**DATA STRUCTURES**

**Le strutture dati presenti nella versione lite di Python sono le seguenti:**

* **Stringhe**
* **Liste**
* **Tuple**
* **Sets**
* **Dizionari**

**Tutte le strutture dati hanno dei metodi che servono a ricevere informazioni da esse o manipolarle. Ci sono tre tipi di metodi.**

1. **Metodi che alterano la struttura dati e non restituiscono un valore di ritorno**
2. **Metodi che non alterano la struttura dati ma restituiscono un valore di ritorno**
3. **Metodi che alterano la struttura dati e restituiscono un valore di ritorno.**

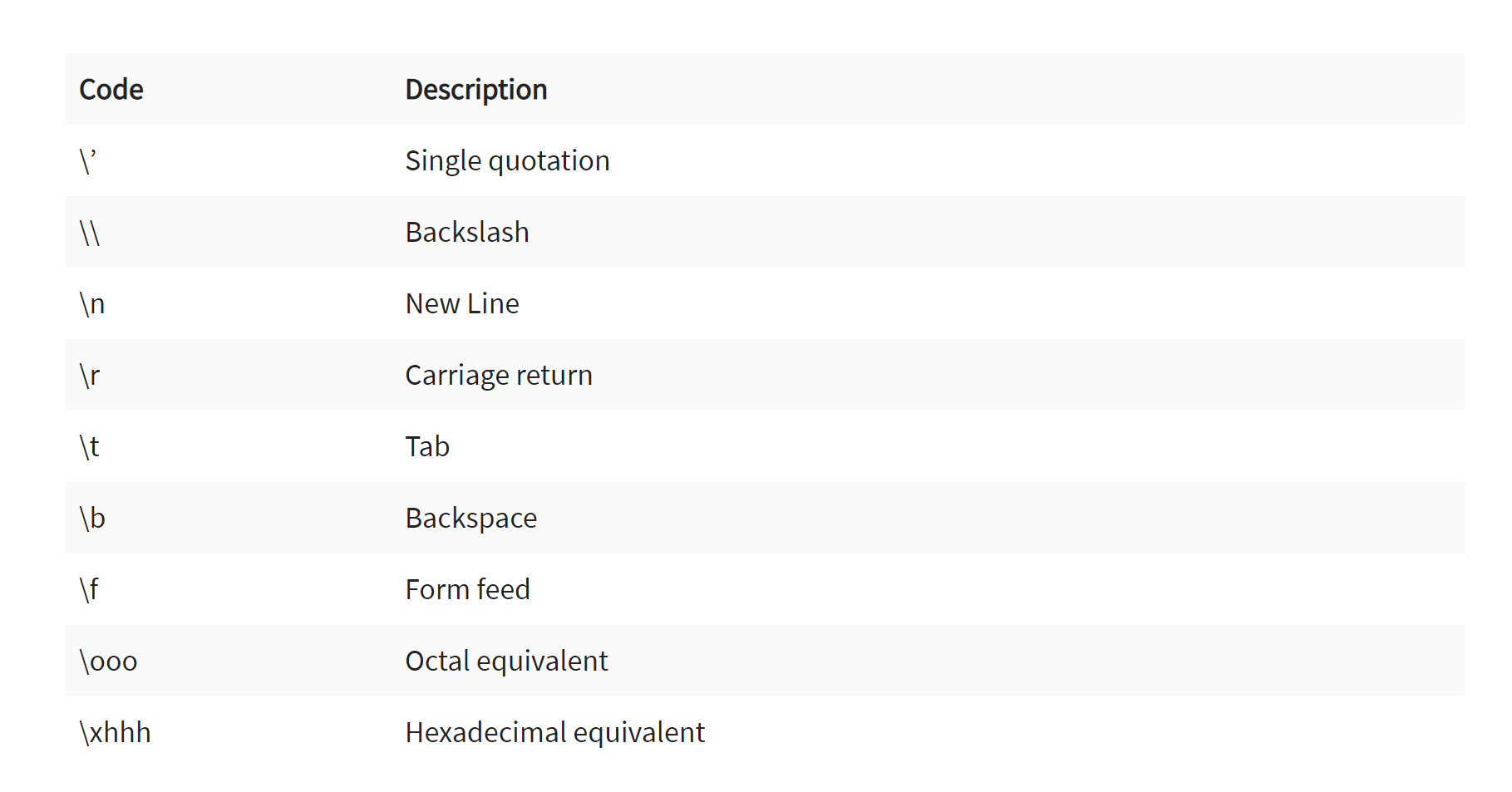
**Per vedere una struttura dati quali metodi accetta, si può utilizzare su di essa la funzione dir.**

