

Programmazione con Python

Input e output

Corso: Tecnologie informatiche

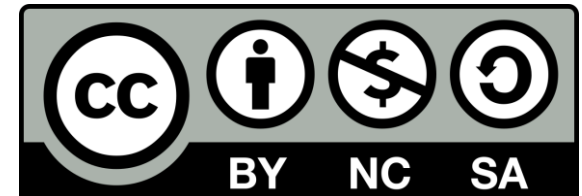
Docente: Leonardo Essam Dei Rossi

Licenze e crediti

Questo materiale è disponibile sul sito Web del docente per il corso di [Tecnologie informatiche](#) per le studentesse e gli studenti dell'anno scolastico 2025/2026.

Versione: 1.0.0 (A)
Ultima modifica: 11/05/2026 09:41

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#)



Sezione #1

Gestione dell'input

Definizione formale

Definizione: *input*

Un programma può essere rappresentato dalla funzione $f : I \rightarrow O$ dove I rappresenta il dominio di tutti i possibili dati di ingresso (*lo spazio degli input*) e O rappresenta il codominio dei possibili risultati (*output*). Un input è quindi un singolo elemento $x \in I$ usato come input di f tale che $f(x) \in O$.

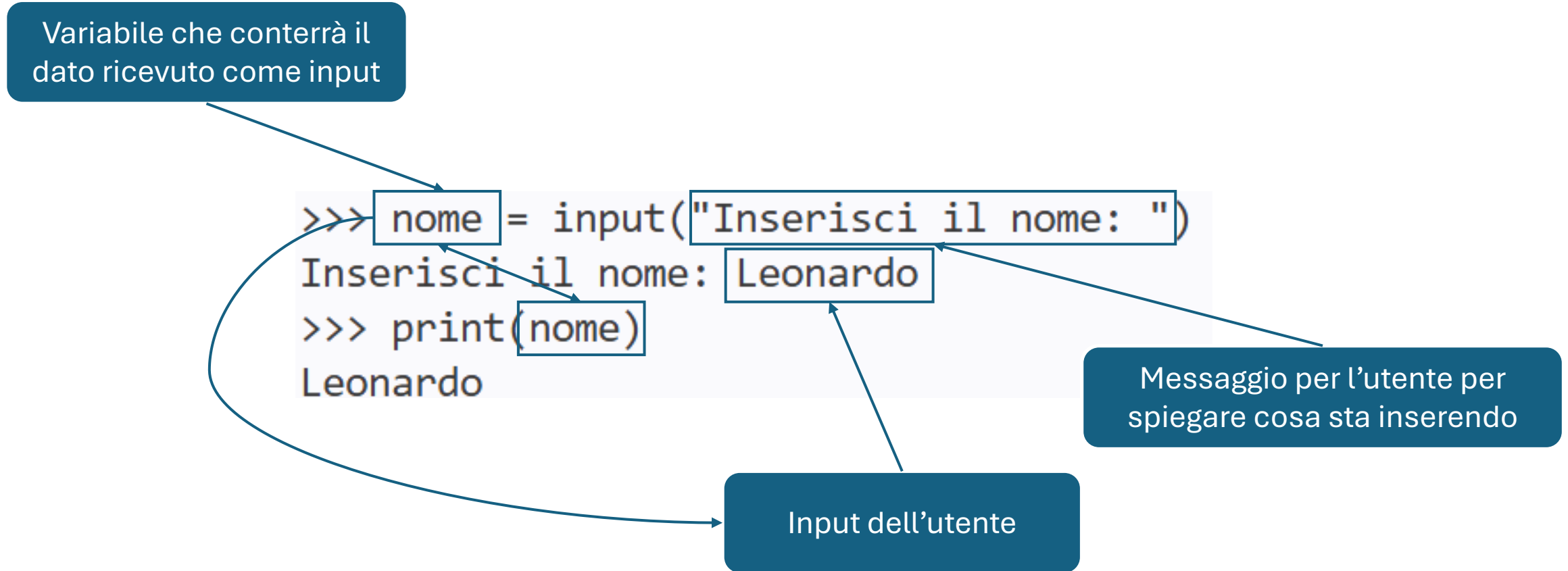
- L'input può essere di tre tipologie:
 - **Umano:** l'utente che interagisce tramite le periferiche del computer;
 - **Hardware:** sensori, componenti del computer o memorie esterne;
 - **Software:** un altro programma che comunica con il programma corrente.

