Training Catalogue
03/06/2020

KAÏNA-COM
TRAINING CATALOGUE

Python Fundamentals

Leading programming language that enables development of tools and applications various purposes

KPYT002 – Python Fundamentals

Reference	KPYT002
Experience	☒ Beginner ☐ Intermediate ☐ Advanced
Duration	Training Program: • 2 days
Training Method	☐ I: i-learning, individual training (web-based training) ☒ V: v-learning, virtual class ☐ C: c-learning, classroom training **KAÏNA-COM** LE CARRÉ HAUSSMANN II, 6 Allée de la Connaissance 77127 Lieusaint - France
Price	1.390,50 € HT
Prerequisite	Basic concepts of programming.
Audience	Developer programmers.

Continued on next page

KPYT002 – Python Fundamentals, Continued

Objective

The course will enrich applicable knowledge in programming using Python. The course will include theoretical knowledge and vast hands-on practice in every field that will be acquired. Participants will gain ability in programming through scripts in the Python programming language. The course will include the following topics:

- The unique characteristics of the language including syntax rules, advantages and disadvantages of the programming language in comparison to other programming languages.
- Installation of a full working environment and creation of input and output.
- The participants will have full capability of use and development within the Python environment including the use of editor and debugger.
- Use of variables and casting of variables (int, float, complex, boolean).
- Use of 'if' statements – if, else, elif.
- Boolean operators and numerical operators.
- Use of strings variables; use of string functions including slicing.
- Types of loops and related commands such as: break, continue, pass, else.
- Use of modules including – module creation and import.
- Advanced data types: List, Tuple, Sets, Dictionary – creation, manipulations on variables.
- Reading and writing files: creation, open, close, read, write, append, etc.
- File types – Random and sequence.
- Functions – definition, usage, capabilities, return values.
- Exception Handling
- Python modules – Introduction, creation, import.
- Performance consideration while developing with Python.

All hands-on topics that will be learnt throughout the course will be exercised to gain practical expertise. Summary exercises will include programming and developing that requires combined use of the various topics.

Continued on next page

KPYT002 – Python Fundamentals, Continued

Course Contents

Course Contents :

Table 1: KPYT002 - Course Contents

Chapter	Description
Day 1	• Introduction – Programming languages, uniqueness of Python as a programming language, interpreter • Installing of full working environment • Familiarization of working environment, creation of input and output • Use of comments, Docstrings • Class practice • Presentation of Python variables: – int, float, Complex, String, Boolean – Use of casting when needed – Strings – functions, slicing – Class practice • Flow control – conditions – if, elif, else • Boolean operators • Numeric operators including power and modulo – Class practice • Presentation of the debugger and practical use • Class practice – writing scripts and use of the debugger for troubleshooting • Loops – for and while (Syntax, Differences) • Additional commands for control: break, continue, pass, else – Class practice • Advanced data types: – List – creation, assignment, access – List functions and operators – Slicing / Sort and reverse sort – Class practice

Continued on next page

KPYT002 – Python Fundamentals, Continued

Course Contents, continued

Chapter	Description
Day 2	• More advanced data types: – Tuple, Sets, Dictionary – creation, assignment, access – Class practice• Reading and writing files (Random and Sequence) – Creation, open, close – Read, write, append – Class practice• Functions – definition, usage, capabilities, return values, documentation – Class practice • Exception Handling – Introduction – When is exception handling actually required? – Examples – Class practice • Python modules – Introduction – Using modules, modules creation, import modules – Class practice • Performance consideration while developing with Python – Improve Python performance and code efficiency – function call in loop definition – list vs. dictionary – more examples – Class practice and scripts run-time measurements
The End	• Summary • Q&A • Evaluation