

6 – Finding Items

Trovare elementi in una lista è un'operazione molto comune in Python. Esistono diverse tecniche per **verificare la presenza di un elemento**, ottenere il suo **indice** o cercare più valori secondo una condizione.

Verificare se un elemento esiste

L'operatore `in` permette di controllare rapidamente se un elemento è presente in una lista:

```
numeri = [10, 20, 30, 40]  
  
print(20 in numeri)    # True  
print(50 in numeri)    # False
```

Al contrario, `not in` verifica l'assenza:

```
print(50 not in numeri)  # True
```

Trovare l'indice di un elemento

Il metodo `index()` restituisce la posizione della **prima occorrenza** dell'elemento:

```
numeri = [10, 20, 30, 20]
posizione = numeri.index(20)
print(posizione) # 1
```

Se l'elemento non è presente, genera un `ValueError`.

Contare le occorrenze

Il metodo `count()` restituisce quante volte un elemento appare nella lista:

```
conteggio = numeri.count(20)
print(conteggio) # 2
```

Trovare elementi con condizioni

Puoi usare un ciclo `for` o list comprehension per cercare elementi secondo criteri specifici:

```
numeri = [5, 10, 15, 20, 25]

# tutti i numeri maggiori di 15
maggiori = [n for n in numeri if n > 15]
print(maggiori) # [20, 25]
```

Funzioni built-in utili

- `min(lista)` → restituisce il valore minimo
- `max(lista)` → restituisce il valore massimo
- `sum(lista)` → somma di tutti gli elementi numerici

```
print(min(numeri)) # 5
print(max(numeri)) # 25
print(sum(numeri)) # 75
```

Riassunto

- Usa `in` e `not in` per verificare la presenza o assenza di elementi.
- `index()` restituisce la posizione della prima occorrenza.
- `count()` permette di contare le ripetizioni.
- Cicli e list comprehension consentono di filtrare gli elementi secondo condizioni specifiche.

Vuoi continuare con il **paragrafo 7 – Sorting Lists?**