Certified Python Course

100% Practical Training | Notes Included | Certificate

Our Website
www.softprocomp.com

SOFTPRO
DMTI
PROFESSIONAL
CERTIFIED

9833900330

Hands-On Learning Experience

Industry-Relevant Curriculum

Career Support & Certification

Flexible & Accessible LMS

python

Certified Python Programming Course

– From Basics to 20+ Projects LEVEL 1

🚀 **100% Practical Training | Notes Included | Certificate |**

About the Course

Learn Python from scratch with step-by-step practical training. This course is specially designed for students, job seekers, and working professionals.

Why Choose DMTI SOFTPRO?

✓ 34+ Years of Training Experience

✓ 100% Practical Learning Approach

✓ Live Projects & Case Studies

✓ 100% Placement Assistance

✓ Expert Trainers

✓ Offline + Online Training Available

Course Highlights

✓ Beginner to Level 1

✓ 20 Real Projects Included

✓ GUI Applications

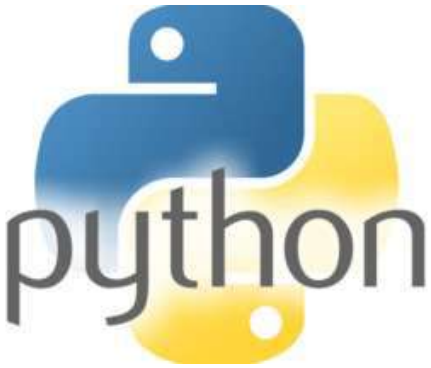
✓ File Handling & Automation

COMPLETE COURSE SYLLABUS ↘ ↘ ↘ ↘ ↘

🔹 Module 1 — Introduction to Python (Session 1–2)

🟢 Session 1 — Python Basics

- What is Python?
- Why learn Python?
- Applications of Python
- History of Python
- Installing Python (step-by-step)
- Setting up IDLE / VS Code



🟢 Session 2 — First Program & Fundamentals

- Writing first Python program
- print() function
- Python identifiers
- Variables introduction
- Basic data types (int, float, string)

🔹 Module 2 — Python Syntax & Data Types (Session 3–5)

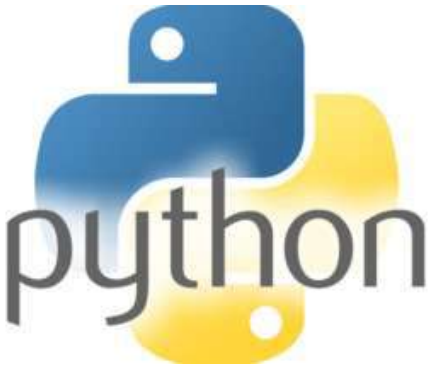
🟢 Session 3 — Syntax & Variables

- Lines and indentation
- Multi-line statements
- Assigning values to variables
- Multiple assignment
- Naming rules

🟢 Session 4 — Data Types Deep Dive

- Strings
- Lists
- Tuples
- Type conversion

🟢 Session 5 — Basic Operations on Data



- String operations
- List operations
- Tuple basics
- Practical examples

🔹 Module 3 — Operators & Decision Making (Session 6–7)

🟢 Session 6 — Operators

- Arithmetic operators
- Comparison operators
- Assignment operators
- Logical operators
- Membership & identity operators

🟢 Session 7 — Decision Making

- If statements
- If-else statements
- Elif statements
- Nested conditions
- Inline conditions

🔹 Module 4 — Loops & Sequences (Session 8–9)

🟢 Session 8 — Loops

- What are loops?
- For loop
- While loop
- Break & Continue
- Loop with else

🟢 Session 9 — Sequences & Comprehension

- Strings as sequences
- Indexing & slicing
- Negative indexing
- List comprehension

🔹 Module 5 — Collections & Data Structures (Session 10–12)

🟢 Session 10 — Lists Advanced

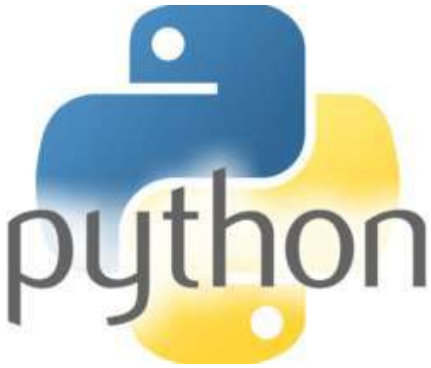
- Creating lists
- Updating lists
- Deleting elements
- Sorting & operations
- Matrix (2D lists)