Definizione formale

- Affinché un input possa essere elaborato da un programma, la sua definizione è spesso vincolata da delle regole che ne stabiliscono le caratteristiche:
 - **Tipizzazione:** l'input deve appartenere alla classe o al tipo di dato previsto dal programma;
 - **Precondizioni (dominio di validità):** l'input deve rispettare specifici limiti semantici o matematici affinché il programma garantisca un comportamento corretto;
 - **Sintassi/formato:** se l'input è complesso, deve rispettare regole sintattiche o di codifica rigorose affinché possa essere correttamente analizzato (*parsing*) dal programma.

Input dall'utente

- Per ricevere un valore dall'utente (input) si usa l'omonima funzione:



Esempio

Esempio: programma che legge due numeri (v. 1)

Si vuole realizzare un programma che legge come due numeri a e b e stampa la loro somma.

Svolgimento:

```
a = input("Inserisci a: ")      # Leggo il valore di a (1)
b = input("Inserisci b: ")      # Leggo il valore di b (2)
c = a + b                       # Faccio la somma
print(c)                        # Stampo il risultato
```

Esempio

Esempio: programma che legge due numeri (v. 1)

Si vuole realizzare un programma che legge come due numeri a e b e stampa la loro somma.

Svolgimento:

```
a = input("Inserisci a: ")      # Leggo il valore di a (1)
b = input("Inserisci b: ")      # Leggo il valore di b (2)
c = a + b                       # Faccio la somma
print(c)                        # Stampo il risultato
```

Output:

```
12                               # ??????
```

Q: *Cos'è successo?*

Esempio

- La funzione `input()` restituisce il dato inserito dall'utente;
- Questo dato ha come tipo una stringa!

Q: *Come risolviamo?*

Esempio

Esempio: programma che legge due numeri (v. 2)

Si vuole realizzare un programma che legge come due numeri a e b e stampa la loro somma.

Svolgimento:

```
a = int(input("Inserisci a: "))           # Leggo il valore di a (1)
b = int(input("Inserisci b: "))         # Leggo il valore di b (2)
c = a + b                               # Faccio la somma
print(c)                                # Stampo il risultato
```

Output:

```
3                                     # :)
```

Casting

- Python di natura è un linguaggio **debolmente tipizzato**:
 - Il tipo di dato di una variabile può cambiare a *runtime*;
 - Il tipo di dato di una variabile viene determinato automaticamente:
 - Alla dichiarazione della variabile;
 - Ad ogni successiva ri-assegnazione.
- Per «forzare» il tipo di dato (*vedasi esempio di prima*), esistono delle funzioni per fare il *cast* al tipo di dato desiderato.

Casting

Funzione	Commento
<code>int(...)</code>	Costruisce un numero intero a partire da una stringa di un numero intero (es. "3"), una stringa di un numero in virgola mobile (es. "3.14", rimuovendo tutte le cifre decimali). Se la stringa non è un numero succede <code>ValueError</code> .
<code>float(...)</code>	Uguale a <code>int(...)</code> , con l'aggiunta che vengono mantenute le cifre decimali.
<code>str(...)</code>	Costruisce una stringa dato un qualsiasi tipo di dato(intero, decimale, ...).

Sezione #2

Gestione dell'output

Definizione formale

Definizione: *output*

Un programma può essere rappresentato dalla funzione $f : I \rightarrow O$ dove O è lo spazio degli output (*codominio*) e un singolo output è definito come l'elemento $y \in O$ tale che $y = f(x)$. L'output è quindi il risultato della specifica trasformazione matematica e logica applicata ai dati in ingresso (*input*).

- L'ambiente esterno che riceve l'output può essere:
 - **Umano:** informazioni presentate tramite interfacce utente, come testo su uno schermo, elementi grafici, suoni o feedback aptico;
 - **Hardware:** segnali elettrici inviati a periferiche o attuatori;
 - **Software:** dati strutturati passati ad altri sistemi sotto forma di valori di ritorno, codici di stato, payload, file di log o pacchetti su socket di rete.

