



Python est devenu un langage de choix pour les scientifiques, en raison de sa simplicité de mise en œuvre et de la richesse de son écosystème, notamment grâce à ses nombreuses et performantes bibliothèques de calculs numériques bien souvent développées par les scientifiques eux-mêmes. Python est probablement le seul langage à offrir aux informaticiens un environnement open-source complet dédié à la recherche scientifique, aux sciences de l'ingénieur et aux mathématiques.



Public visé

- Besoins en calculs et simulations mathématiques
- Informaticiens, scientifiques, chercheurs et ingénieurs (physiques, chimie, sciences des matériaux, génétique, aérospatiale...)



Pré-requis

- Connaître l'algorithmie
- Première expérience du langage Python
- Connaître l'algorithmie



Durée et tarif

- **5 jours** soit 35 heures
- **3000 € HT** par personne



Formation conventionnée

- Financement possible via OPCO



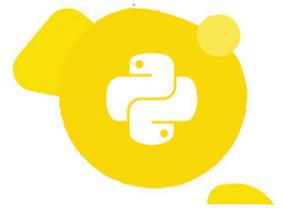
Qualiopi
processus certifié

Objectifs pédagogiques

- Se repérer dans l'écosystème Python Scientifique
- Créer et utiliser un notebook Jupyter
- Réaliser des opérations sur des tableaux numériques
- Manipuler des données issues de fichiers tabulaires
- Calculer des statistiques à partir de séries temporelles
- Prédire une valeur par interpolation
- Créer des graphiques 2D statiques
- Créer des graphiques 2D interactifs
- Créer des cartes à partir de données géographiques
- Manipuler des images
- Classifier automatiquement un jeu de données
- Mettre en place une méthode d'apprentissage supervisé
- Paralléliser une boucle for
- Mesurer les performances d'un programme



Programme



Poser les fondations

Écosystème scientifique autour du langage Python :

- Présenter les bibliothèques phares et domaines couverts
- Apprendre à choisir vos outils
- Installer Anaconda, une distribution utilisée pour l'analyse scientifique de données
- Utiliser un éditeur de code, les notebooks Jupyter, le gestionnaire de package Conda

Limites du Python pour le calcul numérique :

- Problèmes liés au stockage en mémoire
- Problèmes d'erreurs numériques

Tableaux

Discuter des limites des listes Python. Apprendre à accéder à divers jeux de données (volumineux) à travers la présentation de bibliothèques Python spécialisées :

Numpy

- Comprendre l'intérêt des tableaux Numpy
- Manipuler des vecteurs et matrices à plusieurs dimensions
- Offrir et gérer des types de données avancés
- Mixer la souplesse de langage Python et les performances du C

Pandas

- Travailler avec des données tabulaires limitée à 2 dimensions
- Analyser des structures de données complexes
- Manipuler des matrices dont chaque colonne peut être d'un type différent
- Offrir de nombreuses fonctionnalités de manipulation, réduction, recherche et statistiques

Xarray

- Étendre davantage les fonctionnalités de manipulation de tableaux
- Travailler avec des tableaux multidimensionnels et multi-types
- Proposer des fonctions d'analyse avancées qui étendent les forces de base de NumPy et Pandas

Gestion des données temporelles

- Explorer les fonctionnalités offertes par la bibliothèque Pandas pour la manipulation de données temporelles
- Calculer des statistiques temporelles
- Réaliser des interpolations de données manquantes
- Découvrir les fonctionnalités offertes par le projet Scipy

Optimisations et calculs parallèles

- Comprendre les bases de la parallélisation en Python
- Comment paralléliser des applications via la bibliothèque Dask

Machine-learning

- Se familiariser avec les concepts de base du Machine-Learning
- Méthodologie pour la mise en place de ce type d'analyse
- Utiliser les implémentations mises à disposition par la bibliothèque Scikit-Learn