**Vediamo quindi, una per una, le strutture dati presenti in Python. Le strutture dati possono essere mutabili o immutabili, iterabili o non iterabili.**

**STRIGHE**

**Le strighe sono strutture dati IMMUTABILI. Per capire bene come scriverle, occorre ricordare quali sono i caratteri di Escape:**

**Caratteri di Escape**

****

**Le stringhe possono essere definite in 4 modi diversi. Vediamo un esempio direttamente in Python.**

**Codice Python:**

# STRINGHE

stringa1 = 'Ciao a tutti! Come state?'

print(stringa1)

print()

stringa2 = "Ciao a tutti! Come state?"

print(stringa2)

print()

stringa3 = str('Ciao a tutti! Come state?')

print(stringa3)

print()

stringa4 = """Ciao a tutti! Come state?"""

print(stringa4)

**Console:**

**Immagine che contiene testo

Descrizione generata automaticamente**

**Perché si usano modi diversi per creare una stringa? Questo perché in base a come viene creata può non essere obbligatorio l’utilizzo dei caratteri di escape. Vediamo un paio di esempio.**

* **Utilizzo dell’apostrofo all’interno della nostra stringa.**

**Supponiamo di voler utilizzare un apostrofo all’interno della nostra stringa.**

**Partiamo dal primo metodo di creazione della stringa, ossia quello tra singoli apici:**

**Esempio stringa:**

stringa1 = 'Ciao a tutti! Come state?'

**Creiamo quindi una stringa contenente un apostrofo:**

**Codice Python:**

stringa1 = 'Ieri un'ape mi ha punto. Sono finito all'obitorio'

print(stringa1)

**Console:**

**Immagine che contiene testo

Descrizione generata automaticamente**

**Python ci restituisce errore. Come si può risolvere?**

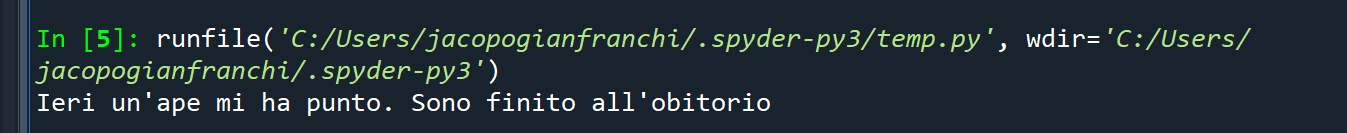
**Un primo modo è l’utilizzo del carattere di escape \:**

**Codice Python:**

stringa1 = 'Ieri un\'ape mi ha punto. Sono finito all\'obitorio'

print(stringa1)

**Console:**

****

**Un secondo modo è quello di inserire la stringa all’interno dei doppi o tripli apici:**

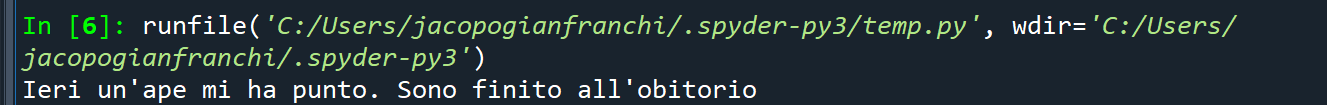
**Doppi apici**

**Codice Python:**

stringa1 = "Ieri un'ape mi ha punto. Sono finito all'obitorio"

print(stringa1)

**Console:**

****

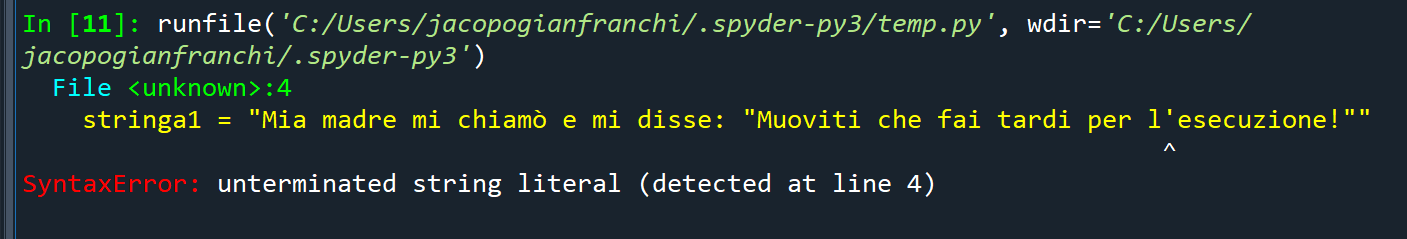
**Ovviamente se volessimo inserire le virgolette doppie all’interno di una stringa inserita tra doppi apici si sarebbe verificato lo stesso problema precedente:**

**Codice Python:**

stringa1 = "Mia madre mi chiamò e mi disse: "Muoviti che fai tardi per l'esecuzione!""

