Programming in Python

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 **Windows** 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 **(virtual)** or **physical Windows** system, including the **Python** programming language environment and all lab files.

COURSE TOPICS

Writing Python Scripts (Review)
- history, versions, ports
- layout of a **Python** procedure
  - running on a **Windows** platform
  - 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
- interactive / debugging mode

Programming in Python

Basic, Intermediate and Advanced Techniques

COURSE TOPICS

Advanced Data Structures Definition and Access
- using **sys.stdin** and **fileinput.input( )**
- generator functions
- arrays that contain arrays
- arrays that contain dictionaries
- special properties of dictionaries
- dictionaries that contain arrays
- dictionaries that contain dictionaries
- (command line) option processing
- functions with named parameters

Python Interaction with the Operating System : part 1
- the **os** Module
  - environment variables
  - launching external programs
  - paths, directories and filenames
  - dates and times
- the **Time** module
- handling (system) signals

Python Interaction with the Operating System : part 2
- file handling functions (**os** and **os.path**)
- using **ARGV** value(s)
- process and thread creation
- converting to daemon level execution

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

Programming in Python

Intermediate and Advanced Techniques

COURSE TOPICS

An Introduction to Python Classes
OO programming
defining classes
initializers, constructors, and destructors
instance methods
properties
packaging
Serialization (**pickle**)

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

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

Unit testing
- **unittest Python** module
  - writing test cases
  - defining assertions
  - exceptions
  - edge cases

Programming in Python

Intermediate and Advanced Techniques

COURSE TOPICS

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

Database Interaction
- **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

COURSE DURATION

This course normally requires **two to five** days, dependent upon the topics selected, approximately 50 % lecture, and 50 % lab.

COURSE PREREQUISITES

It is helpful if the participants have had some experience with the **Python** language (basic constructs, data types, functions). No experience with the **Linux** system is required.