INTRO TO PYTHON PROGRAMMING

Intro to Python Programming is a clear, beginner-friendly guide that helps you build real programming confidence through practical examples and step-by-step explanations. It takes you from core fundamentals to working with files, data structures, and common libraries, giving you the skills to write useful scripts and solve real-world problems with Python.

www.meerkat.pub

DR. KEVIN ROARK | 2026 EDITION

Book Overview

Intro to Python Programming is a comprehensive, student-friendly textbook designed for beginners with no prior coding experience. Whether you're pursuing a degree in Computer Science, aiming for a tech career, or simply exploring programming, this book provides a clear and engaging path to mastering Python.

Organized into eight focused modules, the textbook blends theory with hands-on learning. Students will explore Python fundamentals, data structures, file handling, web interaction, automation, and even an introduction to machine learning. Every concept is reinforced with real-world examples, practical demos, lab assignments, and video-style walkthroughs that bring the material to life. Each chapter is crafted to build confidence while promoting computational thinking and problem-solving skills.

With a strong emphasis on scripting, data manipulation, and practical coding techniques, Intro to Python Programming prepares students for both academic success and future careers in technology. This accessible, modern textbook supports a variety of learning styles and is ideal for classroom and self-paced use.

This digital textbook is complemented by over fifty professionally produced video tutorials that explicitly demonstrate Python syntax, clarify complex topics, and guide learners through comprehensive coding examples. These visual resources serve to enhance understanding and make programming more accessible to individuals from diverse backgrounds.

Throughout the course, learners engage in practical exercises, including scripting-focused labs, data analysis projects, and creative assignments tied to real-world themes. From processing text files and analyzing CSV data to web scraping, automating system tasks, and building simple data-driven applications, students gain hands-on experience that reinforces core Python concepts. These projects are designed not only to strengthen programming skills but also to build confidence in problem-solving, encouraging learners to approach challenges with curiosity and enthusiasm.

Key Features of Introduction to Python

Beginner-Friendly, Easy-to-Read Format. This textbook is thoughtfully written, clear, and purposeful, making it ideal for students who are just starting and have no prior programming experience. The book breaks down complex topics into simple, manageable sections to help you learn with confidence.

Includes Over 50 Video Walkthroughs. Each major topic is paired with engaging video demonstrations that guide students through coding examples, syntax explanations, and best practices. This approach helps turn abstract ideas into a clear, tangible understanding.

Flexible Delivery Options. Designed to support multiple teaching formats:

- Self-paced learning
- Instructor-facilitated courses
- Face-to-face classroom instruction
- Synchronous or asynchronous online delivery

Hands-On Labs and Creative Projects. Each module features themed lab assignments designed to reinforce concepts through engaging practice and creativity. This approach helps students connect with what they've learned in a meaningful and personal way.

Designed specifically for college students and adult learners. The book is thoughtfully organized to align with college-level learning outcomes. It's an excellent choice for academic programs and workforce training, providing support and relevance every step of the way.

Available online and as an iOS/Android app. Students can easily access the textbook on any device, via the web or the iOS/Android app. The platform remembers where they paused and visually shows their progress through each module, making learning more seamless and encouraging.

Who This Book Is For

- **First-Time Programmers -** Whether you're just starting college or exploring programming for the first time, this book assumes no prior experience and builds from the ground up.
- **Career Switchers and Industry Trainees -** Ideal for workforce training programs or individuals transitioning into tech, the book offers clear explanations, hands-on practice, and real-world context.
- **High School Dual Credit and College-Level Courses -** Designed to meet the needs of dual-credit programs and introductory college courses, supporting both academic rigor and student accessibility.

Course/Textbook Overview:

Module 1: Basics of Python

This module lays the foundation for your journey by introducing essential concepts that every programmer needs to understand. You'll start by learning how to write simple Python scripts and understand the structure and flow of a Python program. From there, you'll explore how to use comments to make your code readable and maintainable, and how indentation is more than just good style in Python - it's required for the code to run correctly.

Next, you'll dive into variables and data types, where you'll learn how Python stores and handles data like text, numbers, and more. You'll also be introduced to the math module, which adds powerful mathematical capabilities to your programs, and you'll practice user input to make your code interactive.