🟢 Session 11 — Tuples & Sets

- Tuple operations
- Tuple vs list
- Sets introduction
- Set operations

🟢 Session 12 — Dictionaries

- Key-value structure
- Adding / updating values
- Deleting elements
- Built-in dictionary functions



🔹 Module 6 — Strings Mastery (Session 13)

🟢 Session 13 — Strings Advanced

- String indexing
- String formatting (f-string)
- String methods
- Unicode strings

🔹 Module 7 — Date, Time & Functions (Session 14–16)

🟢 Session 14 — Date & Time

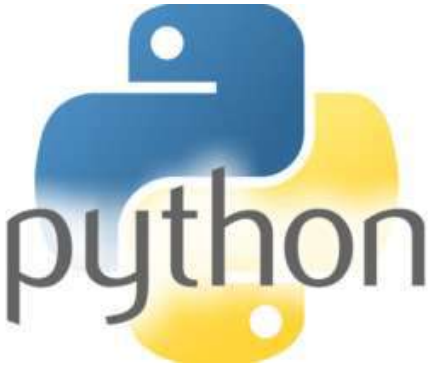
- time module
- Getting current time
- Formatting time
- Calendar module

🟢 Session 15 — Functions Basics

- Defining functions
- Calling functions
- Parameters & arguments
- Return statement

🟢 Session 16 — Advanced Functions

- Default arguments
- Keyword arguments
- Variable-length arguments
- Lambda functions Global vs local variables



🔹 Module 8 — Modules & File Handling (Session 17–18)

🟢 Session 17 — Modules

- Import statement
- from import
- Built-in modules
- Installing packages (pip)

🟢 Session 18 — File Handling

- Opening files
- Reading & writing files
- File modes
- Working with directories

🔹 Module 9 — Exception Handling (Session 19)

🟢 Session 19 — Error Handling

- What is exception?
- try-except
- try-finally
- Raising exceptions

🔹 Module 10 — OOP Concepts (Session 20–21)

🟢 Session 20 — Classes & Objects

- What are classes?
- Creating objects
- Constructor (**init**)

🟢 Session 21 — Inheritance

- Parent & child classes
- Reusing code
- Practical examples

🔹 Module 11 — Projects (Session 22–30)

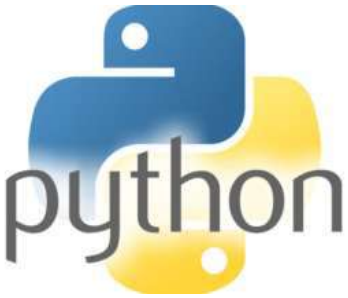
🟢 Session 22–23 — Basic Projects

- Calculator
- Grade Tracker
- Guess Game

🟢 Session 24–25 — Logic Projects

- To-Do List
- Contact Manager
- Task Manager

🟢 Session 26–27 — Intermediate Projects

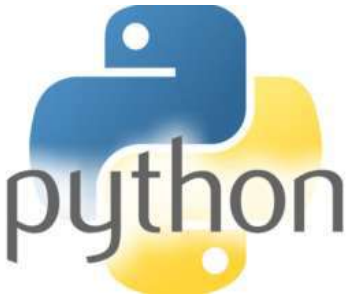


- Chatbot
- Text Analyzer
- Currency Converter

🟢 Session 28–30 — GUI Projects

- GUI Calculator
- GUI To-Do App
- GUI Weather App
- GUI Quiz App
- Expense Tracker

🎯 **Final Outcome for Students:** After completing this course, students will:



- ✓ Write Python programs confidently
- ✓ Build real-world applications
- ✓ Understand logic & coding structure
- ✓ Create GUI applications
- ✓ Work on automation & small AI logic
- ✓ Be job-ready / freelance-ready

