Python Programming 3 Days

This course introduces Python 3 depending on your requirement.

Target Audience:

This course is for programmers starting new projects or working on or existing project written in Python 3.

Prerequisites

- No previous knowledge of Python is assumed,
- Experience of another scripting language, such as Perl or PHP, will be an advantage, but is not needed

Delegates will learn how to

- Use the Python to write and run Python 3 apps
- Understand the differences between Python 2 and Python 3
- Recognize simple and complex variable types and select appropriately
- Use Python 3 operators and built-in functions
- Understand procedural control flow in Python 3
- Program file input/output, including persistent data objects.
- Create well organized and efficient code using functions and modules
- Use Object Oriented techniques in Python 3.
- Run and control other programs from Python

Course Outline.

Introduction to Python 3

- What is Python?
- Why Python?
- Running Python interactively
- Python scripts
- Python help
- Anatomy of a Python script
- Modules
- Functions and built-ins

Fundamental Variables

- Python is Object Oriented
- Python variables
- Type specific methods
- Operators and type
- Python types
- Switching types
- Python lists introduced

- Python tuples introduced
- Python dictionaries introduced

Flow Control

- Python conditionals
- Indentation
- What is truth?
- Boolean and logical operators
- Sequence and collection tests
- While loops
- Loop control statements
- For loops
- enumerate
- Counting 'for' loops
- Zipping through multiple lists
- Conditional expressions
- Unconditional flow control

String Handling

- Python strings
- The print function
- String concatenation
- 'Quotes'
- String methods
- String tests
- String formatting
- Other string formatting aids
- Slicing a string
- String methods - split and join

Collections

- Python lists
- Tuple and list slicing
- Adding items to a list
- Removing items by position
- Removing list items by content
- Sorting
- List methods
- Sets
- Exploiting sets
- Set operators
- Python dictionaries
- Dictionary values
- Removing items from a dictionary
- Dictionary methods

- View objects

Regular Expressions

- Python regular expressions
- Regular expression objects
- Regular expression substitution
- Regular expression split
- Matching alternatives
- Anchors
- Class shortcuts
- Flags
- Repeat quantifiers
- Quantifiers
- Parentheses groups
- Back-references
- Global matches

Data Storage and File Handling

- New file objects
- Reading files into Python
- Filter programs – file input module
- Writing to files from Python
- Standard streams
- Random access
- Python pickle persistence
- Pickle protocols
- Build some shelves
- Compression
- Database interface overview
- Example - SQLite from Python
- Connecting Python to MySQL
- Overview of SQLAlchemy – Object Relational Mapper

Functions

- Python functions
- Function parameters
- Variadic functions
- Assigning default values to parameters
- Named (keyword) parameters
- Enforcing named parameters
- Returning objects from a function
- Variables in functions
- Nested functions
- Variables in nested functions
- Function documentation

- Lamdba functions
- Lambda as a sort key
- Lambda in re.sub

Advanced Collections

- Advanced list functions - filter
- List comprehensions
- Set and dictionary comprehensions
- Lazy lists
- Generators
- Generator objects and next - coroutines
- List comprehensions as generators
- Copying collections - problem
- Copying collections - slice solution?
- Copying collections - deepcopy solution

Modules and Packages

- What are modules and packages?
- Multiple source files
- How does Python find a module?
- Importing a module
- Importing names
- Directories as packages
- Writing a module
- Module documentation
- Testing a module
- Python debugger
- Python profiler
- Distributing libraries - distutils

Classes and OOP

- Classes and OOP
- Object-Oriented terminology
- Object-Oriented Programming
- Using objects
- Defining classes
- Defining methods
- Constructing an object
- Special methods
- Operator overload special methods
- Properties
- Properties and decorators
- Class methods
- Inheritance
- Inheritance terminology

Error Handling and Exceptions

- Controlling warnings
- Exception handling
- Exception syntax
- Multiple exceptions
- Exception arguments
- The finally block
- Order of execution
- The Python exception hierarchy
- A common mistake
- The raise statement
- Raising our own Exceptions
- assert