

Introduzione alla Programmazione e a Python

Francesco Gobbi

I.I.S.S. Galileo Galilei, Ostiglia

20 marzo 2025

Cos'è la Programmazione

- ▶ La programmazione consiste nell'insieme di istruzioni che permettono ad un computer di eseguire compiti specifici.
- ▶ È alla base dello sviluppo di software, applicazioni e sistemi informatici.
- ▶ Consente di automatizzare processi e risolvere problemi complessi in modo efficiente.

Introduzione alla Programmazione in Python

- ▶ **Python** è un linguaggio di programmazione ad alto livello, noto per la sua sintassi semplice ed elegante.
- ▶ Ideale per chi si avvicina per la prima volta al mondo della programmazione.
- ▶ Utilizzato in numerosi campi, dalla creazione di script per l'automazione alla realizzazione di applicazioni complesse.

Programmi Compilati vs Interpretati

▶ Programmi Compilati:

- ▶ Il codice sorgente viene tradotto integralmente in codice macchina tramite un compilatore.
- ▶ Esempi: C, C++.
- ▶ Vantaggi: esecuzione più veloce, ottimizzazione del codice.

▶ Programmi Interpretati:

- ▶ Il codice viene eseguito riga per riga da un interprete.
- ▶ Esempi: Python, JavaScript, Ruby.
- ▶ Vantaggi: maggiore flessibilità, facilità di debugging e sviluppo.

Tipologia di Programma Python

- ▶ **Script:** piccoli programmi per automatizzare compiti ripetitivi.
- ▶ **Applicazioni Web:** sviluppo di siti e servizi online con framework come Django o Flask.
- ▶ **Applicazioni Scientifiche:** analisi dati e calcoli numerici tramite librerie come NumPy e SciPy.
- ▶ **Machine Learning e Data Science:** implementazione di modelli predittivi con scikit-learn, TensorFlow, PyTorch.
- ▶ **Sviluppo di GUI:** creazione di interfacce grafiche con Tkinter, PyQt o Kivy.

Campi di Utilizzo di Python

- ▶ **Sviluppo Web:**

- ▶ *Esempio:* realizzazione di un sito web dinamico con il framework Django.

- ▶ **Data Science e Analisi Dati:**

- ▶ *Esempio:* analisi e visualizzazione di dati con Pandas e Matplotlib.

- ▶ **Machine Learning:**

- ▶ *Esempio:* sviluppo di algoritmi di classificazione con scikit-learn.

- ▶ **Automazione e Scripting:**

- ▶ *Esempio:* script per l'automazione di compiti ripetitivi o amministrativi.

- ▶ **Sviluppo di Giochi:**

- ▶ *Esempio:* creazione di giochi semplici con Pygame.

Perché Utilizzare Python

- ▶ **Sintassi Chiara:** facilita la lettura e la scrittura del codice.
- ▶ **Grande Comunità:** numerose risorse, documentazione e supporto online.
- ▶ **Ampia Gamma di Librerie:** strumenti per ogni esigenza, dalla matematica alla grafica.
- ▶ **Versatilità:** utilizzo in vari ambiti, dal web al machine learning.
- ▶ **Facilità di Integrazione:** interoperabilità con altri linguaggi e tecnologie.

Programmazione ad Oggetti in Python

- ▶ La programmazione ad oggetti (OOP) permette di modellare problemi reali attraverso entità chiamate **oggetti**.
- ▶ Ogni oggetto possiede **attributi** (stati/proprietà) e **metodi** (comportamenti/funzioni).
- ▶ In Python, quasi tutto è un oggetto: numeri, stringhe, funzioni e persino moduli.
- ▶ La creazione di **classi**(già esistenti o nuove) consente di definire nuovi tipi di oggetti e di riutilizzare il codice in modo modulare.

Programmazione ad Oggetti vs Programmazione Classica

- ▶ **Programmazione Classica (Procedurale):**
 - ▶ Organizza il codice in funzioni e procedure.
 - ▶ I dati e le funzioni sono separati.
- ▶ **Programmazione ad Oggetti:**
 - ▶ Combina dati e comportamenti in un'unica struttura (l'oggetto).
 - ▶ Favorisce l'astrazione, l'incapsulamento, l'ereditarietà e il polimorfismo.
 - ▶ Migliora la manutenibilità e l'estendibilità del codice.

Concetti Astratti di Classe ed Oggetto

- ▶ **Classe:** è un modello, un insieme, un raccoglietore, per contenere e creare oggetti.
 - ▶ Es.: *Animale*, *Veicolo*, *Libro*.
- ▶ **Oggetto:** è un'istanza di una classe, dotata di valori specifici per gli attributi.
 - ▶ Es.: da *Animale* possiamo avere l'oggetto *Fido* (un cane) o *Whiskers* (un gatto).
- ▶ La Classe quindi è composta di oggetti che hanno tra loro sono identificati dallo stesso tipo di attributi.
 - ▶ La classe *Animale* conterrà tutti gli animali che hanno come caratteristiche: nome, tipoRazza, età e genere.
 - ▶ Se un oggetto animale non ha tutti i seguenti attributi, vorrà dire che questo non potrà facilmente essere riconducibile alla classe *Animale*.
- ▶ Le classi definiscono **attributi** (le caratteristiche) e **metodi** (le azioni o comportamenti).
- ▶ **Esempio Astratto:** una classe *Libro* potrebbe avere attributi come titolo, autore, anno e metodi come *leggi* o *descrivi*.

Esempio Concreto in Python: Classe Persona

```
1 class Persona:
2     def __init__(self, nome, eta):
3         self.nome = nome      # Attributo: nome
4         self.eta = eta        # Attributo: et
5
6     def saluta(self):
7         # Metodo: saluta, che stampa un messaggio di
8         # benvenuto
9         print(f"Ciao, sono {self.nome} e ho {self.eta}
10             anni.")
11
12 # Creazione di un oggetto (istanza) della classe
13     Persona
14 p = Persona("Luca", 25)
15 p.saluta()  # Output: Ciao, sono Luca e ho 25 anni.
```

Listing 1: Definizione e utilizzo della classe Persona in Python

Esempio di Codice Python

```
1 print("Ciao Mondo!")
```

Listing 2: Script Python di esempio