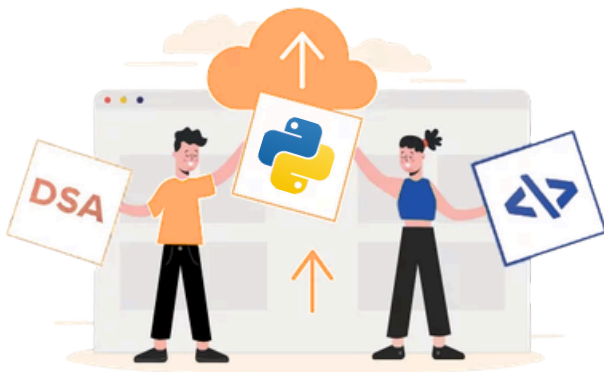


PYTHON + DSA

From **Beginner** to **Interview-Ready**
In Just **16 Weeks**



PHASE 1

PYTHON FOUNDATION

WEEK 1 — Python Basics

Setup, syntax, variables, data types, I/O, operators

Practice: 25–30 basic programs

WEEK 2 — Control Flow

Conditions, loops, range, break/continue

Practice: 30–40 logic programs

WEEK 3 — Strings & Collections

Strings, lists, tuples, sets, dictionaries

Practice: 35–40 collection problems

WEEK 4 — Functions & Essentials

Functions, recursion, lambda, comprehensions, files, exceptions

Practice: 30–35 problems + mini project

PHASE 2

PYTHON FOR DSA

WEEK 5 — OOP

Classes, objects, inheritance, polymorphism, encapsulation

Outcome: Structured Python programs

WEEK 6 — DSA Foundations

Big-O, recursion, iterators, generators, collections, heapq

Outcome: Ready for formal DSA

PHASE 3

CORE DSA

WEEK 7 — Arrays & Lists

Concepts: Traversal, insertion, deletion, prefix sums, two pointers

WEEK 8 — Searching & Sorting

Concepts: Linear/binary search, bubble, selection, insertion, merge, quick sort

WEEK 9 — Strings & Hashing

Concepts: Frequency maps, sets, dictionaries, anagrams

WEEK 10 — Linked Lists

Concepts: Singly, doubly, circular, reverse, cycle detection

WEEK 11 — Stack & Queue

Concepts: Stack, queue, deque, parentheses, monotonic stack

WEEK 12 — Recursion & Backtracking

Concepts: Subsets, permutations, combinations, N-Queens

PHASE 4

ADVANCED DSA

WEEK 13 — Trees & BST

Concepts: Traversals, height, depth, search/insert/delete

WEEK 14 — Heap & Priority Queue

Concepts: Min/max heap, Top-K, priority problems

WEEK 15 — Graphs

Concepts: BFS, DFS, adjacency list/matrix, components

WEEK 16 — Dynamic Programming + Revision

Concepts: Memoization, tabulation, 1D/2D DP, interview revision