**LE STRINGHE**

# 

[INTRODUZIONE](#_heading=h.tyjcwt)

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

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

[COSA FA IL CODICE? [5 minuti]](#_heading=h.1ksv4uv)

[COSA FA IL CODICE? [5 minuti]](#_heading=h.3j2qqm3)

[TERMINI CHIAVE DELLA LEZIONE](#_heading=h.2xcytpi)

[DOMANDE [5 minuti]](#_heading=h.1ci93xb)

[DA RICORDARE IN PYTHON](#_heading=h.2bn6wsx)

[ESERCIZI](#_heading=h.1pxezwc)

# 

# 

# INTRODUZIONE

Finora abbiamo imparato che in programmazione esistono diversi tipi di dato (int, float, bool,string). In particolare, sappiamo che per rappresentare **parole, caratteri alfabetici e punteggiatura** dobbiamo utilizzare le stringhe.

**Esempio**: messaggio = “Ciao come va? Io tutto bene.”

In questa lezione andremo ad approfondire le **stringhe** considerandole **come** una **sequenza di caratteri**.

Per definire una stringa si usano gli apici (‘Ciao’) o le virgolette (“Ciao”).

*Esempio:*  
nome = “Pippo”

| 0 | 1 | 2 | 3 | 4 |
| --- | --- | --- | --- | --- |
| P | i | p | p | o |
| -5 | -4 | -3 | -2 | -1 |

indice:

**carattere:**

indice negativo:

Ogni carattere della stringa è identificato da un indice, così come avviene per gli elementi delle liste.

Le stringhe non sono però semplici liste di caratteri, ma hanno metodi dedicati e sono **immutabili**, non è possibile cioè modificarne direttamente il contenuto.

# COSA FA IL CODICE? [5 minuti]

**Lavoro in coppia**

Osservate il seguente codice:

stringa = "Automobile"

**print**(stringa[0])

**print**(stringa[-1])

**print**(stringa[-2])

**print**(stringa[0:4])

**print**(stringa[4:8])

**print**(stringa[4:])

**print**(stringa[:4])

| Secondo voi, cosa viene visualizzato con le diverse funzioni **print**? |
| --- |
| Vengono stampate le lettere della stringa automobile in diversi ordini |
| Copiate e incollate il codice in VSCode (salvate con nome *144\_Stringhe.py)* e controllate cosa fa. Fa quello che vi aspettavate? Provate a dare una spiegazione in caso contrario. |
| Il codice ha stampato le lettere in certi modi che non mi aspettavo |

# 

Osservate ora le seguenti istruzioni:

stringa[0] = "a"

**print**(stringa)

| Cosa verrà visualizzato dall’istruzione **print**(stringa)? |
| --- |
| La lettera a ovvero l’indice 0 |
| Ora copiate e incollate il codice in VSCode (in fondo al file *144\_Stringhe.py*) e controllate cosa fa. Fa quello che vi aspettavate? Provate a dare una spiegazione in caso contrario. |
| Il codice risulta con un errore, essendo che le stringhe non possono essere modificate |

# 

# 

# COSA FA IL CODICE? [5 minuti]

**Lavoro in coppia**

Osservate il seguente codice:

stringa = "Automobile"

nuova\_stringa="a"+stringa[1:10]

**print**(stringa,nuova\_stringa)

| Secondo voi, cosa fa il codice? |
| --- |
| Stampa la stringa + la “nuova\_stringa” che è un insieme di “a” + dal primo al decimo indice della stringa normale |
| Copiate e incollate il codice in VSCode (salvate con nome *145\_ModificaStringa.py)* e controllate cosa fa. Fa quello che vi aspettavate? E’ possibile modificare una stringa? |
| Si, il codice ha fatto quello che mi aspettavo |

# 

# 

# COSA FA IL CODICE? [5 minuti]

**Lavoro in coppia**

Osservate il seguente codice:

stringa="Conteggio"

stringa\_finale=""

for s in stringa:

stringa\_finale=stringa\_finale+" "+s+" \*"

