

d.m err = strconv.Atoi(ms)

..

ok = true
return

}

var s string

fmt.Scan (&s)

var d data

d, ok := fromString(s)

if !ok {

fmt.Println("Error")

else {

...

}

ESERCIZI

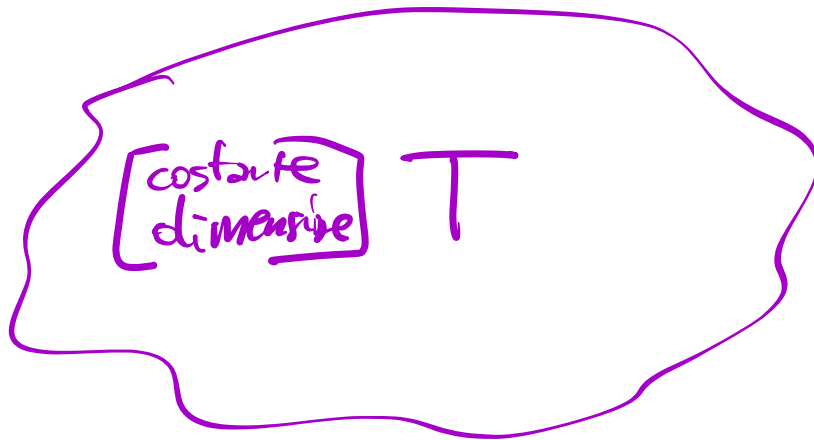
- Funzione che data una data restituisce i giorni dall'inizio di quell'anno
- Funzione che data un anno restituisce la data di Natale di quell'anno
- Funzione che data una data restituisca il giorno della settimana (0 = lunedì, 1 = martedì, ...)
- Funzione che data un anno stabilisce in che giorno si tengono le elezioni presidenziali

americane in quell'anno
(il primo martedì che
segue il primo lunedì
di novembre)

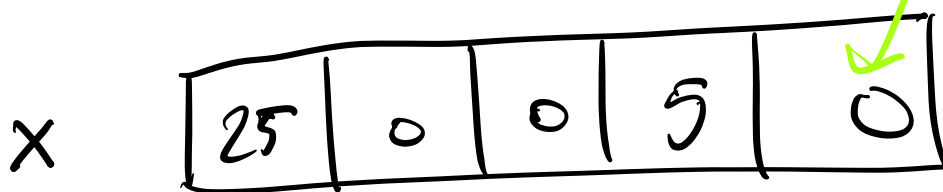
ARRAY E SLICE

- VETTORE

$$\underline{x} \in \mathbb{R}^3 \quad \underline{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$



var x [5] int



elemento

x[0] x[1] ...

x[4]

$$x[3] = 5$$

$$x[0] = x[3] * x[3] + x[4]$$

var

S

[10] string

P

[105] bool

x

[10] *int

y

[100] persona

y[7]. cognome

y[15]. data Di Nascita. 2

CALCOLO DELLA VARIANZA

$$\mu = \frac{x_1 + \dots + x_n}{n}$$

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

```
var    a [100] int  
        n  int
```

```
fmt. Print ("Quante persone ci sono?")
```

```
fmt. Scan (&n)
```

```
if    n > 100 {  
    fmt. Println ("NOOOO!")  
    return
```

for $i := 0; i < n; i++$
 $\text{sum} += a[i]$

}

$\text{sum} := 0$

for $i := 0; i < n; i++$
 $\text{sum} += a[i]$

}

$\text{mean} := \text{float64}(\text{sum}) /$
 $\text{float64}(n)$

$\text{sum_sq} := 0.0$

for $i := 0; i < n; i++$

$\text{sum_sq} +=$

$(\text{float64}(a[i]) - \text{mean})^2$

$(\text{float64}(a[i]) - \text{mean})^2$

(float64(a[i]) / mean)

}
sigma = math.Sqrt(summedScart /
float64(n))

fmt.Println("Media = ", media)
fmt.Println("Sq m = ", sigma)

ARRAY & CICLE

↓
for {
i := 0; i < len(a); i++ }
...
use a[i]
...
}

II.

```
for i := range a {  
  ...  
  usare a[i]  
  ...  
}
```

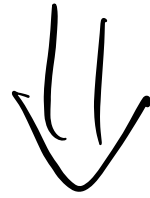
III.

```
for i, x := range a {  
  ...  
  usare i, a[i]  
  usare x (a[i])  
  SOLO IN LETTURA  
  ...  
}
```

```

for i := 0 ; i < n ; i++ {
  somma += a[i]
}

```



```

for _, x := range a {
  somma += x
}

```

$\forall x \in a$

LETTERALI ARRAY

$a := [8] \text{int } \{1, 2, 7, 4, 5, 2, 3, 2\}$

$b := [8] \text{int } \{1, 2\}$

$\{1, 2, 2, 2, 2, 2, 2, 2\}$

c := [...] int {1, 2, 3, 4}

d := [...] int {0: 43, 9: 5,
3: 71}

s := [100] string {0: "adma2",
22: "Trump" }

MISCELLANEA

len(x) → numero di
elementi
(lunghezza)
di x

var x, y [8] int

y = x ← copia

func soma (a [100] int) int