

PYTHON COURSE CONTENT

1. PYTHON OVERVIEW

- Introduction to Python
- The Birth of Python
- Python Timeline
- Python Vs other languages
- Features of Python
- Versions of Python

2. PYTHON ENVIRONMENT

- Installation of Python on WINDOWS and LINUX
- Path settings for Python
- Python Documentation
- Getting Help
- Working with Python Command Line Shell
- Working with Editors and IDE's
- Basic Syntax
- Running Python Scripts on WINDOWS
- Running Python Scripts on LINUX
- Working with python cloud
- Executing Python Scripts on Android
- Executing Python Scripts on IOS
- Working with IDLE

3. GETTING STARTED

- Keywords
- Data Types
- Fundamental Data Types
- Collection Types
- Number systems
- Mutable objects vs. Immutable objects
- Iterable objects vs. non Iterable objects
- Variables
- Naming Conventions
- print(), type(), id(), input() Functions
- Type Conversion functions

4. OPERATORS

- Magic methods
- Working mechanism of operators in python
- Arithmetic Operators
- Relational Operators
- Logical Operators
- Assignment Operators
- Short Hand Assignment Operators
- Bitwise Operators
- Membership Operators
- Identity Operators
- Precedence of operators
- Evaluating expressions

5. FLOW CONTROL

- About Flow Control
- Elements of flow control
- Block/Clause
- Conditional Statements

6. COLLECTIONS

- Introduction to collections
- LIST
 - list objects creation
 - list indexing and slicing

- Loop Statements
- Break statement
- Continue statement
- Pass statement
- Working with infinite loops
- Nested conditional statements
- Nested loops

- applying iterations on list objects
- nested lists
- eliminating duplicate elements of list
- working with methods of list
- sorting elements of a list
- searching for a required element in list
- list comprehensions

•TUPLE

- tuple objects creation
- tuple indexing and slicing
- applying iterations on tuple objects
- nested Tuples
- storing list objects in Tuples
- differences between list and tuple
- working with methods of tuple

•STR

- What is String?
- Single-quoted string literals
- Triple-quoted string literals
- String Indexing
- String Slicing
- Working with String Functions
- Working with String Methods
- String multiplication and concatenation
- Reversing a string
- Searching for a required character/substring

•SET

- creating set objects
- applying iterations on set objects
- performing set operations on set objects
- working with methods of Set
- set comprehensions

•DICT

- creating dictionary objects
- working with dictionary keys and values
- applying iterations on dictionary objects
- working with methods of dictionary
- merging dictionaries
- dictionary comprehensions

7. FUNCTIONS

- Defining a function
- Calling a function
- Function Parameters
- Types of parameters
- Default parameters, non-default parameters
- Arbitrary arguments
- Types of arguments
- key word and non-key word arguments
- Return statement in functions
- Handling return values
- Global variables and Local variables
- Passing collections to a function
- First Class, Higher Order and Partial Functions
- Lambda functions/ anonymous functions

9. PACKAGES

- Introduction to packages
- Defining packages
- Importing from packages
- Defining sub-packages
- Importing from sub packages

11. FILE HANDLING

- What is a file?
- Opening a file

8. MODULES

- What is a module?
- Types of modules
- The import statement
- Module Aliases/renaming a module
- from ... import
- Built in properties of a module
- dir() function
- Creating user defined modules
- Module search path
- Command line arguments
- Working with pre-defined standard modules
- Built-in modules: math; os; sys;

10. EXCEPTION HANDLING

- Syntax Errors
- Runtime Errors
- What is Exception?
- Need of Exception handling
- Predefined Exceptions
- Predefined Exceptions Hierarchy
- try, except and finally clauses
- Named except block
- Default except block
- Handling Multiple Exceptions
- Nested try, except and finally blocks
- User defined Exceptions
- Raise, assert statements
- Return statement in try, except and finally

12. OBJECT ORIENTED PROGRAMMING

- Introduction to OO programming
- Encapsulation

- Various file modes
- Reading data from a file
- Writing data to a file
- Closing a file
- Working with the methods of file object
- Replacing the content of file
- Working with Directories
- Handling IO Exceptions
- With statement

- Defining Classes
- Creating objects
- Class variables and instance variables
- Constructors
- Defining methods
- Non static methods
- Static methods
- Class methods
- Diff between functions and methods
- Inheritance
- Types of inheritances
- Polymorphism (over loading & over riding)
- Super() statement
- Data hiding
- Accessing hidden properties of a class
- Custom exception classes
- Built in properties of class
- Inner classes
- Sorting custom objects

13. GARBAGE COLLECTION

- Introduction to garbage collection
- Referenced versus unreferenced objects
- Object reference count
- Garbage collector
- Objects Garbage collection
- Destructors
- Del statement
- Collections Garbage collection

14. REGULAR EXPRESSIONS

- Introduction to regular expression
- Simple character matches
- Special characters
- Character classes
- Quantifiers
- Forming regular expressions
- Matching at beginning or end
- Compiling regular expressions
- Match(), Search() and sub() functions
- Splitting a string
- Replacing text

15. GUI PROGRAMMING

- Overview
- The main window object
- Widgets
- Colors and fonts

16. PYTHON CLOSURES

- Nested functions in python
- Nonlocal variable in a nested function
- Defining a Closure Function
- When to use closures?

- GUI layout
- Event handling

- Benefits of Python Closure

17. DECORATORS

- Functions inside functions
- Functions as parameters
- Functions returning functions
- Simple decorators in python
- Decorators with parameters

18. ITERATORS IN PYTHON

- What are iterators in Python?
- Iterating Through an Iterator in Python
- How for loop actually works?
- Building Your Own Iterator in Python

19. PYTHON GENERATORS

- What are generators in Python?
- How to create a generator in Python?
- Why generators are used in Python?