By the end of this module, you'll be writing your own basic scripts, using input, performing calculations, and getting comfortable with Python's syntax and logic. This is where your Python journey truly begins!

By the end of this module, you will be able to:

- Understand the purpose and structure of a basic Python program.
- Use comments and proper indentation to write clear, readable, and functional code.
- Declare and use variables to store different types of data, including integers, floats, and strings.
- Identify and use fundamental Python data types and perform basic type conversion.
- Utilize arithmetic and mathematical operations to perform calculations.
- Access and apply functions from Python's built-in math module.
- Gather user input using the input() function and display output using print().
- Understand the difference between = (assignment) and comparison operators.
- Recognize and use Python's identity and bitwise operators in simple contexts.
- Write and test small Python scripts that interact with users and perform basic data processing.

Module 2: Control Structures, Functions, and Basic Error Handling

In this module, you will learn how to write Python programs that make decisions, perform repeated actions, and handle basic errors gracefully. These capabilities are essential to creating interactive and reliable software. You'll start by exploring conditional logic with if, elif, and else statements, allowing your programs to behave differently based on user input or calculated values. Then, you'll move into loop structures that let your programs run blocks of code multiple times automatically.

You will also be introduced to exception handling, which allows you to anticipate and manage common errors like invalid user input or division by zero. Finally, you'll begin using functions to encapsulate logic into reusable blocks, laying the groundwork for more modular and scalable code.

Through a series of guided pages, hands-on demos, and creative labs, you'll gain practical experience writing your own decision-driven, loop-controlled, and error-resistant programs.

By the end of this module, you will be able to:

- Write and evaluate conditional expressions using if, elif, and else.
- Use comparison and logical operators to control program flow.
- Implement while and for loops to perform repetition.
- Apply a break and continue to control loop behavior.
- Write basic functions to organize and reuse code.
- Use try-except blocks to handle exceptions and prevent program crashes.
- Accept and validate user input to build interactive programs.
- Build small programs that incorporate decisions, loops, and functions in meaningful ways.

Module 3: Python Data Structures

In this module, you will explore Python's most commonly used data structures - lists, tuples, dictionaries, and sets - and learn how to use them effectively to store and manage collections of data. You'll practice working with these structures through real-world examples such as course schedules, inventory systems, and student groups. The second part of the module introduces file handling, where you'll gain hands-on experience reading from and writing to different types of files, including text files, CSV, JSON, and XML. This module provides foundational skills needed for data processing, application development, and automation tasks in Python.

By the end of this module, you will be able to:

- Describe and apply Python's core data structures: lists, tuples, dictionaries, and sets.
- Use indexing, slicing, and built-in methods to work with sequence data.
- Select appropriate data structures for solving different types of problems.
- Read from and write to text files using Python's file handling techniques.
- Load and parse data from structured file formats, including CSV, JSON, and XML.
- Apply set operations to solve problems involving grouping and comparison.
- Handle file-related errors gracefully using appropriate exception-handling techniques.
- Build Python programs that integrate in-memory data and external files to solve practical tasks.

Module 4: Object-Oriented Programming and Python Libraries

In this module, you'll explore how Python is used to model real-world systems, clean messy data, and analyze large datasets. You'll begin by learning the fundamentals of object-oriented programming (OOP), including how to design classes and create objects that represent things like students or employees. Then, you'll dive into data cleaning and transformation, using string methods and regular expressions to extract and prepare real-world data for analysis.

The final section introduces you to three of Python’s most powerful libraries for scientific and data-driven programming: NumPy, Pandas, and Matplotlib. You'll use NumPy for efficient numerical operations, Pandas for working with tabular data, and Matplotlib to create charts that help visualize trends and patterns.

Whether you're preparing for roles in software development, data analysis, or engineering, this module provides essential tools for working with data and structuring your programs efficiently.

By the end of this module, you will be able to:

- Design and implement custom Python classes using object-oriented principles.
- Extract and transform unstructured data using string methods and regular expressions.
- Explain and apply the core ideas behind ETL (Extract, Transform, Load) workflows.
- Use NumPy to perform fast, efficient numerical computations.
- Use Pandas to read, filter, and analyze data stored in CSV files or tabular form.
- Use Matplotlib to create professional data visualizations, including bar and pie charts.
- Build Python programs that integrate object-oriented code with real-world data processing and visualization tasks.

