

Cos'è un ciclo

- Costruzione che permette di eseguire un blocco di istruzioni ripetutamente fino a quando una determinata condizione viene soddisfatta o finché una certa regola è rispettata

Componenti di un ciclo

- **Condizione:** Un ciclo è controllato da una condizione che viene valutata ad ogni iterazione.
 - Fino a quando la condizione è vera, il ciclo continua ad eseguire il suo blocco di istruzioni.
 - Quando la condizione diventa falsa, il ciclo si interrompe e il controllo passa alle istruzioni successive.
- **Contatore:** Spesso, i cicli utilizzano un contatore o una variabile di controllo per tener traccia delle iterazioni.
 - Questo contatore viene generalmente inizializzato prima dell'inizio del ciclo e viene aggiornato ad ogni iterazione.
 - Il contatore può essere utilizzato per controllare quante volte il ciclo deve essere eseguito.
 - Viene normalmente inizializzata (assegnata un valore), testata (avanza/diminuisce finché non raggiunge il valore finale di condizione del ciclo) e aggiornata (aumentata/diminuita) dopo ogni singola iterazione
- **Blocco di istruzioni:** Un ciclo contiene un blocco di istruzioni, che è ciò che verrà eseguito ripetutamente.
 - Questo blocco di istruzioni può essere costituito da qualsiasi sequenza di comandi necessari per compiere una specifica operazione.

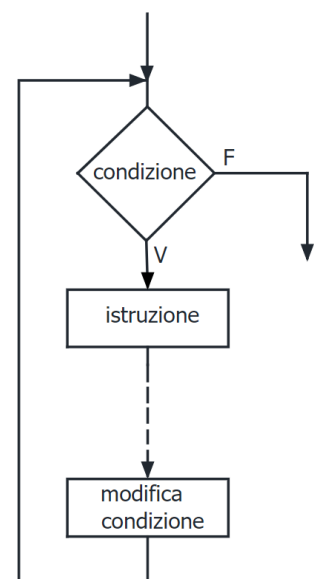
Differenza tra condizioni (if) e cicli (do-while/while/loop)

- Il corpo di un ciclo può essere eseguito zero, una o più volte
- Il corpo di una condizione viene eseguita al più una volta

Tipi di cicli

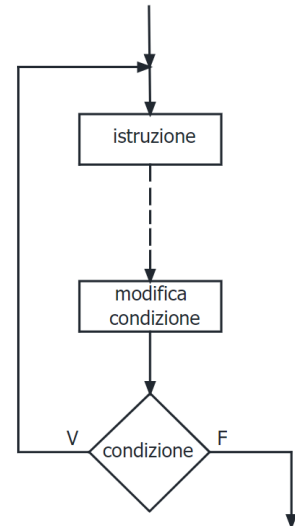
1) While:

- Quando si vuole eseguire un blocco di codice finché una condizione specifica è vera.
- Non viene conosciuto a priori il numero di iterazioni da fare
- La condizione viene controllata prima di ogni iterazione;
 - se la condizione è inizialmente falsa, il corpo del ciclo non viene eseguito
 - se la condizione rimane vera, il ciclo non termina (ciclo infinito)
- Va mantenuto un contatore (*i*) a mano per ricordarsi di incrementarlo ($i++$ oppure $i = i + 1$)



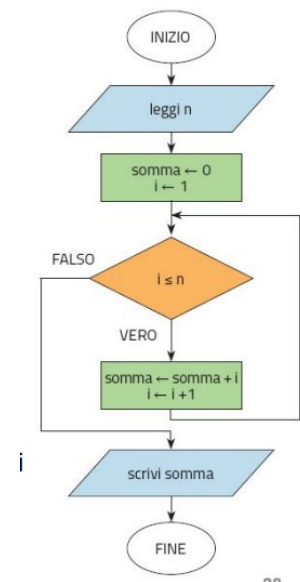
2) Do-While:

- Simile a *while*, ma garantisce che il blocco di codice venga eseguito almeno una volta, poiché la condizione viene verificata alla fine dell'iterazione.
- La condizione viene controllata dopo una prima iterazione; il corpo del ciclo viene comunque eseguito dopo una prima iterazione
- Va mantenuto un contatore (*i*) a mano per ricordarsi di incrementarlo ($i++$ oppure $i = i + 1$)



3) For

- Quando si sa esattamente quante volte si desidera eseguire il blocco di codice.
- Solamente un modo più lungo a parole per fare quello che fanno i due cicli precedenti (perché scrivi subito condizione/contatore/incremento ed eviti di usare a mano un contatore)
- Viene controllata la condizione del ciclo;
 - se la condizione è vera, si esegue il corpo del ciclo e si aggiorna il contatore
 - se la condizione diventa falsa, si esce dal *for* e si continua l'esecuzione con l'istruzione successiva
- Dato che conosciamo già l'intervallo in cui operare, qui sappiamo già che useremo una variabile contatore da far crescere o decrescere all'interno di un certo intervallo



Riassunto

- Il ciclo for quando conosci il numero di iterazioni che devi fare.
- Il ciclo while quando non conosci il numero di iterazioni che devi fare antepo-
nendo il controllo di verifica prima di iniziare.
- Il ciclo do-while quando non conosci il numero di iterazioni e posponi il controllo facendo in modo che il ciclo parta almeno una volta.