Programming in Python

Basic, Intermediate, and Advanced Techniques

COURSE DESCRIPTION

This course presents both the programming interface and the techniques that can be used to write procedures in **Python** on **Unix** / **Linux** systems.

COURSE OBJECTIVES

Each participant will be able to use **Python** techniques and commands to write scripts to perform various user and administrative tasks.

EXECUTION

Each participant will have their own functioning complete **Linux** or **Windows (virtual)** server, including the **Python** programming language environment and all lab files.

COURSE TOPICS

Programming Environment Tools (Linux)

- Review of (**bash**) shell features
- Key capabilities of the **vim** editor
  - modes
  - initialization file (**.vimrc**)
  - key mapping (usage for **Python**)

Interpreted Languages

- Overview of features and benefits
  - **shell scripting**
  - **awk** / **gawk**
  - **Perl**
  - **Ruby**
  - **tcl/Tk**
  - **Python**

Programming in Python

Basic, Intermediate, and Advanced Techniques

COURSE TOPICS

Introduction to Python Scripting
- History, versions, ports
- **Python** capabilities
- Comparison with other interpreted languages

Writing Python Scripts
- **Layout of a Python procedure**
  - signature
  - comments
  - module **import**ation
  - column format
  - documentation (**docstrings**)
    - **pydoc** - generating **man** or **html** pages
  - syntax checking via **pylint**

- **Execution methods**
  - one-liners
  - scripts
  - command line interaction
  - batch file interaction

- **Datatypes**
  - numeric
  - lists (indexed arrays and tuples)
  - sequences
  - strings
  - dictionaries
  - Defining and using objects
  - Operators
    - relational
    - conditional
    - patterns (regular expressions)
  - Serialization (**pickle**)

Programming in Python

Basic, Intermediate, and Advanced Techniques

COURSE TOPICS

Writing Python Scripts

Python programming constructs
- looping statements
- decision statements

Python file **I/O**
- using **ARGV** value(s)
- using **sys.stdin** and **fileinput.input( )**

Python interaction with the **Linux** system
- file handling functions (**os** and **os.path**)
- process and thread creation

Subroutines and functions
- Definition and declaration
- Passing arguments and calling methods
- Return values

Linux usage of **Python** interfaces
- **ftp**
- **telnet**

An Introduction to Python Classes
- **OO** programming
- defining classes
- initializers, constructors, and destructors
- instance methods
- properties
- packaging

Programming in Python

Basic, Intermediate, and Advanced Techniques

COURSE TOPICS

Operating System Services

- the **os** Module
- environment variables
- launching external programs
- paths, directories and filenames
- dates and times
- the **Time** module

Distributing Modules

- installing packages
- ways to distribute code
- overview of distutils
- preparing for distribution
- creating a source distribution
- creating built distributions
- **setup.py** options
- **setup.py** commands

Network Programming

- sockets
- socket options
- client concepts
- server concepts
- multi-tasking network server
- multi-threading network server

Multiprogramming

- multi-processing (advantages / disadvantages)
- daemon transition (native)
- **POSIX** threads
- the **Python** thread manager
- **threading** module

Programming in Python

Basic, Intermediate, and Advanced Techniques

COURSE TOPICS

Database
- **Python cx_Oracle** (database interface)
- connection object
- cursor object
  - embedding **SQL** statements
- fetching objects

Graphical User Interface (GUI)
- **Python Tkinter (**interface to **Tcl/Tk)**
- widget access
  - main window
  - frame,button,menu,boxes (15 types)
  - main event loop

Processing XML Data in Python
- **XML** data layout
- reading / parsing **XML** data in **Python**
  - via the **DOM** library
  - via **SAX** parsing
  - via **ElementTree**
- writing **XML** data in **Python**

COURSE DURATION

This course normally requires **two** to **five** days, approximately 50 % lecture, and 50 % lab time (duration depends upon the topics that are covered).

COURSE PREREQUISITES

It is helpful if the participants have had some experience with an interpreted language (shell scripting, **Perl**, etc), but is not mandatory. No experience with the **Linux** system is required.