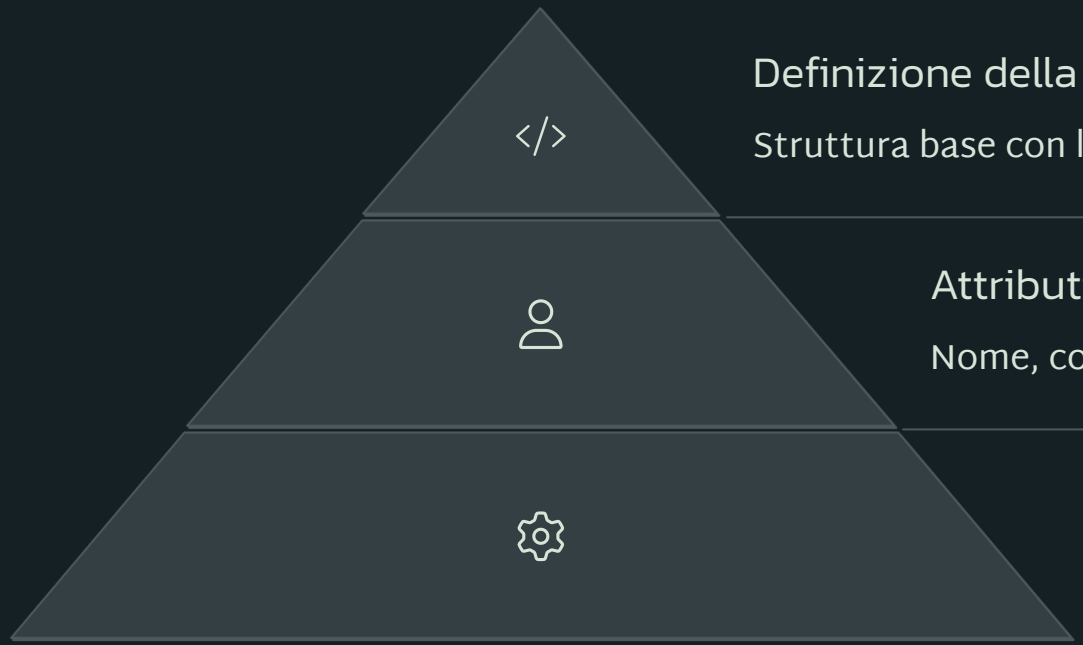
Programmazione Orientata agli Oggetti in Python

Benvenuti a questo corso introduttivo alla programmazione orientata agli oggetti in Python. Esploreremo i concetti fondamentali con esempi pratici.

Creare una Classe Persona



Definizione della classe

Struttura base con la parola chiave class

Attributi della persona

Nome, cognome, età come proprietà fondamentali

Costruttore __init__

Inizializzazione degli attributi al momento della creazione

La classe Persona è il nostro primo esempio di astrazione. Rappresenta una persona reale con attributi come nome, cognome ed età.

Il Metodo saluta()

Definizione del metodo

Aggiunta della funzione saluta() alla classe Persona

Personalizzazione

Utilizzo degli attributi per creare un messaggio personalizzato

Invocazione

Chiamata del metodo sull'istanza della classe

Il metodo saluta() genera un messaggio personalizzato. Utilizza gli attributi dell'oggetto per creare un'interazione unica.

Creare Istanze della Classe

Creazione di oggetti

persona1 = Persona("Mario", "Rossi", 30)

