

10 MARZO 2025

Lezione 4



Indice



01 Ciclo while

02 Ciclo for

1. Ciclo while

■ Utilizzo

Serve per eseguire blocchi di codice in maniera ciclica se e solo se si verifica una certa condizione.

Ogni qualvolta la condizione è vera, quel blocco di codice verrà eseguito.

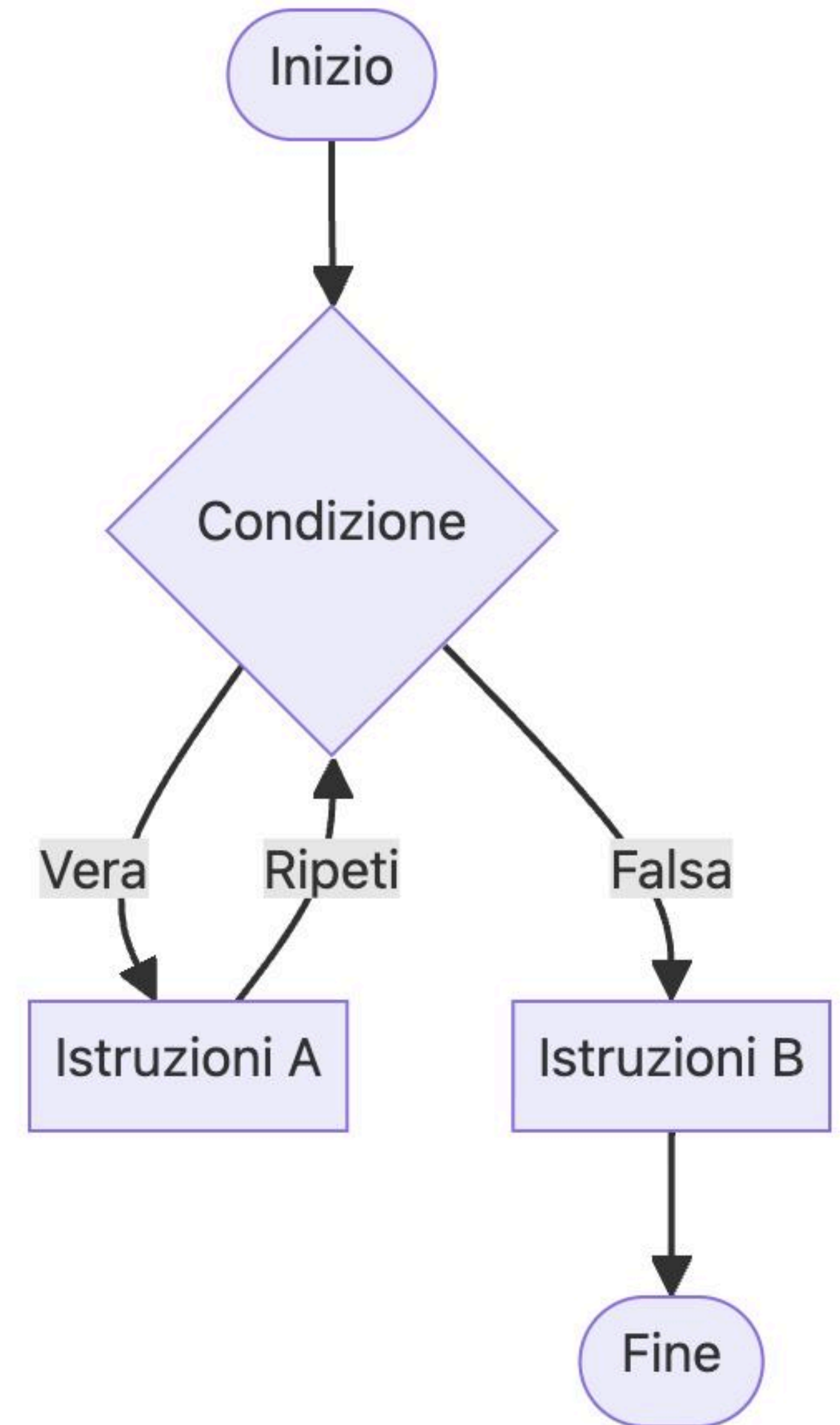
La condizione è rappresentata da una variabile booleana True o False.

```
# Esempio (no while)
print("Ciao a tutti!")
print("Ciao a tutti!")
print("Ciao a tutti!")
print("Ciao a tutti!")
print("Ciao a tutti!")
```

```
Ciao a tutti!
Ciao a tutti!
Ciao a tutti!
Ciao a tutti!
Ciao a tutti!
```

