

Course duration

- 5 days

Course Benefits

- Write Python scripts with confidence.
- Code with ease using Python Standard Library script modules.
- Complete math operations with Python.
- Access data in a Python program using string indexing and slicing strings.
- Leverage data structures like sequences, dictionaries, and sets.
- Use flow control tools to manage the execution of your program.
- Strengthen your code with exception handling, testing, and debugging.
- Quickly match, locate, and manage text using regular expressions.
- Handle date and time with confidence.
- Read, write, and edit files using Python.
- Create reusable code to save time later.
- Work with various data sources.
- Use Python's object-oriented features to create flexible code that is easier to maintain.

Available Delivery Methods

Public Class

Public expert-led online training from the convenience of your home, office or anywhere with an internet connection. Guaranteed to run .

Private Class

Private classes are delivered for groups at your offices or a location of your choice.

Course Outline

1. Python Basics
 1. Getting Familiar with the Terminal
 2. Running Python
 3. Running a Python File
 4. Exercise: Hello, world!
 5. Literals
 6. Exercise: Exploring Types
 7. Variables

8. Exercise: A Simple Python Script
9. Constants and Deleting Variables
10. Writing a Python Module
11. print() Function
12. Collecting User Input
13. Exercise: Hello, You!
14. Reading from and Writing to Files
15. Exercise: Working with Files
2. Functions and Modules
 1. Defining Functions
 2. Variable Scope
 3. Global Variables
 4. Function Parameters
 5. Exercise: A Function with Parameters
 6. Returning Values
 7. Exercise: Parameters with Default Values
 8. Returning Values
 9. Importing Modules
 10. Methods vs. Functions
3. Math
 1. Arithmetic Operators
 2. Exercise: Floor and Modulus
 3. Assignment Operators
 4. Precedence of Operations
 5. Built-in Math Functions
 6. The math Module
 7. The random Module
 8. Exercise: How Many Pizzas Do We Need?
 9. Exercise: Dice Rolling
4. Python Strings
 1. Quotation Marks and Special Characters
 2. String Indexing
 3. Exercise: Indexing Strings
 4. Slicing Strings
 5. Exercise: Slicing Strings
 6. Concatenation and Repetition
 7. Exercise: Repetition
 8. Combining Concatenation and Repetition
 9. Python Strings are Immutable
 10. Common String Methods
 11. String Formatting
 12. Exercise: Playing with Formatting
 13. Formatted String Literals (f-strings) (introduced in Python 3.6)
 14. Built-in String Functions
 15. Exercise: Outputting Tab-delimited Text
5. Iterables: Sequences, Dictionaries, and Sets
 1. Definitions

2. Sequences
3. Lists
4. Sequences and Random
5. Exercise: Remove and Return Random Element
6. Tuples
7. Ranges
8. Converting Sequences to Lists
9. Indexing
10. Exercise: Simple Rock, Paper, Scissors Game
11. Slicing
12. Exercise: Slicing Sequences
13. min(), max(), and sum()
14. Converting between Sequences and Strings
15. Unpacking Sequences
16. Dictionaries
17. The len() Function
18. Exercise: Creating a Dictionary from User Input
19. Sets
20. *args and **kwargs
6. Virtual Environments, Packages, and pip
 1. Exercise: Creating, Activating, Deactivating, and Deleting a Virtual Environment
 2. Packages with pip
 3. Exercise: Working with a Virtual Environment
7. Flow Control
 1. Conditional Statements
 2. Compound Conditions
 3. The is and is not Operators
 4. all() and any() and the Ternary Operator
 5. In Between
 6. Loops in Python
 7. Exercise: All True and Any True
 8. break and continue
 9. Looping through Lines in a File
 10. Exercise: Word Guessing Game
 11. The else Clause in Loops
 12. Exercise: for...else
 13. The enumerate() Function
 14. Generators
 15. List Comprehensions
8. Exception Handling
 1. Exception Basics
 2. Generic Exceptions
 3. Exercise: Raising Exceptions
 4. The else and finally Clauses
 5. Using Exceptions for Flow Control
 6. Exercise: Running Sum
 7. Raising Your Own Exceptions

- 9. Python Dates and Times
 - 1. Understanding Time
 - 2. The time Module
 - 3. Time Structures
 - 4. Times as Strings
 - 5. Time and Formatted Strings
 - 6. Pausing Execution with time.sleep()
 - 7. The datetime Module
 - 8. datetime.datetime Objects
 - 9. Exercise: What Color Pants Should I Wear?
 - 10. datetime.timedelta Objects
 - 11. Exercise: Report on Departure Times
- 10. File Processing
 - 1. Opening Files
 - 2. Exercise: Finding Text in a File
 - 3. Writing to Files
 - 4. Exercise: Writing to Files
 - 5. Exercise: List Creator
 - 6. The os Module
 - 7. os.walk()
 - 8. The os.path Module
 - 9. A Better Way to Open Files
 - 10. Exercise: Comparing Lists
- 11. PEP8 and Pylint
 - 1. PEP8
 - 2. Pylint
- 12. Advanced Python Concepts
 - 1. Lambda Functions
 - 2. Advanced List Comprehensions
 - 3. Exercise: Rolling Five Dice
 - 4. Collections Module
 - 5. Exercise: Creating a defaultdict
 - 6. Counters
 - 7. Exercise: Creating a Counter
 - 8. Mapping and Filtering
 - 9. Mutable and Immutable Built-in Objects
 - 10. Sorting
 - 11. Exercise: Converting list.sort() to sorted(iterable)
 - 12. Sorting Sequences of Sequences
 - 13. Creating a Dictionary from Two Sequences
 - 14. Unpacking Sequences in Function Calls
 - 15. Exercise: Converting a String to a datetime.date Object
 - 16. Modules and Packages
- 13. Regular Expressions
 - 1. Regular Expression Tester
 - 2. Regular Expression Syntax
 - 3. Python's Handling of Regular Expressions

- 4. Exercise: Green Glass Door
- 14. Working with Data
 - 1. Virtual Environment
 - 2. Relational Databases
 - 3. Passing Parameters
 - 4. SQLite
 - 5. Exercise: Querying a SQLite Database
 - 6. SQLite Database in Memory
 - 7. Exercise: Inserting File Data into a Database
 - 8. Drivers for Other Databases
 - 9. CSV
 - 10. Exercise: Finding Data in a CSV File
 - 11. Creating a New CSV File
 - 12. Exercise: Creating a CSV with DictWriter
 - 13. Getting Data from the Web
 - 14. Exercise: HTML Scraping
 - 15. XML
 - 16. JSON
 - 17. Exercise: JSON Home Runs
- 15. Testing and Debugging
 - 1. Testing for Performance
 - 2. Exercise: Comparing Times to Execute
 - 3. The unittest Module
 - 4. Exercise: Fixing Functions
 - 5. Special unittest.TestCase Methods
- 16. Classes and Objects
 - 1. Attributes
 - 2. Behaviors
 - 3. Classes vs. Objects
 - 4. Attributes and Methods
 - 5. Exercise: Adding a roll() Method to Die
 - 6. Private Attributes
 - 7. Properties
 - 8. Exercise: Properties
 - 9. Objects that Track their Own History
 - 10. Documenting Classes
 - 11. Exercise: Documenting the Die Class
 - 12. Inheritance
 - 13. Exercise: Extending the Die Class
 - 14. Extending a Class Method
 - 15. Exercise: Extending the roll() Method
 - 16. Static Methods
 - 17. Class Attributes and Methods
 - 18. Abstract Classes and Methods
 - 19. Understanding Decorators

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *would be useful* for this Python class:

- Some programming experience.