print(stringa1)

**Console:**

****

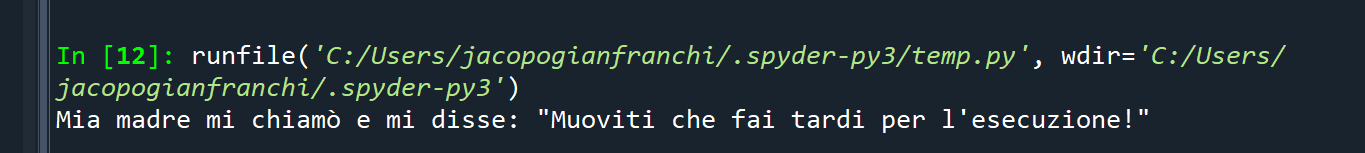
**Per risolvere utilizziamo il solito carattere di escape \ :**

**Codice Python:**

stringa1 = "Mia madre mi chiamò e mi disse: \"Muoviti che fai tardi per l\'esecuzione!\""

print(stringa1)

**Console:**

****

**Passiamo ora ai tripli apici**

**I tripli apici sono comodi anche per mandare il testo a capo.**

**Nel caso di apici singolo o doppi, infatti, per mandare a capo il testo è necessario l’utilizzo del carattere di escape \n.**

**Vediamo un esempio con gli apici doppi:**

**Codice Python:**

stringa1 = "Ieri un'ape mi ha punto.

Sono finito all'obitorio"

print(stringa1)

**Console:**

**Immagine che contiene testo

Descrizione generata automaticamente**

**Errore. Proviamo quindi con il carattere di escape \n:**

**Codice Python:**

stringa1 = "Ieri un'ape mi ha punto.\nSono finito all'obitorio"

print(stringa1)

**Console:**

**Immagine che contiene testo

Descrizione generata automaticamente**

**Con i tre apici il problema si risolve molto più facilmente:**

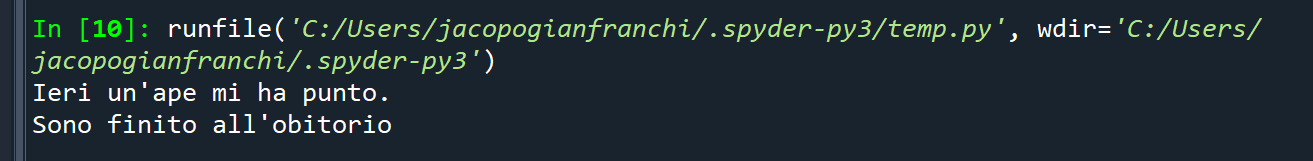
**Codice Python:**

stringa1 = """Ieri un'ape mi ha punto.

Sono finito all'obitorio"""

print(stringa1)

**Console:**

****

**Ok**

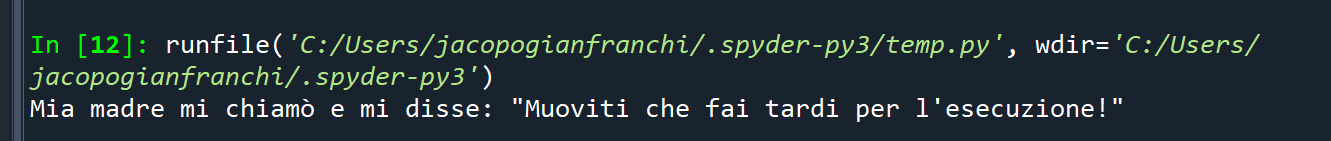
**Ovviamente le doppie virgolette possono essere inglobate nelle triple senza bisogno del carattere di escape:**

**Codice Python:**

stringa1 = """Mia madre mi chiamò e mi disse: "Muoviti che fai tardi per l'esecuzione!"""

print(stringa1)

**Console:**

****

**LISTE**

**Le liste sono dei “raccoglitori” di dati. Possiamo inserire al loro interno elementi di qualsiasi tipo, anche se solitamente il tipo degli elementi deve essere omogeneo.**

**Codice Python:**

lista = ["mamma","papà","nonna","nonno","fratello","sorella"]  
  
print(lista)

**Console:**

**Immagine che contiene testo

Descrizione generata automaticamente**

**Il numero di elementi di una lista si calcola con la funzione len:**

**Codice Python:**

lista = ["mamma","papà","nonna","nonno","fratello","sorella"]  
  
print(lista)  
  
print()  
  
lunghezza\_lista = len(lista)  
  
print("lunghezza\_lista:",lunghezza\_lista)

