Introduction to programming with Python

Hands-on course of 2 days - 14h
Ref.: THO - Price 2024: €1 450 (excl. taxes)

<div style="color:teal">━━━</div>

EDUCATIONAL OBJECTIVES

At the end of the training, the trainee will be able to:

Structure programs using an algorithm

Master lexical and syntactical elements of a language in order to write a program

Compile and execute a program

Debug and test a program

HANDS-ON WORK

This course is 60% hands-on work, performed in Visual Basic (course code INP), Java (course code INJ), C# (course code OGR) or Python (course code THO), as needed.

THE PROGRAMME

last updated: 07/2021

1) A program

- What is a program?
- What is a language? The various paradigms.
- What is an algorithm? Pseudo-language.
- Compilers. Executables.

Exercise : Overview of different languages. Writing a first algorithm in pseudo-language.

2) Genesis of an initial program

- Writing a program: Syntax and instructions.
- Compiling and executing the program.
- What is a library? Its role, its usage.

Exercise : Writing, compiling, and executing a first Python program.

3) Programming rules

- Naming convention. Syntax convention.
- Using comments. Why add comments to your code?
- Improving program legibility: Indenting the code, breaking down the code, etc.

4) Variables.

- What is a variable?
- Primitive types: integers, character strings, real numbers, others.
- Declaring, defining, and initializing a variable. Constants.
- Entry, display, assignment, type conversion.
- Organizing your data into tables.

Hands-on work : Working with variables.

5) Operators and expressions

- Different operators (addition, equality, etc.).
- Combining operators.
- Boolean expressions.

Exercise : Handling operators and Boolean expressions.

6) Control structures

- Alternative selections (if, if-then-else, etc.).
- Instruction blocks (notion of Start... End).

- Iterative loops (while-repeat, repeat-until, for-from-to).
- Embedding instructions.
Exercise : Control structures

7) Procedures and functions.
- Definitions: Procedure, function. Benefits.
- Passing parameters.
- A function's return value. Calling functions.

8) Maintenance, debugging, and program testing
- How to read and interpret different error messages.
- Using a debugger: Executing a program step-by-step, stopping points, inspecting variables.
- Unit tests.
Hands-on work : Using a debugger to check how the programs are running.

DATES

Contact us