

Master Python in 30 Days: Full Roadmap

Week 1: Python Fundamentals

Day 1: Introduction to Python

- What is Python & why use it
- Install Python & VS Code
- Hello World & print()
- Python syntax & comments

Day 2: Variables & Data Types

- Variables
- Strings, Integers, Floats, Booleans
- Type casting
- type() function

Day 3: Operators & Input

- Arithmetic, Comparison, Logical, Assignment
- Taking input from user
- f-strings and formatting

Day 4: Conditional Statements

- if, elif, else
- Nested conditions
- Short-hand if-else

Day 5: Loops

- for and while loops
- break, continue, range()
- Nested loops

Day 6: Data Structures I

- Lists

- Tuples
- Basic operations, slicing

Day 7: Data Structures II

- Dictionaries
- Sets
- Common methods & loops with structures

Week 2: Intermediate Python + Functions

Day 8: Functions

- Defining functions
- Parameters & return
- Default, *args, **kwargs

Day 9: Scope & Recursion

- Local vs global
- global keyword
- Recursive functions

Day 10: String Mastery

- String methods
- Slicing, formatting, in, not in
- join(), split(), replace(), etc.

Day 11: List & Dictionary Comprehensions

- Writing concise loops
- Nested comprehensions

Day 12: Error Handling

- Try, Except
- Finally, Else
- Raising errors

Day 13: File Handling

- open(), read(), write()
- Modes: r, w, a, r+
- Working with files safely

Day 14: Modules & Packages

- import, from, as
- Built-in modules: random, math, os
- Creating your own modules

Week 3: OOP + Advanced Topics

Day 15: OOP Basics

- Classes and Objects
- __init__ constructor
- Instance vs class variables

Day 16: OOP Deep Dive

- Inheritance
- Encapsulation
- Polymorphism

Day 17: Lambda, Map, Filter, Reduce

- Lambda functions
- map(), filter(), reduce()
- When and why to use them

Day 18: Iterators and Generators

- __iter__, __next__
- yield, generator functions
- Memory efficiency

Day 19: Decorators

- Function decorators
- Chaining decorators
- Use-cases

Day 20: Working with JSON & APIs

- JSON module
- requests module (GET, POST)
- API call basics

Day 21: Virtual Environments & PIP

- venv, pip install, requirements.txt
- Working on isolated projects

Week 4: Real-World Python Applications

Day 22: Web Scraping

- requests, BeautifulSoup
- Parsing HTML, extracting data

Day 23: Automation with Python

- Automate files, folders
- os, shutil, datetime

Day 24: Introduction to Tkinter GUI

- Creating basic GUI apps
- Buttons, Labels, Inputs

Day 25: Working with CSV & Excel

- csv module
- pandas for reading/writing Excel

Day 26: Data Analysis with Pandas

- Series, DataFrames
- read_csv, filtering, sorting

Day 27: Data Visualization

- matplotlib basics
- Plotting graphs, labels, titles

Day 28: Project Day 1

- Build a mini project: To-Do App, Web Scraper, or Weather App using API

Day 29: Project Day 2

- Add enhancements to your project
- Make it user-friendly

Day 30: Review & GitHub

- Revise all concepts
- Push your project to GitHub
- Write a README file