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

- Theoretical course
  - PDF course material (in English) supplemented by a printed version for face-to-face courses.
  - Online courses are dispensed using the Teams video-conferencing system.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

Prerequisites

- There is no specific prerequisites

Target Audience

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

Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the traineein his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed by quizzes offered at the end of various sections to verify that the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
  - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites,in agreement with their company manager if applicable.

Plan

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

Renseignements pratiques

Inquiry : 4 days