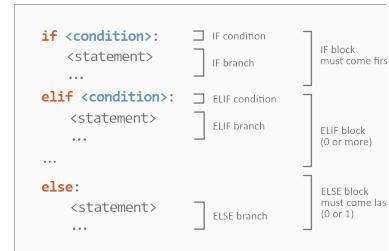
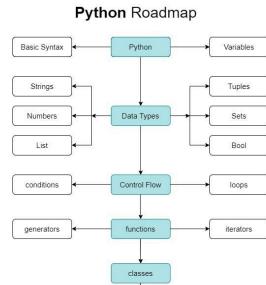
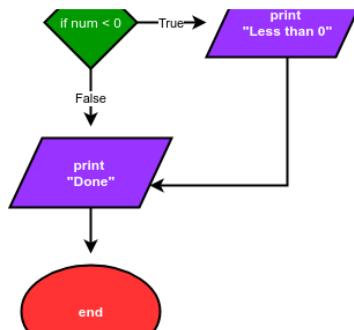


100 Days of Code — Python (Day-by-Day Plan)

Beginner → Advanced Python • OOP • DSA • Algorithms • Projects

★ PHASE 1 — Python Basics (Day 1–20)



DAY 1–5: Introduction & Core Syntax

Day 1 — Installation & Setup

- Install Python, IDE (VS Code / PyCharm)
- Learn how `python` runs scripts
- Explore Python docs: <https://docs.python.org/3/tutorial/>

Day 2 — Variables, Data Types

- `int`, `float`, `bool`, `str`, type conversion
- Practice: Build a simple greeting program

Day 3 — Operators

- Arithmetic, comparison, logical, assignment
- Mini-task: Expression evaluator

Day 4 — Strings

- Indexing, slicing, built-in methods
- Practice: Reverse a string, count vowels

Day 5 — Conditionals

- `if`, `elif`, `else`
- Mini-project: Simple grading system

DAY 6–10: Loops, Functions, Data Structures

Day 6 — Loops

- `for`, `while`, `break`, `continue`
- Task: Print patterns

Day 7 — Lists

- Methods: `append`, `pop`, `sort`, slicing
- Task: Build menu-based list editor

Day 8 — Tuples, Sets

- Tuple immutability
- Set operations
- Task: Remove duplicates from list

Day 9 — Dictionaries

- Keys, values, nested dict
- Task: Create a phonebook dictionary

Day 10 — Functions

- Parameters, return values
- Practice: Calculator functions

DAY 11–15: Intermediate Concepts

Day 11 — Scope, Recursion

- `global`, recursion basics

Day 12 — Error Handling

- `try`, `except`, `finally`
- Task: Safe input converter
- Reference: <https://docs.python.org/3/tutorial/errors.html>

Day 13 — File Handling

- Read/Write files
- Task: Notes app (store text)

Day 14 — Modules & Packages

- `import`, `from xyz import`
- Standard library tour
- Reference: <https://docs.python.org/3/library/>

Day 15 — Virtual Environments

- `venv`, `pip`

- Create your first real project folder structure

DAY 16–20: Foundation Projects

Day 16 — Number Guessing Game

Day 17 — To-Do App (CLI)

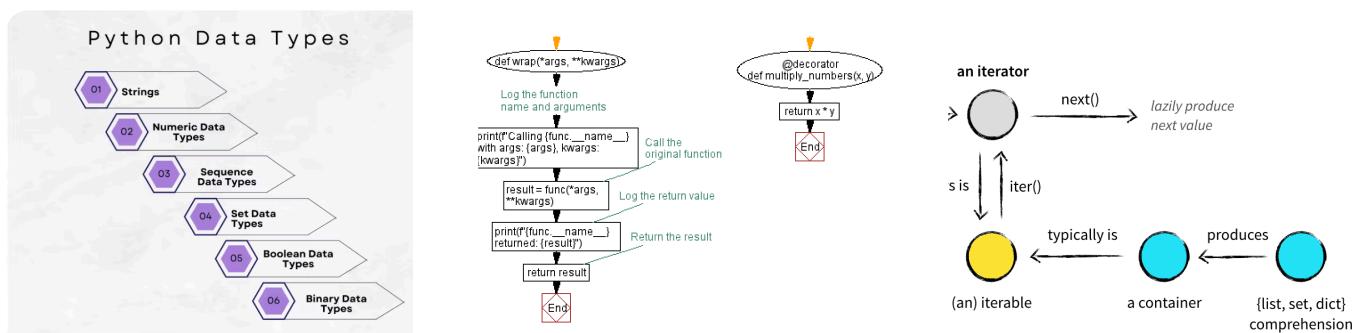
Day 18 — Rock Paper Scissors

Day 19 — Text Analyzer

Day 20 — Beginner Project Day (Choose 1)