**print**(stringa\_finale)

| Secondo voi, cosa fa il codice? |
| --- |
| Crea un ciclo per nella stringa conteggio e poi lo stampa con degli spazi e asterischi tra ogni lettera |
| Copiate e incollate il codice in VSCode (salvate con nome *146\_Conteggio.py)* e controllate cosa fa. Fa quello che vi aspettavate? Provate a dare una spiegazione in caso contrario. |
| Il codice è stato eseguito come mi aspettavo eccetto per lo spazio iniziale |

# 

# 

# COSA FA IL CODICE? [5 minuti]

**Lavoro in coppia**

Osservate il seguente codice:

stringa1="Ciao, come stai"

stringa2=stringa1.**replace**(“Ciao”,“Buongiorno”)

**print**(stringa1)

**print**(stringa2)

| Secondo voi, cosa fa il codice? |
| --- |
| nella creazione della stringa 2 il codice sostituisce “Ciao” con “Buongiorno”, stampa la prima stringa con “Ciao” e poi la seconda stringa con “Buongiorno” |
| Copiate e incollate il codice in VSCode (salvate con nome *147\_Replace.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 |

# 

# TERMINI CHIAVE DELLA LEZIONE

| **TERMINE** | **SIGNIFICATO** | **ESEMPIO IN PYTHON** |
| --- | --- | --- |
| stringa**[i]** | Identifica l’elemento nella posizione i della stringa | stringa=”pippo”  print(stringa[1])  → i |
| **len**(stringa) | Restituisce la lunghezza della stringa | stringa=”pippo”  print(len(stringa))  → 5 |
| stringa1 **+** stringa2 | Concatenazione di stringhe | s1=”Ciao,”  s2=” come va?”  print(s1+s2)  → Ciao, come va? |
| stringa **\*** n | Concatena la stessa stringa n volte | s=”Ciao ”  print(s\*3)  → Ciao Ciao Ciao |
| stringa**[i:j:step]**  **(slicing)** | Estrazione di una sottostringa dall’indice i all’indice j escluso. Il passo, come nel range, indica di quanto muoversi nella stringa.  Se i o j non sono presenti, si intende da inizio a fine stringa. | s = “Buongiorno”  print(s[:4]  → “Buon”  print(s[4:])  → “giorno”  print(s[::2])  → “Bogon” |
| **for** c **in** stringa: | Iterazione su ogni carattere della stringa | for c in “Ciao”:  print(c)  → C  i  a  o |
| stringa **in** stringa2 | restituisce True se stringa2 contiene stringa | “i” in “Ciao”  → True  “z” in “Ciao”  → False |
| stringa.**split**(“c”) | Il metodo split divide una stringa in una lista di stringhe utilizzando il parametro come carattere separatore. | s = “Ciao, come va?”  s.split(“ “)  → [“Ciao,”, “come”, “va?”]  s.split(“,“)  → [“Ciao”, “ come va?”] |
| stringa.**replace**(s1,s2) | Ritorna una stringa uguale a stringa dove la sottostringa s1 viene sostituita con s2 | s=”Ciao Marco”  s.replace(“Ciao”,”Buongiorno”)  → “Buongiorno Marco” |

# DOMANDE [5 minuti]

**Lavoro in coppia**

Rispondete alle seguenti domande:

1. E’ possibile cambiare il contenuto di una stringa?

No, sono immutabili

1. Le stringhe possono contenere elementi di tipo diverso?

Si e si possono anche concatenare

1. Il metodo replace permette di cambiare direttamente una sottostringa con un’altra all’interno della stringa iniziale?

Si, ma bisogna creare una nuova stringa perchè non si può modificare l’originale

1. Se voglio cambiare il contenuto di una stringa, dato che non posso farlo direttamente, come posso agire?

Si può creare una nuova stringa e usare il .replace

# DA RICORDARE IN PYTHON

Come abbiamo visto le stringhe sono delle sequenze di caratteri. Un aspetto importante che differenzia le **stringhe** dalle liste è che le prime sono **IMMUTABILI** e le seconde MUTABILI. Questo significa che non si possono modificare gli elementi della stringa, cosa che invece è possibile con le liste.  
In uno degli esempi visti in precedenza, il programma ha dato un errore per questo motivo.