L'output è lo stesso, ma il ciclo while è compatto ed è stato creato per eseguire cicli

```
# Esempio (con while)
i = 0
while i<5:
    print("Ciao a tutti!")
    i += 1
```



Break & Continue

```
i = 1
while True:
    print(i*5)
    i += 1
    if i > 10:
        break
```

Condizione sempre vera quindi
il ciclo viene eseguito infinite
volte.

Condizione **break**, stoppa
il ciclo.

In questo caso è
all'interno dell'**if** quindi il
ciclo si ferma ad una
iterazione precisa, quando
la condizione per entrare
nell'**if** è verificata.

```
i = 0
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)
```

Condizione **continue**,
salta una iterazione del
ciclo e riparte da quella
dopo.

In questo caso quando la
condizione per entrare
nell'**if** si verifica, vengono
saltate tutte le istruzioni
successive e si riparte
dall'iterazione seguente.

2. Ciclo for

■ Utilizzo

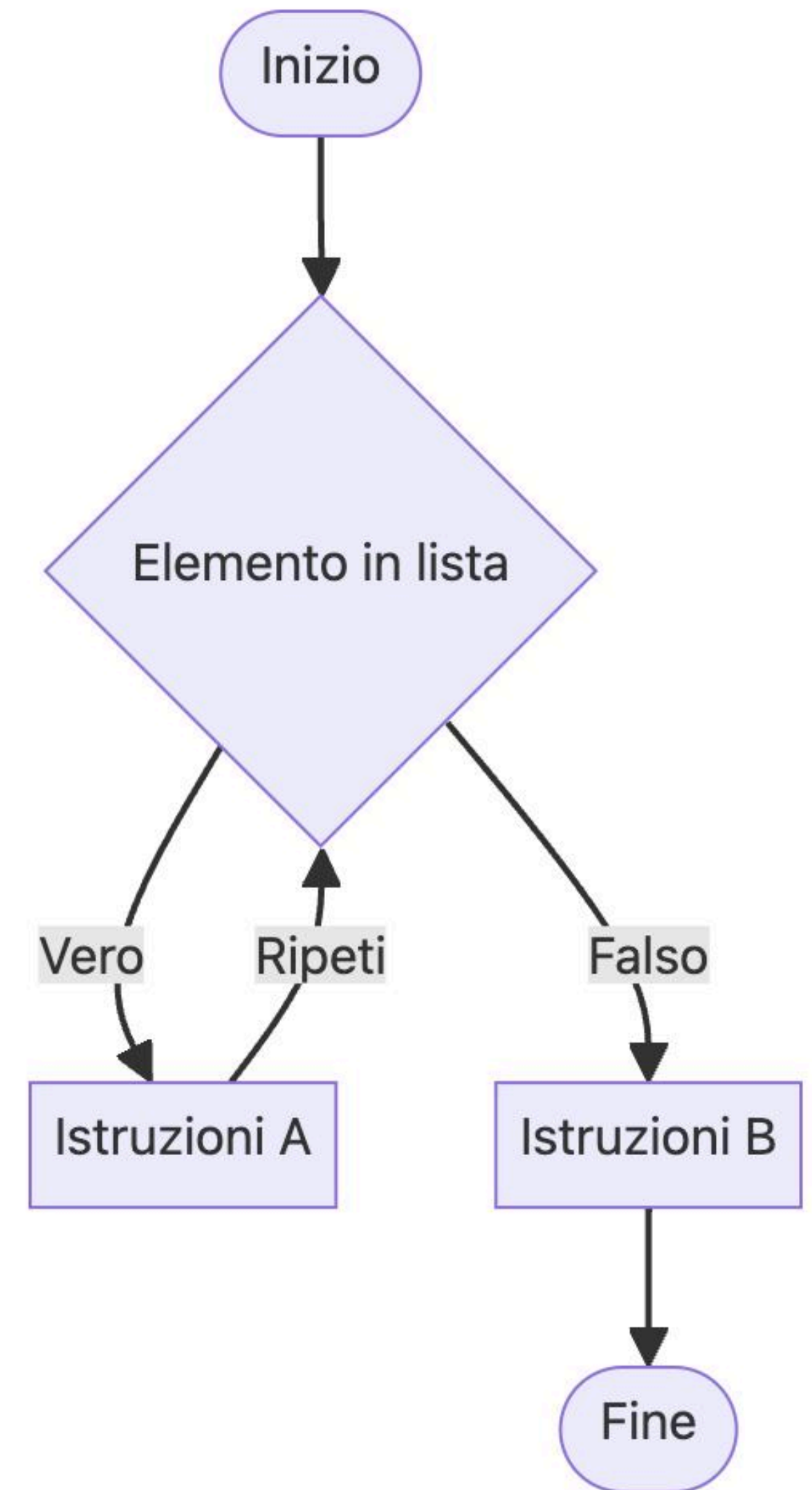
Anch'esso è utilizzato per eseguire dei blocchi di codice ciclicamente.

