L8 - Python

Programming with the Python Language

Objectives

- Master the Python language basics
  - Modular approach
  - Object Oriented features
  - Exception mechanism
- Understand the specifics of the Python interpreter

Course environment

- Printed course material (in English).
- One PC for two trainees.
- Documentation and exercise solutions on CDROM.

Prerequisites

- There is no specific prerequisites

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Course Outline

First Day

Python overview

- History
- Installing Python
- The Python interpreter
- The Python command line

Python language basics

- Comments and instructions
- Variables, data and assignment
  - Identifiers and keywords
  - Basic types
- Expressions and operators
  - Arithmetic operators
  - Relational operators
  - Choice operators
- Simple input/output
- Data structures
  - Sequences
  - Dictionaries

    - Sets

Python program structure

- Complex instructions
    - Instruction sequences
    - Conditions and switches
    - Loops and iterators
- Functions and procedures
    - Parameters
    - Local and global variables
    - Default parameter values
    - Calling functions
- Name spaces
- Modules
    - Creating modules
    - Importing functions from modules

Second Day

Advanced data structures

- Character strings
    - Indexing and slicing
    - Concatenation and repetition
    - Unicode strings
    - Converting strings
    - Formatting
    - Character strings and byte strings
    - Lists
    - Advanced slicing
    - Insertion and extraction
    - List operations
    - List copy
- Tuples
- Dictionaries
    - Creating a dictionary
    - Dictionary operations
    - Keys and data types

Object oriented programming

- Overview
    - Classes and instances
    - Attributes and operations
    - Relations and links
    - Inheritance and polymorphism
- Rationale
    - Divide and conquer
    - The encapsulation paradigm
    - Modularity and security
- Advantages
    - Increased security
    - Incremental development
    - Code reuse

Python as an Object Oriented Language

- Python class design
  - Everything is an object
- Defining classes
  - Instance and class attributes
  - Static and instance methods
  - Constructors

Third Day

Exceptions

- Exceptions and errors
  - Error types
  - Exception types
  - Assertions
- Handling exceptions
  - Try blocks
  - Except (catch) blocks
  - Getting information about the exception
  - The finally block
- Raising exceptions
- The with statement

Input-Output

- User interaction
  - Writing to the terminal
  - Reading from the terminal
- Files
- Persistent objects
  - Explicit serializing with repr
  - Implicit serializing with pickle

Graphical interfaces in Python

- A lot of graphic toolkits
  - PyQt
  - PyGTK
  - wxPython
  - Tkinter
- Graphical object programming with Tkinter
  - Event-driven programming
  - The Tkinter widgets
  - Widget layout
  - Drawing graphics on a canvas
  - Creating a custom widget

Fourth Day

The Python standard library

- The standard modules
  - sys
  - logging

    - urllib and json
- Network programming
    - Sockets
    - Client and server programs
- Multithread programming
    - Creation de threads
    - Sharing data and mutual exclusion
    - Synchronisation and communication

Advanced language features

- Advanced functions
    - Returning multiple values
    - Getting list or tuple parameters
- The special (__xxxx__) methods and attributes
- Metaclasses
- Callable objects
- Containers
    - Creating a container
    - Indexing a container
    - Iterating through a container
- New numeric types
- Dynamic programming
    - Functions creating new functions (lambda)
    - Executing and evaluating character string