**Esempi:**

stringa = "ciao"  
stringa[0] = "c"  
**print**(stringa)

→ Output: TypeError: 'str' object does not support item assignment.

lista = ["Ciao", 3]  
lista[0] = "Buongiorno"  
**print**(lista)

→ Output: ["Buongiorno", 3]

# 

# ESERCIZI

**Lavoro in autonomia**

1. *[148\_Slicing1.py]* Chiedi in input una stringa all’utente e mostra a video i suoi elementi dal secondo al quarto compresi. Se la lunghezza della stringa è minore di 4 stampa “Numero caratteri insufficienti"”.

Esempio: se la stringa inserita è "Casetta", l'output sarà "ase".

1. [*149\_Codici.py*] Dati i seguenti codici: c1 = 'AFCDDR-CF-2020' e c2 = 'SEDTYR-XC-2019', controlla se i codici terminano con "2020"
2. [*150\_QuantiCaratteri.py*] Prendi in input una stringa e conta quante volte un carattere inserito dall’utente c’è nella stringa.
3. [*151\_Sottostringa.py*] Data la stringa s2=”Automobile” e una stringa s1 presa in input dall’utente, controlla se s2 contiene la stringa s1. In caso affermativo stampa la concatenazione di s1 e s2 separate da uno spazio, altrimenti stampa la concatenazione di s2 e s1 separate da uno spazio
4. [*152\_StringaToLista.py*] Data una stringa in input dall’utente contenente degli spazi che separano le diverse parole, crea una lista con le parole che la compongono.
5. [*153\_CreaStringa.py*] Creare una stringa che abbia:
   * come primo carattere \*
   * poi una successione di 5 spazi (usando il costrutto for)
   * come ultimo carattere \*
6. [*154\_UnisciStringhe.py*] Creare una lista contenente alcune parole e costruire la stringa che contiene tutti gli elementi di lista separati da uno spazio. Ad esempio se la lista è costituita dalle parole "ciao", "come", "stai" la stringa finale sarà costituita da "ciao come stai"
7. [*155\_PrimaParte.py*] Visualizzare tutti i caratteri della stringa **PRIMA** della posizione indicata dal numero inserito dall'utente (posizione esclusa).
8. [*156\_GeneraPassword.py*] Chiedere all'utente il proprio nome e generare una password formata dal nome nel quale sia inserito a metà un numero (casuale) di 2 cifre.
9. [*157\_SostituisciCaratteri.py*] L'utente inserisce una stringa. Costruire una seconda stringa nella quale tutte le vocali *a* siano sostituite da un asterisco. Ad esempio, inserendo *Maria* verrà visualizzato M\*ri\*
10. [*158\_ConcatenaStringaInvertita.py*] Visualizzare una stringa finale composta dalla stringa iniziale seguita dalla stessa stringa invertita. Ad esempio, se l'utente inserisce *sedia*, il programma visualizza *sediaaides*
11. [*159\_InserisciPunti.py*] Visualizzare la stringa separando i caratteri con un numero progressivo di punti, ovvero inserendo *sedia* si visualizza *s.e..d...i....a*
12. [*160\_Slicing2.py*] Utilizzando lo slicing, creare una nuova stringa contenente solo il primo e l'ultimo elemento di una stringa inserita dall’utente.
13. [*161\_Slicing3.py*] Utilizzando lo slicing, creare una nuova stringa contenente tutti gli elementi di una stringa inserita dall’utente tranne il primo e l'ultimo.
14. *[162\_PariDispari.py*] Data una stringa inserita dall’utente creare due stringhe una con i soli caratteri in posizione pari e una con i caratteri in posizione dispari.
15. [*163\_SostituisciVocali.py*] L'utente inserisce una stringa. Costruire una seconda stringa nella quale tutte le vocali sono sostituite da un asterisco. Ad esempio, inserendo *Maria* verrà visualizzato M\*r\*\*
16. [*164\_SostituisciNome.py*] Data la seguente frase: *“Marco ieri è andato in piscina mentre Marco domani andrà in montagna.”*, scrivi un programma che stampi la stessa frase sostituendo a Marco un nome inserito dall’utente.