Definizione formale

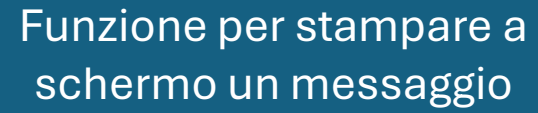
- Come l'input, anche l'output generato da un programma è vincolato da delle regole e da precise proprietà logiche:
 - **Tipizzazione (type):** L'output deve rispettare rigorosamente il tipo di dato dichiarato nella firma del programma o della funzione;
 - **Postcondizioni:** sono i vincoli logico-matematici che l'output deve *assolutamente* soddisfare affinché il programma possa dirsi corretto;
 - **Effetti collaterali (side effects):** in molti paradigmi di programmazione, l'output di un programma non è composto solo dal suo valore di ritorno esplicito, ma include anche le alterazioni di stato osservabili nel sistema. Nella programmazione funzionale pura, invece, una funzione non ha effetti collaterali, e il suo output formale coincide esclusivamente con il valore restituito.

Output verso l'utente

```
>>> print("Ciao, Mondo!")  
Ciao, Mondo!
```

Output verso l'utente

Funzione per stampare a schermo un messaggio



```
>>> print("Ciao, Mondo!")  
Ciao, Mondo!
```

Output verso l'utente

Funzione per stampare a schermo un messaggio

```
>>> print("Ciao, Mondo!")  
Ciao, Mondo!
```

Output stampato a schermo

Output verso l'utente

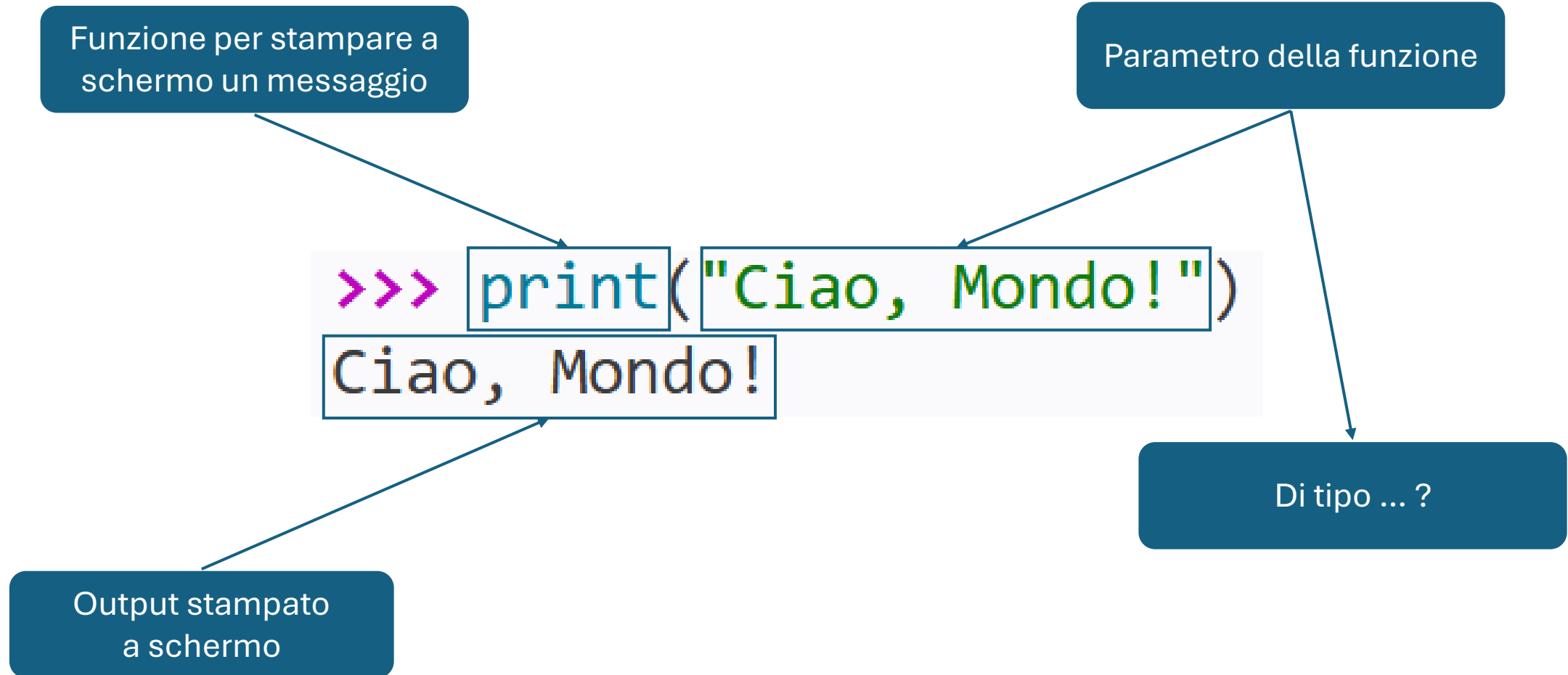
Funzione per stampare a schermo un messaggio

