



PYTHON TRAINING




MODULE-WISE, CHAPTER-WISE COURSE CONTENT

Module 1

Python Fundamentals & Core Concepts

- 1 Python Introduction
- 2 Installation & Environment Setup
- 3 Python Basics
- 4 Variables and Data Types
- 5 Data Structures
- 6 Control Statements
- 7 File Handling

Module 2

Python Analytics & Advanced Programming

- 1 Pandas Fundamentals
- 2 Data Analysis with Pandas
- 3 Data Transformations
- 4 NumPy for Numerical Computing
- 5 Data Cleaning Techniques
- 6 SQL Database Connectivity
- 7 Functions, Modules and Packages
- 8 Exception Handling
- 9 Object-Oriented Programming (OOP)
- 10 Regular Expressions
- 11 Additional Python Concepts

Module 3

Real-Time Project: Banking / Finance

- 1 Project Overview
- 2 Data Ingestion
- 3 Data Cleaning & Transformation
- 4 Analysis & Reporting
- 5 Final Project Implementation

 Duration: 5 Weeks
 Type: LIVE. Interactive.
 100% Practical, Step by Step.



Tools & Technologies Covered

 Python
  Anaconda
  Jupyter Notebook
  Pandas
  NumPy
  Matplotlib
  SQL
  pyodbc
  Regular Expressions
  Python Packages
  AI Assistants

Training Highlights

 LIVE & Interactive Sessions
  Instructor-led Demonstrations
  Hands-on Coding Practice
  Practical Examples
  Assignments & Exercises

 Error Handling & Troubleshooting
  AI-assisted Debugging Exposure
  Real-Time Project Implementation
  Interview-Oriented Concept Discussions
  Resume & Career Guidance

About Python Training

Build practical Python skills for **Data Analytics, Data Engineering, Data Science, Automation and application development** through step-by-step, instructor-led training.

This program starts with Python fundamentals and progressively covers programming concepts, data structures, file handling, Pandas, NumPy, SQL database connectivity, data transformations, exception handling, object-oriented programming and other practical Python concepts.

The training concludes with a **real-time Banking / Finance project**, helping participants understand how Python concepts can be applied to practical data-processing and analytics scenarios.

Your document already covers Python roles for Data Analysts, Data Engineers and Data Scientists, so this positioning fits the existing curriculum well.

Why Learn Python?

Python is widely used across modern data and software roles because of its readable syntax, extensive ecosystem and ability to work with databases, files, APIs, analytics libraries and automation workflows.

With Python, you can learn to:

- Analyze and transform datasets
- Automate repetitive data-processing activities
- Work with CSV, text and other file formats
- Connect Python applications to SQL databases
- Clean and prepare data using Pandas
- Perform numerical operations using NumPy
- Apply programming concepts to real-world projects

Python Skills for Modern Data Roles

Python Fundamentals → Python Programming → Pandas & NumPy → Data Cleaning → SQL Integration → Data Transformations → Visualization → Real-Time Analytics Project → Career Preparation.

How We Train

Learn → Practice → Implement → Troubleshoot → Build Project

Every concept is taught through a practical, step-by-step approach.

- ✓ LIVE & Interactive Sessions
- ✓ Instructor-led Demonstrations
- ✓ Hands-on Coding Practice
- ✓ Practical Examples
- ✓ Assignments & Exercises
- ✓ Error Handling & Troubleshooting
- ✓ AI-assisted Debugging Exposure
- ✓ Real-Time Project Implementation
- ✓ Interview-Oriented Concept Discussions

Who Should Enrol?

This program is suitable for:

- ✓ Beginners who want to start programming with Python
- ✓ SQL Developers and Database Professionals
- ✓ Aspiring Data Analysts
- ✓ Aspiring Data Engineers
- ✓ BI / Reporting Professionals
- ✓ Software Developers looking to learn Python
- ✓ Students and fresh graduates entering data careers
- ✓ Working professionals looking to add Python to their existing skill set

Prerequisite: Basic computer knowledge is sufficient to begin. Previous Python programming experience is not required.