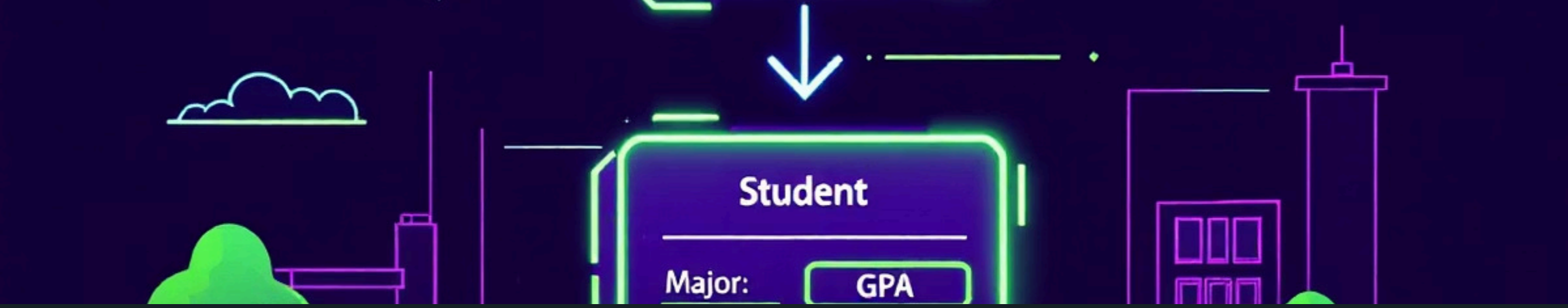
Oggetti indipendenti

Ogni oggetto ha i propri valori per gli attributi

Interazione con oggetti

Chiamata di metodi su ciascuna istanza: persona1.saluta()

Possiamo creare più oggetti dalla stessa classe. Ogni oggetto è una istanza indipendente con propri attributi e comportamenti.



Creazione di una Sottoclasse

Definizione della sottoclasse

class Studente(Persona): per ereditare da Persona

Ereditarietà

Studente eredita tutti gli attributi e metodi di Persona

Specializzazione

Aggiunta di funzionalità specifiche per lo Studente

La sottoclasse Studente estende la classe Persona. Mantiene tutte le caratteristiche di Persona aggiungendo funzionalità specifiche.

Override del Metodo saluta()

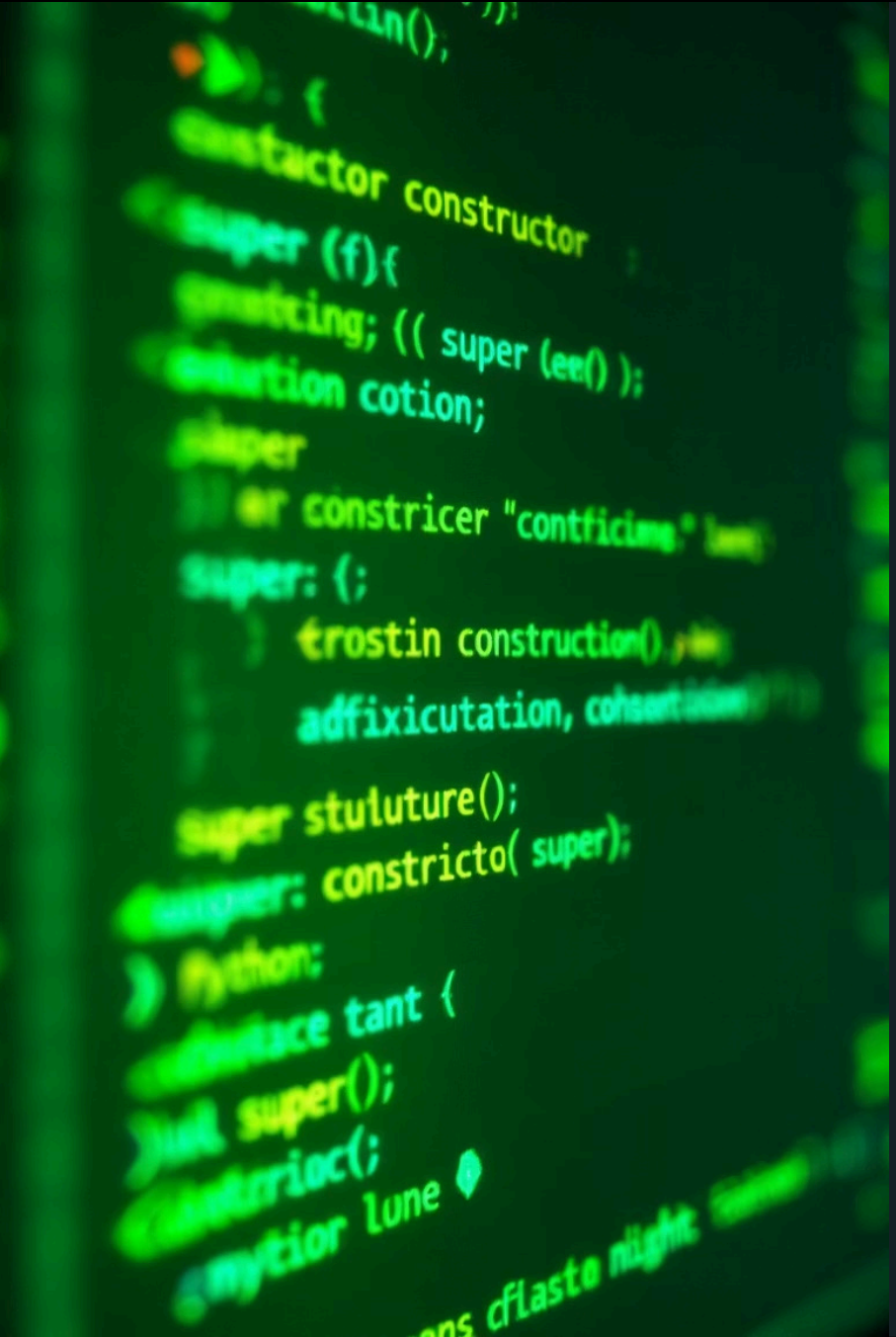
Metodo originale

```
def saluta(self):
    return f"Ciao, sono {self.nome}"
```

