30 Days Python Learning Plan

30-day learning plan to get you started with learning Python

Week 1: Getting Started

Day 1-3: Introduction and Setup

Day 1: Introduction to programming concepts and Python's role.

Day 2: Installing Python, choosing an IDE (like Visual Studio Code), and exploring basic IDE features.

Day 3: Writing and running your first "Hello, World!" program.

Day 4-6: Variables, Data Types, and Operators

Day 4: Learn about variables and different data types (integers, floats, strings, booleans).

Day 5: Explore basic operators (+, -, *, /, %) and their usage.

Day 6: Dive into more advanced operators (// for floor division, ** for exponentiation).

Week 2: Building Foundations

Day 7-9: Control Structures

Day 7: Understand if statements and conditional branching.

Day 8: Learn about loops: for and while.

Day 9: Practice using loops and conditional statements.

Day 10-12: Lists and Tuples

Day 10: Introduction to lists: creating, indexing, and slicing.

Day 11: Learn about tuples and their immutability.

Day 12: Practice with list and tuple manipulation.

Week 3: Functions and File Handling

Day 13-15: Functions

Day 13: Introduction to functions: defining, calling, and returning values.

Day 14: Function parameters and scope.

Day 15: Practice creating and using functions effectively.

Day 16-18: File Handling and Input/Output

Day 16: Learn how to read from and write to files.

Day 17: Explore standard input and output methods.

Day 18: Combine file handling and input/output concepts in a small project.

Week 4: Object-Oriented Programming (OOP)

Day 19-21: Introduction to OOP

Day 19: Understand the core principles of Object-Oriented Programming.

Day 20: Learn about classes, objects, attributes, and methods.

Day 21: Practice creating simple classes and objects.

Day 22-24: Advanced OOP Concepts

Day 22: Dive deeper into inheritance and polymorphism.

Day 23: Explore encapsulation and access modifiers.

Day 24: Apply advanced OOP concepts in a project.

Week 5: Intermediate Concepts and Projects

Day 25-27: Error Handling and Modules

Day 25: Learn about exceptions and how to handle errors.

Day 26: Explore modules, libraries, and importing functions.

Day 27: Practice error handling and using external modules.

Day 28-30: Basic Data Manipulation and Projects

Day 28: Introduction to NumPy for numerical computations.

Day 29: Introduction to Pandas for data analysis.

Day 30: Work on a small project that incorporates data manipulation