

DENOMINAZIONE INSEGNAMENTO / ATTIVITÀ FORMATIVA

INTRODUZIONE ALLA PROGRAMMAZIONE IN PYTHON

INTRODUCTION TO PYTHON PROGRAMMING

INFORMAZIONI INSEGNAMENTO / ATTIVITÀ FORMATIVA

A.A.	2026-2027	CdS	SCIENZE BIOLOGICHE
Codice	8068363	Canale	Unico
CFU	2	Lingua	Italiano

DOCENTE RESPONSABILE DELL'INSEGNAMENTO / ATTIVITÀ FORMATIVA

Manuela Helmer Citterich

OBIETTIVI FORMATIVI E RISULTATI DI APPRENDIMENTO ATTESI

Italiano	OBIETTIVI FORMATIVI: acquisizione delle conoscenze di base della programmazione in Python, dall'esplorazione delle caratteristiche del linguaggio all'organizzazione del codice e all'utilizzo di librerie per l'analisi e la visualizzazione dei risultati. CONOSCENZA E CAPACITÀ DI COMPRENSIONE: acquisizione di una solida comprensione dei concetti fondamentali di Python e familiarità con i principi logici della programmazione. CAPACITÀ DI APPLICARE CONOSCENZA E COMPRENSIONE: sviluppo della capacità di risolvere problemi semplici mediante la scrittura e l'esecuzione di programmi Python. AUTONOMIA DI GIUDIZIO: sviluppo della capacità di valutare soluzioni alternative a problemi di programmazione e di individuare errori o limiti nel codice. ABILITÀ COMUNICATIVE: sviluppo della capacità di presentare risultati e visualizzazioni in modo chiaro, utilizzando grafici e rappresentazioni appropriate. CAPACITÀ DI APPRENDIMENTO: acquisizione di competenze per consultare e utilizzare in modo efficace documentazione, esempi e risorse tecniche relative a Python.
<i>English</i>	LEARNING OUTCOMES: <i>The aim of the course is to teach how to work with essential tools in Python, starting from exploring the language's features, to code organization, and the utilization of libraries for results visualization.</i> KNOWLEDGE AND UNDERSTANDING: <i>To acquire a solid understanding of fundamental Python concepts and become familiar with logical principles of programming.</i>

	<i>APPLYING KNOWLEDGE AND UNDERSTANDING:</i> <i>Develop problem-solving skills.</i> <i>MAKING JUDGEMENTS:</i> <i>Competency in evaluating solutions for programming problems.</i> <i>COMMUNICATION SKILLS:</i> <i>Ability to communicate results through the utilization of plots.</i> <i>LEARNING SKILLS:</i> <i>Acquire skills in effectively utilizing documentation.</i>
--	--

PREREQUISITI

Italiano	nessuno
English	none

PROGRAMMA E CRONOPROGRAMMA

Italiano	Lezione 1 (2 ore): Perché programmare. Cos'è Python e a cosa serve. Linguaggi interpretati e compilati. Ambiente di sviluppo (Spyder). Prime istruzioni in Python. Commenti nel codice. Variabili e tipi di dato. Esercitazione: scrittura dei primi programmi, utilizzo dell'interprete, dichiarazione di variabili ed esecuzione di semplici operazioni. (Materiale didattico: Pacchetto di slide 1) Lezione 2 (2 ore): Conversione dei tipi di dato. Operazioni con numeri e stringhe. Caratteri di escape. Formattazione delle stringhe. Funzione input(). Esercitazione: sviluppo di semplici programmi che acquisiscono input dall'utente, manipolano stringhe ed eseguono operazioni aritmetiche. (Materiale didattico: Pacchetto di slide 2) Lezione 3 (2 ore): Valori booleani e operatori logici. Istruzioni condizionali (if, elif, else). Esercitazione: realizzazione di programmi basati su condizioni logiche e risoluzione di semplici problemi decisionali. (Materiale didattico: Pacchetto di slide 3) Lezione 4 (2 ore): Cicli for, funzione range. Introduzione alle strutture dati: liste, tuple, dizionari e insiemi. Esercitazione: utilizzo dei cicli per elaborare sequenze di dati e primi esercizi con le strutture dati. (Materiale didattico: Pacchetto di slide 4) Lezione 5 (2 ore): Operazioni sulle strutture dati. Lettura e scrittura di file. Esercitazione: manipolazione di liste e dizionari, lettura di dati da file di testo e scrittura dei risultati su file. (Materiale didattico: Pacchetto di slide 5) Lezione 6 (2 ore): Gestione del file system. Definizione e utilizzo delle funzioni. Parametri e valore di ritorno (return). Esercitazione: sviluppo di programmi modulari mediante funzioni e gestione di file e cartelle. (Materiale didattico: Pacchetto di slide 6)
----------	---