**Console:**

**Immagine che contiene testo

Descrizione generata automaticamente**

**La lista “lista” ha quindi 6 elementi.**

**Le posizioni degli elementi vanno da 0 a len(lista)-1**

**Codice Python:**

lista = ["mamma","papà","nonna","nonno","fratello","sorella"]  
  
print(lista)  
  
  
print()  
  
elemento\_1 = lista[0]  
elemento\_2 = lista[1]  
elemento\_4 = lista[3]  
ultimo\_elemento= lista[len(lista)-1]  
  
print(elemento\_1,elemento\_2,elemento\_4,ultimo\_elemento)  
  
print()

**Console:  
Immagine che contiene testo

Descrizione generata automaticamente**

**Possiamo anche utilizzare gli indici negativi per prendere gli elementi. L’ultimo elemento si trova in posizione -1, il penultimo in posizione -2, etc.**

**Codice Python:**

lista = ["mamma","papà","nonna","nonno","fratello","sorella"]  
  
print(lista)  
  
  
print()  
  
elemento\_1 = lista[-6]  
elemento\_2 = lista[-5]  
elemento\_5 = lista[-2]  
ultimo\_elemento= lista[-1]  
  
print(elemento\_1,elemento\_2,elemento\_5,ultimo\_elemento)  
  
print()

**Console:**

**Immagine che contiene testo

Descrizione generata automaticamente**

**Cosa succede se inserissimo una posizione rispetto alla quale non abbiamo elementi? Python restituirà errore:**

**Codice Python:**

lista = ["mamma","papà","nonna","nonno","fratello","sorella"]

print(lista[-23])

**Console:**

**Immagine che contiene testo

Descrizione generata automaticamente**

**SLICING**

**Tramite lo slicing è possibile estrarre un sottoinsieme da una lista. Anche qui si può ragionare con numeri positivi e numeri negativi. Supponiamo che, dalla nostra solita lista (lista = ["mamma","papà","nonna","nonno","fratello","sorella"]), si voglia estrarre la sottolista [“nonna”,”papa”,”fratello].**

**Codice Python:**

lista = ["mamma","papà","nonna","nonno","fratello","sorella"]  
  
# [ nonna, papà , fratello ]  
  
  
lista1 = lista[-5:-3]  
  
lista1.append(lista[-2])  
  
print(lista1)

**Console:**

**Immagine che contiene testo

Descrizione generata automaticamente**

**Possiamo scrivere lo slicing in modo da prendere tutti gli elementi da un certo elemento in poi. Ad esempio, supponiamo di voler creare una sottolista che possiede tutti gli elementi a partire dall’elemento nonna, ossia dal secondo elemento in poi:**

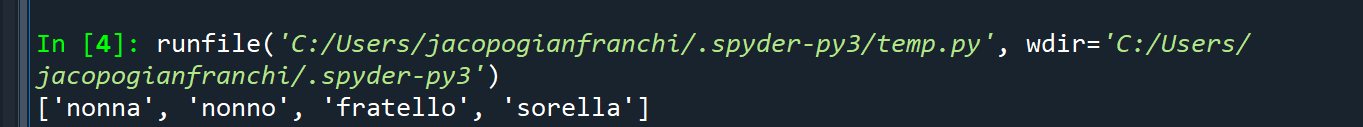
**Codice Python:**

lista = ["mamma","papà","nonna","nonno","fratello","sorella"]

sottolista = lista[2:]

print(sottolista)

**Console:**

****

**Supponiamo ora di voler creare una sottolista che contiene tutti gli elementi dal primo a “nonno”, ossia gli elementi dal primo al quarto. Ricordiamo che l’estremo destro non viene considerato. Quindi:**

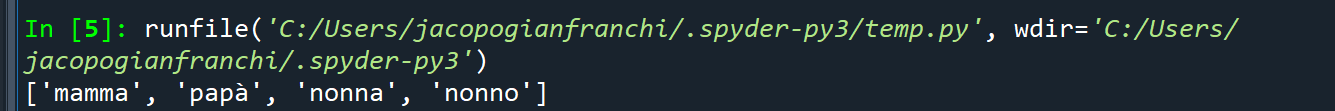
**Codice Python:**

lista = ["mamma","papà","nonna","nonno","fratello","sorella"]

sottolista = lista[:4]

print(sottolista)

**Console:**

****

**Cosa succede se inseriamo un estremo superiore maggiore della lunghezza della lista?**

**Per prima cosa vedremo quindi i metodi applicabili alle liste:**

**Codice Python:**

lista = ["mamma","papà","nonna","nonno","fratello","sorella"]

print(dir(lista))

**Console:**

**Immagine che contiene testo

Descrizione generata automaticamente**

**I metodi applicabili alle liste sono quindi: append, clear, copy,count,extend,index,insert,pop,revove,reverse,sort.**