Course : Python: Advanced

Practical course - 4d - 28h00 - Ref. PYA
Price : 2630 CHF E.T.

★★★★⯨   4,4 / 5

BEST

🎯 Teaching objectives

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

- ✅ Rigorous implementation of recognized design patterns
- ✅ Use the advanced techniques of the Python language: Context Manager, metaclasses, closures, advanced functions
- ✅ Optimize the performance of your programs with the help of monitoring and parallelism
- ✅ Package and deploy Python artifacts
- ✅ Exploit libraries that help make the language a success: scientific computing, Artificial Intelligence, XML, networks

Practical details

Teaching methods

The course is made up of theory, illustrated by examples of code, then strengthened by creating a mini-project (70% hands-on work).

Course schedule

(1) Important reminders about the language.

- Assignment by reference and mutable/immutable data types.
- Passing arguments, default values and local variables.
- Class and instance variables.
- Advanced slices and data structures.
- Introspection.
- Advanced elements of control structures: the else clause of "for, while, try/except" statements.

Hands-on work

Optimization: intersection of lists and computing algorithm complexity.

2 Advanced features

- Advanced use of decorators (generation to consumption, consumer pipeline).
- Decorators and design patterns.
- Closure.

Hands-on work
Chaining data consumers. Subscription to events via decorators.

3 Advanced Object-Oriented Programming

- The properties.
- Iterators.
- Multiple inheritance and its shortcomings.
- Context managers.
- Abstract base classes and methods (ABCs).
- Metaclasses.

Hands-on work
Implement a metaclass to create singleton classes.

4 Deployment and quality

- Installing third party libraries (pip, easy_install).
- The Python Package Index (PyPI).
- Packaging your libraries (distutils, setuptools).
- Deploying a standalone environment (virtualenv and buildout).

Hands-on work
Packing a library and putting it on PypI.

5 Parallelism: Optimizing the performance of your programs

- Profiling your programs with Timeit and cProfile.
- Parallelization: Avoid multithreading and go for multiprocessing.
- Distributed computing with the Celery library.

Hands-on work
Distribute and consolidate (Map Reduce) computing with Celery.

6 Libraries that help make the language a success

- Scientific computing and statistics with Numpy, Scipy, Matplotlib and Pandas.
- Artificial Intelligence and learning algorithms with Scikit-Learn.
- Finding information in XML files with ElementTree.
- Network: tcp relay with Twisted and SNMP monitoring with PySNMP.

Hands-on work
Extracting information from XML log files, filters and statistics on the collected data and graphical representation using information trends.

Dates and locations

REMOTE CLASS
2026 : 24 Mar., 21 Apr., 26 May, 23 June, 23 June,

1 Sep., 1 Sep., 27 Oct., 17 Nov., 8 Dec., 8 Dec.

1 Sep., 1 Sep., 27 Oct., 17 Nov., 8 Dec., 8 Dec.