

Programmazione ad Oggetti in Python

Francesco Gobbi

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

31 marzo 2025

Python: Linguaggio di Programmazione ad Oggetti

- ▶ Python è un linguaggio di programmazione ad oggetti: ogni entità (numeri, stringhe, funzioni, etc.) è un oggetto.
- ▶ Questo paradigma permette di organizzare il codice in moduli, classi e oggetti, rendendolo più modulare e riutilizzabile.

Python: Linguaggio Interpretato

- ▶ Python è un linguaggio interpretato: il codice viene eseguito riga per riga dall'interprete.
- ▶ Non richiede una fase di compilazione, facilitando il testing e il debugging.
- ▶ Questo approccio consente una rapida prototipazione e sperimentazione.

Cos'è una Classe?

- ▶ Una classe è un modello o una "ricetta" per creare oggetti.
- ▶ Definisce attributi (caratteristiche) e metodi (comportamenti) comuni a tutti gli oggetti che ne sono istanze.
- ▶ Quindi la **classe** è un insieme di oggetti che hanno tutti i medesimi attributi e metodi, ovvero le stesse caratteristiche per descrivere un oggetto e gli stessi comportamenti che possono essere utilizzati sull'oggetto stesso.
- ▶ **Esempio concettuale:**
 - ▶ *Automobile*: la classe può definire attributi come *colore*, *modello* e metodi come *accelera* o *frena*.

Cos'è un Oggetto?

- ▶ Un oggetto è un'istanza concreta di una classe, con valori specifici per gli attributi definiti dalla classe.
- ▶ Ogni oggetto è unico anche se appartiene alla stessa classe.
- ▶ **Esempio concettuale:**
 - ▶ *miaAuto*: un'istanza della classe *Automobile* che potrebbe avere attributi specifici come *colore* = *rosso* e *modello* = *sportivo*.

Metodi: Funzioni Legate a Classe e Oggetto

- ▶ I metodi sono funzioni definite all'interno di una classe che rappresentano i comportamenti degli oggetti.
- ▶ Esistono due tipologie principali:
 - ▶ **Metodo di istanza:** opera su un oggetto specifico.
 - ▶ *Esempio concettuale:* il metodo *accelera* di *miaAuto* aumenta la velocità solo di quell'automobile.
 - ▶ **Metodo di classe:** opera sull'intera classe, non su una specifica istanza.
 - ▶ *Esempio concettuale:* un metodo che conta quante automobili sono state create dalla classe *Automobile*.
- ▶ I metodi aiutano a incapsulare il comportamento degli oggetti all'interno della classe.

Riferimenti agli Oggetti in Python

- ▶ In Python ogni valore è un oggetto.
- ▶ La variabile `x` in:

```
1 x = 10  # Il numero 10 e' un oggetto di tipo int, e 'x'  
        ' e' un'etichetta che punta a quell'oggetto.
```

Listing 1: Assegnamento e riferimento a un oggetto

- ▶ Qui, il valore 10 è un oggetto, mentre `x` è semplicemente una variabile che funge da etichetta per far riferimento a quell'oggetto desiderato.
- ▶ La gestione della memoria in Python avviene tramite il conteggio dei riferimenti agli oggetti.

Riassegnazione delle Variabili in Python

- ▶ In Python, le variabili sono etichette che possono essere riassegnate per puntare a oggetti differenti, anche di tipi diversi.
- ▶ **Esempio concettuale:**
 - ▶ Inizialmente, si assegna `x = 10`: qui `x` punta ad un oggetto di tipo `int` che contiene il valore 10.
 - ▶ Successivamente, si può assegnare `x = "Ciao"`: ora `x` punta ad un oggetto di tipo `stringa` contenente la stringa "Ciao".
- ▶ Questo comportamento dimostra la natura dinamica delle variabili in Python: la stessa etichetta può puntare a oggetti di tipi differenti nel corso del programma.
- ▶ Fare attenzione al nome delle variabili/etichette utilizzate e come queste vengono utilizzate durante il codice.

Esercizio: Cosa punta la variabile x?

- ▶ Considera la seguente sequenza di assegnazioni:
 1. `x = 10`
 2. `x = "Ciao"`
 3. `x = [1, 2, 3]`
- ▶ **Domanda:** Qual è l'oggetto a cui la variabile `x` fa riferimento alla fine di queste assegnazioni?

Soluzione: Riassegnazione di Variabili

- ▶ La variabile `x` alla fine punta all'oggetto `list` `[1, 2, 3]`.
- ▶ **Spiegazione:**
 - ▶ In Python ogni assegnazione ad una variabile sovrascrive il riferimento precedente.
 - ▶ Dopo `x = 10`, `x` puntava a un oggetto di tipo `int` con valore 10.
 - ▶ Con `x = "Ciao"`, `x` è stato riassegnato per puntare ad un oggetto di tipo `str` contenente "Ciao".
 - ▶ Infine, con `x = [1, 2, 3]`, `x` punta all'oggetto di tipo `list` `[1, 2, 3]`.
 - ▶ Questo esempio evidenzia come la stessa etichetta (`x`) possa riferirsi a oggetti di tipi diversi nel tempo.

Cos'è un riferimento ad oggetto?

- ▶ In Python ogni valore è un oggetto.
- ▶ Le variabili non contengono direttamente il valore, ma conservano un **riferimento** all'oggetto.
- ▶ **Un riferimento è un indirizzo in memoria che punta alla posizione in memoria (RAM solitamente) in cui è memorizzato l'oggetto.**

Come funziona il riferimento ad oggetto

- ▶ Quando si assegna un valore a una variabile, Python:
 - ▶ Crea l'oggetto in memoria.
 - ▶ Associa alla variabile un riferimento che punta a quell'oggetto.
- ▶ **Esempio:** `x = 10`
 - ▶ L'oggetto 10 (di tipo `int`) viene creato.
 - ▶ La variabile `x` contiene un riferimento che punta a questo oggetto.
- ▶ Le variabili possono essere riassegnate per puntare a oggetti differenti.

Il conteggio dei riferimenti

- ▶ Python tiene traccia del numero di riferimenti ad ogni oggetto.
- ▶ Quando un oggetto non ha più riferimenti (il conteggio scende a zero), viene automaticamente eliminato dalla memoria.
- ▶ Questo meccanismo è noto come **garbage collection**.

Esempio pratico: assegnazione e riassegnazione

- ▶ Considera la seguente sequenza:
 1. `x = 10` (`x` punta all'oggetto 10 di tipo `int`)
 2. `y = x` (`y` punta allo stesso oggetto a cui punta `x`)
 3. `x = "Python"` (`x` viene riassegnata e ora punta a un oggetto "Python" di tipo `str`)
- ▶ In questo scenario:
 - ▶ La variabile `y` continua a puntare all'oggetto 10.
 - ▶ La variabile `x` ora punta a un nuovo oggetto, la stringa "Python".
- ▶ Questo esempio evidenzia come le variabili siano semplicemente etichette che possono essere "riprogrammate" per puntare a oggetti diversi.

Conclusioni

- ▶ Python adotta il paradigma della programmazione ad oggetti, migliorando modularità e riusabilità.
- ▶ È un linguaggio interpretato che permette una rapida iterazione nello sviluppo.
- ▶ Ogni valore in Python è un oggetto, mentre le variabili sono semplici etichette che fanno riferimento a questi oggetti.
- ▶ La riassegnazione delle variabili evidenzia la flessibilità del linguaggio e la gestione della memoria tramite il conteggio dei riferimenti.