

## Initiation au Python Scientifique

Mise à jour janvier 2025

**Inter 1800€ HT/participant**  
**Intra 4500€ HT\* pour 6 participants**  
\*hors de frais de déplacement et sur-mesure

Python est actuellement l'un des seuls langages à proposer un environnement complet dédié à la recherche scientifique. Cette formation propose une initiation aux enjeux et techniques principaux du calcul numérique.

Elle s'achève par trois applications autour de thématiques emblématiques : l'analyse statistique de donnée, l'apprentissage non-supervisé, le deep-learning.

**Durée: 21.00 heures (3.00 jours)**

### À QUI S'ADRESSE CETTE FORMATION ?

#### Profil du participant

- Toute personne travaillant dans des secteurs ayant de forts besoins en calculs et simulations mathématiques
- Informaticiens, scientifiques, chercheurs et ingénieurs travaillant dans les domaines comme les sciences physiques, la chimie, les sciences des matériaux, la génétique, la mécanique des fluides, l'aérospatiale, les statistiques, le monde de la finance, etc...

#### Prérequis

- Expérience en programmation, de préférence en python

### OBJECTIFS PÉDAGOGIQUES

1. Maîtriser les éléments de lexique et de syntaxe du langage Python pour implémenter un algorithme
2. Se repérer dans l'écosystème Python Scientifique
3. Créer et utiliser un notebook Jupyter
4. Importer une bibliothèque tierce
5. Écrire et utiliser ses propres fonctions
6. Réaliser des opérations sur des tableaux numériques
7. Manipuler des données issues de fichiers tabulaires
8. Créer des graphiques 2D statiques
9. Créer des graphiques 2D interactifs

### CONTENU (PROGRESSION PÉDAGOGIQUE)

## Rappel sur le langage Python

Cette partie comprend quelques rappels sur le langage Python. Nous traiterons : les types de données (nombre, chaîne de caractères, liste, dictionnaire, tuple), les structures de contrôles (if / else, while, for), les fonctions, les packages et modules. Nous en profiterons pour évoquer quelques limites du langage Python pour le calcul numérique avec notamment :

- les problèmes liés au stockage en mémoire
- les problèmes d'erreurs numériques

## Écosystème

Cette partie présente l'écosystème python pour le calcul scientifique. Nous passerons en revue les différentes bibliothèques scientifiques en Python, leur domaine d'utilisation et comment elles permettent de contourner les limites exposées dans la partie 1. Nous installerons **Anaconda**, l'une des distributions les plus utilisées pour l'analyse scientifique de données. Nous verrons comment utiliser un éditeur de code ou les notebooks Jupyter ainsi que le gestionnaire de package, Conda.

## Chargement et manipulation des données

Nous verrons comment les bibliothèques **Numpy** et **Pandas** permettent de charger des jeux de données, parfois même très volumineux. Nous apprendrons à accéder de manière efficace et intuitive à ces données.

## Visualisation graphique

Au travers des bibliothèques **Matplotlib** et **Plotly**, nous expérimenterons quelques techniques de visualisation des données. Nous verrons comment tirer quelques conclusions à partir de simples observations. Les graphiques réalisés pourront être statiques ou interactifs.

## En option, une quatrième journée pourra permettre d'aborder au choix :

*Chacune de ces applications sera l'occasion de mettre à nouveau en pratique les concepts étudiés dans les parties 3 et 4.*

### le machine-learning

À l'aide des bibliothèques **Scipy**, **Scikit-learn** et **TensorFlow**, nous étudierons et mettrons en pratique quelques techniques de calcul numérique : analyse statistique et calcul des moments d'une variable aléatoire ; partitionnement automatique des données avec l'algorithme des k-moyennes ; introduction au deep-learning et réseaux de neuronaux.

### le calcul parallèle

Après avoir compris les bases de la parallélisation en Python (multi-threading et multi-processing), nous verrons comment paralléliser divers applications via la bibliothèque **Dask**.

### l'analyse de données géographiques

Nous verrons ensemble les spécificités des données géographiques avant de les manipuler à l'aide des bibliothèques **GeoPandas** (pour les données vectorielles) et **Rasterio** (pour les données raster). Leur représentation graphique pourra être réalisée de manière statique ou interactive.

### le traitement d'images

Nous passerons en revue les bibliothèques Python dédiées à l'analyse d'images. À l'aide de la bibliothèque **Scikit-image** notamment, nous mettrons en place quelques traitements classiques d'analyse d'images dont : la binarisation, l'application de filtres de convolution, la segmentation d'image.

*De plus, dans le cadre d'une formation intra-entreprise, un cas pratique ou jeu de données représentatif peut être fourni par l'entreprise afin que le formateur l'intègre dans le cursus.*

## ORGANISATION

### Formateur

Formation assurée par un expert-formateur Python

### Moyens pédagogiques et techniques

- Accueil des stagiaires dans une salle dédiée à la formation
- Documents supports de formation partagés
- Exposés théoriques
- Étude de cas concrets
- Quiz en ligne
- Mise à disposition en ligne de documents supports à la suite de la formation

### Dispositif de suivi de l'exécution de d'évaluation des résultats de la formation

- Feuilles de présence dématérialisée.
- Questions orales ou écrites (QCM).
- Mises en situation.
- Formulaire d'évaluation de la formation.

### Délais d'accès

La convocation est envoyée 7 jours avant le début de la formation après réception du bon de commande signé.

### Coordonnées de l'équipe pédagogique

- Responsable formation, handicap et votre formateur : Cécile Chardonneau [formation@makina-corpus.com](mailto:formation@makina-corpus.com)
- Suivi facturation : Nathalie Carles Salmon [administration@makina-corpus.com](mailto:administration@makina-corpus.com)