



Python for Beginners

Prof. Motti Zohar (email: motti.zohar@univ.haifa.ac.il)

Reception time: Wednesday 16:30 -17:30 (schedule via email, appointments via zoom)

General

Python for Beginners is an online course tailored for students eager to dive into the world of Python programming. This course will guide you through the essentials of Python, from setting up and navigating Python environments to writing your first scripts. You will gain hands-on experience in scripting, explore the fundamentals of Object-Oriented Programming (OOP), and develop Python scripts using popular libraries and packages. Whether you are new to coding or looking to enhance your skills, this course provides a solid foundation for Python programming and practical implementation.

סטודנטים יקרים וסטודנטיות יקרות,

אם יש לכם לקות למידה או מוגבלות/בעיה רפואית שעשויה להשפיע על לימודיכם, ושבגינה אתם זקוקים להתאמות, אנא פנו לדיקנאט הסטודנטים, למדור נגישות ולקוויות למידה:
דוא"ל: LDA@univ.haifa.ac.il טל: 04-8249265
לאבחון והתאמות בגין לקות למידה ו/או הפרעת קשב יש לפנות ליה"ל:
דוא"ל: mhait@univ.haifa.ac.il טל: 04-8249022
אם יש ברשותכם מכתב התאמות מהאוניברסיטה ואתם זקוקים להנגשה בקורס אנא פנו אליי (למרצה) בשעות הקבלה או במייל, בסמוך לתחילת הקורס.

Dear students,

If you have a disability that may affect your studies and for which you may require accommodation, please contact me (the lecturer) or the Accessibility and Learning Disabilities Department at the Dean of Students office:

E-mail: LDA@univ.haifa.ac.il; Phone number: 04-98249265.

If you have received an accommodation letter, and need academic adjustments, please contact me (the lecturer) as soon as possible.



The objectives of the course

1. Develop foundational proficiency in Python programming from environment setup to executing scripts effectively.
2. Understand essential programming principles including variables, data types, functions, and control structures.
3. Gain practical experience in structured problem-solving by designing, coding, debugging, and testing Python applications.
4. Explore Object-Oriented Programming (OOP) concepts such as inheritance, abstraction and encapsulation to build reusable and modular code.
5. Apply Python for data handling and analysis using popular libraries such as NumPy, Pandas & Matplotlib.
6. Prepare the students for real-world applications of Python across domains like data science, automation, and GIS-based spatial analysis.

Duties and credit

1. Final exam: **100% of the final grade.** Passage of the final exam with a grade of 61 and above
Pay attention! The exam will be held at the Campus with NO access to the internet and AI tools
2. 12 weekly exercises: **recommended, not mandatory**

Pay attention!

In case of any contradiction between what is written in the syllabus and what is said or written elsewhere (e.g., in recording or presentation), the syllabus is the authoritative source.



Syllabus

Lesson/meeting	Theme	Units
(27.10.2025 19:00 via Zoom)	Opening meeting	Zoom meeting: Overview of course structure, grading, and expectations. Learn to install Python, set up IDEs, navigate materials, submit assignments, and access support, ensuring a smooth start to your learning journey.
1 (28.10.2025)	Introduction	▪ Unit 1: Course Overview – Introduction to course objectives, expectations, and study materials. Learn how to navigate the online platform and organize your learning process effectively. ▪ Unit 2: What is Python? – Understand the fundamentals of Python, its history, applications, and advantages. Get familiar with coding environments like IDLE and PyCharm. ▪ Unit 3: Writing Your First Script – Learn how to write, execute, and debug basic Python scripts using IDLE and PyCharm, building confidence in your coding abilities
2 (04.11.2025)	Python basics	▪ Unit 1: Introduction to Programming – Explore programming concepts: compilation vs. interpretation, Python's advantages and limitations, and working in environments like IDLE, PyCharm, and ArcGIS Pro. ▪ Unit 2: Core Python Elements – Learn about numbers, variables, expressions, functions, user input, and comments. Develop basic interactive scripts.
3 (11.11.2025)	Data Structures	▪ Unit 1: Sequences – Understand different sequence types (lists, tuples, strings) and their properties, including indexing, slicing, and iteration. ▪ Unit 2: Lists – Learn about list creation, modification, iteration, and common list operations like sorting and filtering. ▪ Unit 3: Tuples – Explore immutable sequences, when to use tuples, and how they compare to lists.
4 (18.11.2025)	Data Structures	▪ Unit 1: Sets – Discover how sets store unique elements, perform set operations, and apply them in real-world scenarios. ▪ Unit 2: Dictionaries – Learn how key-value pairs work, dictionary methods, and efficient data retrieval techniques. ▪ Unit 3: Strings – Master string manipulation techniques, including formatting, concatenation, slicing, and built-in string functions.
5 (25.11.2025)	Conditionals & Loops	▪ Unit 1: Conditional Statements – Implement decision-making using if, else, and elif statements. Learn Boolean logic and logical operators. ▪ Unit 2: Loops for Efficiency – Master for and while loops for repetitive tasks and learn how to exit loops using break and continue statements. ▪ Unit 3: Key Statements – Explore multi-line statements, return, try-except error handling, with statements for resource management, and other key Python constructs.
6 (02.12.2025)	Functions	▪ Unit 1: Defining and Calling Functions – Learn how to create reusable functions, pass parameters, and return values to write clean, modular code.



