Updated on 18/03/2025

Sign up

Python training: Object programming

5 days (35 hours)

Presentation

This Python: Object Programming course will give you the knowledge you need to program with the Python language. This course will cover the syntax, tools and best practices for developing in Python, so that you can benefit from the full power of this technology. We will present frameworks, libraries, tools and functionalities most commonly used in business, such as access, XML file manipulation, user interface and web interface creation.

In this training, as in all our training courseswe'll be using the latest stable version (Python 3.13 at the date of this article).

Objectives

- Master the syntax of the Python language
- Acquire the essential notions of Object Programming
- Design graphic interfaces
- Know how to apply Python module functions
- Master program testing and evaluation tools

Target audience

- Developers
- Architects
- Engineers
- IT Project Manager

Prerequisites

- Basic programming skills.

Prerequisites

- Python interpreter installed
- A development environment such as PyCharm or Visual Studio Code
- PDF reader and archiving software (.zip)
- Network access rights for installing libraries

Our Python training program: Object programming

Language presentation

- The origins of language
- Installing the Python interpreter on different systems
- Development environment for Python

Language syntax

- Instructions and expressions
- Basic types
- The collections
- Control structures
- The functions
- Special treatment for collections
- Python and the functional approach
- Date management
- Structuring code (modules, packages, imports)

Exception handling

- The concept of exceptions
- Intercept and handle exceptions
- When to use exceptions

Code quality

- Static code analysis tools (Pylint, Flake8...)
- Unit test modules
- Test-driven development

Object-oriented programming

- Object-oriented programming concepts
- Introducing UML
- Notions of class, attributes, methods
- Python implementation
- Visibility and encapsulation in Python
- Special methods in Python
- Heritage and abstraction
- Notions of polymorphism
- Polymorphism and duck typing

Files and the file system

- Reading and writing in a text file
- The file system, historical (but indispensable) tools
- Using the Pathlib module
- An update on structured files

Python and databases

- Accessing databases with Python
- Code structuring
- Transaction management case study

Regular expressions

- Introduction to the concept of regular expressions
- Regular expressions in Python

Creating graphical user interfaces with Python

- Object-oriented programming concepts
- Representation: introduction to UML
- Declaring and using classes
- Visibility and encapsulation in Python
- Special methods
- Relationships
- UML: the limits of graphical representation
- Heritage and abstraction
- Notions of polymorphism
- Polymorphism and duck typing
- Data Classes

Exception handling

- Programming principles for graphical user interfaces
- Overview of the various libraries

- Containers and widgets
- Event management
- Code organization

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives with regard to the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.