**LE LISTE**

# 

[INTRODUZIONE](#_heading=h.tyjcwt)

[COSA FA IL CODICE? [5 minuti]](#_heading=h.3dy6vkm)

[COSA FA IL CODICE? [5 minuti]](#_heading=h.17dp8vu)

[COSA FA IL CODICE? [5 minuti]](#_heading=h.lnxbz9)

[DOMANDE [5 minuti]](#_heading=h.z337ya)

[TERMINI CHIAVE DELLA LEZIONE](#_heading=h.1y810tw)

[DA RICORDARE IN PYTHON](#_heading=h.1ci93xb)

[ESERCIZI](#_heading=h.2bn6wsx)

# 

# 

# INTRODUZIONE

In questa lezione vedremo come gestire un insieme di dati mediante una struttura dati chiamata ***lista*** che ci permette, con un unico nome, di indicare una sequenza di dati. I dati all’interno della lista vengono identificati con un ***indice*** che rappresenta la posizione in cui si trovano. L’elemento in prima posizione ha indice 0.

Per indicare che i dati sono contenuti in una lista si utilizzano le parentesi quadre.

*Esempio:*  
anni = [2020, 2021, 2022, 2023]

| 0 | 1 | 2 | 3 |
| --- | --- | --- | --- |
| 2020 | 2021 | 2022 | 2023 |
| -4 | -3 | -2 | -1 |

indice:

**elemento:**

indice negativo:

L’elemento 2020 si trova nella posizione di indice 0

→ per richiamarlo scriviamo anni[0]

Analogamente il 2023 si trova nella posizione di indice 3

→ per richiamarlo scriviamo anni[3]

E’ possibile usare anche gli indici negativi indicizzando la lista a partire dal fondo, per esempio possiamo indicare il 2023 come elemento in posizione -1

→ per richiamarlo scriviamo anni[-1]

In questa lezione incontrerete alcune funzioni per gestire le liste. Al solito, per immaginare il loro scopo, potete rifarvi alla traduzione della parola inglese.

Per esempio:

* append() significa *accodare*
* len() sta per *length* che significa *lunghezza*

# COSA FA IL CODICE? [5 minuti]

**Lavoro in coppia**

Osservate il seguente codice:

studenti = ["Arianna","Bruno","Dario","Elena","Mario","Teresa"]

for nome in studenti:

**print**(nome)

**print**(studenti)

**print**(studenti[0])

**print**(studenti[1])

**print**(studenti[1:4])

| Secondo voi, cosa viene visualizzato dall’istruzione **print**(nome)? |
| --- |
| Stampa i nomi della lista studenti |
| Secondo voi, cosa viene visualizzato dall’istruzione **print**(studenti) ? |
| Stampa l’intera lista studenti in una sola riga |
| Secondo voi, cosa viene visualizzato dalle istruzioni **print**(studenti[0]) e **print**(studenti[1])? |
| Stampa il primo e secondo nome quindi Arianna e poi Bruno |
| Secondo voi, cosa viene visualizzato dall’istruzione **print**(studenti[1:4])? |
| Stampa i nomi della lista studenti dal secondo al quarto |
| Copiate e incollate il codice in VSCode (salvate con nome *129\_Liste.py)* e controllate cosa fa. Fa quello che vi aspettavate? Provate a dare una spiegazione in caso contrario. |
| Si il codice ha fatto esattamente quello che mi aspettavo |

# 

Osservate ora le seguenti istruzioni:

studenti[0]="Giovanni"

studenti.append("Giorgia")

**print**(studenti)

| Cosa verrà visualizzato dall’istruzione **print**(studenti)? |
| --- |
| Il codice stamperà l’intera lista sostituendo il nome Giovanni con Arianna e aggiungendo il nome Giorgia |
| Ora copiate e incollate il codice in VSCode (in fondo al file *129\_Liste.py*) e controllate cosa fa. Fa quello che vi aspettavate? Provate a dare una spiegazione in caso contrario. |
| Si, fa quello che mi aspettavo |

# 

Osservate ora le seguenti istruzioni:

studenti[1]=2

**print**(studenti)

| Cosa verrà visualizzato dall’istruzione **print**(studenti)? |
| --- |
| Sostituirà il nome 1 con il nome 2 |
| Ora copiate e incollate il codice in VSCode (in fondo al file *129\_Liste.py*) e controllate cosa fa. Fa quello che vi aspettavate? Provate a dare una spiegazione in caso contrario. |
| No, il codice non ha fatto quello che mi aspettavo, semplicemente ha sostituito il secondo nome con il numero “2” |

# COSA FA IL CODICE? [5 minuti]

**Lavoro in coppia**

Osservate il seguente codice:

studenti = ["Arianna","Bruno","Dario","Elena","Mario","Teresa"]

for i in range(**len**(studenti)):

