

About Python

- Python course is a great introduction to both fundamental programming concepts and the Python programming language. By the end, you'll be familiar with Python syntax and you'll be able to put into practice what you'll have learned in a final project you'll develop locally.
- Python Programming is intended for software engineers, systems analysts, program managers and user support personnel who wish to learn the Python programming language.

Python Duration

- This course duration is **35** hrs++.

Training Objectives

The learning objectives of this course are:

- To understand why Python is a useful scripting language for developers.
- To learn how to design and program Python applications.
- To learn how to use lists, tuples, and dictionaries in Python programs.
- To learn how to identify Python object types.
- To learn how to use indexing and slicing to access data in Python programs.
- To define the structure and components of a Python program.
- To learn how to write loops and decision statements in Python.
- To learn how to write functions and pass arguments in Python.
- To learn how to build and package Python modules for reusability.
- To learn how to read and write files in Python.
- To learn how to design object-oriented programs with Python classes.
- To learn how to use class inheritance in Python for reusability.
- To learn how to use exception handling in Python applications for error handling.

Training Pre - Requisites & Who Should Learn Python

- As such, there are no pre-requisites for learning Python, Knowledge of high level language and scripting language, but certainly not a mandate.
- Market for Python is growing across the world and this strong growth pattern translates into a great opportunity for all the IT Professionals.

- Here are the few IT Professional, who are continuously enjoying the benefits moving into Python domain:
 - Developers, Java Programmers and Architects
 - BI /ETL/DW professionals
 - Senior IT Professionals
 - Testing professionals
 - Mainframe professionals
 - System/Network Administrators
 - Freshers
-

I. Python Basics

Chapter 1: Getting started with Python programming

1.1 Introduction to Python

1.1.1 Python features

1.1.2 Scope of python

1.1.3 Python products

1.1.4 Python in today's context

1.2 Python Download, Installation and Environment Setup

1.3 First python program execution "Hello World"

1.4 The world of programming

1.5 Python programming syntax

Chapter 2: Variables, keywords and Operators

2.1 Variables

2.1.1 Memory mapping of variables

2.1.2 Application memory

2.1.3 Variable nomenclature

2.1.3 Properties and scope of variables

2.2 Keywords in Python

2.3 Operators

2.3.1 Arithmetic operators

2.3.2 Operator precedence

2.3.3 Logical operators

2.3.4 Membership Operators

2.4 Basics I/O and Type casting

2.5 `__builtins__` functions and getting help

Chapter 3: Control flow statements

3.1 Flow of program control

3.2 Decision making statements: if-elif-else

3.3 for loop

3.3.1 Making of 'for' loop

3.3.2 Repetition using for loop: range() function

3.3.3 Iteration using for loop

3.4 while loop

3.4.1 Making of 'for' loop

3.4.2 Infinite loop

3.5 Loop control keywords: break, continue, and pass

Chapter 4: Numbers and Functions

4.1 Introduction to functions

4.1.1 Function definition and return

4.1.2 Function call and reuse

4.1.3 Function parameters

4.2 Function recipe and docstring

4.3 Programming with functions

4.4 Namespaces and scope of variable

4.5 Numbers - int, float, long, complex

Chapter 5: Strings

5.1 Introduction to Python 'string' data type

5.2 Properties of a string

5.3 String built-in functions

5.4 Programming with strings

5.5 String formatting

Chapter 6: Lists

6.1 Introduction to Python 'string' data type

6.2 Properties of a list

6.3 List built-in functions

6.4 Programming with lists

6.5 List comprehension

Chapter 7: Tuples, Dictionary and Sets

7.1 Tuples as Read only lists

7.2 Moving from list to dictionary

7.3 Dictionary built-in functions

7.4 Sets and sets properties

7.5 Set built-in functions

Practice, Test & Revision

II. Advanced Python

Chapter 8: More of Python functions

- 8.1 Recursive functions
- 8.2 *args, **kwargs, argv
- 8.3 Modules and Packages
- 8.4 Iterators and Generators
- 8.5 Function decorators

Chapter 9: Object oriented programming with Python

- 9.1 OOPs concepts: Classes and objects
- 9.2 Making of a class and module namespace
- 9.3 Static and instance variables
- 9.4 Deep understanding of self and __init__()
- 9.5 Inheritance and Overriding
- 9.6 Overloading functions
- 9.7 Operator overloading
- 9.8 Encapsulation: Hiding attributes

Chapter 10: Exception Handling in Python

- 10.1 Understanding exceptions
- 10.2 try, except, else and finally
- 10.3 raising exceptions with: raise, assert
- 10.4 Creating your own exception classes
- 10.5 Logging and Debugging

Chapter 11: File handing - Part 1

- 11.1 Working with files

- 11.2 File objects and Modes of file operations
- 11.3 Reading, writing and use of 'with' keyword
- 11.4 read(), readline(), readlines(), seek(), tell() methods
- 11.5 Handling comma separated value files
- 11.6 CSV reading and writing with DictWriter

Chapter 12: File Handling - Part 2

- 12.1 JSON parsing
- 12.2 File compression - zipping and unzipping
- 12.3 Pickling

Chapter 13: Regular expression

- 13.1 Pattern matching
- 13.2 Meta characters for making patterns
- 13.3 re flags
- 13.4 Project 1: Pattern matching over files

Chapter 14: Multithreading

- 14.1 Introduction
- 14.2 Starting a new Thread using 'Thread' module
- 14.3 The Threading Module
- 14.4 Synchronizing Threads

Chapter-15: Python Debugging and Automation Framework

- 15.1 'dis' module
- 15.2 'pdb' module
- 15.3 'profile' module
- 15.4 Planning for automation framework

15.5 Developing Automation framework

15.6 Real time planning for automation testing and do automation

15.7 Creating networking and network test automation