

Unicode

$$17 \cdot 2^{16}$$

Go

↓ UTF-32 = rune

TOTO
UNICODE

BYTE

4

ISO-8859-1
(Latin 1)

(*)

0-255

1

UTF-16

(*)

0-65535

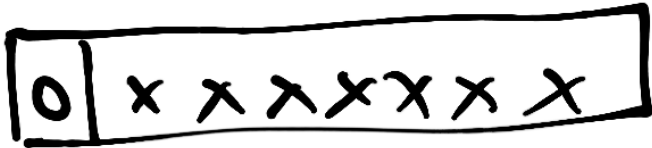
2

UTF-8

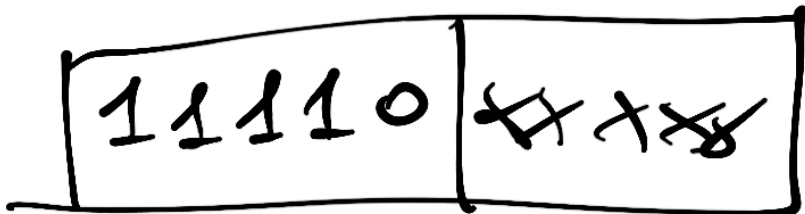
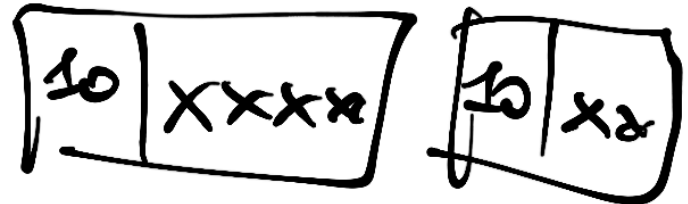
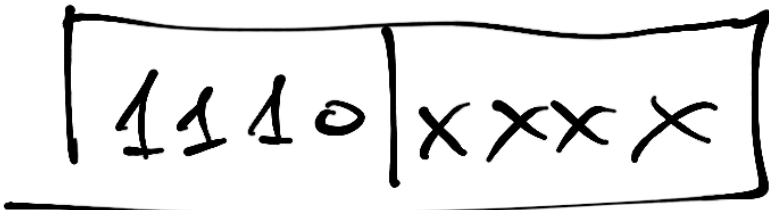
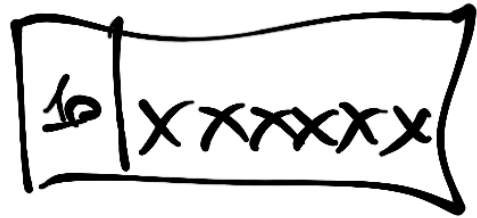
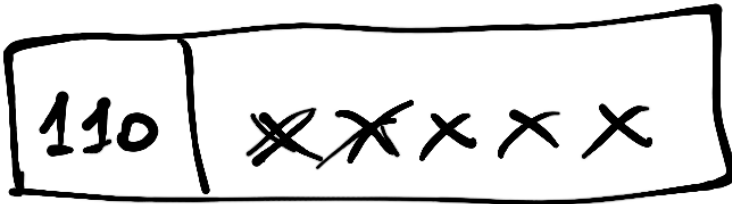
↑
string

UTF-8

1 character \leftrightarrow 1-4 byte



1 byte
(ASCII)



string = seq. di byte

- si usa di solito per
rappresentare sequenze
di caratteri codificati
in UTF-8

var s string

s = "Paolo"

s = "città"

s = "🍷 sei qui!"

#byte(s) ≥ #character(s) ≥ #glibc(s)

len(s)

S = "ciao \n ma mwa"

S = "ciao \u0000 \u001F\u2068" "comms"

S = "\x1F\x06\x15xA1"

DICHIARAZIONE

var s string

LUNGHEZZA (in byte)

len(s)

CONCATENAZIONE

s1 + s2

ACCESSO AI BYTE

s[i] \rightarrow uint8

\uparrow
i-esimo byte di s

$0 \leq i < \text{len}(s)$

if s[0] == 'm'

...

<u>var</u>	s	<u>string</u>
------------	---	---------------

fmt.Scan (&s)

fmt.Scan (&t)

<u>var</u>	x	<u>int</u>
------------	---	------------

<u>var</u>	s	string
------------	---	--------

fmt.Scan (&x)

→ fmt.Scan (&s)

fmt.Scan (&s)

397

Prolog

SLICING DI STRINGHE

$s[i:j]$

la stringa tra la pos. i
(compresa) e la pos. j
(esclusa)

$t = s[i:j]$

$s[i:]$

$s[:j]$

S := "mamua"
0 1 2 3 4

t := S[:3] // "mam"

v := S[1:2] // "a"

c := S[1:2][0] // 'a'

v := S[0:5] // "mamua"

SCANZONE byte PER byte

var s string

...

for $i := 0; i < \text{len}(s); i++$ {
... $s[i]$...

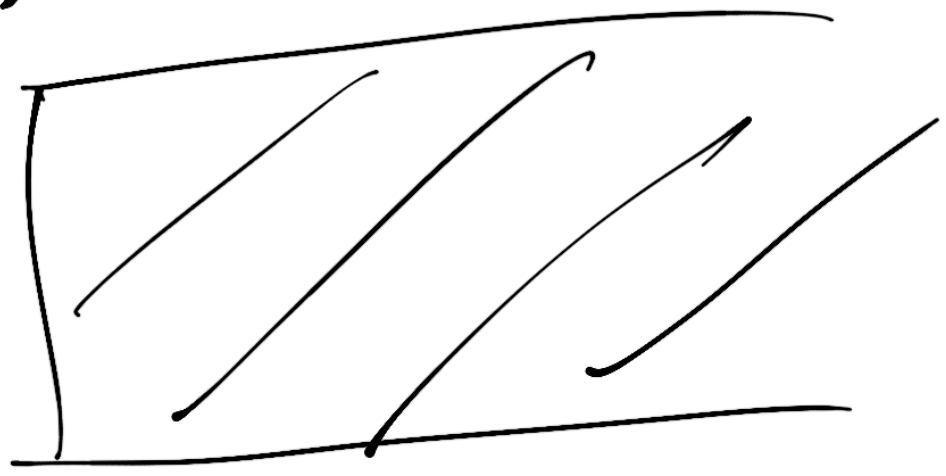
↓

for $i := \text{len}(s) - 1; i \geq 0; i--$ {
... $s[i]$...

}

for-range

for ^{indice} i , ^{carattere} $x :=$ range ^{stringa} s {



}