	Lezione 7 (2 ore): Introduzione alle librerie NumPy e Pandas. Gestione di array e DataFrame. Esercitazione: caricamento, esplorazione e manipolazione di semplici dataset mediante NumPy e Pandas. (Materiale didattico: Pacchetto di slide 7) Lezione 8 (2 ore): Introduzione a Matplotlib. Creazione e personalizzazione di grafici per la visualizzazione dei dati. Esercitazione: realizzazione di grafici a partire da dataset di esempio e interpretazione delle principali visualizzazioni. (Materiale didattico: Pacchetto di slide 8)
English	<i>Lesson 1 (2 hours): Why programming matters. What Python is and its main applications. Interpreted and compiled programming languages. Development environment (Spyder). First Python instructions. Code comments. Variables and data types. Practical session: writing first Python programs, using the interpreter, declaring variables, and performing simple operations. (Reference materials: slide package 1)</i> <i>Lesson 2 (2 hours): Data type conversion. Operations with numbers and strings. Escape characters. String formatting. The input() function. Practical session: developing simple programs that receive user input, manipulate strings, and perform arithmetic operations. (Reference materials: slide package 2)</i> <i>Lesson 3 (2 hours): Boolean values and logical operators. Conditional statements (if, elif, else). Practical session: implementing programs based on logical conditions and solving simple decision-making problems. (Reference materials: slide package 3)</i> <i>Lesson 4 (2 hours): for loops and the range function. Introduction to data structures: lists, tuples, dictionaries, and sets. Practical session: using loops to process data sequences and solving introductory exercises involving data structures. (Reference materials: slide package 4)</i> <i>Lesson 5 (2 hours): Operations on data structures. Reading from and writing to files. Practical session: manipulating lists and dictionaries, reading data from text files, and writing results to output files. (Reference materials: slide package 5)</i> <i>Lesson 6 (2 hours): File system management. Defining and using functions. Parameters and return values (return). Practical session: developing modular programs using functions and working with files and directories. (Reference materials: slide package 6)</i> <i>Lesson 7 (2 hours): Introduction to the NumPy and Pandas libraries. Working with arrays and DataFrames. Practical session: loading, exploring, and manipulating simple datasets using NumPy and Pandas. (Reference materials: slide package 7)</i> <i>Lesson 8 (2 hours): Introduction to Matplotlib. Creating and customizing data visualizations. Practical session: generating graphs from example datasets and interpreting the resulting visualizations. (Reference materials: slide package 8)</i>

TESTI ADOTTATI E BIBLIOGRAFIA DI RIFERIMENTO

Italiano	Pacchetti di slide e materiale didattico forniti durante il corso. Documentazione ufficiale di Python e documentazione delle librerie utilizzate nelle esercitazioni. Per ciascun argomento
----------	---

	del programma, il pacchetto di slide e/o la documentazione di riferimento sono indicati nella sezione Programma.
<i>English</i>	<i>Slide packages and teaching materials provided during the course. Official Python documentation and documentation for the libraries used during the practical sessions. For each topic of the programme, the relevant slide package and/or documentation is indicated in the Programme section.</i>

DESCRIZIONE DELLE MODALITÀ DI SVOLGIMENTO E METODI DIDATTICI ADOTTATI

☒ In presenza ☐ A distanza

Italiano	Il corso sarà strutturato attraverso lezioni teoriche ed applicazioni pratiche. Studentesse e studenti avranno l'opportunità di applicare direttamente le conoscenze acquisite attraverso esercitazioni basate su casi di studio nel campo della biologia.
<i>English</i>	<i>The course will be structured through theoretical lectures and practical applications. Students will have the opportunity to apply the knowledge acquired through practical exercises based on real case studies in biology.</i>