**print**(i, studenti[i])

| Secondo voi, cosa fa il codice? |
| --- |
| Fa un ciclo contando quanti nomi ci sono all’interno della lista e la stampa |
| Copiate e incollate il codice in VSCode (salvate con nome *130\_Len.py)* e controllate cosa fa. Fa quello che vi aspettavate? Provate a dare una spiegazione in caso contrario. |
| Si, il codice ha eseguito come immaginavo io |

# 

# COSA FA IL CODICE? [5 minuti]

**Lavoro in coppia**

Osservate il seguente codice:

# 

studenti = ["Arianna","Bruno","Dario","Elena","Mario","Teresa"]

studente\_cercato=**input**("Scrivi il nome dello studente: ")

if studente\_cercato in studenti:

**print**(studente\_cercato, "è uno studente di questa scuola")

else:

**print**(studente\_cercato,"non è un nostro studente")

| Secondo voi, cosa fa il codice? |
| --- |
| Tramite l’input dell’utente il codice cerca se lo studente fa parte della lista studenti o no |
| Copiate e incollate il codice in VSCode (salvate con nome *131\_In.py)* e controllate cosa fa. Fa quello che vi aspettavate? Provate a dare una spiegazione in caso contrario. |
| Si, il codice ha fatto quello che mi aspettavo |
| Provate a sostituire in con l’istruzione not in. Cosa succede di diverso? |
| Il codice ora stampa solo se il nome non è nella lista |

# 

# DOMANDE [5 minuti]

**Lavoro in coppia**

Rispondete alle seguenti domande:

1. Le liste possono cambiare il loro contenuto?

Si, il contenuto può essere rimosso,sostituito, cambiato e aggiunto

1. Le liste possono contenere elementi di tipo diverso (es stringhe insieme a numeri)?

Si e possono essere modificati entrambi

1. Che differenza c’è tra i due cicli che abbiamo visto per lo scorrimento delle liste:

for x in studenti e for x in range(len(studenti))?

For x in studenti da come output una lista dei nomi mentre

usando For x in range(len(studenti)) si avrà come output la lista dei nomi e in aggiunta la lunghezza di essa.

# TERMINI CHIAVE DELLA LEZIONE

| **TERMINE** | **SIGNIFICATO** | **ESEMPIO IN PYTHON** |
| --- | --- | --- |
| lista | sequenza di dati che sono salvati in un unico contenitore | lista = [3,2,5] |
| lista.**append**(elemento) | Aggiunge elemento alla fine della lista | lista = [3,2,5]  lista.append(8)  print(lista)  → [3,2,5,8] |
| lista**[i]** | Identifica l’elemento nella posizione i della lista | lista = [3,2,5,8]  print(lista[2])  → 5 |
| lista**[i:j]** | Accede a una parte di lista che va dall’elemento di indice i all’elemento di indice j (escluso).  Questa operazione di chiama **slicing.** | lista=[3,2,5,8,10]  print(lista[1:3])  → [2,5] |
| lista**[-i]** | Identifica l’elemento in posizione i partendo dalla fine | lista=[3,2,5,8,10]  print(lista[-2])  → 8 |
| lista.**insert**(n,elemento) | Inserisce elemento nella posizione n della lista | lista = [3,2,7,8]  lista.insert(1,10)  print(lista)  → [3,10,2,7,8] |
| lista.**remove**(elemento) | Rimuove elemento da lista. Se c’è più di un elemento uguale a quello indicato, elimina il primo che trova (quello nella posizione più vicina allo 0). | lista = [3,10,2,7]  lista.remove(2)  print(lista)  → [3,10,7] |
| **del**(lista[i)) | Rimuove l’elemento di posizione i nella lista | lista = [3,10,7,8]  del(lista[3])  print(lista)  → [3,10,7] |
| **len**(lista) | Restituisce il numero di elementi della lista | lista = [3,2,5]  print(len(lista))  → 3 |

# 

# 

# DA RICORDARE IN PYTHON

Le liste sono strutture di dati che consentono di **assegnare un gruppo di valori a un identificatore**. Questo evita l'uso di più variabili per memorizzare un gruppo di valori.

Le liste possono **crescere e ridursi** di dimensioni e memorizzare gruppi di valori di **tipi di dati misti.**

Gli elementi delle liste possono essere modificati: le **liste** sono infatti strutture dati **mutabili**.

Una lista si rappresenta elencando gli elementi che la compongono in sequenza, separati da una virgola. La sequenza è racchiusa in una coppia di parentesi quadre.

