

Python Mentorship Outline

Introduction

- Python, interpreters and compilers
- What can be used?
- Installing and setting up IDE
- Environment variable

Python Basics

- Syntax, Variables, and data types
- Control structures(loops and conditions)
- Functions and their usage
- Input and output operations
- Lists, tuples, dictionaries, and sets
- Generators and Iterators

Object-Oriented Programming

- Introduction to OOP concepts(classes, objects, inheritance, polymorphism)
- Why do we need OOP?
- Implementation of class
- Encapsulation, abstraction, inheritance, and polymorphism in Python
- Class Properties and Special methods

Advanced Python Topics

File Handling

- Reading from and writing to files
- Working with different file formats (CSV, JSON, etc.)
- Encodings
- Applying generators for memory-efficient iteration

Error Handling and Debugging

- Exception handling in Python
- Debugging techniques and tools (using pdb, traceback)

Functional Programming in Python

- Understanding functional programming paradigms
- Lambda functions, map, filter, and reduce functions
- Using functools standard library
- Understanding decorators as higher-order functions

Python Standard Library

- Overview of commonly used modules in the standard library (os, sys, datetime, etc.)
- Exploring modules for different functionalities (file handling, data manipulation, etc.)

Working with External Libraries and APIs

- Installing and using external libraries (requests, pandas, numpy, etc.)
- Making API requests and handling responses

Context Managers

- Explaining the context management protocol in Python
- Creating context managers using the with statement and contextlib module
- Implementing custom context managers

Concurrent Programming in Python

- Understanding threads and multi-threading in Python
- Implementing threading using the threading module
- Synchronization and thread safety concepts

Multiprocessing

- Introduction to parallel processing and multi-core utilization
- Utilizing the multiprocessing module for concurrent execution
- Managing shared resources and communication between processes

Asynchronous Programming with asyncio

- Introduction to asynchronous programming and event loops
- Using asyncio for asynchronous I/O operations
- Writing asynchronous code using coroutines and async/await

Networking and Socket Programming

- Overview of sockets and networking basics
- Implementing socket programming in Python with the socket module
- Creating TCP and UDP clients and servers

Miscellaneous Advanced Topics

- Understanding regular expressions for pattern matching and using `re`
- Utilizing Python debuggers (pdb) for code inspection
- Profiling code for performance optimization using cProfile and line_profiler
- Cryptographic Services
- What is behind requests?(`urllib`)
- Development Tools
 - Typing
 - Pydoc
 - unittest