

Public	Ingénieurs, chercheurs, techniciens.
Durée	2 jours - 14 heures
Pré-requis	Connaissances de base en Python.
Objectifs	Analyser les performances d'un code Python existant. Utiliser Cython pour améliorer les performances d'un code Python. Gérer finement les structures composées pour aller plus loin dans l'optimisation. Appliquer l'ensemble du processus d'optimisation à un exemple.
Méthodes pédagogiques	Pour bien préparer la formation, le stagiaire remplit une évaluation de positionnement et fixe ses objectifs à travers un questionnaire. La formation est délivrée en présentiel ou distanciel (e-learning, classe virtuelle, présentiel et à distance). Le formateur alterne entre méthodes démonstratives, interrogatives et actives (via des travaux pratiques et/ou des mises en situation). La validation des acquis peut se faire via des études de cas, des quiz et/ou une certification. Cette formation est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par Audit Conseil Formation.
Moyens techniques	1 poste de travail complet par personne De nombreux exercices d'application Mise en place d'ateliers pratiques Remise d'un support de cours Passage de certification(s) dans le cadre du CPF Remise d'une attestation de stage
Modalité d'évaluation des acquis	Evaluation des besoins et objectifs en pré et post formation Evaluation technique des connaissances en pré et post formation Evaluation générale du stage
Délai d'accès	L'inscription à cette formation est possible jusqu'à 5 jours ouvrés avant le début de la session
Accessibilité handicapés	Au centre d'affaires ELITE partenaire d'ACF à 20 m. Guide d'accessibilité à l'accueil.

ANALYSE DES PERFORMANCES D'UN CODE PYTHON EXISTANT

- Mesurer le temps d'exécution d'un code Python et découvrir des outils de visualisation graphique.
- Mesurer la consommation mémoire d'un code.

TRAVAUX PRATIQUES

- Comparaison des différentes solutions pour un même problème.

UTILISATION DU PYTHON POUR AMÉLIORER LES PERFORMANCES D'UN CODE PYTHON.

- Générer avec Cython de fonctions C accessibles depuis Python.
- Déterminer où ajouter du typage des paramètres et des variables pour une optimisation poussée.
- Typier des paramètres et des variables.
- Accéder en Python à une bibliothèque C existante grâce à Cython.

TRAVAUX PRATIQUES

- Cas concret d'un exemple d'optimisation de code de production.

GESTION DES STRUCTURES COMPOSÉES POUR ALLER PLUS LOIN DANS L'OPTIMISATION.

- Gérer les chaînes de caractères et les chaînes d'octets.
- Utiliser efficacement les objets NumPy dans Cython pour les tableaux de nombres.
- Gérer l'allocation mémoire lors de la création de tableaux.

TRAVAUX PRATIQUES

- Des exemples concrets sur la gestion des structures composées pour l'optimisation du code de production.

NOUS CONTACTER

Siège social

16, ALLÉE FRANÇOIS VILLON
38130 ÉCHIROLLES

Téléphone

04 76 23 20 50 - 06 81 73 19 35

Suivez-nous sur les réseaux sociaux, rejoignez la communauté !



ACF Audit Conseil Formation



@ACF_Formation

Dernière mise à jour : 15/04/2026

PROFIL Formateur : Les formateurs sont recrutés selon plusieurs critères :
Expérience, pédagogie, dynamisme et prévoyance.