Parametro della funzione

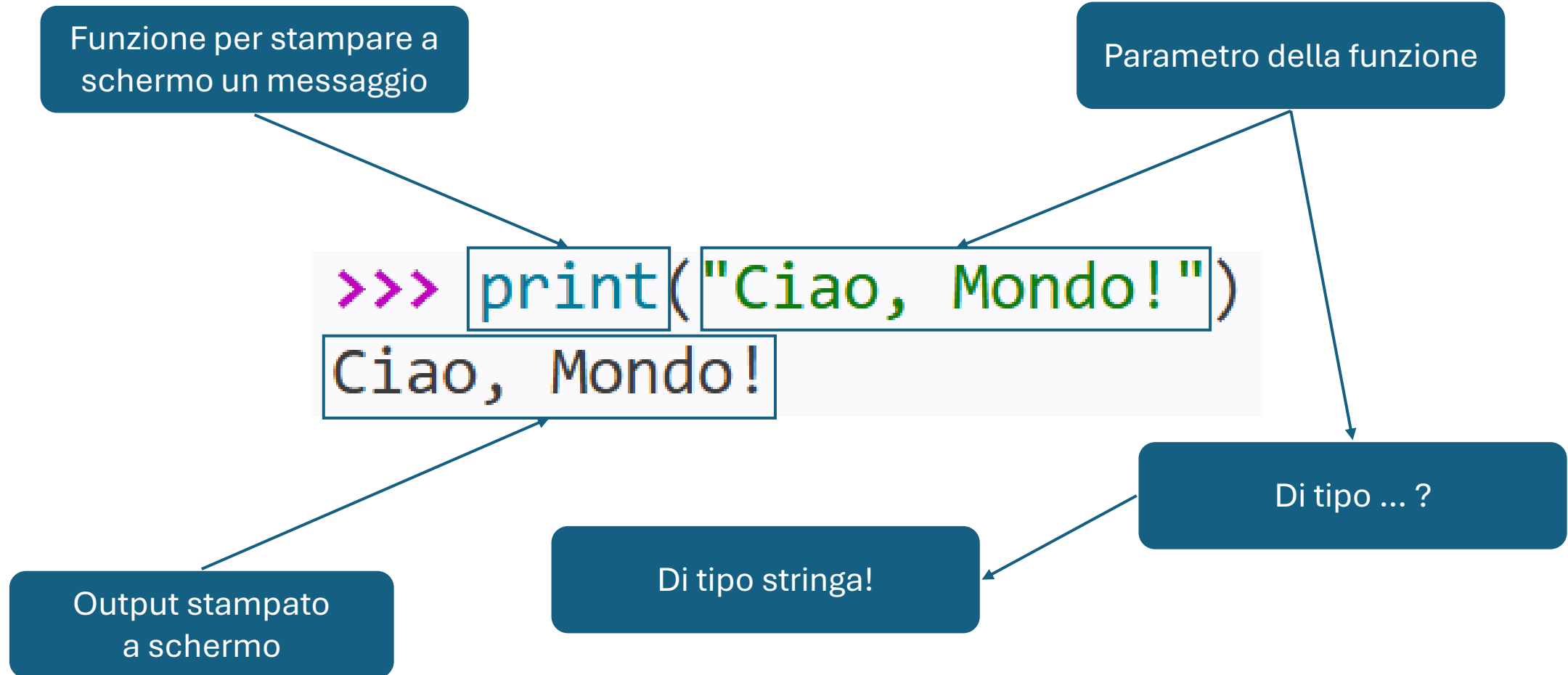
```
>>> print("Ciao, Mondo!")  
Ciao, Mondo!
```