- Password generator
- Simple contact manager

★ PHASE 2 — Intermediate Python (Day 21–40)



DAY 21–25: More Python Power

Day 21 — List Comprehensions

Day 22 — Lambda, map, filter, reduce

Day 23 — Iterators

Day 24 — Generators

Day 25 — Decorators (Beginner → Intermediate)

Reference for functions: <https://docs.python.org/3/tutorial/controlflow.html#define-functions>

DAY 26–30: Standard Libraries

Day 26 — `os`, `sys`

Day 27 — `json`, `csv`

Day 28 — `datetime`

Day 29 — Regex (`re`)

Day 30 — API Calls (`requests`)

Mini Project: Weather App using API

DAY 31–35: Applied Python

Day 31 — Automation Basics

Day 32 — Command-line arguments

Day 33 — Logging

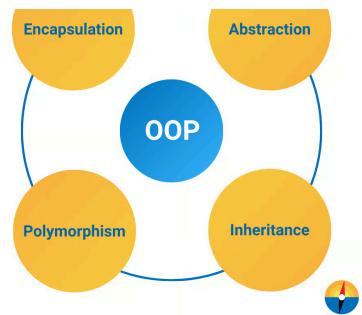
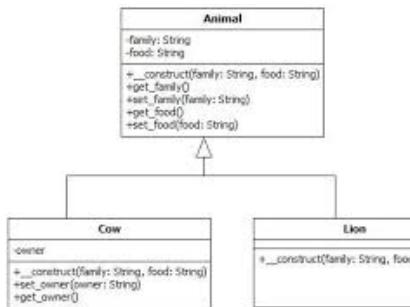
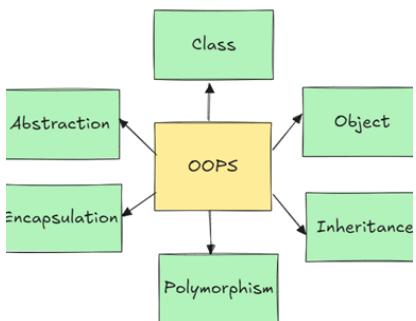
Day 34 — Packaging a small script

Day 35 — Build: Markdown → HTML Converter

DAY 36–40: Intermediate Projects

- Expense Tracker
- Email Sender
- Web Scraper
- JSON-based Database App
- API Tester Tool

★ PHASE 3 — OOP: Object-Oriented Programming (Day 41–55)



Reference Docs: <https://docs.python.org/3/tutorial/classes.html>

Day 41 — Classes & Objects

Day 42 — Instance vs Class Attributes

Day 43 — Methods: instance, class, static

Day 44 — Encapsulation

Day 45 — Inheritance

Day 46 — Polymorphism

Day 47 — Abstraction

Day 48 — Magic/Dunder Methods

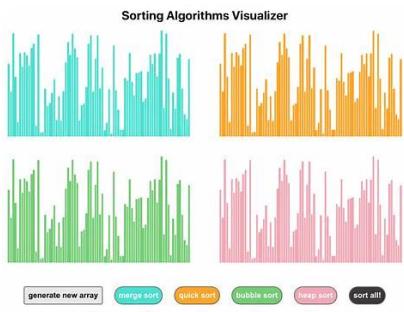
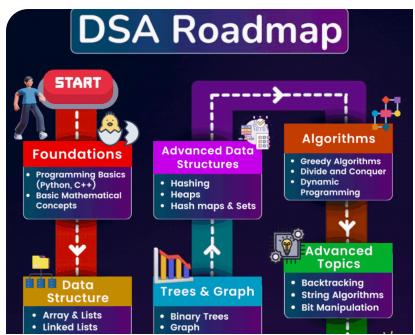
Day 49 — OOP Principles (SOLID)

