

Python

--- Conducted by OmniLin (omnilin.com/)

About this course

The course is basically designed for developers, learners to familiarize with Python Programming. The training period allows the developer to understand different features of Python Programming used for coding and debugging.

Enrol yourself in this training ! Learn Python!

Prerequisites

Before joining this Programming Courses students should have very basic knowledge of any one programming language like "C", "C++", or any other programming language basics.

What are you going to get from this course? (AT A GLANCE)

1. To gain deep understanding of Object Oriented Programming
2. To learn basics of Python
3. To create and manage strings, arrays, collections and enumerators using Python
4. To learn development tools for python
5. To carry out and learn the concept of python project structure
6. To create simple to complex application using python
7. To create classes which are efficient and appropriate for given programming scenarios
8. To gain deep understanding of python language features like Data Structures, exceptions etc.

Syllabus overview

Fundamentals of object oriented programming

- Introduction
- Basic Concepts Of Object Oriented Programming
- Benefits of OOPS

Python Evaluation

- Python history
- Python Features
- Difference from Java

Overview of Python language

- Installing python
- Simple python program
- Whitespaces & indentation
- python Identifiers

- comments
- modules

Constants, variables and datatypes

- Constant
- Variable
- Data Type
- Reading from keyboard
- command line input
- Multiple Statements on a Single Line

Operators and expressions

- Operators
- Relational Operators
- Logical Operators
- Shorthand Operator
- Expressions
- Type Conversions

If-else , the control flow

- If statement
- Else statement
- Truth value testing

Looping

- While loop
- Lists
- For loop
- range() function
- Continue statement
- Else loop

Data Structures

- Lists
- Using lists as stack and queue
- List Comprehensions
- Tuples
- Sets
- Dictionaries

Strings

- Different methods available for Strings
- Strip the strings
- Finding text

Functions

- Defining a function
- Local and global variables
- Default argument value
- Keyword arguments

- Keyword only argument
- Docstrings
- Higher-order function
- map function

File handling

- File opening
- Closing a file
- Reading a file
- Writing in a file

Exceptions

- NameError
- TypeError
- How to handle exceptions?
- Raising exceptions
- Using finally for cleanup

Class

- Your first class
- `__init__` method
- Inheritance
- Multiple Inheritance
- Deleting an object
- Getters and setters in Python

Modules

- Introduction
- Importing modules
- Submodules
- `__all__` in `__init__.py`
- Default modules
- Module `os`
- Requests Module
- Collections module
- Counter
- `defaultdict`
- `namedtuple`

Iterators, generators and decorators

- Iterators
- Generators
- Generator expressions
- Closures
- Decorators

Simple testing in Python

- What we should test ?
- Unit testing
- unittest module
- Test case

Introduction to Flask

- What is flask?
- What are template engines?
- An "Hello world" application in flask
- Using arguments in Flask

Project structure

- Documentation of code
- Structure of code