



pythonTM

Core & Advanced

We are ready to serve Latest IT Trends, Are you ready to learn.??

The Leader in Software Training
9963799240 / 7730997544

Ameerpet / Kondapur
Hyderabad

START DATE :

TIMINGS :

DURATION :

TYPE OF BATCH :

FEE :

FACULTY NAME :

What is Python?

- Python is a General Purpose, High Level, Procedure Oriented, Object Oriented, Scripting Programming Language.
- It comes with a lot of modules and packages to reducing modularity of the code and helps to improve the application performance.

Why Python?

- Python is friendly & easy to learn
- Python programs can be developed and tested quickly since there is no compilation phase.
- Python is very easy to debug, Python runs everywhere and Python is a Open source.

Who should learn this course?

- ✓ Automation(Robotics- Artificial Intelligence) engineer
- ✓ Data Scientist(Data Analysis and Data Reporting)
- ✓ Administration Activities(DBA, System Administration, OS Admin, Network Admin)
- ✓ Web Application Development(web developer)
- ✓ DevOps Engineer & Cloud Engineer(AWS/Azure/Sales force/Google Cloud)
- ✓ Test-Engineer(Selenium)
- ✓ IOT(Internet Of Things) & Embedded application development
- ✓ GUI application Development & Scientific Application
- ✓ Games Development & Animation applications
- ✓ Hacking(Implement Cryptography and Network Security Algorithms)
- ✓ Business People
- ✓ Beginner's also Learning Python
- ✓ GIS (Geographic Information System)

Core Python

Python Overview:

What is Python? The Birth of Python, History of Python, Features of Python, Versions of Python, Applications of Python.

Python Implementations – CPython, Iron Python, Jython, pypy

Distributions – python.org, anaconda python, pypy

The Python Environment:

Installation of Python, Python Documentation, Getting Help, How to develop python applications/projects, Python Editors and IDE's, Basic Syntax, Running a Python Script, Python Scripts on UNIX/Windows.

Getting Started:

Keywords, Data Types, Variables, assign values to variables, multiple assignments, Space Indentation, Quotes, Comments, Print(), Type(), Id() Functions, input(), raw_input() functions, How to read the data from keyboard, Type conversions, Number systems, Mutable and Immutable objects.

String Handling:

What is String? Single-quoted string literals, Triple-quoted string literals, String Indexing, String Slicing, Working with String Functions, Working with String Methods.

Operators:

What is Expression, What is Operator?

Types of Operators:

- ✓ Arithmetic Operators
- ✓ Relational Operators
- ✓ Logical Operators
- ✓ Assignment Operators
- ✓ Short Hand Assignment Operators

- ✓ Bitwise Operators,
- ✓ Membership Operators,
- ✓ Identity Operators,

Operator Precedence.

Difference between 'is' operator and '== ' operator.

Flow Control:

About Flow Control Statements, Elements of control flow statements

Types of Control Flow Statements,

Conditional Statements:

- ✓ simple if
- ✓ if...else
- ✓ if...elif...else

Looping Statements:

- ✓ while loop,
- ✓ infinite while loop
- ✓ while ... else,
- ✓ for loop,
- ✓ for loop ... else
- ✓ nested loops,

Break Statement

Continue Statement

Pass Statement.

Collection Objects/Collection Data Types:

what is Collection Object?



Types of Collection Objects:

- ✓ Sequence Collection Objects
- ✓ Non Sequence Collection Objects

Sequence Collection Objects:

- ✓ List:

what is list, creating list, accessing/deleting/updating list elements, indexing, slicing and matrix, working with built-in list functions and methods. list comprehension.

- ✓ Tuple:

what is tuple, creating tuple, accessing/deleting/updating tuple elements, indexing, slicing and matrix, working with built-in tuple functions and methods.

Non-Sequence Collection Objects:

- ✓ Set:

what is Set, creating Set, working with built-in set functions and methods and set comprehension, mathematical set operations:

- ✓ Frozenset:

what is Frozenset, creating Frozenset, working with built-in Frozenset functions and methods

- ✓ Dict(sequence and non sequence):

what is dictionary, creating a dictionary, accessing/deleting/updating dictionary elements, working with built-in dictionary functions and methods, dictionary comprehension.

Functions:

Defining a function,

Types of Functions:

- ✓ Built-in Functions
- ✓ User-Defined Functions

Calling a function, Function Parameters,

Types of Parameters:

- ✓ Default Parameters
- ✓ Non-Default Parameters

Function Arguments,

Types of Arguments:

- ✓ Normal Arguments
 - Keyword Arguments
 - Non-Keyword Arguments
- ✓ Arbitrary Arguments
- ✓ kw args

Returning Values, Function recursion, Function Variables,

Types of Function Variables:

- ✓ Local Variable
- ✓ Global Variable

Variables scope, Call by value, Call by reference, Passing collections to a function, passing functions to function, Lambda function.