Learning Outcomes

By the end of this training, participants should be able to:

- ✓ Write and execute Python programs
- ✓ Work confidently with Python data types and data structures
- ✓ Implement conditions, loops, functions and reusable modules
- ✓ Read, write and process files using Python
- ✓ Create and manipulate Pandas DataFrames
- ✓ Clean missing, duplicate and inconsistent data
- ✓ Merge, join and transform multiple datasets
- ✓ Connect Python with SQL databases
- ✓ Execute SQL queries and process results using Python
- ✓ Work with exceptions and error handling

- ✓ Understand classes, objects and OOP concepts
- ✓ Use regular expressions for text processing
- ✓ Apply Python concepts in a real-time project

Tools & Technologies Covered

Python | Anaconda | Jupyter Notebook | Pandas | NumPy | Matplotlib | SQL | pyodbc | Regular Expressions | Python Packages | AI Assistants

Course Modules:

Module 1 – Python Fundamentals & Core Concepts

Build a strong programming foundation covering Python architecture, installation, variables, operators, data types, lists, dictionaries, tuples, conditions, loops and file handling.

Module 2 – Python Analytics & Advanced Programming

Move into practical data processing with Pandas, NumPy, DataFrames, SQL connectivity, data cleaning, transformations, functions, modules, packages, exception handling, OOP, regular expressions and additional programming concepts.

Module 3 – Real-Time Project: Banking / Finance

Apply the concepts learned throughout the course to a practical project involving data ingestion, cleaning, transformation, analysis and business-ready outputs.

- ✓ **Duration: 6 Weeks**
- ✓ **Type: LIVE. Interactive.**
- ✓ **100% Practical, Step by Step.**

Module 1: Python Concepts

Ch 1: Python Introduction

- ✓ Python Introduction
- ✓ Python Versions
- ✓ Python Job Roles
- ✓ Python for Data Analysts
- ✓ Python for Data Engineers
- ✓ Python for Data Scientists

Ch 2: Python Architecture

- ✓ Python Architecture
- ✓ PVM: Python Virtual Machine
- ✓ Compiler
- ✓ Byte Code
- ✓ Execution Process
- ✓ Resource Allocations
- ✓ Python Implementations

Ch 3: Python Installations

- ✓ Python Introduction
- ✓ Python Installations
- ✓ Anaconda Installation
- ✓ Python IDE & Usage
- ✓ Jupyter Notebooks

Ch 4: Python Print Statement

- ✓ Python Print Statement
- ✓ print(), print()
- ✓ Testing Case Sensitivity
- ✓ Single Line print()
- ✓ Multi Line print()
- ✓ print() with single quotations
- ✓ Debug with AI (AI Assistants)

Ch 5: Python Variables

- ✓ Python Variables
- ✓ Assigning values
- ✓ Purpose & Rules
- ✓ Variable Value Reads
- ✓ Multiple Variables & Print()

Ch 6: Python Operators

- ✓ Arithmetic Operators
- ✓ Python String Literals
- ✓ Single, Double Quotes
- ✓ Format Strings (f string)
- ✓ Comparison, Indexing Operators

Ch 7: Python Data Types

- ✓ Python Data Types
- ✓ Integer, Float, String Data Types
- ✓ Type Casting
- ✓ Type Identification
- ✓ Multi Value Assignments

Ch 7: Python Lists

- ✓ Creating Python Lists
- ✓ Printing List Items
- ✓ Print List Slices
- ✓ Length & Type
- ✓ Empty Lists, Append
- ✓ Loops, List Updates

Ch 8: Python Dictionaries

- ✓ Python Dictionary
- ✓ Indexing Dictionaries
- ✓ Edit Key Values
- ✓ Lists inside Dictionaries
- ✓ Delete & Clear

