Python Intermediate

Duration – 3 days
DAY-1

Python Introduction

- What's Python?
- Why do people use Python?
- Some quotable quotes
- A Python history lesson
- The features list.
- Python portability
- Summary

Using the Interpreter

- Python's Interactive Prompt
- Scripting
- Program Execution Model
- Program Architecture: modules.
- How to run Python programs
- Using Python IDEs

working with Variables in Python

- Python Variables
- Naming Conventions & Rules
- Types as Objects

Decision making & Looping.

- Comparison Operations
- The if Statement
- The if Ternary Expression
- The while Loop.
- The for Loop

Python Strings

- Generating Strings in Python
- Immutable
- Common String Methods
- Type Conversion in Python
- Formatting String Output
- Format Specifier
- Variable Substitution
- String Indexing
- String Slicing

- String Iteration

Python's Tuples

- Immutable
- Common Tuples Methods
- Tuples Operations
- Tuples Indexing
- Tuples Slicing
- Tuples Iteration

Python's Lists

- Common List Methods
- The range () Function
- List Operations

Python List Comprehension

- Basic List Comprehensions
- Compound List Comprehensions

Python set data type.

- Understanding & using set data type

Python Dictionaries

- Python Dictionaries
- Assigning Values to Dictionaries
- Dictionary Methods
- Dictionaries vs Lists & Tuples

Basic Input/Output with Files

- Opening Files
- Working with Files
- Controlling Output Location

DAY-2

Creating Python Functions

- Function Basics
- Defining Functions
- Function Polymorphism
- Argument Defaults
- Lambdas
- Local Variables
- Understanding __built-in__
- Preventing Variable Modifications
- Argument Matching Methods
- Keyword Argument Methods

Classes and Objects

- Introduction to OOP using python.
- Classes and class attributes
- Instances and instance attributes
- Binding and method invocation
- Composition, Sub-classing, and Derivation
- Inheritance
- Built-in functions for classes, instances, and other objects
- Privacy and Delegation
- An overview of built-in python classes and modules

Exceptions

- About Exceptions
- Learning how exceptions work in Depth
  - Handling exceptions
  - Raising exceptions
  - Catching exceptions
- Python's Default Exception Handler
- Using Try/Except/Else/Finally Exceptions
- Generating User Defined Exceptions
- Using Asserts
- Exception Classes

Regular Expression in Python

- Using the re module
- Searching with regular expressions
- Replacing with regular expressions
- Reusing regular expressions with recompile
- The match Function.
- The search Function.
- Regular-expression patterns

REST API access.

* Working with JSON
* Using urllib3
* Accessing URLs with urllib3
* Using requests module
* GET, POST using requests module

PANDAS

* Getting Started
* Series

* Data Frames
* Read CSV
* Read JSON
* Analyzing Data Module

NUMPY

* Creating Arrays
* Array Indexing
* Array Slicing
* Data Types
* Copy vs View
* Array Shape
* Array Reshape

Day-3

DJANGO / Flask

Installation and Introduction to Django

- Learn about python programming (inbuilt data structure & Object oriented Programming)
- Learn and explore various web dev tools like VsCode Editor, Git & GitHub,
- Sqlite3, Postman, Virtual Environment, DBeaver, and Other Essential Tools
- Install Django framework and its dependencies.
- Setup Django environment
- Create your first sample Django project.
- Learn about project structure in Django.

Django Admin, Commands and Shell

- Acquire the knowledge to reuse initial built-in Django applications.
- Learn about Django admin.
- Learn about various Django Commands
- Learn about Django Shell

Routing and Views in Django

- Map web URLs to view functions.
- Familiarize with various HTTP methods, including GET, POST, PUT/PATCH, and DELETE
- Create simple views in Django with HTTP response or JSON response and learn about different status codes