		▪ Unit 2: Recursive Functions – Understand recursion, how it differs from iteration, and its role in solving complex problems efficiently.
7 (09.12.2025)	Polymorphism	▪ Unit 1: Introduction to OOP – Learn about classes, objects, and how OOP principles improve code structure and reusability. ▪ Unit 2: Encapsulation, Inheritance & Abstraction – Discover the power of class inheritance, encapsulation, and abstraction for designing scalable applications.
8 (16.12.2025)	Object Oriented Programming (OOP)	▪ Unit 1: Implementing Classes & Objects – Create classes with constructors, properties, and methods to represent real-world entities. ▪ Unit 2: Extending Classes – Learn how to design complex classes, reuse code through inheritance, and implement internal and external class structures.
(23.12.2025 19:00 via zoom)	Mid-course meeting	Zoom meeting: Review of Lessons 1-8 – Recap key concepts, clarify doubts, and prepare for the next phase of the course.
9 (30.12.2025)	Advanced themes in Python IDEs	▪ Unit 1: Debugging & Testing – Use debugging tools, error tracing, and testing techniques to improve code quality and performance. ▪ Unit 2: Exception Handling – Handle runtime errors using try-except blocks to make programs more robust and reliable. ▪ Unit 3: Project Development – Learn how to structure Python projects using Jupyter Notebooks, ArcGIS Notebooks, and integrate Python with GIS tools.
10 (06.01.2026)	Files and Databases	▪ Unit 1: Handling Files – Read, write, and process CSV and text files efficiently in Python. ▪ Unit 2: Database Connectivity – Connect to Microsoft Access and SQL Server using PYODBC, write queries, and manage relational data.
11 (13.01.2026)	Extending python	▪ Unit 1: NumPy Library – Use NumPy for numerical computing, including array operations and mathematical computations. ▪ Unit 2: Pandas Library – Learn how to manipulate structured data using Pandas for analysis and visualization.
12 (20.01.2026)	Extending python cont.	▪ Unit 1: Matplotlib for Data Visualization – Generate plots and graphs to visualize data effectively. ▪ Unit 2: SciPy for Scientific Computing – Work with SciPy's advanced mathematical and statistical functions.
13 (26.01.2026 19:00 via zoom)	Course Summary	Zoom meeting: Summarize key topics, address remaining questions, and discuss the format of the final exam.



Bibliography

Books accompanying the course:

1. Hetland, M.L., Lie Hetland, M., and Nelli, F. 2024. Beginning Python : from Novice to Professional. Apress Media LLC, New York, NY. 4th edition
2. Nelli, F. 2023. Python Data Analytics : With Pandas, NumPy, and Matplotlib. Apress : Imprint: Apress, Berkeley, CA. 3rd edition

Usefull weblinks:

1. [Python Website](https://www.python.org/) – The official Python website provides the latest Python downloads, documentation, community updates, and learning resources. It is a primary reference for staying updated on Python's development and new features (<https://www.python.org/>)
2. [Python Tutorial](https://docs.python.org/3/tutorial/) – This is the official Python tutorial from the documentation site. It offers a structured, in-depth introduction to Python for beginners, covering syntax, data types, functions, and more with practical examples (<https://docs.python.org/3/tutorial/>)
3. [W3Schools Python Tutorial](https://www.w3schools.com/python/) – A beginner-friendly resource with interactive exercises and hands-on coding examples. It simplifies complex Python concepts, making it a great supplementary learning tool for practicing and reinforcing Python basics (<https://www.w3schools.com/python/>)
4. [ArcGIS Pro Python Reference](https://pro.arcgis.com/en/pro-app/arcpy/main/arcgis-pro-arcpy-reference.htm) – Essential for students interested in GIS applications, this reference explains how to use Python for automating tasks in ArcGIS Pro with the ArcPy library, enabling geospatial data analysis and automation (<https://pro.arcgis.com/en/pro-app/arcpy/main/arcgis-pro-arcpy-reference.htm>)
5. [Python for Non-Programmers](https://wiki.python.org/moin/BeginnersGuide/NonProgrammers) – A beginner's guide tailored for those with no prior programming experience. It includes easy-to-follow resources, explanations, and learning pathways to help newcomers grasp Python fundamentals (<https://wiki.python.org/moin/BeginnersGuide/NonProgrammers>)
6. [Python Language References](#) – A detailed, technical resource that provides formal language definitions, syntax rules, and advanced programming concepts. It is ideal for



understanding Python's core functionalities and debugging complex issues

(<https://docs.python.org/3/reference/index.html#reference-index>)

7. [Python for Programmers](#) – A guide specifically designed for experienced programmers transitioning to Python. It highlights key differences from other languages and provides concise explanations to quickly master Python's features and best practices (<https://wiki.python.org/moin/BeginnersGuide/Programmers>)

Web Forums:

1. [Stack Overflow](#) – A leading programming Q&A forum where Python learners can find solutions, debug code, and get expert advice on common errors, best practices, and advanced programming challenges (<https://stackoverflow.com/>)
2. [GIS Stack Exchange](#) – A specialized forum for GIS professionals using Python. Get answers on ArcPy, QGIS, spatial analysis, and geospatial data processing, making it essential for Python-based GIS applications (<https://gis.stackexchange.com/>)