Metodo sovrascritto

```
def saluta(self):
    return f"Salve, sono lo studente {self.nome}"
```

L'override permette di modificare il comportamento ereditato. La sottoclasse può implementare una versione personalizzata del metodo.



Estendere il Costruttore con super()

Definizione costruttore Studente

__init__ con parametri estesi

Chiamata super().__init__()

Inizializzazione degli attributi della classe base

Inizializzazione attributi aggiuntivi

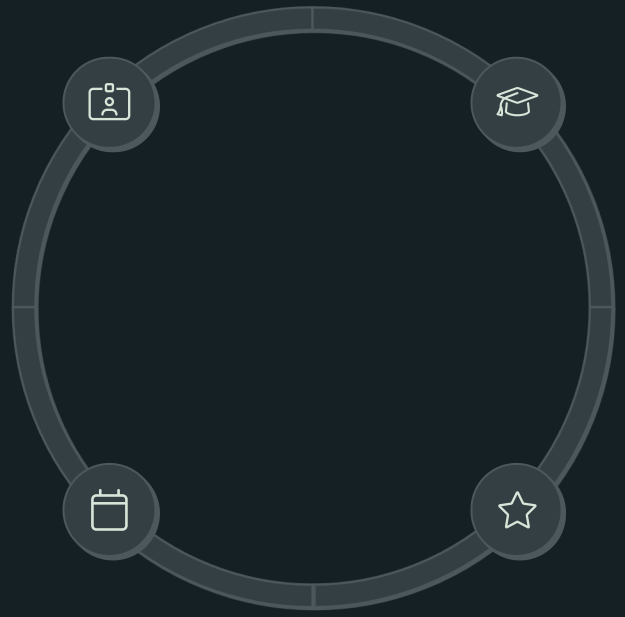
Aggiunta dei nuovi attributi specifici dello Studente

La funzione super() richiama il costruttore della classe genitore. Consente di estendere il comportamento senza duplicare il codice.

Attributi Specifici dello Studente

Matricola
Identificativo unico dello studente

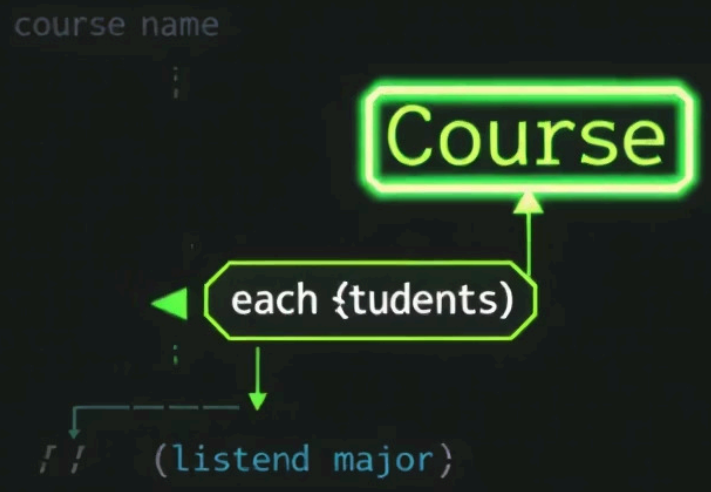
Corso di laurea
Indirizzo di studio dello studente



Anno di iscrizione
Anno accademico di inizio

Media voti
Rendimento accademico

Lo Studente ha attributi specifici oltre a quelli ereditati. Questi attributi rappresentano le caratteristiche uniche di uno studente.



