

# Python Course Content

## Core Python

### Introduction

- History
- Features
- Setting up path
- Working with Python
- Basic Syntax
- Variable and Data Types
- Operator

### Conditional Statements

- If
- If- else
- Nested if-else

### Looping

- For
- While
- Nested loops

### Control Statements

- Break
- Continue
- Pass

### String Manipulation

- Accessing Strings
- Basic Operations
- String slices

## Lists

- Introduction
- Accessing list
- Operations
- Working with lists
- Function and Methods

## Tuple

- Introduction
- Accessing tuples
- Operations
- Working
- Functions and Methods

## Dictionaries

- Introduction
- Accessing values in dictionaries
- Working with dictionaries
- Properties
- Functions

## Functions

- Defining a function
- Calling a function
- Types of functions
- Function Arguments
- Anonymous functions
- Global and local variables

## Modules

- Importing module
- Math module
- Random module
- Packages

- Composition

## Input-Output

- Printing on screen
- Reading data from keyboard
- Opening and closing file
- Reading and writing files
- Functions

## Exception Handling

- Exception
- Exception Handling
- Except clause
- Try ? finally clause
- User Defined Exceptions

## ADVANCED PYTHON:--

### Module 1:

Recap of Beginners Python This module will focus on a revision of: Data types, Sequences, Mapping types, Program structure, Files and console I/O, Conditionals, Loops and Bulletins

### Module 2: OS Services The Os module

- Environment variables
- Launching external commands
- Walking directory trees
- Paths, directories, and filenames
- Working with file systems
- Dates and times

### Module 3: Plythonic Programming the Zen of Python

- Common idioms
- Lambda functions

- List comprehensions
- Generator expressions
- String formatting

#### **Module 4:** Modules and Packages Initialization code

- Namespaces
- Executing modules as scripts
- Documentation
- Packages and name resolution
- Naming conventions
- Using imports

#### **Module 5:** Classes defining classes

- Instance methods and data
- Initializes
- Class methods and Static methods
- Inheritance and Multiple inheritances
- Pseudo-private variable

#### **Module 6:** Meta Programming Implicit properties

- Gobals() and locals()
- Attributes
- The inspect module
- Decorators
- Monkey patching

#### **Module 7:** Programmer Tools Analyzing programs

- Using pylint and unittest
- Testing and Debugging
- Profiling

#### **Module 8:** Distributing Modules Distribution concepts

- setuptools
- creating setup.py

- building
- & running installers

### **Module 9:** Database Access The DB API

- Available Interfaces
- Connecting to a server
- Creating and executing a cursor
- Fetching data
- Parameterized statements
- Metadata
- Transaction control

### **Module 10:** GUI Programming The mainwindow object

- Widgets
- Colors and fonts and GUI layout
- Event handling

### **Module 11:** Network Programming Sockets, Clients and Servers

- Application protocols
- Forking servers and Threaded servers
- Binary data
- The struct module

### **Module 12:** Threads Why use threads?

- Python threads modules
- The threading and queue module
- The python thread manager
- Debugging threaded programs

### **Module 13:** XML and JSON Working with XML, DOM and SAX

- Introducing ElementTree
- Parsing XML
- Navigating the document

- Creating a new XML document
- JSON
- Parsing JSON into Python
- Converting Python into JSON

#### **Module 14:** Extending Python About non-Python modules

- Overview of a C extension
- Creating functions
- Registering functions
- Initialization code
- Loading the module

#### **Module 15:** Using the sh module The sh namespace

- Running a simple command
- Keyword arguments
- Running commands in the background
- Piping and redirection
- Working with STDIN
- Exit codes
- Advanced features