



Python Syllabus

Introduction to Python 3

- Origin and Goals of Python
- Overview of Python Features
- Getting and Installing Python
- Accessing Python Documentation: Python Enhancement Proposals
- Python's Strengths
- Using Python with Other Programming Languages

Using Python

- Executing Python Programs from the Command Line
- Python Command Line Options
- Environment Variables that Influence Python
- Creating Python GUI Applications (PEP)
 - Standalone vs. Web-Enabled Interfaces
- The Python Standard Library

Language Fundamentals

- Python's Lexical Analyzer
- Using Whitespace to Structure Programs
- Identifiers and Keywords
- Python's Execution Model
 - Naming Objects and Binding
- Python's Data Model



Python Syllabus

- Immutable and Mutable Objects
- Values
- Types
- Creating and Using Variables (PEP)

Flow Control Constructs

- if/elif/else Statements
- Creating Loops with while and for
- Understanding Iterators
- Returning Values with return Statements
- Loop Modification with break and continue
- Returning Generator Iterators with the yield Statement
- Retrieving Iterators with next()

Expressions

- Unary and Binary Arithmetic Operations
- Comparison and Boolean Operations
- Conditional Expressions
- Lambda Expressions
- Order of Operations and Operator Evaluation
- Expression Lists
- Assignment Operations



Python Syllabus

Exception Handling

- Types of Python Exceptions
- Handling Exceptions with try/except/finally
- Triggering Exceptions with raise
- Defining New Exception Types
- Implementing Exception Handling in Functions, Methods and Classes
- Working with the Regular Expression Error Exception

Using the String Object

- Using ASCII and Unicode Strings
- Manipulating Strings with String Methods
- Using the format() Function to Format Strings
- Using Escape Sequences
- Working with Raw Strings

Organizing Code

- Defining Functions
- Calling Functions
- Creating Anonymous Functions
- Altering Function Functionality with Decorator Functions
- Creating Classes with the class Statement
- Creating Objects as Class Instances
- Using Preexisting Classes as the Basis of a New Class
- Using Modules to Group Related Functions, Classes and Variables



Python Syllabus

- Locating and Importing Modules
- Using Packages to Group Modules Together

Arrays, Collections and Dictionaries

- Sequenced Data Structures
 - Arrays
 - Collections
 - Dictionaries
- Creating and Accessing Lists
- Manipulating Lists
- Creating and Accessing Tuples
- Understanding the Differences Between Lists and Tuples
- Using Dictionaries to Create Data Records
- Manipulating Dictionaries Using Dictionary Methods
- Creating Sets
- Performing Set Operations
 - Union
 - Intersect
 - Difference
- Differences Between Sets and Dictionaries
- Using Generators to Return Iterators

Working with Arguments

- Passing Arguments to Functions by Reference and by Value
- Defining Functions with Required Arguments



Python Syllabus

- Defining Functions with Default Arguments
- Defining Flexible Functions that Take Variable Length Arguments

Regular Expressions

- Regular Expression Syntax
- Using Regular Expressions in Python
- Altering Regular Expression Processing with Regular Expression Modifiers
- Using Regular Expression Operators
- Scanning Through Strings Using the search() and match() Methods
- Creating Reusable Patterns by Using the compile() Method

Object Oriented Programming Concepts

- The Object Oriented Programming Paradigm
- Encapsulating Information
- Classes vs. Instances of Objects
- Built-in Class Attributes
- Implementing Class Inheritance
- Using Objects in Code

I/O Handling

- Sending Output to STDOUT Using the print() Method
- Reading Input with the input() Method
- Creating File Objects with the open() Method
- Controlling File Access Modes
- Working with File Object Attributes
- Closing File Objects with the close() Method



Python Syllabus

- Reading and Writing to File Objects with read() and write()
- Using File Processing Functions from the OS Module

Data Management

- Embedding SQLite Databases in Applications
- Best Practices for Data Management
- Storing Data in Local Databases
- Discussing and Understanding the DB API
- Understanding and Using Common SQL Statements
- Connecting to a SQLite Database
- Using Cursors to interact with Data from a Database
- Implementing Error Handling with Database Connections