



Python Mastery Roadmap (2025 Edition)

✓ 1. Python Basics

- Introduction & Syntax
 - Variables & Data Types
 - Input/Output (`input()`, `print()`)
 - Comments (single-line & multi-line)
 - Type Conversion (`int()`, `str()`, `float()`)
-

✓ 2. Operators & Expressions

- Arithmetic, Assignment, Comparison
 - Logical (`and`, `or`, `not`)
 - Identity (`is`, `is not`) and Membership (`in`, `not in`)
 - Bitwise Operators
-

✓ 3. Control Flow

- `if`, `elif`, `else`
 - Nested `if`
 - `match-case` (Python 3.10+ modern switch-case)
-

✓ 4. Loops

- `for` loops
 - `while` loops
 - `break`, `continue`, `pass`
 - Loop `else` block
 - Nested loops
 - `enumerate()`, `range()`
-

✓ 5. Data Structures

● Strings

- Indexing, slicing, formatting (f-strings)
- String methods (`split()`, `join()`, `replace()`, `strip()`)

● Lists

- Indexing, slicing
- List methods (`append()`, `extend()`, `insert()`, `pop()`, `remove()`, `sort()`)
- List comprehension

● Tuples

- Immutable data
- Tuple unpacking

● Sets

- Unique elements, set operations (`union`, `intersection`, `difference`)

● Dictionaries

- Key-value access
 - Methods (`items()`, `keys()`, `values()`, `get()`, `update()`)
-

✓ 6. Functions

- Defining and calling functions
 - Parameters & arguments
 - Default, keyword, variable-length arguments (`*args`, `**kwargs`)
 - `return` statement
 - Scope and `global`
-

✓ 7. Functional Programming

- `lambda`
 - `map()`, `filter()`, `reduce()`
 - `zip()`, `any()`, `all()`
-

✓ 8. Error Handling

- `try`, `except`, `else`, `finally`
- Built-in exceptions (`ValueError`, `TypeError`, etc.)
- `raise` keyword

✓ 9. File Handling

- Reading & writing files
- Modes: `'r'`, `'w'`, `'a'`, `'x'`, `'b'`
- Working with `.txt`, `.csv`, `.json`

✓ 10. Object-Oriented Programming (OOP)

- Classes & Objects
- `__init__()` method
- `self` keyword
- Instance vs class variables
- Inheritance
- Method overriding
- Encapsulation & Abstraction
- `@classmethod`, `@staticmethod`
- `super()`

✓ 11. Modules & Packages

- `import` statements
- `__name__ == '__main__'`

- Creating your own module
 - Python Standard Library overview
-

✓ 12. Python Advanced Concepts

- `*args`, `**kwargs` in-depth
 - Decorators
 - Generators (`yield`)
 - Iterators
 - Context Managers (`with` statement, `__enter__`, `__exit__`)
 - `dataclasses` (Python 3.7+)
 - `typing` module (type hints, `List[str]`, `Dict[str, int]`, etc.)
-

✓ 13. Python 3.10+ Modern Features

- `match-case` pattern matching
 - Structural pattern matching
 - The Walrus Operator (`:=`)
 - `zoneinfo` (Python timezone handling)
 - New typing improvements (`TypeAlias`, `Literal`, `Final`, `Union`)
-

✓ 14. Testing & Debugging

- Writing unit tests using `unittest`
 - Assertions
 - Debugging with `pdb`
 - Linting with `flake8` / `pylint`
-

✓ 15. Real-World Projects (Optional after each milestone)

- **Student Record System** (after loops, dicts)
- **To-Do List CLI App** (functions, file handling)
- **Quiz App** (OOP, random, score tracking)
- **Budget Tracker** (file I/O, classes)
- **Basic Contact Manager** (search, sort, save to file)
- **Resume Parser** (advanced string/dict parsing)
- **Weather App using API** (modules, requests)
- **Portfolio Website Backend (Flask/Django)** (when web dev begins)