C'è sempre una variabile, contatore, che determina il numero di ripetizioni del ciclo.

La sua funzione principale, cioè la funzione per la quale viene utilizzato più spesso, è l'iterazione su sequenze di dati (liste, tuple, set e dizionari).

```
verdura = ["finocchio", "cavolo", "zucca"]  
for x in verdura:  
    print(x)
```

```
finocchio  
cavolo  
zucca
```



Parola chiave In

```
print("zucca" in verdura)  
print("carciofo" in verdura)
```

True
False

```
numeri = [3, 6, 8, 12, 13, 14, 21, 28]  
somma = 0  
for n in numeri:  
    somma += n  
print("La somma è: " + str(somma))
```

La somma è: 105

La parola chiave **in** è utilizzata per ottenere un'espressione booleana che determina la presenza di un elemento.

La funzione range()

```
# Tabellina del 5
for x in range(5, 51, 5):
    print(x)
```

```
5
10
15
20
25
30
35
40
45
50
```

Parameter	Description
start	Optional. An integer number specifying at which position to start. Default is 0
stop	Required. An integer number specifying at which position to stop (not included).
step	Optional. An integer number specifying the incrementation. Default is 1

```
range(start, stop, step)
```

In questo caso restituisce una lista di numeri da 5 a 50 con passo 5 (la tabellina del 5)

Per scorrere un blocco di codice un numero specifico di volte, possiamo utilizzare la funzione range().

La funzione range() restituisce una sequenza di numeri.

Break & Continue

```
verdura = ["finocchio", "cavolo", "zucca", "zucchina", "carota"]
for x in verdura:
    print(x)
    if x == "zucca":
        break
```

Condizione **break**, stoppa il ciclo.

In questo caso è all'interno dell'**if** quindi il ciclo si ferma ad una iterazione precisa, quando la condizione per entrare nell'**if** è verificata.

```
finocchio
cavolo
zucca
```

Condizione **continue**, salta una iterazione del ciclo e riparte da quella dopo.

In questo caso quando la condizione per entrare nell'**if** si verifica, vengono saltate tutte le istruzioni successive e si riparte dall'iterazione seguente.

```
verdura = ["finocchio", "cavolo", "zucca", "zucchina", "carota"]
for x in verdura:
    if x == "zucca":
        continue
    print(x)
```

```
finocchio
cavolo
zucchina
carota
```