01. Introduction

October 13, 2025

1 1. Introduction

Python is a powerful, versatile programming language known for its simplicity and readability. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, making it a popular choice for a wide range of applications, from web development and data science to automation and machine learning.

In this course, we will delve deeper into Python's more sophisticated features and libraries, building on your foundational knowledge to enhance your coding skills and expand your capabilities. Our comprehensive curriculum includes the following topics:

- Introduction
- Recap of Python data types - Understanding mutable vs. immutable types
- Control Flow - Advanced Control Flow Techniques
  - Using the `else` clause effectively
  - The walrus operator `:=`
  - Match expressions for pattern matching
- More on Functions and Iterables - Expanding Functional Knowledge
  - Functions with variable-length arguments
  - Variable unpacking
  - Comprehensions for lists, sets, and dictionaries
  - Iterators and generators
  - Anonymous functions (lambdas)
  - Built-in functions: `filter`, `map`, `enumerate`, `sorted`, `zip`
  - Advanced iteration with the `itertools` module
- Advanced Data Structures - Enhancing Data Handling
  - Advanced structures: `namedtuple` and `defaultdict`
  - Understanding and using `deque`, `OrderedDict`, and `Counter`
- Decorators - Advanced Function Manipulation
  - Understanding functions' roles and uses
  - Creating simple decorators
  - Implementing decorators with arguments
- Object-Oriented Programming - Deepening OOP Skills
  - Dunder (magic) methods and attributes, including operator overloading
  - Static and class methods
  - Getter/Setter methods and the property class
  - Introduction to metaclasses
  - Common design patterns in Python
- Context Managers - Efficient Resource Management

  - Using the `with` statement
  - Creating custom context managers with the `contextlib` module
- Concurrent Execution - Optimizing Performance
  - Using the `threading` module
  - Utilizing the `multiprocessing` module
- Communicating with external processes
  - The `subprocess` module