Output stampato a schermo

Output verso l'utente



Output verso l'utente

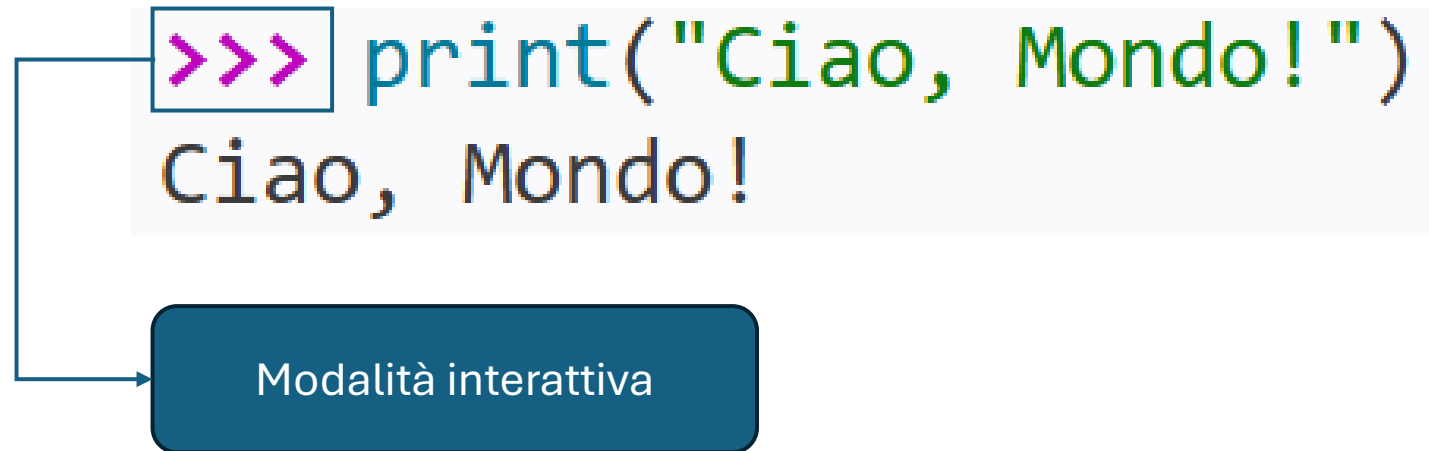


Output verso l'utente

```
>>> print("Ciao, Mondo!")  
Ciao, Mondo!
```

Q: *In che modalità di Python siamo?*

Output verso l'utente



Q: *In che modalità di Python siamo?*

Output multiplo

- La funzione `print()` usa solitamente un parametro, ma ne ammette anche più di uno in modo da permettere la stampa di stringhe multiple.

Definizione: *funzione* `print()`

Sintassi: `print("...")`, `print("...", "...")`

La funzione presenta come parametro opzionale un insieme di valori, se non viene stampato nessun valore verrà stampata una linea vuota. Altrimenti, i valori verranno stampati uno dopo l'altro separati da uno spazio.

Alcuni esempi della funzione `print()`

- Stampare valori numerici:
 - `print(3 + 4)`
 - Valuta l'espressione $3 + 4$ e visualizza 7.
- Passare più valori alla funzione:
 - `print("Valore ottenuto", 6 * 7)`
 - Visualizza: «Valore ottenuto 42»
- Tutti i valori passati alla funzione vengono visualizzati uno dopo l'altro, separati da uno spazio.

Alcuni esempi della funzione print()

- Di default, la funzione print crea una nuova riga («va a capo») ogni volta che viene invocata:
 - `print("Ciao, ")`
 - `print("Mondo!")`
- Sampa due linee di testo:
 - Ciao,
 - Mondo!

Sezione #3

Esercizi

Esercizi

- **Esercizio 1**

Scrivere un programma in Python che stampi a schermo il proprio nome, dove il proprio nome è salvato in una variabile.

- **Esercizio 2**

Scrivere un programma in Python che stampi a schermo un nome inserito dall'utente, il nome deve essere prima salvato in una variabile.

- **Esercizio 3**

Scrivere un programma in Python che legga due numeri interi dall'utente e stampi a schermo il risultato delle quattro operazioni fondamentali.