

# Cicli while

## Introduzione

Buongiorno e benvenuti! Oggi parleremo dei **cicli**. Sono costrutti importantissimi nella programmazione perché, un po' come gli if else, permettono di eseguire il codice in un ordine ben preciso ma soprattutto, permettono di eseguire parti di codice più volte in un ciclo se una tale espressione o condizione è vera.

Sintassi:

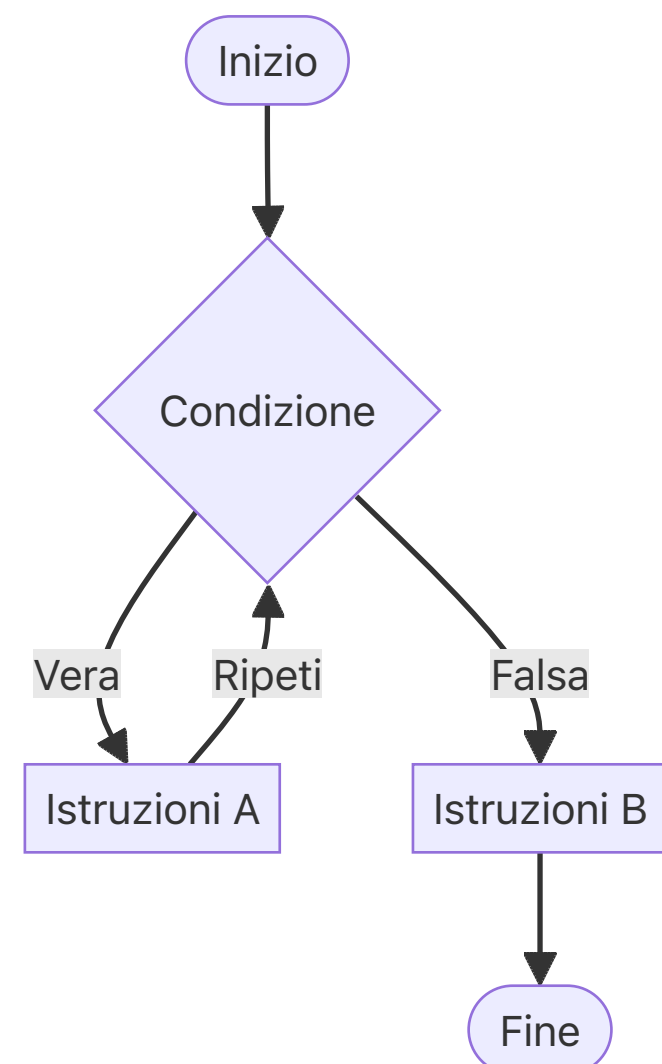
**while** condizione:

    blocco di codice eseguito se la condizione è vera

    blocco di codice eseguito quando la condizione è, o diventa falsa

Anche questa condizione deve essere espressa tramite operatori booleani e deve risultare o **True** o **False**.

## Diagramma di flusso



## Esempio

Come posso stampare 5 volte la stringa "Ciao a tutti!"?

```
In [19]: # Esempio (no while)
print("Ciao a tutti!")
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!

```
In [6]: # Esempio (con while)
i = 0
while i<5:
    print("Ciao a tutti!")
    i += 1
```

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

**ATTENZIONE!!!** Se la condizione è SEMPRE True, il ciclo non terminerà MAI! Il ciclo sarà infinito. Bisogna stare attenti perché potrebbe creare problemi.

```
In [ ]: # Meglio non eseguire
# while True:
#     print("Infinito")
```

## Esercizio 1

Stampare la tabellina del 3 utilizzando il ciclo while.

```
In [2]: i = 0
while i <= 10:
    print(str(3*i))
    i += 1
```

0  
3  
6  
9  
12  
15  
18  
21  
24  
27  
30

## Esercizio 2

Aggiungere ad ogni elemento della lista **lista** un valore **x** e stampare la nuova lista.

```
In [52]: x = 10
lista = [10, 20, 30]
i = 0
while i < len(lista):
    lista[i] = lista[i] + x
    i += 1
print(lista)
```

## Esercizio 3

Moltiplicare ogni elemento della lista **lista** per il valore dell'indice in cui si trova.

```
In [58]: x = 10
lista = [32, 12, 8, 5, 2]
i = 0
while i < len(lista):
    lista[i] = lista[i]*i
    i += 1
print(lista)
```

## Esercizio 4

Stampare tutti gli elementi di **l1** in una sola stringa usando il ciclo while. Inserire gli elementi della lista in una stringa singola e stamparla.

**l1 = [5, 10, 15, 20, 25]**

```
In [26]: l1 = [5, 10, 15, 20, 25]

indice = 0
lista_str = ""
while indice < len(l1):
    lista_str += str(l1[indice]) + " "
    indice += 1

# Stampa della somma
print(f"Esercizio 0 - La lista è: {lista_str}")
```

## Break & Continue

### Break

Con la parola chiave **break** possiamo fermare l'esecuzione del ciclo anche se la condizione è comunque **True**.

### Esempio

Come possiamo notare, nonostante la condizione suggerisca che verranno stampati tutti i numeri da 1 a 5 (dato che è vera sino a che i=5 e falsa da i=6 in poi), il ciclo stamperà da 1 a 3 grazie all'istruzione **break**.

```
In [74]: i = 1
while i < 6:
    print(i)
    if i == 3:
        break
    i += 1
```

1  
2  
3

### Continue

Con la parola chiave **continue** possiamo fermare l'esecuzione dell'attuale iterazione del ciclo e continuare dalla successiva.

### Esempio

Come possiamo vedere, verranno stampati tutti i numeri tra 1 e 6, ma non verrà stampato il 3 a causa del **continue**.

```
In [69]: i = 0
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)
```

1  
2  
4  
5  
6

## Esercizio 5

Utilizzare un **while True** (costrutto che abbiamo visto che significa loop infinito) per scrivere la tabellina del 5.

```
In [3]: i = 1
while True:
    print(i*5)
    i += 1
    if i > 10:
        break
```

5  
10  
15  
20  
25  
30  
35  
40  
45  
50