Day 50 — Design Patterns (Intro)

DAY 51–55: OOP Mini Projects

- Bank Management System
- Library Management System
- Student Record App
- Inventory Management
- Quiz Application

★ **PHASE 4 — DSA + Algorithms (Day 56–80)**



Inorder Traversal	4	2	5	1	6	3	7
Preorder Traversal	1	2	4	5	3	6	7
Postorder Traversal	4	5	2	6	7	3	1
Level order Traversal	1	2	3	4	5	6	7

DAY 56–60: Core Concepts

Day 56 — Arrays

Day 57 — Strings

Day 58 — Linked Lists

Day 59 — Stacks & Queues

Day 60 — Hashmaps (Dictionaries Internals)

DAY 61–70: Trees, Graphs, Recursion

Day 61 — Trees

Day 62 — Binary Trees (Traversals)

Day 63 — Binary Search Tree (BST)

Day 64 — Heaps & Priority Queues

Day 65 — Graph Basics

Day 66 — BFS & DFS

Day 67 — Recursion Deep Dive

Day 68 — Backtracking

Day 69 — Greedy Algorithms

Day 70 — Dynamic Programming (Intro → Medium)

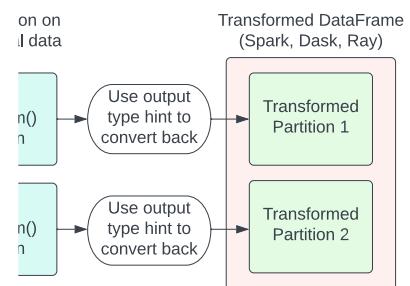
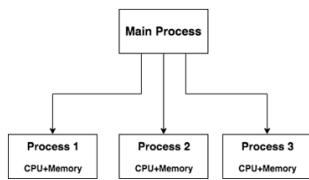
DAY 71–80: DSA Projects

- Implement your own Linked List
- Implement Stack & Queue using OOP
- Path Finder (BFS/DFS)
- Maze Solver (Backtracking)
- Trie Auto-suggest System
- Sorting Visualizer
- Mini LeetCode Challenge (10 Problems)

★ PHASE 5 — Advanced Python (Day 81–90)



Multiprocessing



Day 81 — Context Managers (with)

Day 82 — Closures

Day 83 — Type Hints (typing module)

Day 84 — Multithreading

Day 85 — Multiprocessing

Day 86 — AsyncIO

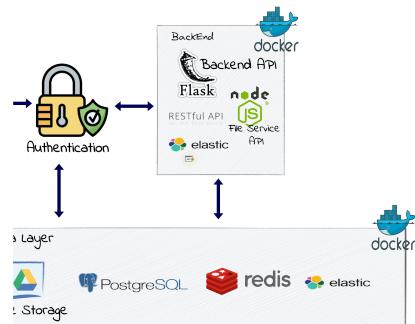
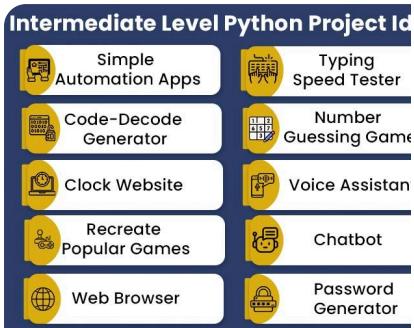
Day 87 — Python Optimization

Day 88 — Working with Databases

Day 89 — Building Python Packages

Day 90 — CLI Tools & Distribution

★ PHASE 6 — Capstone Projects (Day 91–100)



Choose 3–5 major projects for portfolio:

Web Development (Flask / FastAPI)

- Blog Website
- Authentication System
- API Backend

Automation

- Email Automation System
- File Organizer
- Auto Backup Tool

Data & Machine Learning

- CSV Analyzer
- Movie Recommendation (basic ML)
- Data Dashboard (Streamlit)

Final Days

- Day 96: Debug + Optimize
- Day 97: Documentation
- Day 98: Deploy (Render / Railway / GitHub Pages)
- Day 99: Portfolio Preparation
- Day 100: Celebrate + Write your journey