

TIPO string

string = slice immutabile di byte

```
var x string
x = "Paolo Boldi"
fmt.Println(x)
```

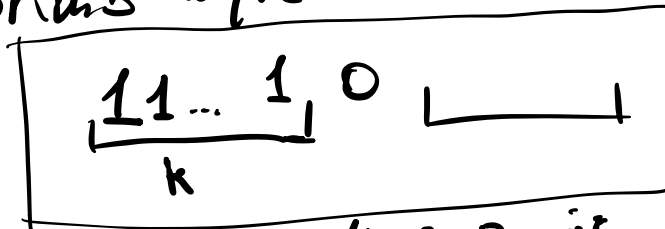
UTF-8 → ARRAY DI BYTE

UTF-8

1) È un formato di codifica delle stringhe

2) In questo formato, ogni carattere è rappresentato da 1-4 byte

3) Il primo byte di un carattere è

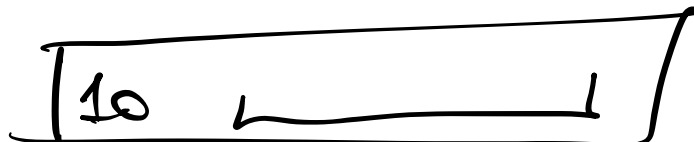


k = n° complessivo di byte

che inizia anche i caratteri ASCII



4) I successivi byte iniziano tutti con 10



CONTINUATION BYTES

ES. CARATTERE CHE OCCUPA 3 BYTE



ACCESSO A BYTE DI
UNA STRINGA

`len(x)`

lunghezza in
byte di una
stringa

`x[0], x[1], ..., x[n-1]`

i byte che compongono
la stringa

IMMUTABILI

var `x` string "Polo Boldi" ~~NO~~
`x[0] = 'x'`

var `x` string "Polo Boldi"
:
`x` "Bella Ciao"
:
OK

INPUT DI STRINGHE

var x string
funct. Scan (&x)

SI FERMA A
PRIMO
WHITESPACE

ITERAZIONE

SU

STRINGHE

ASCII

```
for i=0; i<len(x); i++  
    ... x[i] ...  
}
```

ANCHE NON-ASCII

INDICE DI BYTE RUNA

```
for i, r := range x {  
    ...  
}
```

SOTTO STRINGHE

INIZIO FINE

x[i:j]

with string a
all'indice
all'indice j

i (incluso)
(escluso)

$x =$ ^{0 1 2 3 4 5 6 7 8 9 10}
"Polo Boldi"
 $y = x[3:8]$

ESERCIZI

Dati una stringa

- 1) contare le lettere minuscole
- 2) cercare la posizione della prima "/"
- 3) contare le parole [ATTENZIONE!]