

06 MARZO 2025

Lezione 3



Indice



01 If else

02 Liste

03 Tuple

04 Sets

1. Istruzione if else

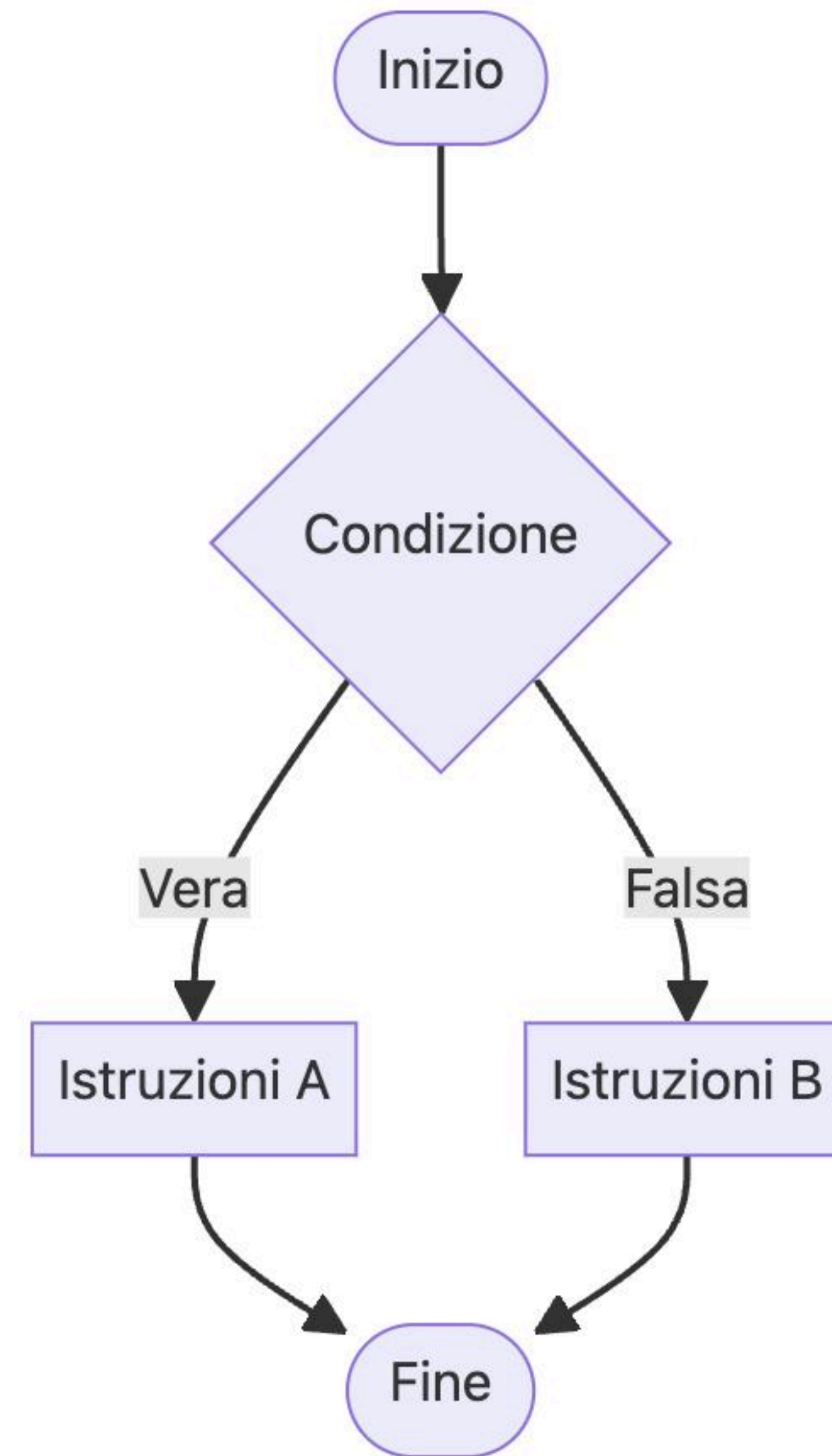
■ Utilizzo

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

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

```
a = 10
b = 20
if a < b:
    print("a è minore di b")
else:
    print("a è maggiore o uguale a b")
```

a è minore di b



Condizioni multiple

■ If elif

Se si hanno più condizioni sulla stessa variabile bisogna utilizzare la parola chiave **elif**.

```
a = 200
b = 33
if b > a:
    print("b è maggiore di a")
elif a == b:
    print("a e b sono uguali")
else:
    print("a è maggiore di b")
```

a è maggiore di b

■ If innestati

Se serve, è possibile inserire istruzioni **if else** all'interno di altri blocchi **if else**.

In gergo si chiamano **nested**.

```
x = 41

if x > 10:
    print("Maggiore di 10,")
    if x > 20:
        print("e anche maggiore di 20!")
    else:
        print("ma non maggiore di 20.")
```

Maggiore di 10,
e anche maggiore di 20!

2. Liste

Le **liste** sono usate per immagazzinare in una sola variabile più dati.

