

Introduzione a C++

~~C++~~

↑ 2to livello, vicino al
la guaglio unno
print ("cizo") ← python

System.out.println ("cizo"); ← Java

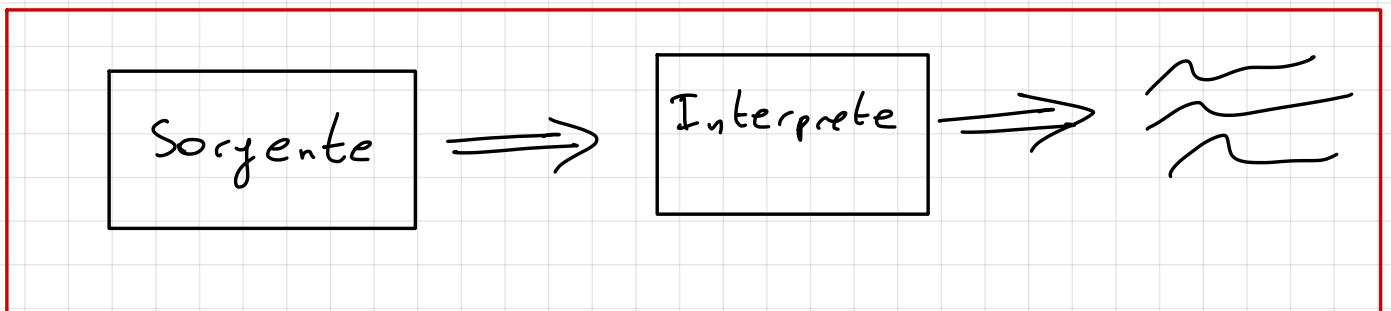
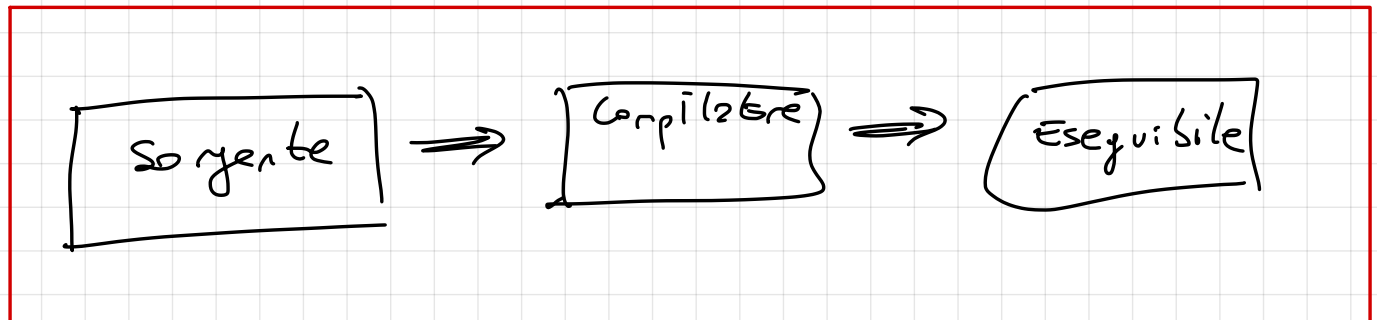
printf ("cizo"); ← C

cout << "cizo"; ← C++

↑ "basso", livello perche
piu vicino al
linguaggio macchina

⇒ Es. Assembly!

Compilato



Interpretato

↑ inter
switch (a) {
 case 0:
        ~~~~~  
        break;  
    case 1:  
        ~~~~~  
 break;
}

\hookrightarrow while () {
} while ();
do {
}

fino a questo è vero
 "fino a che"
 "fino a quando"
 Continui a farlo.
 Ci entra sempre!!

Int

cont = 0;

or

while (cont == 0 || cont >= 1) {

cont <= cont;

cont++;

}

cont = cont + 1;

```
int cnt = 0;
```

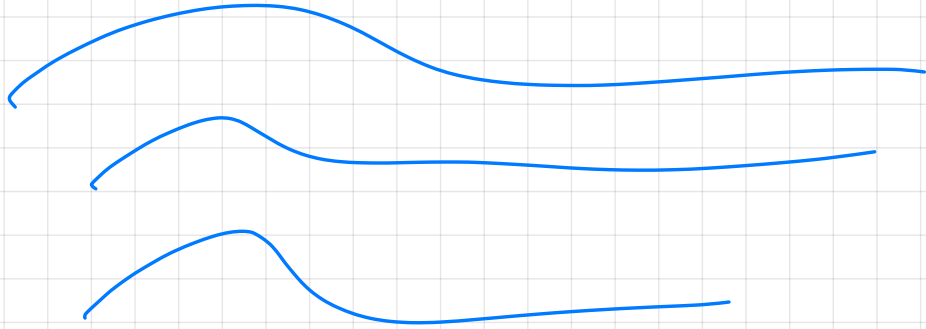
```
do {
```

```
    cout << cnt;
```

```
    cnt++;
```

```
} while ( cnt <= 1 );
```

```
for ( int i = 0 ; i < 5 ; i ++ ) {
```



```
}
```

```
int i = 0;
```

```
while ( i < 5 ) {
```

```
    i ++;
```

```
}
```

And

8x 8x

cond 1	cond 2	result
vero	falso	falso
falso	vero	falso
falso	falso	falso
vero	vero	vero

OR

||

shift + sx di 1

cond 1	cond 2	result
vero	falso	vero
falso	vero	vero
falso	falso	falso
vero	vero	vero

! = \longrightarrow \downarrow verso

!(true) \longrightarrow false

int a = 3;

(a == 3)

\hookrightarrow boolean!

vero
~~false~~

bool b = (a == 3);

true

cout << b; // true

int a = 1;

true!!!

if (a) {


entra!

