

Corso ITS:

PROGETTISTA E SVILUPPATORE SOFTWARE:

*FULL STACK DEVELOPER E CLOUD SPECIALIST*

Modulo: **Programmazione in Python**

Docente: *Andrea Ribuoli*

---

**Martedì 15 Aprile 2025**

**09:00 - 13:00**

**13:30 - 16:30**

---

### Concetto di oggetto

- nella programmazione ad oggetti non si gestiscono solo numeri e stringhe
- il programmatore ne implementa il comportamento realizzando i **metodi**
- la **programmazione orientata agli oggetti** (*object oriented programming*) è un paradigma che vuole semplificare la realizzazione di programmi altrimenti molto complessi
- abbiamo già visto *stringhe, liste, insiemi e dizionari*
- ciascuno di questi oggetti ha un proprio insieme di *metodi*

### Concetto di classe

- gli oggetti con comportameno simile sono raggruppati in **classi**
- il programmatore ne implementa il comportamento realizzando i **metodi**
- durante lo sviluppo di un programma in OOP si **creano** oggetti
- in Python una **classe** (*class*) descrive un insieme di oggetti che hanno lo stesso comportamento

### Concetto di incapsulamento

- l'insieme di tutti i metodi messi a disposizione da una *classe* è l'**interfaccia pubblica** della classe
- si parla di **incapsulamento** (*encapsulation*) a fronte di questa strategia di **information hiding**
- semplifichiamo per altri programmatori l'uso delle classi da noi create

## Variabili di istanza

- un oggetto è dunque un **esemplare** di una classe (anche detto **istanza** di una classe)
- un oggetto memorizza i propri dati in **variabili di istanza** (o di *esemplare*)
- gli utilizzatori della classe non vedono direttamente tali variabili
- per convenzione, il nome di tali variabili inizia con l'**underscore** ( `_` )

## Metodi e costruttori

- un metodo viene definito come parte della definizione di una **classe** ( `class Abcd :`  )
- la prima variabile parametro di un metodo si chiama sempre **self**
- i metodi accedono alle variabili di istanza antepoendo al nome il parametro **self** seguito dal punto `self.`
- si differenziano metodi **modificatori** piuttosto che metodi **di solo accesso**

## Riferimenti ad oggetti

- in Python gli oggetti non sono **interni** alla variabile
- la variabile conserva la **posizione in memoria** dell'oggetto
- parliamo più propriamente di **riferimento ad un oggetto**
- è dunque possibile che **due (o più)** variabili facciano riferimento allo stesso oggetto
- due variabili facenti riferimento al medesimo oggetto vengono dette **alias**
- operatore **is** (o **is not**)