

```

rev := ""
for _, r := range s {
    rev = string(r) + rev
}

```

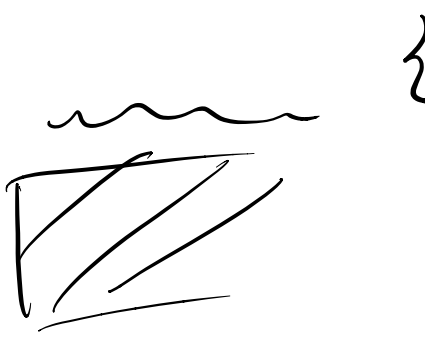
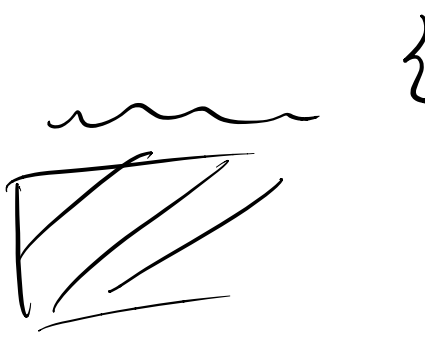
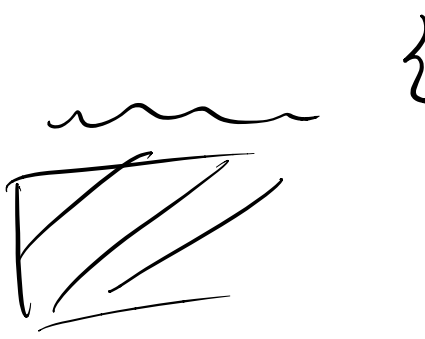
p a p a
 ↑ ↑ ↑ ↑

rev "a`pap"

bool

true

false

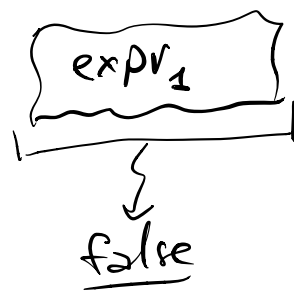
if  if  if 

&&

||

!

SHORT-CIRCUIT EVALUATION

if  && 

if $x \neq 0$ $\&\&$ $y/x == 5$

 ↓
 false

~~↑~~

if $y/x == 5$ $\&\&$ $x \neq 0$

SELEZIONE MULTIARIA

switch
case

espressione
espr, espr, ...

relettore

opzionale
true

case

espr, espr, ...

default:

}

opzionale

(x14)

switch {

case $x > y$:
 funt. Println ("A")

case $y > x$:
 funt. Println ("B")

default:
 funt. Println ("C")

}

```
if x > y {  
      
}  
else if y > x {  
      
}  
else {  
      
}
```

STRING

CONFRONTI FRA STRINGHE

==	!=
>	>=
<	<=

"Polo"	==	"Polo"
"Polo"	==	"polo"
"Polo"	>	"Abate"
"Polo"	>	"Zutten"

true
false
true
false

Polo	Abate
↑	↑

Abate
↓ ↓ ↓ ↓
Abate

b := bufio.NewScanner(os.Stdin)

for b.Scan() {
 s := b.Text()

}

const

- Sintassi simile a var
ma serve per dichiarare
delle costanti, e deve
essere seguita da
un'inizializzazione

const PI float64 = 3.14

const MAX_VAL int = 1000

const MAX = 100

ESERCIZI

A leggere una riga e contare
 quante sono le parole
 (separate da spazi)
 [contare gli inizi di parola]

Per stringhe ASCII

B Per stringhe generiche

C $13 \times 4 = 52$

$0 \rightarrow 12$

cuori

$0 \rightarrow 51$

$13 \rightarrow 25$

quarti

scrivete un programma che
 legge un numero da 0 a
 51 e stampa la carta
 corrispondente

7♥

J♦