*Esempi:*

dati = [1, 2, "mele", "fragole"]

frutta = ["mele", "pere", "arance"]

anni = [2020, 2021, 2022, 2023]

Come usare le strutture che conosciamo per manipolare le liste:

| **for** elemento **in** lista: | scorre gli elementi della lista, si usa quando il valore dell'indice dell’elemento non è necessario |
| --- | --- |
| **for** i **in range**(len(studenti)): | scorre gli elementi della lista, si usa quando serve conoscere l’indice (posizione) dell’elemento della lista |
| **if** elemento **in** lista: | verifica se un elemento è nella lista |
| **if** elemento **not in** lista: | verifica se un elemento NON è nella lista |

# 

# ESERCIZI

**Lavoro in autonomia**

1. [*132\_ManipolaListe.py*] Crea una la lista (chiamala *studenti*) e inserisci i nomi Arianna, Bruno, Dario, Elena, Mario, Teresa. Poi scrivi le istruzioni per eseguire le seguenti operazioni:
   * aggiungi in fondo alla lista il nome tuo e del tuo vicino di banco
   * inserisci in posizione di indice 3 il nome di un altro tuo compagno di classe
   * rimuovi lo studente in seconda posizione
   * cancella la studentessa “Teresa”
   * visualizza posizione e nome di ogni studente della lista
2. [*133\_Studenti.py*] Crea una lista (chiamala *studenti*) che sia inizialmente vuote e una seconda lista con i nomi di 5 tuoi compagni di classe (chiamala *studenti\_classe*). Chiedi poi all’utente di inserire 5 nomi (con un ciclo for). Verifica se ogni nome inserito è presente in *studenti\_classe*. In caso affermativo aggiungilo in fondo alla lista studenti, altrimenti stampa “questo studente non va inserito nella classe”. Prima di inserire lo studente nella lista *studenti*, verifica anche che non sia già stato inserito precedentemente.
3. [*134\_ListaCas.py*] Crea una lista di 10 numeri generati casualmente (scegli tu gli estremi dell’intervallo in cui generi i numeri e il nome della lista). Visualizza gli elementi dalla prima all'ultima posizione e poi dall'ultima posizione alla prima.
4. [*135\_ListaQuadrati.py*] L’utente inserisce un numero N, il programma genera N numeri casuali (tra 1 e 20) e li salva in una lista. Poi, il programma visualizza tutti i numeri generati e poi tutti i numeri elevati alla seconda.
5. [*136\_ListaRaddoppia.py*] Crea una lista di 10 numeri interi casuali in un intervallo a tua scelta e raddoppia i valori di quelli dispari.
6. [*137\_SommaLista.py*] L’utente inserisce un numero N, il programma genera N numeri casuali (tra 1 e 20) e li salva in una lista. Poi, il programma calcola la somma dei numeri contenuti nella lista. *Suggerimento: puoi partire dall’esercizio 114\_SommaN.py e inserire al suo interno la gestione delle liste e la generazione casuale.*
7. [*138\_PosNeg.py*] L’utente inserisce un numero N, il programma genera N numeri casuali (tra -20 e 20) e li salva in una lista. Poi, il programma calcola separatamente la somma dei numeri positivi e quelli negativi della lista. *Suggerimento: puoi partire dall’esercizio 115\_SommaPositiviNegativi.py e inserire al suo interno la gestione delle liste e la generazione casuale.*
8. [*139\_PariDispari.py*] Chiedi in input all’utente quanti numeri (N) vuole nella lista, poi genera casualmente N numeri casuali e inseriscili in una lista. Stabilisci quanti elementi sono pari e quanti dispari nella lista. Visualizza la lista generata e il numero di pari e di dispari. *Suggerimento puoi partire dall’esercizio 126\_PariDispari.py e inserire il codice per gestire le liste e la generazione casuale.*
9. [*140\_istogramma.py*] Crea una lista di numeri e stampa un istogramma basato su questi numeri, usando asterischi per disegnarlo. (Scegli tu come creare la lista). Ad esempio, data la lista [3,7,9] il programma deve produrre questo grafico:

\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

1. [*141\_MaxLista.py*] Generare casualmente una lista di 20 numeri tra 1 e 100. Determinare il numero massimo della lista. Visualizza la lista generata e il valore massimo trovato. *Suggerimento: puoi partire dall’esercizio 124\_MassimoNumero.py e inserire al suo interno la gestione delle liste e la generazione casuale.*
2. [*142\_Prodotto.py]* Crea una lista di 10 numeri interi random tra 1 e 10 e calcola il prodotto dei numeri.
3. [*143\_SopraMedia.py*] Crea una lista di 10 numeri interi random tra 1 e 10 e conta i componenti che hanno un valore superiore alla media (calcolata su tutta la lista).