6-Month Python Daily Checklist (Printable)

Daily Rule (Tick all):

☐ Studied concept (30–45 min)
☐ Wrote code manually (90–120 min)
☐ No copy-paste without understanding
☐ Notes updated

🔹 MONTH 1 – Foundations (Days 1–30)

Week 1 – Basics

☐ Day 1: Install Python, VS Code, print(), comments
☐ Day 2: Variables & data types
☐ Day 3: Operators
☐ Day 4: Input & type casting
☐ Day 5: if / elif / else
☐ Day 6: Practice problems (calculator, grading)
☐ Day 7: Rewrite all from memory

Week 2 – Loops & Logic

☐ Day 8: for loop
☐ Day 9: while loop
☐ Day 10: break / continue / pass
☐ Day 11: Nested loops
☐ Day 12: Pattern problems
☐ Day 13: Mini project – Number guessing game
☐ Day 14: Debug & refactor

Week 3 – Data Structures

☐ Day 15: Lists
☐ Day 16: List methods & slicing

☐ Day 17: Tuples & Sets
☐ Day 18: Dictionaries
☐ Day 19: Dictionary problems
☐ Day 20: Mixed DS problems
☐ Day 21: Mini project – Student record system

Week 4 – Functions & Files

☐ Day 22: Functions
☐ Day 23: Parameters & return
☐ Day 24: Scope (local/global)
☐ Day 25: File handling
☐ Day 26: try / except
☐ Day 27: Mini project – File-based To-Do app
☐ Day 28: Revise basics
☐ Day 29: Rewrite all projects
☐ Day 30: Cleanup & GitHub push

🔹 MONTH 2 – Core Python (Days 31–60)

OOP (Critical)

☐ Day 31: Why OOP
☐ Day 32: Classes & objects
☐ Day 33: Constructors
☐ Day 34: Methods & attributes
☐ Day 35: Encapsulation
☐ Day 36: Inheritance
☐ Day 37: Polymorphism
☐ Day 38: Banking system – design
☐ Day 39: Banking system – code
☐ Day 40: Banking system – refactor

Modules & Environment

☐ Day 41: Modules
☐ Day 42: Packages
☐ Day 43: Virtual environments
☐ Day 44: pip & libraries
☐ Day 45: Library system – design
☐ Day 46: Library system – code
☐ Day 47: Library system – cleanup

Data Handling

☐ Day 48: CSV
☐ Day 49: JSON
☐ Day 50: Date & time
☐ Day 51: Logging
☐ Day 52: Expense tracker – design
☐ Day 53: Expense tracker – code
☐ Day 54: Expense tracker – test
☐ Day 55–60: Full revision & refactoring

🔹 MONTH 3 – Advanced Python (Days 61–90)

Pythonic Features

☐ Day 61: List comprehensions
☐ Day 62: Dict & set comprehensions
☐ Day 63: Lambda functions
☐ Day 64: map / filter / reduce
☐ Day 65: Iterators & generators

Deep Concepts

☐ Day 66: Decorators
☐ Day 67: *args & **kwargs
☐ Day 68: Regular expressions
☐ Day 69: Memory & references
☐ Day 70: Garbage collection

APIs & Automation

☐ Day 71: HTTP basics
☐ Day 72: requests library
☐ Day 73: API authentication
☐ Day 74: Web scraping
☐ Day 75: Scraper project – design
☐ Day 76: Scraper – code
☐ Day 77: Scraper – analysis
☐ Day 78: Scraper – optimize
☐ Day 79–90: Rewrite & testing

🔹 MONTH 4 – Software-Level Python (Days 91–120)

Testing & Debugging

☐ Day 91: Debugging strategies
☐ Day 92: Unit testing
☐ Day 93: Writing test cases

Databases

☐ Day 94: SQL basics
☐ Day 95: SQLite with Python
☐ Day 96: CRUD operations
☐ Day 97: DB project – design
☐ Day 98: DB project – code

☐ Day 99: DB project – test
☐ Day 100: DB project – refactor

Performance & Concurrency

☐ Day 101: Time complexity
☐ Day 102: Profiling
☐ Day 103: Multithreading
☐ Day 104: Multiprocessing
☐ Day 105: Automation project – design
☐ Day 106–110: Automation – code
☐ Day 111–120: Cleanup + GitHub

🔹 MONTH 5 – Specialization (Days 121–150)

☐ Day 121: Choose ONE specialization
☐ Day 122–130: Learn core tools
☐ Day 131–140: Major project
☐ Day 141–145: Optimization
☐ Day 146–150: Documentation & testing

🔹 MONTH 6 – Advanced & Job-Ready (Days 151–180)

☐ Day 151–155: Capstone project – design
☐ Day 156–160: Capstone – build
☐ Day 161–165: Security & error handling
☐ Day 166–170: Performance tuning
☐ Day 171–175: Portfolio & GitHub
☐ Day 176–180: Interview prep & revision