Ch 9: Python Tuples

- ✓ Python Tuples
- ✓ Defining, Indexing
- ✓ Length(), Type()
- ✓ Mixed Values in Tuples
- ✓ Overwriting Tuples

Ch 10: Python IF..ELSE Condition

- ✓ If..Else conditions
- ✓ if..elif..else & Shorthand if
- ✓ composite conditions
- ✓ Indent, pass statement
- ✓ in & negation operators
- ✓ range conditions

Ch 11: Python Loops

- ✓ Python For Loop
- ✓ For Loop @ Range
- ✓ For Loop @ Sequence Values
- ✓ While Loop, Exit Conditions
- ✓ Nested Loops
- ✓ Control Statements
- ✓ Break, Continue, Pass

Ch 12: Python File Handling

- ✓ File Handling, Activities
- ✓ Loop, Write, Close Files
- ✓ Appending, Overwriting
- ✓ import os, path.exists
- ✓ f.open, f.write
- ✓ f.read, f.close

Module 2: Python Analytics, Programming

Ch 13: Python Dataframes

- ✓ Dataframes: Creation
- ✓ Pandas Dataframes
- ✓ Dataframes From Single List

- ✓ Dataframes from Dictionary
- ✓ Display Dataframes, List Items
- ✓ Identify, Replace Nulls, NumPy

Ch 14: Python SQL DB Access

- ✓ SQL DB Access with Python
- ✓ import pandas.DataFrame
- ✓ pyodbc module, sql functions
- ✓ SQL DB Cursor Connections
- ✓ SQL Query Executions: DDL, DML
- ✓ Filters, Aggregations with SQL
- ✓ Dataframe Usage with SQL

Ch 15: Dataframe Transformations - 1

- ✓ Dataframe Transformations
- ✓ Concat & Append
- ✓ Merge Function
- ✓ Join with Multiple Dataframes
- ✓ Data Type Checks, Conversions
- ✓ Loops with Dataframes

Ch 16: Dataframe Transformations - 2

- ✓ Pandas - Cleaning Data
- ✓ Replace, Transform Columns
- ✓ Data Discovery & Column Fill
- ✓ Identify & Remove Duplicates
- ✓ dropna(), fillna() Functions

Ch 17: Dataframe Transformations - 3

- ✓ Python Functions for date, time
- ✓ Now() Function in Python
- ✓ date, time Functions
- ✓ Calendar Data Generation

Ch 18: Python Functions & Lambda

- ✓ Python Functions & Usage
- ✓ Function Parameters
- ✓ Default & List Parameters
- ✓ Python Lambda Functions
- ✓ Recursive Functions, Usage
- ✓ Return & Print @ Lambda

Ch 19: Python Modules

- ✓ Import Python Modules

- ✓ Built In Modules & dir
- ✓ datetime module in Python
- ✓ Date Objects Creation
- ✓ strftime Method & Usage
- ✓ imports & datetime.now()

Ch 20: Python User Inputs & TRY

- ✓ Try Except, Exception Handling
- ✓ Raise an exception method
- ✓ TypeError, Scripting in Python
- ✓ Python User Inputs
- ✓ Python Index Numbers
- ✓ input() & raw_input()

Ch 21: Python Dictionary

- ✓ Dictionary Creation, Use
- ✓ Hashing, Copy, Update
- ✓ Deletion, Sorting
- ✓ Len(), Inbuilt Functions
- ✓ Variable Types - python List
- ✓ Cmp() List Method
- ✓ Python Dictionary Str(dict)
- ✓ Programming Concepts
- ✓ Loops and Sets
- ✓ Realtime Usage

Ch 22: Python Packages

- ✓ Package in Python
- ✓ Creating a package
- ✓ Package Imports, Modules
- ✓ Sub Packages Creation
- ✓ Sub Package Imports
- ✓ Popular Packages in Python
- ✓ NumPy & SciPy
- ✓ Libraries in Python
- ✓ Python Seaborn
- ✓ Python framework

