

# Formation : Campus Atlas - Python, perfectionnement

*Cours pratique - 4j - 28h00 - Réf. PYN*

*Prix : 2100 € H.T.*

NEW

À l'issue de la formation, le participant sera capable d'utiliser Python pour développer des applications plus performantes et optimisées. Ce programme de formation est destiné aux salariés des branches professionnelles relevant de l'OPCO Atlas.



## Objectifs pédagogiques

À l'issue de la formation, le participant sera en mesure de :

- ✓ Approfondir la connaissance des concepts avancés de Python
- ✓ Utiliser les techniques avancées du langage Python
- ✓ Optimiser les performances de ses programmes à l'aide du monitoring et du parallélisme
- ✓ Packager et déployer ses artefacts Python
- ✓ Exploiter des bibliothèques majeures du langage

## Public concerné

Pour les adhérents à l'OPCO Atlas : ingénieurs et développeurs.

## Prérequis

Disposer de bonnes connaissances en développement Python.

## Méthodes et moyens pédagogiques

### Travaux pratiques

Exercices pratiques et/ou études de cas.

### Méthodes pédagogiques

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.

## PARTICIPANTS

Pour les adhérents à l'OPCO Atlas : ingénieurs et développeurs.

## PRÉREQUIS

Disposer de bonnes connaissances en développement Python.

## COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

## MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## Modalités d'évaluation

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## Programme de la formation

### 1 Python 3, les fondamentaux du langage - Contenu digital learning préformation

- Introduction.
- Types de données.
- Algorithmique.
- Manipulation de données.

#### Activités digitales

Cette formation en ligne présente les bases essentielles du langage Python pour apprendre à programmer efficacement. Les participants étudieront la structure d'un programme, les types de données, les fonctions et les notions d'algorithmique, avant d'aborder la manipulation de données et la programmation orientée objet. Ils apprendront également à utiliser une base de données avec SQLAlchemy et à appliquer les bonnes pratiques pour développer du code Python propre et maintenable.

### 2 Python 3, concepts avancés - Contenu digital learning préformation

- Modèle objet.
- Objets typés.
- Tests.
- XML.
- Génération de document.

#### Activités digitales

Cette formation en ligne présente le modèle objet de Python et les objets typés, un des axes de développement moderne de Python. Les participants seront alors capable de construire des applications performantes et modernes et de sécuriser le traitement de leurs données. Ils seront amenés à découvrir les meilleures pratiques pour tester leur code et ainsi assurer sa qualité. Ils verront également quelques recettes pour manipuler du XML avec Python, puis pour générer des documents PDF, openDocument ou encore des images.

### 3 Types et structures avancés

- Types complexes et leurs méthodes.
- Slicing et séquences avancées.
- Structures de données spécialisées.
- Optimisation mémoire.

#### Travaux pratiques

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

## MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les formations pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque formation ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le participant a bien assisté à la totalité de la session.

## MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

## ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Pour toute question ou besoin relatif à l'accessibilité, vous pouvez joindre notre équipe PSH par e-mail à l'adresse [psh-accueil@orsys.fr](mailto:psh-accueil@orsys.fr).

## 4 Introspection et métaprogrammation

- Concepts d'introspection.
- Métaprogrammation de base.
- Inspection des objets.
- Manipulation dynamique.

### Travaux pratiques

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

## 5 Programmation fonctionnelle avancée

- Décorateurs avancés.
- Closures et scopes.
- Design Patterns fonctionnels.
- Fonctions d'ordre supérieur.
- Générateurs et itérateurs.
- Programmation réactive.

### Travaux pratiques

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

## 6 Programmation fonctionnelle

- Décorateurs avancés.
- Closures et scopes.
- Fonctions d'ordre supérieur.
- Design patterns fonctionnels.

### Travaux pratiques

Création de décorateurs. Implémentation de patterns.

## 7 Générateurs et itérateurs

- Générateurs avancés.
- Itérateurs personnalisés.
- Programmation réactive.
- Optimisation des flux.

### Travaux pratiques

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

## 8 Concepts POO avancés

- Propriétés et descripteurs.
- Héritage multiple et MRO.
- Classes abstraites.
- Métaclases de base.

### Travaux pratiques

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

## 9 Design patterns

- Patterns créationnels.
- Patterns structurels.
- Patterns comportementaux.
- Bonnes pratiques.

### Travaux pratiques

Patterns créationnels. Patterns structurels et comportementaux.

## 10 Métaprogrammation avancée

- Métaclasses avancées.
- Protocol descriptors.
- Customisation d'imports.
- Metaclass hierarchies.

### Travaux pratiques

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

## 11 Context Managers avancés

- Context managers complexes.
- Nested contexts.
- Async context managers.
- Patterns d'utilisation.

### Travaux pratiques

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

## 12 Écosystème Python avancé

- Data science avec NumPy et Pandas.
- Visualisation avec Matplotlib.
- Machine Learning avec Scikit-learn.
- Web avec FastAPI/Django.
- Cybersécurité avec PyCrypto.
- Networking avec Twisted.

### Travaux pratiques

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

## 13 Optimisation de performance

- Profilage de code.
- Optimisation mémoire.
- Algorithmes efficaces.
- Caching et memoization.

### Travaux pratiques

Profilage d'applications. Optimisation de code. Benchmarking.

## 14 Programmation parallèle

- Multiprocessing.
- Threading avancé.
- Asyncio.
- Pools de workers.

### Travaux pratiques

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

## 15 Applications distribuées

- Architecture distribuée.
- Message queuing.
- Load balancing.
- Scalabilité.

### Travaux pratiques

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

## 16 Projet final

- Architecture complète.
- Performance optimale.
- Tests et qualité.
- Déploiement.

### Travaux pratiques

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

## 17 Python pour la data science - Contenu digital learning post-formation

- Python et la data science.
- Visualisation de données.
- Les statistiques inférentielles avec Python.
- Modélisation multivariée avec Python.

### Activités digitales

Cette formation en ligne montre comment utiliser Python pour la data science et l'analyse de grands volumes de données. Les participants apprendront à manipuler et visualiser les données avec Numpy et Pandas, puis à appliquer des méthodes statistiques et des modèles prédictifs grâce à la bibliothèque Scikit-Learn.

## Dates et lieux

### CLASSE À DISTANCE

2026 : 17 mars, 9 juin, 22 sep., 1 déc.

### METZ

2026 : 22 sep.

### PARIS LA DÉFENSE

2026 : 10 mars, 2 juin, 15 sep., 24 nov.

### NANCY

2026 : 17 mars, 22 sep.