

Python

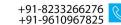
• Introduction To Python

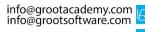
- Application Areas of Python
- Python Implementations
 - 1. Cpython
 - 2. Jython
 - 3. IronPython
 - 4. PyPy
- Python Versions
- Installing Python
- Python Interpreter Architecture
 - 1. Python Byte Code Compiler
 - 2. Python Virtual Machine(PVM)
- Writing and Executing First Python Program
 - Using Interactive Mode
 - Using Script Mode
 - 1. General Text Editor and Command Window
 - 2. IDLE Editor and IDLE Shell

Python Language Fundamentals

- 1. Character set
- 2. Keywords
- 3, Comments
- 4. Variables
- 5.Literals
- 6. Operators Reading input from console
- 7. Parsing string to int, float









Python Conditional Statements

- 1. If statement
- 2.If else statement
- 3.If elif statement
- 4.If elif else statement
- 5. Nested if statement

Looping Statements

- 1.While loop
- 2.For loop
- 3. Nested loops
- 4. Pass, break and continue keywords

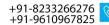
Standard Data Types

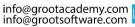
- 1.Int, float, complex, bool, nonetype
- 2.Str, list, tuple, range
- 3.Dict, set, frozenset

String Handling

- 1.What is string
- 2. String representations
- 3. Unicode string
- 4. String functions, methods
- 5. String indexing and slicing
- 6. String formatting























Python List

- 1. Creating and accessing lists
- 2.Indexing and slicing lists
- 3.List methods
- 4. Nested lists
- 5.List comprehension

Python Tuple

- 1.Creating tuple
- 2.Accessing tuple
- 3.Immutability of tuple

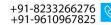
Python Set

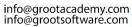
- 1. How to create a set
- 2. Iteration over sets
- 3. Python set methods
- 4. Python frozenset

Python Dictionary

- 1.Creating a dictionary
- 2. Dictionary methods
- 3. Accessing values from dictionary
- 4. Updating dictionary
- 5. Iterating dictionary
- 6.Dictionary comprehension

































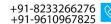
Python Functions

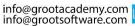
- 1. Defining a function
- 2. Calling a function
- 3. Types of functions
- 4. Function arguments
 - 4.1 Positional arguments, keyword arguments
 - 4.2 Default arguments, non-default arguments
 - 4.3 Arbitrary arguments, keyword arbitrary arguments
- 5. Function return statement
- 6.Nested function
- 7. Function as argument
- 8. Function as return statement
- 9. Decorator function
- 10.Closure
- 11.Map(), filter(), reduce(), any() functions
- 12. Anonymous or lambda function

Modules & Packages

- 1.Why modules
- 2. Script v/s module
- 4.Importing module
- 4. Standard v/s third party modules
- 5. Why packages
- 6. Understanding pip utility

























File I/O

- 1. Introduction to file handling
- 2. File modes
- 3. Functions and methods related to file handling

Regular Expressions(Regex)

- 1. Need of regular expressions
- 2.Re module
- 3. Functions /methods related to regex
- 4. Meta characters & special sequences

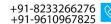
Object Oriented Programming

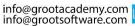
- 1.00P Principles
- 2. Defining a Class & Object Creation
- 3. Inheritance
- 4. Encapsulation
- 5.Polymorphism
- 6.Abstraction
- 7. Garbage Collection
- 8.Iterator & Generator

Exception Handling

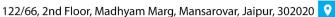
- 1. Difference Between Syntax Errors and Exceptions
- 2. Keywords used in Exception Handling
 - 2.1 try, except, finally, raise, assert
- 3. Types of Except Blocks
- 4.User-defined Exceptions

























Multi-Threading Programming

- 1. Multiprocessing v/s Multithreading
- 2. Need of threads
- 3. Creating child threads
- 4. Functions / methods related to threads
- 5. Thread synchronization and locking

Python Database Connectivity

- 1.Database Drivers and connectors
- 2. Creating connection object
- 3. Understanding cursor object
- 4. Executing SQL statements using cursor
- 5. Fetching records from cursor
- 6. Storing and retrieving Date and Time

Standard Library in Python

- 1.Os and Sys Module to interact with Operating System
- 2. Shutil Module to copy, paste and delete files
- 3. Time & Datetime Module for time management



