

SYNTACTIC SUGAR

- RUOLO DELL'A-CAPPO
- ; ← terminatori di istruzioni
elevatore
(segnale, fine di funz., --)
(break, return, continue)
- {} ovvero inizio e fine istruzioni

STRUTTURE DI CONTROLLO

- if
- switch
- for
- while
- do ... while
↑
- break - continue

DICHIARAZIONE DI VARIABILE

var {id}

{tipo}

:=

{tipo} {id};

NON ESISTE

TCP1 D1 BASE

int
unsigned (int)

long
unsigned long

char
float

double

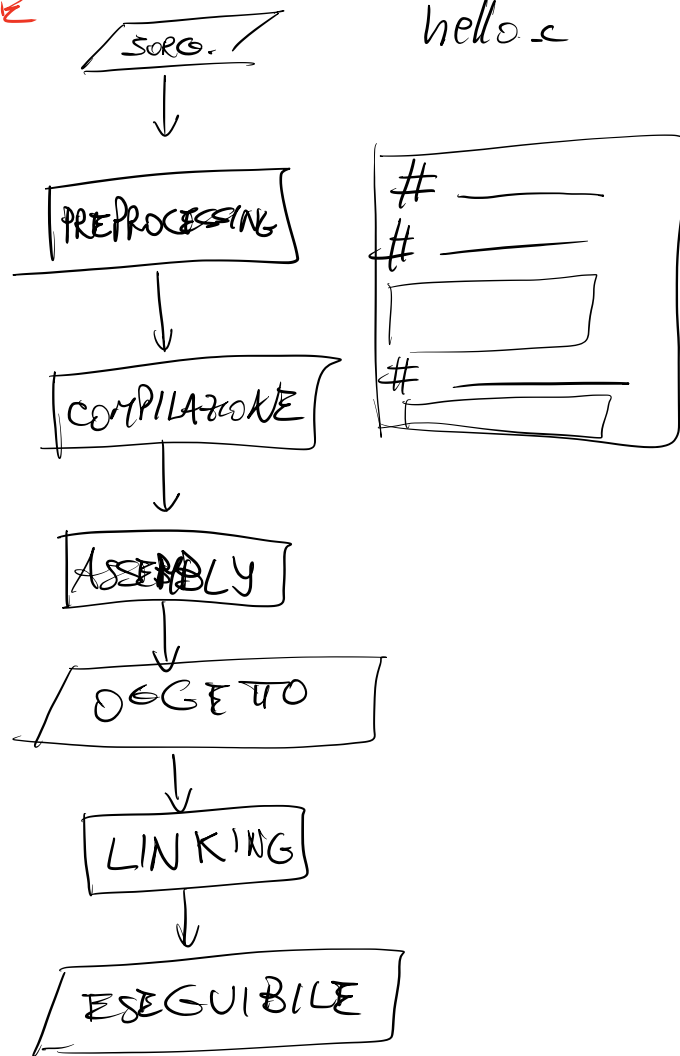
NO

{
→ bool
→ string

↔ interi $\begin{cases} = 0 & \text{false} \\ \neq 0 & \text{true} \end{cases}$

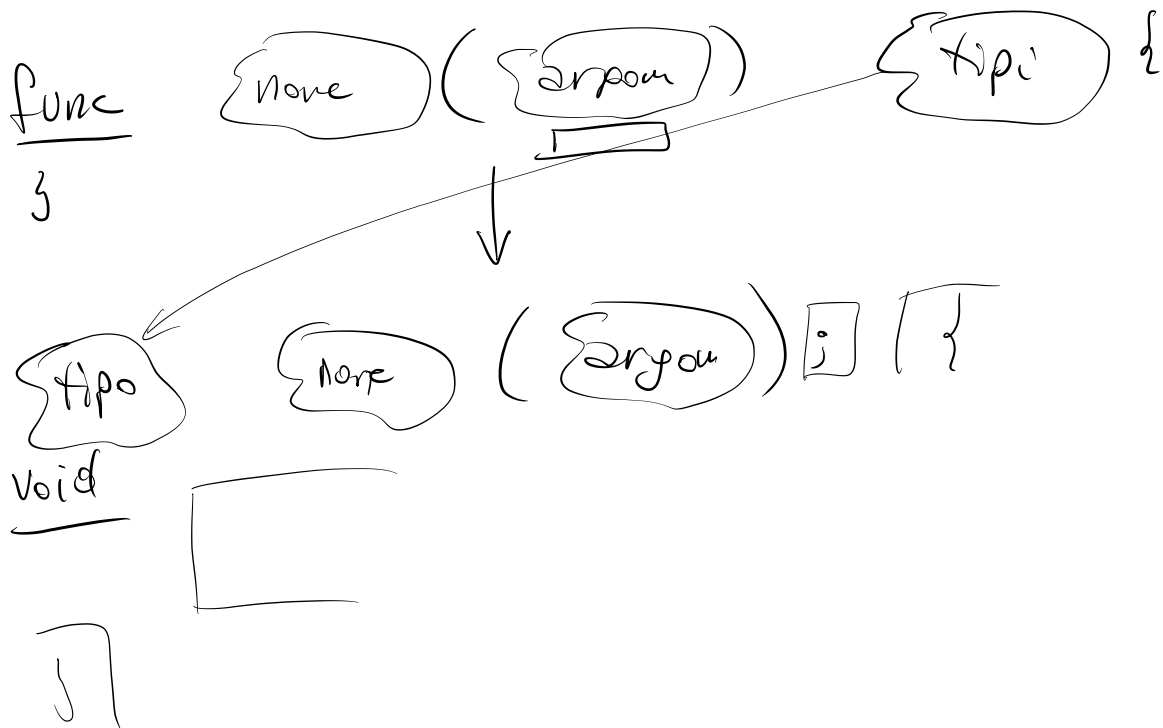
if (i) {
 ...
} else {
 ...
}

COMPILAZIONE



FUNCTION

HEADER



func f (x int) (int, int) {
 a, b = f(x)

}

void

f (int x, int *p1, int *p2)
 f(x, &a, &b)

ESPRESSIONI

double f;
int x;

f = x ;

x = (int)(f);
 ↓
 double

← CASTING
verso il base

ⓧ → Media = (double)(somma) / n

Ⓝ → media = (double)(somma / n)

$$y = \underbrace{a[x++]}$$

↓
espressione con
effetti collaterali

$$x = 6$$

$$y = a[++x]$$

$$y = (a[x++] - a[--y]) / (++i)$$

$$a = \underbrace{3 + (y = 5)}$$

$$a = \underbrace{x > y ? b + 1 : b * 2};$$

$$t[x > y ? i : j] = 1$$

```
if (x > 4) {  
    t[i] = 1  
}  
else {  
    t[j] = 1  
}  
}
```

Tipi Composti

Go	C
struct	sì, identiche
map	no
array	sì, con differenze
slice	no
puntatori	sì, permissivi
tipi funzione	sì
interfacce	no