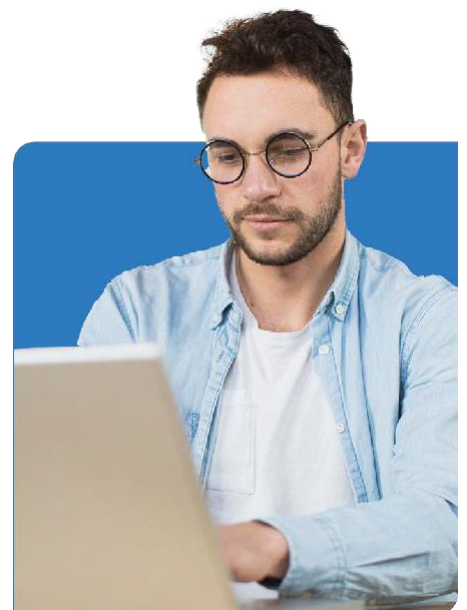




Python - Basic

TDXFSB-103



ThriveDX Python - Basic

Time Commitment

5 days (total of 40 hours / 8 hours per day)

Skill Level

Basic Level

Course Category

Full Stack- Back-end Development

Python is a powerful, widely used and very common object-oriented programming language. Although Python is considered one of the easiest languages to learn due to its simple and clean syntax. Simplicity and efficiency of the language have made it one of the most popular in the world and is on almost every platform that exists.

Python is widely used in the fields of artificial intelligence, data analysis and machine learning - areas that require rapid analysis and calculations due to high computing capabilities of Python.

This course provides participants with basic but practical knowledge of the Python language and prepares them to start working with Statistics, web development and applications (mainly server side), networks, cyber and even interfaces development on huge amounts of information and touching on graphics and games.

Target Audience

The course is intended for those with little to no programming experience, but is also appropriate for people who already know to code and want to add Python to their tool belt.



Prerequisites

There are no prerequisites for this course. Previous experience with another programming language is an advantage.

Objectives

- Learning of general topics and concepts from the world of programming
- Participants will learn to write Python programs from scratch, using one of the environments that is the most common for development in the industry
- Proper learning of proper syntax, proper use of indentations, assimilation of logic in programs, efficient design of code
- Effective work with the variety of types of arrays of the language
- Use and creation of user functions
- Reading and writing text files

Program Structure

Module 1 – Introduction to Python

- Overview of Python: History, versions, and significance in the industry.
- Setting up the Python environment: Installing Python and a development environment (IDE).
- Basic syntax and principles: Understanding Python syntax, indentation, and basic principles.

Module 2 – Variables and Data Types

- Understanding variables: Declaration, assignment, and conventions.
- Data types: Introduction to Python's built-in types (strings, numbers, lists, dictionaries, tuples, sets).
- Type conversion: Converting between different data types.
- Operators and Expressions
- Arithmetic operators: Addition, subtraction, multiplication, division, etc.
- Comparison and logical operators: Equal to, not equal to, greater than, less than, and, or, not.
- Bitwise and assignment operators: Understanding bitwise operations and assignment shortcuts.

Module 3 – String Manipulation

- Creating and accessing strings: Basic operations, indexing, and slicing.
- String methods: Searching, modifying, and case conversion.
- Formatting strings: Old-style (%) formatting, str.format(), and f-strings.

Module 4 – Collections

- Lists: Creation, indexing, slicing, and methods.
- Tuples: Understanding immutability, creation, and usage.
- Dictionaries: Key-value pairs, accessing, adding, and removing elements.
- Sets: Characteristics, creation, and operations (union, intersection, difference).

Module 5 – Control Flow

- Conditional statements: if, elif, and else blocks.
- Looping constructs: for and while loops, iterating over collections, loop control statements (break, continue, pass).

Module 6 – Functions

- Defining and calling functions: Syntax, arguments, and return values.
- Scope and lifetime of variables: Local, global, and nonlocal variables.
- Anonymous functions: Understanding lambda functions and their uses.

Module 7 – Modules and Packages

- Importing modules: Using import statements, from...import syntax.
- Exploring standard modules: math, sys, os, random, etc.

Module 8– File Handling

- Opening and closing files: Reading from and writing to files.
- File modes and operations: Understanding different file modes (read, write, append).
- Working with paths: Using the os and pathlib modules to handle file paths.

Module 9 - Introduction to Object-Oriented Programming**

- Classes and objects: Defining classes, creating instances.
- Attributes and methods: Instance variables, methods, the self parameter.
- Special methods: `__init__`, `__str__`, `__repr__`, and others.

Module 10– Creating GUI Applications with PyQt

- Introduction to GUI Development
- Installing PyQt
- Understanding the PyQt application structure: QApplication, QWidget, and QMainWindow.
- Basic PyQt Widgets
- Exploring basic widgets: QLabel, QPushButton, QLineEdit, QTextEdit, etc.
- Layout management: Understanding layouts (HBoxLayout, QVBoxLayout, GridLayout) to design the form.
- Signals and Slots
- Introduction to event-driven programming: Understanding signals and slots mechanism in PyQt.
- Connecting widgets: Making applications interactive by connecting user actions to functions.
- Advanced Widgets and Dialogs
- Introduction to Qt Designer: Benefits and how it simplifies GUI design.
- Creating forms with Qt Designer: Using drag-and-drop to design the UI.
- Converting .ui files to Python code: Using pyuic5 or pyuic6 command-line tool.
- Application Logic and Structure
- Structuring a PyQt application: Separating business logic from UI code.
- Implementing MVC (Model-View-Controller) or MVP (Model-View-Presenter) patterns in PyQt applications.
- Database Connectivity

Module 11– Connecting to database

- Connecting to databases: Using QSqlDatabase to connect to SQLite, MySQL, or PostgreSQL databases.
- Performing CRUD operations: Displaying, adding, editing, and deleting database records through the GUI.
- Deploying PyQt Applications

Module 12– final project

- Building a complete PyQt application: From concept to deployment. Projects could include a simple text editor, a task manager, or a database-driven application.