}

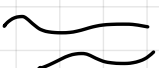
$\text{if } (((5/2) * 3) + 4) == \sim \{$



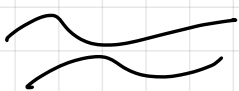
}

$\text{if (true) \{$

 $\}$

→ *entra sempre!!*

$\text{if (false) \{$

 $\}$

→ *non entra!!*

$\text{if (!false) \{$

 $\}$

→ *entra sempre!!*

IF (5) {
 ↑ $\neq 0$
 ~
 ~
 ~
}

}

sempre
vero!

Funzioni

tipo
di ritorno

nome funzione (parametri) {

Corpo della funzione

}

String cicloMagno () {

}

void ricorsiva () {

ricorsiva ();

}

→ non termina mai

tipo di
ritorno

nome - funzione - ricorsiva () {

] caso base

] caso ricorsivo

}

```
void stampaFinoAO ( int contatore ) {
```

```
    if ( contatore == 0 ) {  
        return;  
    }
```

→ Il caso
in cui
usciamo

```
    cout << contatore;  
    contatore --;
```

← quella che
viene fatta
ad ogni
passaggio,

```
    stampaFinoAO ( contatore );
```

↓
In questo
caso stampare
e decrementare

```
}
```

MAIN

```
    stampaFinoAO ( 4 );
```

```
int contatore = 4;
```

Esempio di Void

```
void
```

```
    stampaInformazioni ( string nome ) {
```

```
        cout << "ciao sono Alessandro" << endl;
```

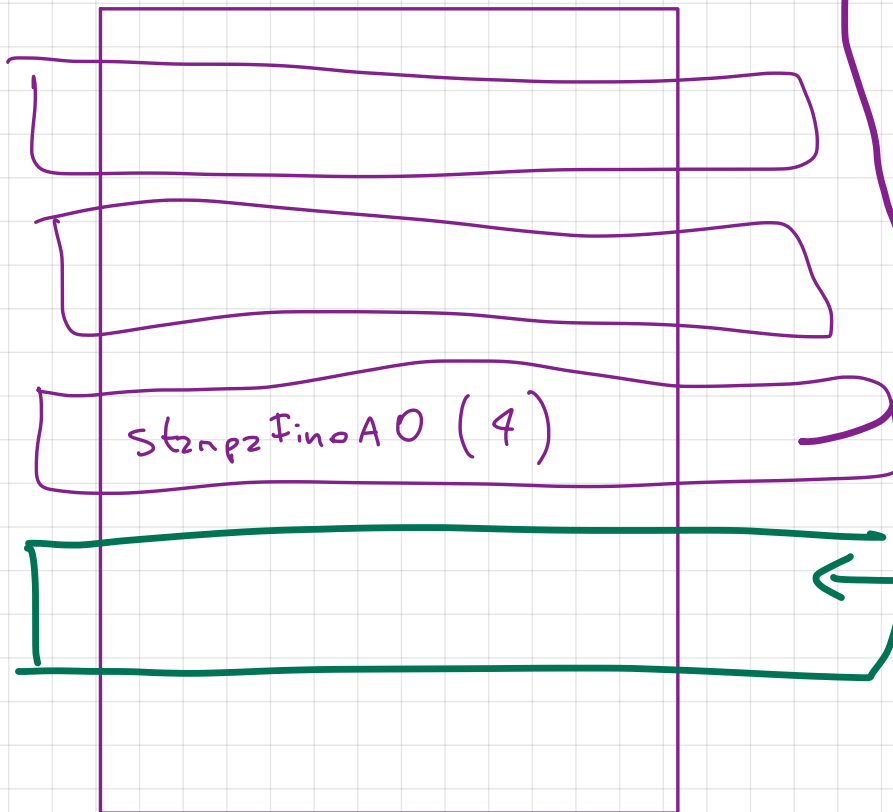
```
        cout << "sto solo scrivendo" << endl;
```

```
        int c = 3;
```

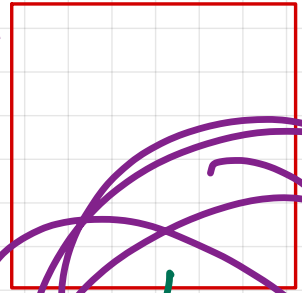
```
        cout << nome;
```

```
}
```

MAIN



Funktion



Return

Una funzione restituisce sempre
solo un valore!!

return ;

non restituisce
null

int funzione() {

return 5 ;

}

restituisce un
valore (intero)

Main

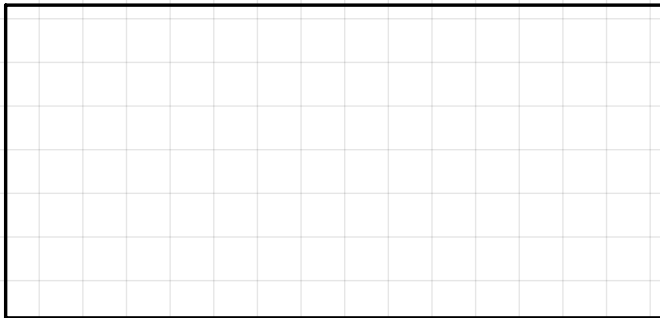
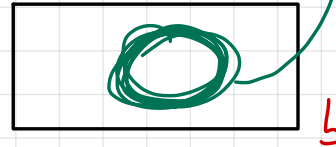
int a = funzione();

salvo 5 dentro a

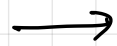
int a = 5;



int* b = a;



cout << &a;



& serves to recover the address of a variable.