**Basic Python Programming**

**Recommended duration: 25 hours**

**Course Description:**

This training program provides a basic introduction to Python programming

language. The focus of this training program will be more on "pythonic" approach towards problem-solving.

**Target Audience (who should attend):**

* Engineers who wish to learn python program to develop applications or to automate their applications/framework.
* Engineers who wish to prototype new applications.

**Pre-requisites:**

Participants should be comfortable with the following technologies:

* Basic programming background with good understanding of programming language ingredients that include variables and datatypes, flow control statements, and function/procedural programming paradigms.
* Knowledge of any scripting language would be beneficial.

**Course Objectives:**

* Understand the basic programming paradigms using Python 2 and Python3
* Understand builtin datatypes, variables, functions and flow control statements.
* Learn how to use string, tuple, list, bytearray, set, dictionary types effectively.
* Learn "pythonic" idioms and anti-idioms
* Learn about building our own modules and use third party modules

**Course Agenda**

**Day1**

**Introduction to Python**

* Introduction to python programming
* An overview of scripting and the pragmatic development approach.
* An overview on python interpreter
* An overview on Python development tools
* Python 2 vs Python 3 (feature differences)

**Python Scripting**

* Python Scripts in Linux/Unix & Windows
* Whitespace Significance
* Line Termination
* Comments in Python
* Basic Output Generation
* Simple User Input
* Python Modules
* Module Search Paths
* Determining the System Search Path
* input()
* Memory management

**Working with Variables in Python**

* Python Variables
* Naming Conventions & Rules
* Types as Objects
* Variable References & Garbage Collection
* Sequence Types
* Membership Statements
* List Iteration
* Sequence Assignments
* Mutable vs Immutable Objects
* Multi Target Assignments

**Numeric Operations in Python**

* More About Python's Numeric Types
* Numeric Tools
* The Decimal Module
* Operator
* Arithmetic
* Logical
* Relational
* Bitwise
* Special Operators
* Operator Precedence

**Python Compound Statements**

* Python Nesting Recap
* Comparison Operations
* The if Statement
* The if Ternary Expression
* The while Loop
* The for Loop
* Traversing Parallel Sets

**Day2**

**Working with Strings**

* An overview of strings in python
* String operators
* Built-in string manipulation functions
* Built-in string methods
* Special string features in python
* Built-in modules for string handling
* Unicode strings and bytearray

**Lists, Tuples and Sets**

* Common List Methods
* The range() Function
* List Operations
* List Indexing
* List Slicing
* Multi-Dimensional Lists (Matrices)
* Basic List Comprehensions
* Compound List Comprehensions
* Immutable tuples
* Common Tuples Methods
* Tuples Operations
* Tuples Indexing
* Tuples Slicing
* Tuples Iteration
* Multi-Dimensional Tuples (Matrices)
* Set operations
* Applications of sets

**Day3**

**Working with dictionaries**

* Introduction to dictionaries
* Creating, assigning, updating dictionaries
* Dictionary operators, functions and built-in methods
* Dictionaries vs Lists & Tuples

**Implementation of mini project using list, tuples and dictionaries**

**Functions**

* Creating user-defined functions
* Passing functions
* Formal arguments
* Variable-length arguments
* Function Polymorphism
* Argument Defaults
* Lambdas
* Local Variables
* Preventing Variable Modifications
* Argument Matching Methods
* Keyword Argument Methods
* Understanding \_\_builtin\_\_
* A walk-through on various built-in functions

**Day4**

**Modular development**

* Creating modules
* Variable scope
* Understanding namespaces
* Importing modules and module attributes
* Module hierarchy

**File and Directory handling**

* File I/O operations
* Built-in file and directory handling libraries
* fileinput
* stat
* filecmp and dircmp
* pickle modules
* Serialization using json

**Day5**

**Standard Python modules**

* Using the sys module
* sys.argv, sys.path, sys.version
* An overview on \_\_builtin\_\_ and \_\_future\_\_ modules
* Using the os module
* Filesystem/directory functions
* Basic process management functions
* Recursive directory iteration using os.walk
* Using the os.path module
* Determining basename, dirname, path manipulation
* Using the time and datetime modules

**Working with xls files using xlrd and xlwt module**

* Installing xlrd and xlwt module
* Reading xls files using xlrd module
* Reading multiple sheets from spreadsheet(xls)
* Reading row-wise data
* Reading column data
* Creating and writing in xls file using xlwt module
* Updating xls file using xlutils