Python 3.6

Workshop Details:

|  |  |
| --- | --- |
| Duration: | 6 Days |
| Description: | Developing Web and Windows applications using Python 3.6 |
| Objectives: | This session coverts the object oriented and procedural programming concepts of Python. This session helps the participants to understand various programming features of Python like File I/O, Threading, Networking, Database connectivity and integration with C language. |
| Participants’ Entry Profile: | Participants attending this course must have development experience on:   * Knowledge of basic programming concepts * OOPs and Procedure oriented programming * Knowledge of Scripts and services |
| Training Methodology: | The workshop will follow Synergetics methodology of   * Concept Visualization * Active Experimentation * Application Development   The workshop will be 100% Hands-On with each participant having access to system during the session |

Setup Requirements:

|  |  |
| --- | --- |
| Hardware and Software Requirements: | Participant’s as well as Trainer’s Machine are required to have:  Hardware   * Intel Pentium 4 [2+ GHz recommended] * 4 GB RAM * 50 GB HDD space * LAN connectivity * Good Internet connectivity and bandwidth   Software [Installed]   * Windows 8.1 or later * Python 3.6 * JetBrains PyCharm 2017 Community Edition |
| Training Lab Requirements: | Whiteboard 6 feet by 4 feet (minimum)  Whiteboard markers – Red, Blue, Green, Black  Video Projector (1024 X 768 resolutions) |
| Virtual Lab Requirements:  [Optional] | Virtual labs can be provided for participants, that provides completely configured platform to work with. |

Course Contents:

Day 1

Introduction to Python 3.6

* What is Python?
* Benefits and feature of Python
* Where to use Python
* Interpretation process
* Installing and configuring Python

Fundamentals of Python

* Data types and operators
* Immutable and mutable types
  + Numbers, strings and Tuples
  + Lists, Dictionary and Sets
* Loops in Python
* Type conversions
* String operations
  + Formatting, encoding and decoding
  + Secret codes
* Python Scripts on UNIX/Windows
* Command Line Parameters and Flow Control
* Built-in Functions, Math Operators and Expressions
* Conditional branching
  + If, elif, then, otherwise
  + Combining conditions

Repeating events

* Turtle
* Loops, nested loops and loop variables
* Loop an unknown number of times
* Looping issues

Sequences and operations

* Lists and tuples
  + Slicing, indexing, searching
  + Iterating sequences
* Operators in sequences
* X range() function
* List comprehensions
* Generator expressions

Functions in Python

* Functions and function parameters
* Positional, keyword and default arguments
* Implementing variable-length argument lists
* Global variables, scopes and return values
* Lambda functions
* Sorting collections of collections, dictionaries and Lists

Day 2

Modules and packages

* Using modules
* Import statement for modules
* Controlling the Import of Everything
* Splitting a Module into Multiple Files
* Module searching path
* Package installation
* Referencing functions from modules by qualification
* Accessing the Standard Library

Object-oriented programming with Python

* Creating classes and objects
* Encapsulating attributes and methods in classes
* Initializing objects with constructors
* Accessing and modifying attributes with methods
* Overloading operators
* Reusing functionality through inheritance
* Extending methods from base classes
* Overriding methods for dynamic behavior
* Tracing the scope in the namespace
* Enhancing functionality with class decorators

Day 3

Exception handling

* Error and exception handling
* Error types
* Gracefully handling exceptions
* Handling multiple exceptions
* Exception hierarchy
* Raising Exceptions
* User-Defined Exceptions
* Defining Clean-up Actions

Reading and writing files

* Reading and writing text files
* Reading and writing binary files
* Importing OS module for directory management
* File objects
* Saving structured data with JSON

Regular expressions

* Compiling a pattern.
* Flags - ignorecase, dotall
* Working with multiple flags.
* Search vs match.
* Raw string notations.
* Special characters
  + Globbling characters
  + Anchors
  + Character sets
* Grouping

Day 4

Working with Databases

* Creating a Relational Database connection
* Instantiating cursors to access data
* CRUD Operations
* Retrieving desired datasets
* Updating with action statements

GUI Programming with Python

* Defining GUI classes and frames using Tkinter
* Placing widgets
* Forms and controls
* Providing menu items
* Responding to mouse clicks
* Binding event handlers
* Overview of wxPython
* Overview of JPython

Day 5

Developing web applications

* Request processing pipeline
* Mapping request to python scripts
* Developing MVC with python
  + Processing requests with Python controllers
  + Integrating the model with databases
* Producing HTML with Django templates

Multi-threading in Python

* Understanding the Thread module
* Creating and starting new thread
* Synchronizing Threads
* Creating thread pool
* Performing Simple Parallel Programming
* Storing thread-state
* Implementing Publish/Subscribe messaging

Day 6

Networking

* Interacting with HTTP Services
* Creating a TCP Server
* Creating a UDP Server
* Creating a Simple REST-Based Interface
* Understanding Event-Driven I/O
* Sending and receiving large arrays

C Extensions

* Accessing C Code Using ctypes
* Writing C Extension Module
* Writing an Extension Function That Operates on Arrays
* Managing Opaque Pointers in C Extension Modules
* Defining and Exporting C APIs from Extension Modules
* Calling Python from C
* Releasing the GIL in C Extensions
* Wrapping C Code with Swig
* Wrapping Existing C Code with Cython
* Using Cython to Write High-Performance Array Operations
* Turning a Function Pointer into a Callable
* Passing NULL-Terminated Strings to C Libraries
* Passing Unicode Strings to C Libraries
* Converting C Strings to Python
* Passing Filenames to C Extensions
* Passing Open Files to C Extensions
* Consuming an Iterable from C