

# Formation Python - Calcul parallèle

**Objectifs :** Développer des compétences avancées pour exploiter efficacement les architectures de calcul parallèle en Python, en maîtrisant les techniques de multithreading, multiprocessing, calcul distribué et traitement GPU afin d'optimiser les performances des applications et workflows de traitement de données.

**Compétences visées :** - Appréhender le paysage de calcul parallèle avec Python

- Maîtriser les concepts et terminologies de la programmation parallèle
- Identifier les goulots d'étranglement et les parties parallélisables de votre programme
- Utiliser efficacement les bibliothèques de calcul parallèle en Python
- Savoir développer des applications parallélisées (multithreading, multiprocessing, calcul distribué, cloud computing)
- Exécuter des calculs sur GPU (cartes graphiques)
- Créer des workflows de traitement de données

**Durée :** 5 jour(s) (35 heures)

**Public :** Développeurs, chefs de projets, data scientists développant des applications scientifiques requérant d'importantes capacités de calculs

**Pré-requis :** Pour suivre ce stage dans de bonnes conditions, il est recommandé d'avoir suivi en amont la formation [Python – Bases et introduction aux bibliothèques scientifiques](#)

**Méthode pédagogique :** Cette formation est majoritairement pratique. L'ensemble des TP sera réalisé sur des machines louées dans le cloud et dotées de cartes graphiques NVIDIA ainsi que sur les postes des participants. Les concepts présentés font l'objet de TP d'illustration afin de permettre leur assimilation. Environ 70% du temps est consacré à la pratique des bibliothèques présentées. L'interactivité est garantie au travers de TP réalisés en majeure partie sur les notebooks Jupyter/Lab.

**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 : 3250 € HT
- Distanciel : 3000 € HT

(-10% pour 2 inscrits, -20% dès 3 inscrits)

**Option(s) :**

- Forfait déjeuners : 100 € HT

**Nos prochaines sessions**

**Distance**

du 24 au 28 novembre 2025

**Lyon**

du 12 au 16 mai 2025

du 3 au 7 novembre 2025

**Paris**

du 23 au 27 juin 2025

du 8 au 12 décembre 2025

**Toulouse**

du 13 au 17 octobre 2025

**Programme :**

**- Etat de l'art de la discipline et concepts de base**

- Historique des supercalculateurs
- Comprendre les différentes architectures disponibles pour le calcul parallèle (CPU, GPU, TPU, ASIC, FPGA, NUMA... )
- Tout n'est pas parallélisable : comprendre les limites de la programmation parallèle
- Présentation du paysage de calcul parallèle avec Python

*Travaux pratiques*

*Identifier les capacités matérielles de votre ordinateur. Mesurer les performances/limites de votre configuration (disques, mémoire, processeurs, ...).*

*Configurer son environnement de calcul parallèle.*

*Administrer une ferme de serveurs avec ansible*

**- Les concepts de la programmation parallèle**

- Comprendre la terminologie: programmation asynchrone, concurrente, distribuée, multithreading, multiprocessing, ...
- Multithreading : paralléliser le code de votre programme - mise en oeuvre des concepts de base
- Comprendre les limites du multithreading en Python
- Multiprocessing : paralléliser votre programme sur plusieurs processeurs et mécanismes de

synchronisation (verrous, sémaphores, barrières, pools de process...)

#### *Travaux pratiques*

*Application des concepts de base aux travers d'exercices pratiques.*

*Mesurer les différences de performances entre les librairies multithreading et multiprocessing.*

*Premier cluster de calcul distribué avec les Managers et Proxy.*

## **- Le calcul sur GPU**

Un GPU ne se programme pas comme un CPU.

- Comprendre les architectures GPU : kernels, mémoire, threads, ...
- Travailler avec des cartes graphiques externes (eGPU)
- Mise en œuvre des principales librairies Python pour GPU: Cupy, PyCUDA, Numba et RapidsAI

#### *Travaux pratiques*

*Identifier quand un GPU devient plus intéressant qu'un CPU.*

*Traitement d'images, calcul matriciel, tester la fiabilité d'un mot de passe, ...*

## **- Calcul distribué**

- Les principales librairies : Celery, Dask et PySpark
- Déployer et superviser un cluster de calcul parallèle avec chacune des librairies
- Exécuter des calculs sur un cluster

#### *Travaux pratiques*

*Batch de tâches avec Celery.*

*Calcul numérique et analyse de données avec Dask (array et dataframe)*

*Analyse de données avec les DataFrames Spark et la librairie Koalas.*

## **- Créer un pipeline de traitement de données**

- Présentation des librairies Luigi et Airflow
- Concevoir et superviser son workflow

#### *Travaux pratiques*

*Réaliser un workflow sur un ensemble de fichiers volumineux et le superviser avec Airflow.*

## **- Tour d'horizon des autres librairies Python pour le calcul parallèle**

- La compilation Just In Time avec Numba
- Greenlets : vers un meilleur multithreading

- MPI4Py : Message Passing Interface
- Pythran : Le transpileur qui convertit votre code Python en C++

*Travaux pratiques*

*Exercices de base illustrant les principes de chacune des librairies*

*Date de dernière modification : 17 mars 2025*