Creazione della Classe Corso

Attributi del corso

Nome, codice, docente e lista degli studenti iscritti

Lista degli studenti

Inizializzata come lista vuota nel costruttore

Struttura base

class Corso con __init__ per inizializzare gli attributi

La classe Corso gestisce un gruppo di studenti. Contiene attributi descrittivi e una lista per memorizzare gli iscritti.

Metodo per Iscrivere Studenti

1

Definizione metodo

iscrivi_studente() con parametro studente

2

Controllo duplicati

Verifica se lo studente è già iscritto

3

Aggiunta studente

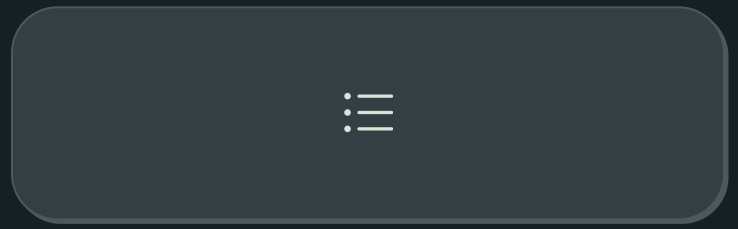
Append alla lista degli studenti iscritti

Il metodo iscrivi_studente() aggiunge uno studente alla lista degli iscritti. Controlla anche la presenza di duplicati per evitare iscrizioni multiple.

Stampare Elenco Iscritti

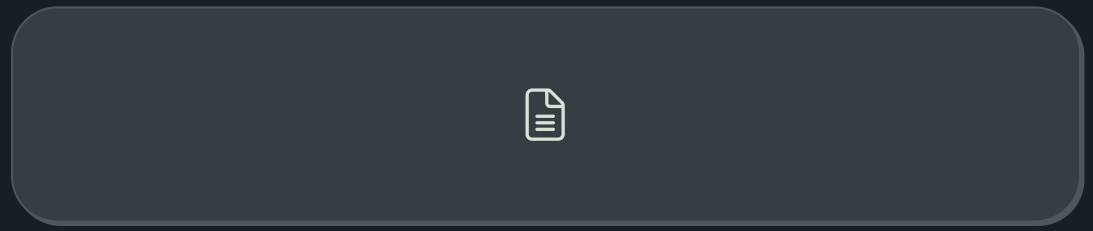
Definizione metodo

stampa_iscritti() per visualizzare tutti gli studenti



Iterazione sulla lista

Ciclo for per scorrere gli studenti iscritti



Formattazione output

Presentazione ordinata delle informazioni

Il metodo stampa_iscritti() visualizza tutti gli studenti del corso. Formatta l'output per una facile lettura delle informazioni.