Caratteristiche principali:

- **Ordinate**: gli elementi hanno un ordine preciso basato sull'indice, che parte da 0. Questo ordine non cambia a meno che non venga modificato con metodi specifici.
- **Modificabili**: puoi aggiungere, cambiare o rimuovere elementi dopo aver creato una lista.
- **Permettono duplicati**: le liste possono contenere più elementi con lo stesso valore.
- **Tipi di dato**: possono contenere qualunque tipo di dato, anche mischiati

```
lista = ["mela", "banana", "ciliegia", "kiwi", "melone"]
```

NOTARE: anche le stringhe (i tipi *string*) sono delle liste!

Metodi

Accesso con indici

```
print(lista[1])  
print(lista[-2])  
print(lista[1:3])  
print(lista[2:])
```

```
banana  
kiwi  
['banana', 'ciliegia']  
['ciliegia', 'kiwi', 'melone']
```

```
frase = "Che bello questo corso, mi sto divertendo un sacco!"  
frase[24:]
```

```
'mi sto divertendo un sacco!'
```

Aggiungere elementi

Metodo	Descrizione
append()	Aggiunge un elemento alla fine della lista.
insert()	Inserisce un elemento in una posizione specificata della lista.
extend()	Estende la lista aggiungendo gli elementi di un'altra iterabile (lista, tupla, ecc.).

Per l'accesso

`stringa[start:stop:step]`

- **start** -> Punto di partenza (vuoto significa "inizia dall'inizio").
- **stop** -> Punto di fine (vuoto significa "arriva fino alla fine").
- **step** -> Indica di quanti passi muoversi (se è -1, significa "vai all'indietro").

Metodi

Rimuovere elementi

Metodo	Descrizione
remove()	Rimuove l'elemento specifico, se ce ne sono di più rimuove solo il primo
pop()	Rimuove l'elemento all'indice specifico, se non si specifica l'elemento rimuove l'ultimo
clear()	Rimuove tutti gli elementi, la lista diventa vuota

Miscellanea

Metodo	Descrizione
count()	Conta le occorrenze di uno specifico elemento
split()	ATTENZIONE si applica a una stringa; ritorna una lista con gli elementi creati dividendo la stringa tramite il separatore passato come parametro.
join()	ATTENZIONE si applica a una stringa; ritorna una stringa unendo tutti gli elementi della lista presa come parametro utilizzando la stringa come separatore

Funzione len(): non è un metodo ma è una funzione; permette di stampare il numero di elementi nella lista.

3. Tuple

Le **tuple**, come le liste, sono utilizzate per immagazzinare in una sola variabile più dati, ma hanno delle caratteristiche diverse.

- **Ordinate**: significa che gli elementi hanno un ordine definito, e tale ordine non cambierà.
- **Immutabili**: il che significa che non possiamo modificare, aggiungere o rimuovere elementi dopo che una tupla è stata creata.
- **Consentono duplicati**: poiché le tuple sono indicizzate, possono contenere elementi con lo stesso valore.
- **Tipi di dato**: possono contenere qualunque tipo di dato, anche mischiati

In breve sono delle strutture identiche alle liste ma i valori immagazzinati non possono essere cambiati!

```
tupla = ("mela", "banana", "ciliegia", "arancia", "kiwi", "melone", "mango")
```


Metodi

Accesso con indici

```
print(tupla[1])  
print(tupla[-2])  
print(tupla[2:5])  
print(tupla[:4])
```

```
banana  
melone  
( 'ciliegia', 'arancia', 'kiwi' )  
( 'mela', 'banana', 'ciliegia', 'arancia' )
```

Miscellanea

Metodo	Descrizione
count()	Conta le occorrenze di uno specifico elemento

Funzione len(): non è un metodo ma è una funzione; permette di stampare il numero di elementi nella tupla.

4. Sets

Set in italiano si traduce con **insieme**.

Anch'essi sono utilizzati per immagazzinare più valori in una singola variabile.

- **Non ordinati**: gli elementi in un set non hanno un ordine preciso e non possono essere usati con indici.
- **Parzialmente immutabili**: non puoi modificare i singoli elementi di un set, ma puoi aggiungere o rimuovere elementi.
- **Niente duplicati**: Un set non può contenere due elementi con lo stesso valore.
- **Tipi di dato**: possono immagazzinare diversi tipi di dato mischiati MA, True - 1 e False - 0, sono considerati lo stesso valore.

```
insieme = {"mela", "banana", "mango", 1, True, 1, 2}
```

Metodi

Aggiungere elementi

Metodo	Descrizione
add()	Inserisce un elemento all'interno dell'insieme (set)
update()	Aggiunge tutti gli elementi di una qualsiasi struttura dati all'interno dell'insieme (set)

Rimuovere elementi

Metodo	Descrizione
remove()	Rimuove l'elemento passato come parametro, se non esiste all'interno dell'insieme <i>produce un errore</i>
discard()	Rimuove l'elemento passato come parametro, se non esiste all'interno dell'insieme NON <i>produce un errore</i>
pop()	Rimuove un elemento random (a caso) dall'insieme
clear()	Rimuove tutti gli elementi nell'insieme

Miscellanea

Metodo	Descrizione
difference()	Restituisce un set con gli elementi presenti solo nel primo set (differenza).
intersection()	Restituisce un set con gli elementi comuni a entrambi i set.
isdisjoint()	Restituisce <code>True</code> se i due set non hanno elementi in comune.
union()	Restituisce un set con tutti gli elementi presenti in entrambi i set (unione).

Funzione len():
non è un metodo ma è una funzione; permette di stampare il numero di elementi nell'insieme.