MODALITÀ DI FREQUENZA

☐ frequenza obbligatoria ☒ frequenza facoltativa

DESCRIZIONE DELLE MODALITÀ DI FREQUENZA

Italiano	La frequenza è facoltativa. Le esercitazioni pratiche sono fortemente consigliate perché favoriscono l'applicazione dei concetti trattati a lezione, ma la mancata frequenza non comporta penalizzazioni nella valutazione. Studentesse e studenti frequentanti e non frequentanti sono valutati con le stesse modalità e sugli stessi contenuti. I materiali didattici messi a disposizione consentono anche a chi non frequenta di preparare l'esame sull'intero programma.
<i>English</i>	<i>Attendance is optional. Practical sessions are strongly recommended because they support the application of the concepts covered in class, but non-attendance does not result in any penalty in the assessment. Attending and non-attending students are assessed in the same way and on the same contents. The teaching materials provided allow non-attending students to prepare for the exam on the entire programme.</i>

MODALITÀ DI VALUTAZIONE

☒ Prova scritta ☐ Prova orale ☐ Prova di laboratorio ☐ Prova pratica
☐ Valutazione in itinere ☐ Valutazione di progetto ☐ Valutazione di tirocinio

COMPOSIZIONE DELLA COMMISSIONE VALUTATRICE

Manuela Helmer Citterich, Romina Appierdo, Gerardo Pepe

DESCRIZIONE DELLE MODALITÀ E DEI CRITERI DI VERIFICA DELL'APPRENDIMENTO

Italiano	L'esame consiste in una prova scritta a risposta multipla, della durata di 1 ora, finalizzata a verificare la conoscenza dei concetti fondamentali di Python trattati durante il corso. Le domande riguardano gli argomenti affrontati a lezione, tra cui sintassi di base, variabili, tipi di dato, strutture dati, operatori, istruzioni condizionali, cicli, funzioni, principali librerie introdotte durante il corso e interpretazione di semplici frammenti di codice. Non sono previste prove in itinere, esoneri parziali, valutazioni di progetto o valutazioni di tirocinio. I criteri di valutazione sono gli stessi per studentesse e studenti frequentanti e non frequentanti. Tutte le studentesse e tutti gli studenti sono valutati sui contenuti del programma e sui materiali didattici indicati nella scheda dell'insegnamento. Graduazione del voto: Non idoneo: la studentessa o lo studente mostra importanti carenze nella conoscenza dei concetti fondamentali di Python, non è in grado di interpretare semplici frammenti di codice e risponde in modo errato ai quesiti di base. 18–21: la studentessa o lo studente dimostra una conoscenza di base degli argomenti del corso e risponde correttamente alle domande relative ai concetti fondamentali. 22–25: la studentessa o lo studente mostra una buona comprensione dei principali costrutti del linguaggio Python e sa interpretarli correttamente in contesti semplici. 26–29: la studentessa o lo studente possiede una conoscenza completa e ben strutturata degli argomenti del corso, interpreta correttamente il codice e risolve quesiti di maggiore complessità. 30 e 30 e lode: la studentessa o lo studente dimostra una conoscenza approfondita e completa degli argomenti trattati, risponde correttamente anche ai quesiti più complessi e mostra piena padronanza dei concetti teorici e applicativi.
English	<i>The assessment consists of a 1-hour multiple-choice written examination designed to evaluate students' understanding of the fundamental Python concepts covered during the course. The questions address the topics presented in class, including basic syntax, variables, data types, data structures, operators, conditional statements, loops, functions, the main libraries introduced during the course, and the interpretation of simple code snippets.</i> <i>There are no midterm examinations, partial exemptions, project-based assessments, or internship evaluations.</i> <i>The assessment criteria are the same for attending and non-attending students. All students are evaluated on the course content and the teaching materials specified in the course syllabus.</i> <i>Grading criteria:</i> <i>Fail: the student shows significant deficiencies in the understanding of fundamental Python concepts, is unable to interpret simple code snippets, and incorrectly answers basic questions.</i> <i>18–21: The student demonstrates a basic understanding of the course topics and correctly answers questions covering the fundamental concepts.</i>

	<i>22–25: The student demonstrates a good understanding of the main Python language constructs and correctly interprets them in simple contexts.</i> <i>26–29: The student demonstrates a comprehensive and well-structured understanding of the course topics, correctly interprets code, and successfully answers more complex questions.</i> <i>30–30 with distinction (cum laude): The student demonstrates an in-depth and comprehensive understanding of all course topics, correctly answers the most challenging questions, and shows excellent mastery of both the theoretical and practical aspects of Python programming.</i>
--	---