

# Formation Python - Bases et introduction aux bibliothèques scientifiques

**Objectifs** : Connaître les concepts du langage Python ainsi que les principales bibliothèques scientifiques : NumPy, Pandas, Matplotlib...

**Compétences visées** : - Acquérir les fondamentaux de Python en installant l'environnement, écrivant des premiers programmes, et en utilisant l'IPython pour interagir avec le langage.

- Maîtriser les bases du langage Python y compris les opérateurs, instructions de contrôle, fonctions, et structures de données
- Structurer un projet Python en modules et packages, en appliquant les principes de programmation modulaire pour organiser le code de manière réutilisable.
- Exploiter la bibliothèque standard de Python pour la gestion des fichiers, des répertoires, et l'interface avec le système d'exploitation
- Mettre en œuvre les concepts de base de la programmation orientée objets
- Adopter des bonnes pratiques de développement Python
- Manipuler des tableaux de données et effectuer des calculs numériques avec NumPy en utilisant ses fonctions pour la corrélation de données, la manipulation de matrices, et d'autres opérations mathématiques
- Visualiser des données en traçant des graphiques avec Matplotlib
- Manipuler et analyser des données avec Pandas, en utilisant des Series et DataFrames pour l'indexation, la transformation, l'agrégation, et la visualisation de données.
- Explorer les bases du Machine Learning et du Deep Learning avec SciKit-Learn, TensorFlow, et PyTorch, pour comprendre les concepts clés et implémenter des modèles simples.

**Durée** : 3 jour(s) (21 heures)

**Public** : Profil scientifique désireux d'acquérir les bases du langage Python pour être autonome lors de l'utilisation des bibliothèques scientifiques

**Pré-requis** : Expérience de la programmation et de l'algorithmique

**Méthode pédagogique** : Chaque chapitre s'achève par des travaux pratiques qui mettent en œuvre les éléments présentés. Les TP utilisent les outils Pycharm ou Spyder selon les souhaits.

Pédagogie active mêlant exposés, exercices et applications pratiques dans le logiciel Python.

**Modalités d'évaluation** : Un formulaire d'auto-évaluation proposé en amont de la formation nous permettra d'évaluer votre niveau et de recueillir vos attentes. Ce même formulaire soumis en aval de la formation fournira une appréciation de votre progression.

Des exercices pratiques seront proposés à la fin de chaque séquence pédagogique pour l'évaluation des acquis.

En fin de formation, vous serez amené(e) à renseigner un questionnaire d'évaluation à chaud.

Une attestation de formation vous sera adressée à l'issue de la session.

Trois mois après votre formation, vous recevrez par email un formulaire d'évaluation à froid sur l'utilisation des acquis de la formation.

**Accessibilité :** Vous souhaitez suivre notre formation Formation par ville et êtes en situation de handicap ? Merci de nous contacter afin que nous puissions envisager les adaptations nécessaires et vous garantir de bonnes conditions d'apprentissage

**Tarifs :**

- Présentiel : 1650 € HT
- Distanciel : 1500 € HT
- (-10% pour 2 inscrits, -20% dès 3 inscrits)

**Option(s) :**

- Forfait déjeuners : 60 € HT

## Nos prochaines sessions

### Distance

- du 22 au 24 avril 2025
- du 25 au 27 juin 2025
- du 22 au 24 octobre 2025

### Lyon

- du 2 au 4 décembre 2024
- du 23 au 25 avril 2025
- du 1 au 3 décembre 2025

### Paris

- du 25 au 27 novembre 2024
- du 3 au 5 mars 2025
- du 11 au 13 juin 2025
- du 24 au 26 novembre 2025

### Toulouse

- du 17 au 19 mars 2025
- du 24 au 26 septembre 2025

**Programme :**

## - Introduction rapide

- Historique
- Installation
- Premier programme
- Principales versions
- Prise en main de IPython

## - Bases du langage

- Opérateurs et expressions
- Instructions de contrôle
- Fonctions
- Structures de données

## - Modules et packages

- Programmation modulaire
- Importation de modules
- Du module au Package

## - Librairie standard

- Gestion des fichiers
- Gestion des répertoires
- Interface avec le système

## - Programmation objets (bases)

- Définition de classes
- Héritages
- Gestion des exceptions
- Surcharge des opérateurs

## - Syntaxe avancée

- Définitions fonctionnelles de listes
- Itérateurs et générateurs
- Décorateurs
- Instructions "with" et Contextlib
- Lambda fonctions

## - Aide au développement

- Documentation de code
- Tests unitaires
- Debugger
- Installation de packages (pip)

## - NumPy

- Base de NumPy (tableaux et types)
- Entrées/Sorties
- Fonctions utiles
  - corrélation de données
  - polynômes
  - programmation fonctionnelle
- Manipulation de matrices

## - Matplotlib

- Structure d'un graphe - éléments esthétiques
- Layout et Annotations
- Graphes en 3D
- Graphes interactifs
- Introduction à Seaborn
- Introduction à VisPy (3D temps-réel)

## - Pandas

- Manipulation de Series et DataFrames
- Indexation, Catégories
- Fonctions numériques et statistiques
- Lecture & écriture de données
- Transformation de données
- Agrégations
- Time-Series
- Visualisation

## - Machine Learning

- Introduction à SciKit-Learn
- Introduction à TensorFlow
- Introduction à PyTorch (Deep Learning)

*Date de dernière modification : 5 novembre 2024*