Module 5 System Administration Tasks

In this module, you’ll explore how Python can streamline and automate routine system administration tasks. From monitoring active processes and logging system usage to scheduling scripts and managing files, you’ll gain hands-on experience writing practical automation tools that work across platforms. You’ll also learn how to detect changes in the file system, use system commands in Python, and perform basic network tasks like IP and subnet calculations. This module provides foundational skills that are valuable for system administrators, DevOps engineers, and anyone responsible for managing computing environments.

By the end of this module, you will be able to:

- Explain how scripting with Python supports system administration and reduces manual workload.
- Use Python’s os, sys, and getpass modules to interact with user and environment data.
- Monitor and control system processes using the psutil library.
- Automate system-level tasks using subprocess to execute commands and schedule jobs.
- Detect file creation, deletion, and modification events using Python.
- Schedule and automate recurring tasks using platform-specific tools (e.g., cron, Task Scheduler).
- Perform fundamental network analysis using Python libraries to calculate IP ranges and subnet information.

Module 6 Scripting for Web Interaction and APIs

In this module, you'll learn how Python connects to the web to retrieve and process live online data. From requesting information from APIs to scraping content from websites, this module teaches you how to write powerful scripts that interact with real-world web sources. You'll begin by learning how HTTP works, how to send and receive data using Python's requests library, and how to parse structured responses using JSON. You'll also explore HTML structure and use BeautifulSoup to extract information from websites.

By completing hands-on demos and labs, you'll build tools to search for quotes, pull weather forecasts, scrape book prices, and more. The module also introduces key concepts in ethical web automation, including respecting robots.txt files, adhering to rate limits, and citing sources. Whether you're building a dashboard or exploring data for research, these skills are essential for modern scripting.

By the end of this module, you will be able to:

- Describe how Python communicates with web servers using HTTP methods such as GET and POST.
- Use the requests library to send HTTP requests and interpret response codes.
- Retrieve and parse data from web APIs using JSON.
- Extract structured information from websites using BeautifulSoup and HTML parsing.
- Apply responsible and ethical practices when accessing online data sources.

Module 7: Building and Testing Robust Scripts

In real-world programming, errors are inevitable - but crashes don't have to be. In this module, you'll learn how to write Python scripts that anticipate and handle problems gracefully. You'll explore Python's exception-handling system using try, except, else, and finally, and learn how to raise your own exceptions using custom classes. You'll also practice defensive programming, validate assumptions with assert, and use the logging module to monitor and debug your programs.

By the end of this module, you'll have the skills to write more reliable, maintainable, and professional Python code that's ready for production.

By the end of these sections, you will be able to:

- Use try, except, else, and finally blocks to manage exceptions in Python
- Create and raise custom exception classes for specific application needs
- Apply assertions to enforce conditions during development and debugging
- Implement defensive programming techniques to prevent and isolate errors
- Configure and use the logging module to track program behavior and diagnose issues
- Develop scripts that are resilient to invalid input, system failures, and logic errors

Module 8: Data Processing, Analysis, and an Introduction to Machine Learning

In today’s data-driven world, collecting, cleaning, analyzing, and interpreting data are essential skills for developers, analysts, and problem solvers. In this final module, you will learn how to use Python to prepare and analyze real-world datasets using the pandas library. You will also explore how to visualize insights using matplotlib, identify trends and correlations, and build your first machine learning models with scikit-learn. This module concludes with a discussion of ethical considerations, including data bias, privacy, and the responsible use of machine learning.

Through hands-on examples and guided labs, you'll gain practical experience with loading data from CSV files, transforming it into usable formats, identifying trends, creating visualizations, and making basic predictions with algorithms like K-Nearest Neighbors and Linear Regression.

By the end of this module, you will be able to:

- Load, inspect, and clean real-world data using the pandas library
- Handle missing values and remove duplicates to prepare datasets for analysis
- Apply transformations such as grouping, filtering, and sorting to gain insights
- Use summary statistics and correlation metrics to explore relationships in data
- Create visualizations using matplotlib to communicate data trends effectively
- Understand the difference between supervised and unsupervised machine learning
- Build and evaluate simple machine learning models using scikit-learn
- Split data into training and testing sets to validate predictions
- Identify common ethical issues in