Publication date : 03/15/2024

Course : Developing in object-oriented Python, API Society certification

Practical course - 4d - 28h00 - Ref. YPT
Price : 3090 CHF E.T.

★★★★½ 4,8 / 5

Python is a programming language widely used in web applications, software development, data science, finance, cartography, artificial intelligence and machine learning. This course will teach you basic syntax, different data types, conditional and repeating structures, functions, modules, packages, file manipulation and exception handling.

Teaching objectives

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

- ✓ Know the syntax of the Python language
- ✓ Knowing and manipulating fundamental data types
- ✓ Use and define functions, modules, packages, exceptions, manipulate files
- ✓ Understand the theory of object-oriented programming and how to put it into practice in Python
- ✓ Know and use the essential modules of the standard library

Intended audience

Developers, engineers, project managers close to development.

Prerequisites

Knowledge of a programming language.

Certification

The certification exam takes place online, off-line and in French, in the month following the training course. It consists of a 20-minute theory test - 40 True/False MCQ-type questions and information to be entered, and a practical programming test (code exercise).

PARTICIPANTS

Developers, engineers, project managers close to development.

PREREQUISITES

Knowledge of a programming language.

TRAINER QUALIFICATIONS

The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

ASSESSMENT TERMS

The trainer evaluates each participant’s academic progress throughout the training using multiple choice, scenarios, hands-on work and more.
Participants also complete a placement test before and after the course to measure the skills they’ve developed.

Practical details

Hands-on work

Individual and group practical work, collective reflection.

Teaching methods

Active pedagogy encouraging personal involvement and exchanges between participants.

Course schedule

1 Introduction

- History (author, date of first version).
- Python versions (branches 2 and 3).
- Language features (multi-paradigm, strong dynamic typing, clear syntax).
- Panorama of the standard library.
- Expansion modules and pip control.
- How the PYC bytecode interpreter works.
- Official CPython interpreter and other interpreters (micropython, brython, pypy, numba).
- Resources (python.org website, access to documentation).
- Help() function and document chains.
- Principle of indentation to delimit instruction blocks.
- Commentary.
- Reserved keywords.
- Naming conventions.
- Interactive interpreter.
- Stand-alone program.
- Basic built-in functions: print(), type(), input(), len().

2 Non-modifiable data types

- Usefulness of non-modifiable types (memory optimization), id() and hash() functions, is operator.
- Principle of ordered sequences (str, tuple and list) and collections (dict, set).
- Boolean (bool), True and False objects.
- Number (int, float, complex), constructors, operators >>, <<, |, &, ^, //, % and **.
- Exponential, binary, octal and hexadecimal notation, hex(), oct(), bin(), chr(), ord() functions.
- Unicode string (str), definition with single and double quotation marks, multiline strings, raw mode.
- Positive and negative signalling, slice, +, * and in operators, iteration.
- Essential str methods: split(), replace(), lower(), upper(), strip(), join().
- Formatted string (%s, %d, %f), format() method and f-string.
- Principle of variable depacking.
- Byte array, manufacturer.
- Tuple (tuple), constructor, indexing, iteration, +, * and in operators, count() and index() methods.
- None object and repr() function.

TEACHING AIDS AND TECHNICAL RESOURCES

• The main teaching aids and instructional methods used in the training are audiovisual aids, documentation and course material, hands-on application exercises and corrected exercises for practical training courses, case studies and coverage of real cases for training seminars.

• At the end of each course or seminar, ORSYS provides participants with a course evaluation questionnaire that is analysed by our instructional teams.

• A check-in sheet for each half-day of attendance is provided at the end of the training, along with a course completion certificate if the trainee attended the entire session.

TERMS AND DEADLINES

Registration must be completed 24 hours before the start of the training.

ACCESSIBILITY FOR PEOPLE WITH DISABILITIES

Do you need special accessibility accommodations? Contact Mrs. Fosse, Disability Manager, at psh-accueil@orsys.fr to review your request and its feasibility.

3 Modifiable data types

- Lists (list), constructor, indexing, integration, +, * and in operators.
- Append(), insert() methods, del() function, sort(), reverse(), remove(), extend(), pop(), clear() methods.
- Pointer manipulation, superficial copying using the copy() method or [:] markers.
- Study deep copying with the copy module's deepcopy() function.
- Sorted function.
- How iterable objects work.
- Reversed() and range() functions.
- Dictionaries (dict), constructor, indexing, in operator, del() function.
- Pointer manipulation, superficial copying using the copy() method;
- Deep copy analysis with the copy module's deepcopy() function.
- Set (set), constructor, operators - | & and ^.

4 Conditional and repetitive structures

- Conditional structure if ... elif ... else.
- Ternary and Morse operator.
- Repetitive while structure.
- Repetitive structure for.
- Break and continue instructions.
- Enumerate function.
- Block else on repetitive structure.
- Intension list (comprehension list), intension dictionary (comprehension dict).

5 Functions, modules and packages

- Defining and calling a function.
- Local, global, predefined namespace (__builtins__) and dir() function.
- Return values, , return instruction.
- Generic functions (duck typing).
- Default values.
- Passage by label.
- Arbitrary number of arguments (*args, **kwargs).
- Anonymous (lambda) functions.
- eval(), exec(), map() and filter() functions.
- Import modules.
- Create a module.
- Block if __name__ == "__main__".
- Package import.
- Create a package (__init__.py).
- Instruction yield.

6 File handling

- Open() function and close() method.
- Readline() and readlines() methods.
- Iterable object.
- Instruction with files.
- Read() and write() methods.
- Tell() and seek() methods.
- Writelines() method.
- Complementary modules: struct, csv, json, xml.
- Serialization with the pickle module.
- Serialization with the shelve module.

7 Object-oriented programming

- Fundamental OOP concepts (code separation, encapsulation, inheritance).
- Notions of object class, object (instance), attribute and method.
- Defining an object class.
- Object instantiation, isinstance() function.
- Constructor (__init__).
- Attributes and methods.
- The self parameter.
- Display overload (__str__).
- Operator overload (__eq__, __add__).
- Property (special property function), accessor and mutator.
- Global, object and class namespaces.
- Class variable.
- Constructor with arbitrary number of arguments (*args, **kwargs).
- Aggregation / Composition.
- Class inheritance (generalization), issubclass(), super() functions and mro() method.

8 Exceptions

- Operating principle.
- Predefined exceptions and inheritance tree.
- Instructions try ... except ... else ... finally.
- Propagation of exceptions.
- Triggering exceptions.
- Exception definition.

9 Standard library modules

- Interaction with the interpreter: module sys.
- Interaction with the operating system: os and pathlib modules.
- Interaction with the file system: os.path module.
- Regular expressions: re module, search(), match(), split() and sub() functions.
- Unit tests: assert statement, unittest module.
- Tour d'horizon de modules de la bibliothèque standard :datetime, math, timeit, urllib, collections, csv, json, sqlite3.
- Introduction to datetime, subprocess, shutil, collections, timeit, urllib, sqlite3...

Dates and locations

REMOTE CLASS
2026 : 24 Mar., 16 June, 29 Sep., 15 Dec.