# Mentor - Himanshu Gupta Contact - > 9532696131

## ADVANCED AI WITH PYTHON

**Unit 1: Introduction to Python (2 days)**  
• Introduction to Python  
• Why Python?  
• Features of Python  
• Python Software Installation  
• Python Program Syntax  
• Case Study

**Unit 2: Basic Python (2 days)**  
• The print() Statement  
• Comments in Python  
• Python Data Structures and Data Types  
• String Operations in Python  
• Simple Input and Output  
• Practice: Real-World Non-AI and AI Questions, Case Study

**Unit 3: Operators in Python (2 days)**  
• Arithmetic Operators  
• Assignment Operators  
• Boolean Operators  
• Bitwise Operators  
• Logical Operators  
• Practice: Real-World Non-AI and AI Questions

**Unit 4: Control Flow Statements (2 days)**  
• if Statement  
• if-else Statement  
• elif (Else If) Statement  
• Nested if Conditions  
• Practice: Real-World Non-AI and AI Questions

**Unit 5: Data Structures (4 days)**  
• Strings  
• Lists  
• List Comprehension  
• Tuples  
• Tuple Comprehension  
• Dictionaries (dict)  
• Sets  
• **Practice: Real-World Non-AI and AI Questions**

**Unit 6: Loop Statements (2 days)**  
• for Loop  
• Looping with Data Structures  
• while Loop  
• while Loop with Data Structures  
• pass, break, and continue Statements  
• Practice: Real-World Non-AI and AI Questions

**Unit 7: Modules and Packages (1 day)**  
• Modules  
• Types of Modules  
• Packages  
• Using User Programs as Modules and Packages  
• Practice: Real-World Non-AI and AI Questions

**Unit 8: Functions (2 days)**  
• Creating Your Own Functions  
• Function Parameters & Variable Arguments  
• Function Argument Types (Default, Keyword, Arbitrary)  
• Recursive Functions  
• Lambda Functions  
• map() and filter() Functions  
• Practice: Real-World Non-AI and AI Questions

**Unit 9: File Handling (2 days)**  
• File Handling Modes  
• Reading Files  
• Writing & Appending Files  
• Creating New Files  
• Deleting Files  
• Handling File Exceptions  
• Practice: Real-World Non-AI and AI Questions

**Unit 10: Object-Oriented Programming (3 days)**  
• Creating Classes  
• Objects  
• Instance Methods  
• Inheritance and its Types  
• Polymorphism  
• Abstraction  
• Encapsulation  
• Practice: Real-World Non-AI and AI Questions

**Unit 11: Error and Exception Handling (2 days)**  
• Types of Errors  
• Exception Handling  
• try, except, finally, assert  
• Practice: Real-World Non-AI and AI Questions

**Unit 12: Advanced Python Concepts (2 days)**  
• Iterators  
• Generators  
• Closures  
• Decorators  
• Practice: Real-World Non-AI and AI Questions

**Unit 13: Pandas Library (3 days)**  
• Introduction to Pandas  
• Series and DataFrame objects  
• Creating, Reading, and Writing DataFrames (read\_csv, to\_csv, read\_excel)  
• Indexing, Selecting, and Filtering Data  
• Handling Missing Data (isna(), fillna(), dropna())  
• Data Aggregation (groupby, pivot\_table)  
• Sorting and Ranking  
• Merging, Joining, and Concatenation of Data Frames  
• Applying Functions (apply, map, applymap)  
• DateTime Operations in Pandas  
• Data Visualization with Pandas (plot())

**Unit 14: NumPy Library (2 days)**  
• Introduction to NumPy and its Importance  
• NumPy Arrays: Creation, Indexing, and Slicing  
• Array Operations: Arithmetic, Universal Functions  
• Array Reshaping and Flattening  
• Random Number Generation (random module)  
• Statistical Functions (mean, median, std, var)  
• Linear Algebra Functions (dot, transpose, inverse)  
• Boolean Indexing and Masking  
• Broadcasting in NumPy

**Unit 15: Matplotlib Library (1 day)**  
• Introduction to Matplotlib  
• Plotting with pyplot  
• Line Plots, Scatter Plots, Bar Charts, Histograms, Pie Charts  
• Customizing Plots: Titles, Labels, Legends, Colors, Styles  
• Subplots and Figure Size Management  
• Saving Plots to Files (savefig)

**Unit 16: Seaborn and Plotly (1 day)**

Seaborn:  
• Introduction to Seaborn  
• Visualizing Categorical Data (countplot, boxplot, violinplot)  
• Visualizing Continuous Data (distplot, kdeplot, histplot)  
• Pairplots, Heatmaps, Correlation Plots  
• Customizing Seaborn Plots (Themes, Colors)

**Plotly (Optional / Interactive Plots):**• Introduction to Plotly  
• Interactive Line, Bar, and Scatter Plots  
• Customizing Plotly Layout and Style  
• Exporting Interactive Plots

**Unit 17: Python with SQL/Database (2 days)**  
• Introduction to Database  
• Configure Steps  
• Creating DB Table  
• Insert Data & Read Data From DB  
• Update Data & Delete Data

**Unit 18: GUI Programming with Python (3 days)**  
• Introduction to GUI Programming  
• Tkinter Components and Events  
• Adding a Button  
• Entry Widgets, Text Widgets  
• Check Button, Radio Button, Combo Box, Message Box  
• Speech Recognition  
• Generate QR Code  
• Bank Application  
• Live Running Train Status

**Unit 19: Networking and Threads & Multi-threading (2 days)**  
• Socket Programming  
• Client and Server  
• Multiple Client and Server  
• Threading Method  
• Synchronization Thread

**Unit 20: Python Graphics (1 day)**  
• Turtle Introduction  
• Turtle Animation  
**Real-World Projects**

**Project 1: Smart Street Light Data Monitoring and Visualization (Jaipur / Delhi Smart City)**  
• Objective: Fetch real-time or simulated data related to street lights in Jaipur using Python and visualize energy usage or fault detection.  
• Concepts Used:  
– File Handling  
– Pandas for data cleaning and aggregation  
– Matplotlib/Seaborn for visualization  
– Conditional Logic for fault classification  
**• Outcome: Identify areas with malfunctioning lights and generate energy reports.**

**Project 2: Build a Bank Management GUI with QR and Voice Features**  
• Objective: Create a GUI-based banking system to perform deposit, withdrawal, balance check, and transaction recording using Tkinter.  
• Concepts Used:  
– GUI Programming (Tkinter Widgets, Events)  
– File Handling for saving transactions  
– QR Code Generation for passbooks or customer ID  
– Speech Recognition for voice-activated commands  
**• Outcome: A complete local GUI app that mimics a mini-bank system.**

**Project 3: Real-World Data Aggregation & Reporting System (e.g., COVID, Rainfall, or Pollution Data)**  
• Objective: Use open CSV/Excel files of real-world data (e.g., rainfall, air quality, COVID) to analyze and report insights.  
• Concepts Used:  
– Pandas, NumPy  
– Data Cleaning (isna, fillna, dropna)  
– Data Aggregation and Pivot Tables  
– Data Visualization  
**• Outcome: Create a dashboard-style summary report that helps decision-making.**

**Project 4: Job Recommendation System Based on Skillset (Rule-Based AI)**  
• Objective: Create a basic rule-based system that suggests job roles based on user input (skills, location, interest).  
• Concepts Used:  
– Dictionaries and Conditions  
– Functions, Loops, and String Matching  
– File Reading (CSV for job database)  
– Optional: GUI for form-based input  
**• Outcome: Personalized career suggestions using basic AI logic.**