Ch 23: Exception Handling

- ✓ Shell Script Commands
- ✓ OS operations in Python
- ✓ File System Shell Methods
- ✓ os - math - cmd - csv - random
- ✓ Numpy (numerical python)

- ✓ Pandas - sys - Matplotlib
- ✓ Common RunTime Errors
- ✓ Python Custom Exception
- ✓ Exception Handling
- ✓ Try...Except...else, Try...finally

Ch 24: Python Class & Objects

- ✓ Class variables, Instances
- ✓ Built in Class Attributes
- ✓ Objects - Constructors
- ✓ Modifiers - Self Variable
- ✓ Python Garbage Collections
- ✓ Hierarchical Inheritance
- ✓ Multilevel, Multiple, Hybrid
- ✓ Overloading & OverRiding
- ✓ Polymorphism– Abstraction

Ch 25: Regular Expressions

- ✓ Regular Expression
- ✓ Regular Expression Patterns
- ✓ Literals - Repetition Cases
- ✓ Groups and Grouping
- ✓ w+ and ^ , \s Expressions
- ✓ re.split function
- ✓ Regular expression methods
- ✓ re.match() in Regular Expr
- ✓ re.search(), re.findall for Text

Ch 26: Multi-Threading

- ✓ Python Multi-Threading
- ✓ Thread Synchronization
- ✓ Python Gil & Programming
- ✓ Thread Control Block (TCB)
- ✓ Stack Pointers & App Usage
- ✓ Program Counters in Realtime
- ✓ Thread State Concept
- ✓ Python Exception Handling

Ch 27: Python TKinter

- ✓ Tkinter GUI Program
- ✓ Components & Events
- ✓ Adding Controls in Tkinter
- ✓ Radio & Check Buttons
- ✓ Tkinter Forms in Realtime

- ✓ List Boxes, Menu, ComboBox
- ✓ Mainloop () & Functions

Ch 28: Python Web & IoT Intro

- ✓ Python Web Frameworks
- ✓ Django: Advantages
- ✓ Web Framework
- ✓ MVC and MVT - Django
- ✓ Web Pages using python
- ✓ HTML5, CSS3 usage
- ✓ PYTHON Bottle & Pyramid
- ✓ Falcon; smart_open in python

Real-Time Project – Banking / Finance Data Analytics

Project Objective

Build an end-to-end Python-based data-processing and analytics solution using realistic Banking / Finance datasets.

Project Activities

Raw Data Sources

- Python Data Ingestion
- Data Validation
- Data Cleaning
- Pandas DataFrames
- Data Transformations
- SQL Database Integration
- Business Analysis
- Final Processed Dataset

Practical Skills Covered

- ✓ Reading data from multiple sources
- ✓ DataFrame creation and manipulation
- ✓ Handling null and duplicate values
- ✓ Data type conversions
- ✓ Filtering and aggregating records
- ✓ Merge, Join and Concat operations
- ✓ SQL database connectivity
- ✓ Exception handling
- ✓ Reusable functions
- ✓ Business-rule implementation
- ✓ Preparing analysis-ready datasets

Ready to Build Your Python Skills?

Learn Python. Work with Data. Build Real-World Solutions.

Start from Python fundamentals and progress toward practical **Python Analytics & Programming** through instructor-led sessions and hands-on implementation.

Program Highlights

- ✓ 6 Weeks of Structured Learning
- ✓ LIVE & Interactive Training
- ✓ 100% Practical, Step-by-Step Sessions
- ✓ Python Analytics + Programming
- ✓ SQL Database Integration
- ✓ Pandas & NumPy
- ✓ Real-Time Banking / Finance Project

BUILD REAL SKILLS. WORK ON REAL PROJECTS. BECOME JOB READY.



Ready to Become an Expert?



Join Our Next Batch. Reach us today!



Call/WhatsApp: +91 9951440801



Course Details: <https://sqlschool.com/python-programmer-training/>

Wish you all the best!