Classe Docente con Metodi Specifici

Definizione classe Docente

Sottoclasse di Persona con attributi specifici

Assegnazione corsi

Metodo per gestire i corsi assegnati

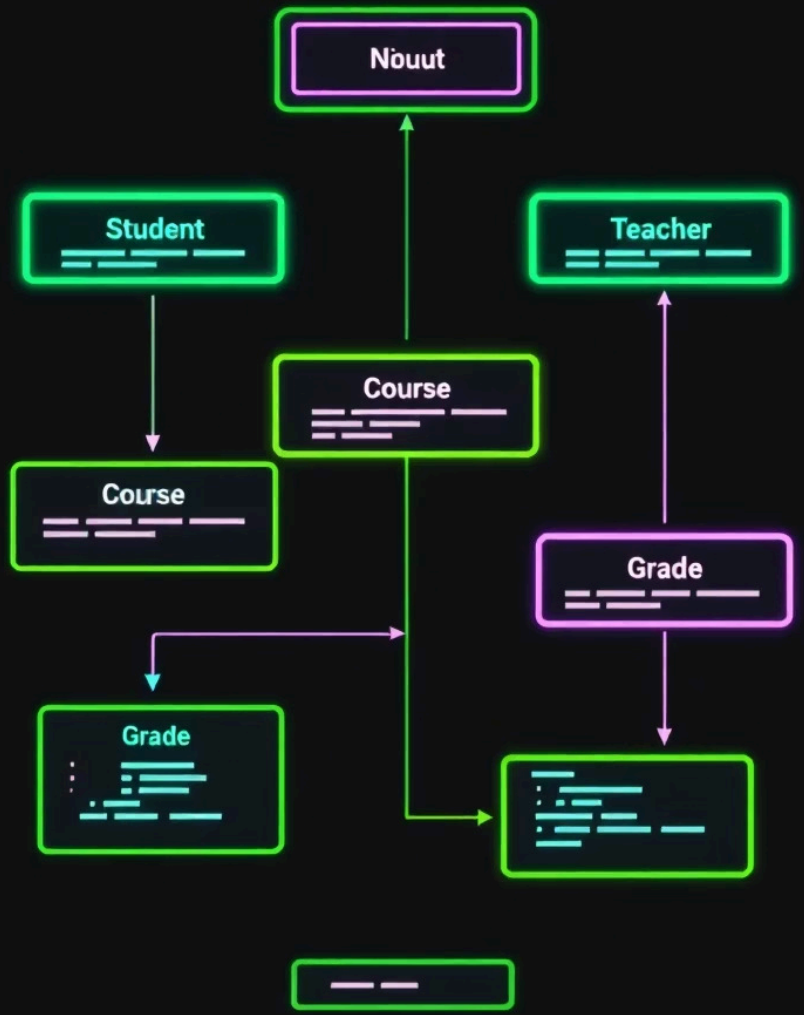
Statistiche corso

Analisi del rendimento degli studenti

Gestione valutazioni

Metodo per registrare i voti degli studenti

La classe Docente estende Persona con funzionalità didattiche. Include metodi per gestire corsi, valutazioni e statistiche di rendimento.



