

<https://www.tankerino.com/it/corsi/2/introduzione-alla-programmazione/lezioni/123/le-liste>

## Le Liste nella Programmazione

Le liste sono una delle strutture dati più fondamentali nella programmazione. Una lista è una collezione ordinata di elementi che possono essere modificati. Ogni elemento all'interno di una lista ha una posizione specifica, chiamata indice.

In Python, una lista può essere creata come segue: `lista = [1, 2, 3, 4]`

Gli indici delle liste iniziano sempre da 0. Quindi, il primo elemento della lista è accessibile tramite `lista[0]`.

Le liste possono contenere elementi di qualsiasi tipo, anche altre liste, rendendole estremamente versatili per la gestione dei dati.

Un aspetto fondamentale da capire è che le liste sono mutabili. Questo significa che gli elementi possono essere modificati dopo la creazione della lista.

Aggiungere un elemento a una lista in Python può essere fatto con il metodo `append`: `lista.append(5)`

## Operazioni con le Liste

Le liste supportano diverse operazioni che consentono di manipolare i dati che contengono. Alcune delle operazioni più comuni sono:

- l'aggiunta
- la rimozione
- l'accesso

agli elementi.

Per rimuovere un elemento da una lista in Python, si può usare il metodo `remove`: `lista.remove(3)`

Se si prova a rimuovere un elemento non presente nella lista, Python restituirà un errore.

## Count

È fondamentale sapere la lunghezza di una lista. In Python, si può usare la funzione integrata `len()`. In C# si usa la funzione `Count()`. Il concetto però rimane immutato e ci restituisce la lunghezza di una lista.

## Slice

È anche possibile accedere a un sottoinsieme di una lista utilizzando il slice. Questo consente di ottenere una nuova lista contenente solo una parte degli elementi della lista originale.

Per ottenere i primi tre elementi di una lista, si può usare lo slice `lista[:3]`.

Lo slice `lista[a:b]` restituisce gli elementi dalla posizione 'a' fino a 'b-1'. Se 'a' è omissso, lo slice inizia dall'inizio della lista. Se 'b' è omissso, lo slice va fino alla fine della lista.

## Iterare sulle Liste

Uno degli aspetti più potenti delle liste è la possibilità di iterare su ogni elemento. Ciò significa eseguire un'azione su ogni elemento della lista, uno alla volta.

In Python, è possibile iterare su una lista usando un ciclo `for`: `for elemento in lista: print(elemento)`

Durante l'iterazione, la variabile 'elemento' rappresenta ogni volta l'elemento corrente della lista.

Iterare su una lista è particolarmente utile quando si vuole eseguire operazioni su ogni elemento, come il calcolo di una somma o l'applicazione di una funzione.

È anche possibile iterare su una lista e ottenere sia l'elemento sia il suo indice usando la funzione `enumerate`.

Per ottenere indice ed elemento si può usare: `for indice, elemento in enumerate(lista): print(indice, elemento)`

## Applicazioni delle Liste

Le liste trovano applicazioni in quasi tutti i campi della programmazione. Dall'archiviazione di dati fino alla gestione di strutture dati più complesse, le liste sono uno strumento indispensabile.

Un esempio pratico potrebbe essere la gestione di una lista di utenti in un'applicazione web, dove ogni utente è rappresentato da una lista di attributi (nome, età, email, ecc.).

Le liste sono anche fondamentali in algoritmi e strutture dati avanzate, come pile, code e alberi, dove servono per rappresentare e manipolare i dati in modi specifici.

La comprensione profonda delle liste e delle operazioni che è possibile eseguire su di esse è fondamentale per qualsiasi programmatore.

In conclusione, le liste sono uno strumento versatile e potente che permette di gestire e manipolare i dati in maniera efficace e intuitiva.

## Liste e Memoria

Capire come le liste sono memorizzate e gestite in memoria può essere un concetto astratto. Per semplificarlo, immagina che una lista sia come un libro, e ogni elemento della lista sia come una pagina di questo libro.

Ogni pagina in un libro ha un numero unico, proprio come ogni elemento in una lista ha un indice unico. Se vuoi trovare una specifica informazione, puoi andare direttamente alla pagina (indice) corretta senza dover sfogliare tutto il libro (lista).

Ogni elemento in una lista è accessibile tramite il suo indice, proprio come ogni pagina in un libro è accessibile tramite il suo numero di pagina.

Quando aggiungi elementi a una lista, è come se stessi aggiungendo nuove pagine al tuo libro. Se il libro (lista) è pieno, puoi legare insieme più libri o pagine, proprio come alcune liste si ridimensionano automaticamente per accogliere più elementi.

Rimuovere un elemento da una lista è come strappare una pagina da un libro. La pagina (elemento) viene rimossa e lo spazio che occupava può essere utilizzato per nuove pagine (elementi). A differenza di un libro fisico, però, le liste possono automaticamente ridurre lo spazio non utilizzato, a

seconda dell'implementazione della lista nel linguaggio di programmazione.

Se stai cercando una specifica informazione, sai che si trova a una certa pagina. Vai direttamente a quella pagina usando il numero di pagina, proprio come accederesti a un elemento in una lista usando il suo indice.

L'accesso diretto a una pagina in un libro o a un elemento in una lista tramite l'indice è molto efficiente perché non richiede di esaminare tutti gli elementi uno per uno.

Se vuoi organizzare le informazioni in un libro in un ordine specifico, dovresti spostare le pagine in modo che seguano l'ordine desiderato. Questo processo è simile all'ordinamento di una lista: il sistema deve passare attraverso ogni elemento e posizionarlo nella posizione corretta all'interno della lista ordinata.

Proprio come un libro organizza le informazioni in pagine numerate, una lista organizza i suoi elementi in posizioni indicizzate. Comprendere questa analogia ti aiuta a visualizzare come le liste funzionano a livello di memoria e come sono gestite le operazioni su di esse, rendendo più intuitiva la comprensione delle strutture dati e della loro gestione in memoria.

(CC BY-NC-SA 3.0) lezione - by tankerino.com

<https://www.tankerino.com>

---

Questa lezione e' stata realizzata grazie al contributo di:



Risorse per la scuola

<https://www.baobab.school>



Siti web a Varese

<https://www.francescobelloni.it>