Examples: range (), xrange (),abs(), all(),any(), format(),enumerate(), map(), filter(), reduce(),round(),zip(),.....etc

Advanced Python

OOPs concepts:

About OO programming, Benefits OOP's concepts

The OOP's concepts are:

- ✓ Encapsulation
- ✓ Inheritance
- ✓ Polymorphism
- ✓ Data Abstraction

What is a Class, Defining a Class, what is Object, Creating a object, Class methods and data, static variables and non-static variables, Static methods, Instance methods, Constructors, Garbage collection, Destructors, Built-in attributes of a class, add and remove the attributes of a class from outside of that class, What is Inheritance, Types of inheritances, what is MRO, Polymorphism (over loading & over riding),Data hiding, access modifiers, Dunder methods.

Modules:

What is a module?, Creating user defined module,

Importing a module in python:

- ✓ normal import
- ✓ from import
- ✓ from import with *

Renaming a module, module search path, reloading a Module, Dir function, working with Standard modules (Math, Random, Datetime and time, Os and sys, String,).

The hidden concept of if `__name__ == 'module'` :

Packages:

What is a package?, Creating user defined package,

Importing a package in python:

- ✓ normal import

- ✓ from import
- ✓ from import with *

Renaming a package.

Multi Threading:

Defining a Thread, starting a Thread, Thread Life Cycle, What is Scheduling, suspend Thread by using sleep(), Threads synchronization.

Synchronization primitives are :

- ✓ Semaphore,
- ✓ Locks,
- ✓ Events,
- ✓ Condition Variables.

Errors and Exception Handling:

Syntax Errors, Runtime Errors, What is Exception?

Types of Exceptions:

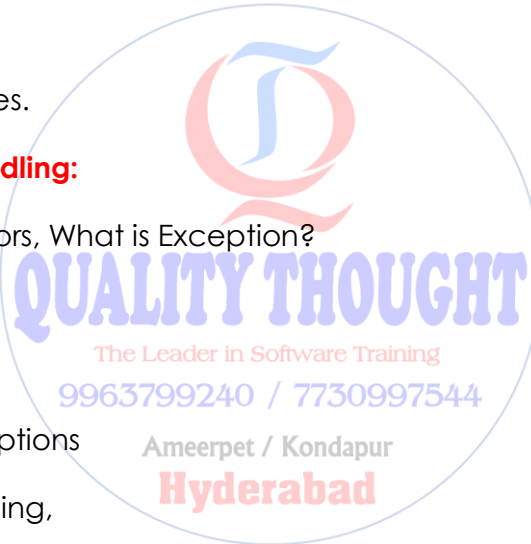
- ✓ Built-in Exceptions
- ✓ User defined Exceptions

Need of Exception handling,

Handling the Exceptions:

- ✓ try block/clause
- ✓ except block/clause
- ✓ finally block/clause

single try block with multiple except blocks, Nested try blocks, Handling Multiple Exceptions, Raising the user defined exceptions.



File Handling:

History of file concept, what is file, types of file formats, order of the file handling, Opening a file, Closing a file, Writing data to files, Reading a data from files, Tell(), Seek() functions.

Serialization and de-serialization:

what is Serialization ,how to implement Serialization in python, what is de-serialization, how to implement Serialization and de-serialization in python.

- ✓ pickle module

Database Access:

Basics of database (What is Data, What is Information, What is DBMS, Types of Data bases), Connections, Executing SQL and queries,

Basic SQL commands are:

- ✓ DDL(Data Definition Languages)
- ✓ DML(Data Manipulation Languages)
- ✓ DCL(Data Control Languages)
- ✓ TCL(Transaction Control Languages)
- ✓ DRL (Data Retrieval Languages).

what is performance tuning?

Performance Tuning Techniques are:

- ✓ By calling prepare() method

Transaction management.

Command Line Arguments:

What is Command Line argument, How to implement Command Line Arguments in python.

Regular Expressions:

What is regular expression?, Wild card characters, Forming regular expressions, Character classes, Quantifiers, Greedy matches, Grouping, Match, Search functions, matching/searching, Sub function splitting a string, Replacing text, Flags.

Advanced concepts in python:

- ✓ **Iterators**
- ✓ **Generators**
- ✓ **Closure's**
- ✓ **Decorators**
- ✓ working with JSON files
- ✓ working with CSV files