Career Opportunities

- ✓ Python Developer
- ✓ Data Analyst
- ✓ Automation Engineer
- ✓ Freelancer

Our Locations

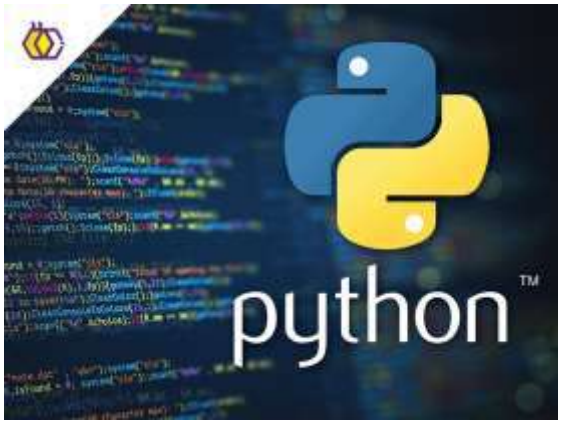
Mumbai – Andheri | Dadar | Matunga | + Online

Special Offer

- 🎉 Fees Original 9990/ -
- 🎉 **Further 👆 SAVE 50%**
- 🎉 Limited Time Discount Available
- 🎉 Includes Notes, Printed Book, GST, Certificate

Enroll Now

Call / WhatsApp: 9833900330 / 9833900990



💼 Future Career Scope

- AI Eng. & Data Science
- Software & App Development
- Automation & Robotics
- Cyber Security
- Freelancing opportunities

🤖 ADD this Module — AI with ChatGPT for Python

🎯 Learn Smart Coding using AI ChatGPT (Session 31–33)

🔹 Session 31 — Introduction to AI + ChatGPT for Python

🟢 Understanding AI & ChatGPT

- What is Artificial Intelligence?
- What is ChatGPT?
- How ChatGPT helps in coding
- Real-world use cases (students, developers, freelancers)

🟢 Using ChatGPT for Python Basics

- How to ask questions to ChatGPT (Prompting basics)
- Generating simple Python programs
- Understanding code explanations from ChatGPT
- Writing beginner programs using ChatGPT

🟢 Practical Activity

- Ask ChatGPT to create:
  - Calculator program
  - Even/odd checker
  - Simple loop program

Session 32 — Debugging & Improving Code using ChatGPT

Debugging with ChatGPT

- What is debugging?
- Finding errors in Python code
- Using ChatGPT to fix errors
- Understanding error messages

Improving Code Quality

- Optimizing code using ChatGPT
- Making code shorter & cleaner
- Converting basic code into advanced version

Practical Activity

- Fix errors in:
  - Calculator program
  - List program
  - File handling program

🔹 Session 33 — Advanced Use of ChatGPT with Python

🟢 Writing Projects using ChatGPT

- Generate project ideas using ChatGPT
- Build:
  - To-Do App
  - Chatbot
  - Mini automation scripts

🟢 Prompt Engineering for Python

- What is a good prompt?
- Examples of strong prompts:
  - “Write a Python program for…”
  - “Explain this code step-by-step”
  - “Debug this code”

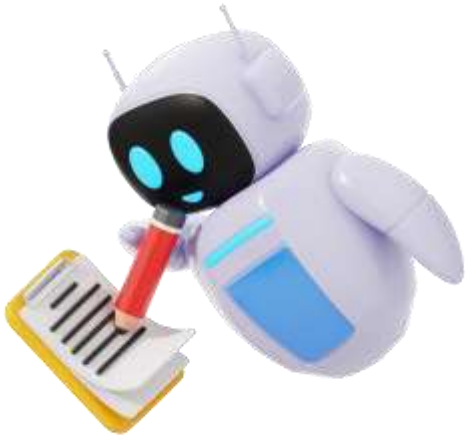
🟢 Final Activity

- Students will:
  - Create 1 project using ChatGPT
  - Debug it
  - Improve it
  - Present it

🎯 Outcome of This Module

Students will be able to:

- ✓ Use ChatGPT to write Python code
- ✓ Debug errors quickly
- ✓ Improve coding speed
- ✓ Build projects faster
- ✓ Work like a smart developer (AI-powered)



Fees: Save 50%

Extra Module Of ChatGPT with Prompt Engineering Python

Special Offer – Call the Institute for More Offers

💼 Future Career Scope

- AI Eng. & Data Science
- Software & App Development
- Automation & Robotics
- Cyber Security
- Freelancing opportunities