Esercizio: Gestione Scuola

Creazione sistema

Implementare le classi per una scuola completa

Relazioni

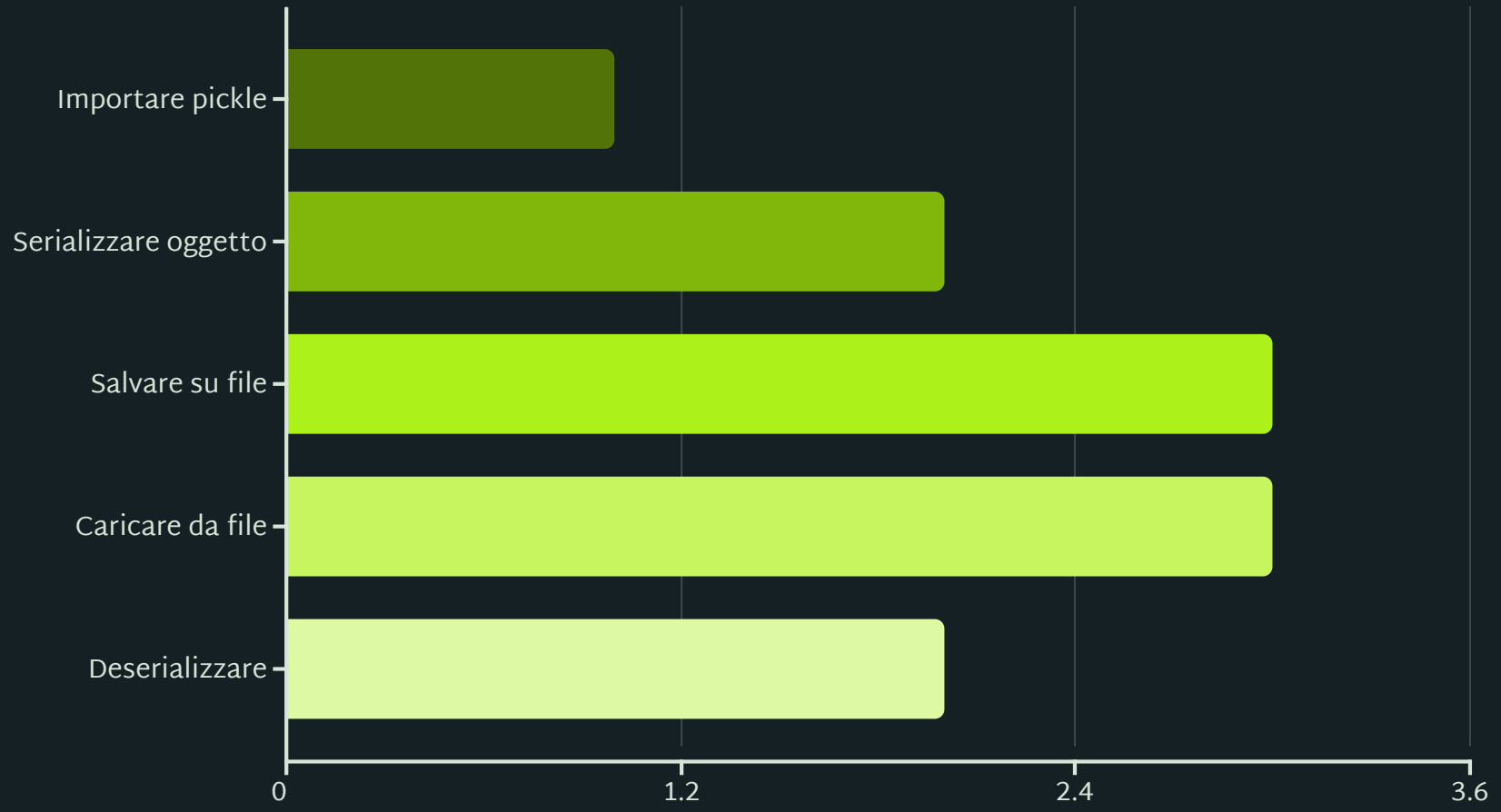
Gestire studenti, docenti, corsi e aule

Funzionalità

Iscrizioni, assegnazioni corsi, orari e valutazioni

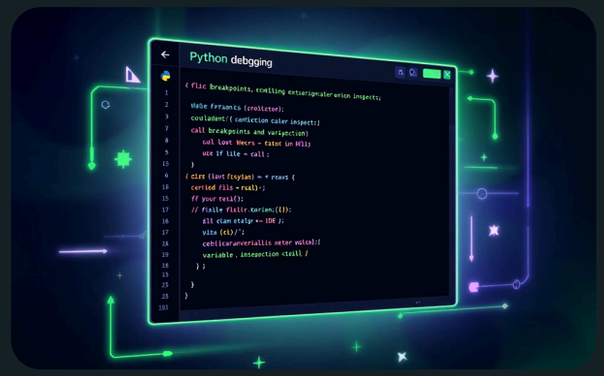
Esercizio pratico per consolidare i concetti. Crea un sistema completo di gestione scolastica utilizzando classi e oggetti.

Salvataggio Oggetti con Pickle



Il modulo pickle permette di salvare oggetti Python su disco. È utile per mantenere lo stato del programma tra diverse esecuzioni.

Debug e Mini-Progetto Finale

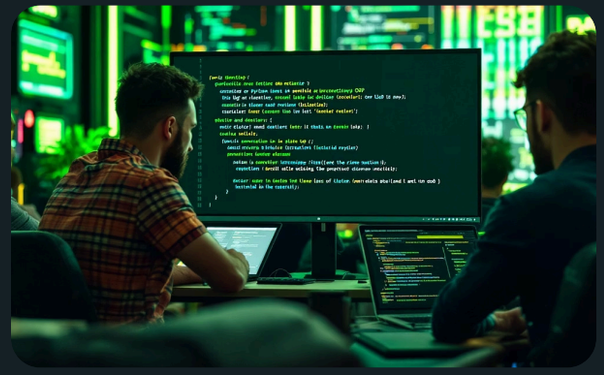


Strumenti di Debug

Tecniche per identificare e risolvere errori nelle classi e negli oggetti.

Presentazione Progetto

Condivisione del mini-progetto sviluppato durante il corso.



Revisione del Codice

Analisi collaborativa per migliorare la qualità e l'efficienza del codice.

Il corso si conclude con tecniche di debug e la presentazione del mini-progetto. Dimostra le competenze acquisite creando un'applicazione completa basata su OOP.