Course : Campus Atlas - Python, advanced

Practical course - 4d - 28h00 - Ref. PYN
Price : 2470 CHF E.T.

NEW

On completion of the course, participants will be able to use Python to develop more efficient and optimized applications. This training program is intended for employees of professional branches covered by the OPCO Atlas.

Teaching objectives

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

- ✓ Deepen your knowledge of advanced Python concepts
- ✓ Use advanced Python language techniques
- ✓ Optimize program performance through monitoring and parallelism
- ✓ Packaging and deploying Python artifacts
- ✓ Use major language libraries

Intended audience

For OPCO Atlas members: engineers and developers.

Prerequisites

Good knowledge of Python development.

Practical details

Hands-on work

Practical exercises and/or case studies.

Teaching methods

70% pratique – 30% théorie. Pour optimiser le parcours d'apprentissage, des modules e-learning peuvent être fournis avant et après la session présentielle ou la classe virtuelle, sur simple demande du participant.

Course schedule

PARTICIPANTS

For OPCO Atlas members: engineers and developers.

PREREQUISITES

Good knowledge of Python development.

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.

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.

1 Python 3, the language basics - Digital learning pre-training content

- Introduction.
- Data types.
- Algorithms.
- Data manipulation.

Digital activities

This online training course introduces the essential basics of the Python language to learn how to program efficiently. Participants will study program structure, data types, functions and algorithmic concepts, before moving on to data manipulation and object-oriented programming. They will also learn how to use a database with SQLAlchemy and apply best practices to develop clean, maintainable Python code.

2 Python 3, advanced concepts - Pre-training digital learning content

- Object model.
- Typical objects.
- Testing.
- XML.
- Document generation.

Digital activities

This online training course introduces the Python object model and typed objects, one of Python's modern development axes. Participants will be able to build high-performance, modern applications and secure data processing. They will also discover best practices for testing their code to ensure its quality, and learn how to manipulate XML with Python, and generate PDF, openDocument and image documents.

3 Advanced types and structures

- Complex types and their methods.
- Slicing and advanced sequences.
- Specialized data structures.
- Memory optimization.

Hands-on work

Manipulation des types. Exercices de slicing avancé. Optimisation des structures.

4 Introspection and metaprogramming

- Introspection concepts.
- Basic metaprogramming.
- Object inspection.
- Dynamic handling.

Hands-on work

Exploration de l’introspection. Création d’outils dynamiques. Tests et validation.

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.

5 Advanced functional programming

- Advanced decorators.
- Closures and scopes.
- Functional design patterns.
- Higher-order functions.
- Generators and iterators.
- Reactive programming.

Hands-on work

Maîtrise des décorateurs. Implémentation de patterns fonctionnels. Programmation réactive.

6 Functional programming

- Advanced decorators.
- Closures and scopes.
- Higher-order functions.
- Functional design patterns.

Hands-on work

Création de décorateurs. Implémentation de patterns.

7 Generators and iterators

- Advanced generators.
- Custom iterators.
- Reactive programming.
- Flow optimization.

Hands-on work

Développement d'itérateurs. Création de générateurs. Tests de performance.

8 Advanced OOP concepts

- Properties and descriptors.
- Multiple inheritance and MRO.
- Abstract classes.
- Basic metaclasses.

Hands-on work

Implémentation de descripteurs. Exercices d'héritage multiple. Architecture avec classes abstraites.

9 Design patterns

- Creative patterns.
- Structural patterns.
- Behavioral patterns.
- Best practices.

Hands-on work

Patterns créationnels. Patterns structurels et comportementaux.

10 Advanced metaprogramming

- Advanced metaclasses.
- Protocol descriptors.
- Customize imports.
- Metaclass hierarchies.

Hands-on work

Exploration des métaclasses. Protocol descriptors avancés. Customisation d'imports.

11 Advanced Context Managers

- Complex context managers.
- Nested contexts.
- Async context managers.
- Usage patterns.

Hands-on work

Implémentation de managers. Tests de scénarios complexes. Optimisation des ressources.

12 Advanced Python ecosystem

- Data science with NumPy and Pandas.
- Visualization with Matplotlib.
- Machine Learning with Scikit-learn.
- Web with FastAPI/Django.
- Cybersecurity with PyCrypto.
- Networking with Twisted.

Hands-on work

Data science et visualisation. Machine learning Appliqué. Web et sécurité.

13 Performance optimization

- Code profiling.
- Memory optimization.
- Efficient algorithms.
- Caching and memoization.

Hands-on work

Profilage d'applications. Optimisation de code. Benchmarking.

14 Parallel programming

- Multiprocessing.
- Advanced threading.
- Asyncio.
- Pools of workers.

Hands-on work

Threading versus multiprocessing. Asyncio en pratique. Optimisation avec Worker Pools.

15 Distributed applications

- Distributed architecture.
- Message queuing.
- Load balancing.
- Scalability.

Hands-on work

Conception d'architecture distribuée. Implémentation du message queuing. Tests de charge et monitoring.

16 Final project

- Complete architecture.
- Optimum performance.
- Testing and quality.
- Deployment.

Hands-on work

Conception et développement. Optimisation et tests. Présentation et retours.

17 Python for data science - Post-training digital learning content

- Python and data science.
- Data visualization.
- Inferential statistics with Python.
- Multivariate modeling with Python.

Digital activities

This online training course shows how to use Python for data science and the analysis of large volumes of data. Participants will learn how to manipulate and visualize data with Numpy and Pandas, then apply statistical methods and predictive models using the Scikit-Learn library.

Dates and locations

REMOTE CLASS
2026 : 17 Mar., 9 June, 22 Sep., 1 Dec.