CYTHON: OPTIMISATION DU CODE PYTHON

Formation proposée par ALAA EL YASSIR

Python

Cette formation permet d’identifier quelles parties d’un programme Python consomment trop de mémoire ou trop de temps processeur, puis comment on peut utiliser l'outil CYTHON pour reprendre ses parties et ainsi optimiser son programme.

◷DURÉE
2 Jour(s)
▣MODALITÉS
Présentiel / Distanciel
♙PARTICIPANTS
Jusqu'à 10 participants
★LANGUE
Français
LIEU
France entière
♙CERTIFICATION
Attestation de formation
▦MISE À JOUR
02/02/2026

VOTRE FORMATEUR



ALAA EL YASSIR
Formateur professionnel
Formateur
Conseiller Pédagogique / Formateur

TARIFS INDICATIFS

Inter-entreprise	**Pas de dates planifiées**
Intra-entreprise	**3500 € HT / groupe**
Individuelle	**Sur devis**
Durée	**2 Jour(s)**

Tarifs indicatifs, sous réserve d'analyse du besoin et de devis.

◎Objectifs pédagogiques

✓- 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.

⌘Compétences visées

Python

♙Public cible et prérequis

Public cible

- - Ingénieurs.
- - Chercheurs.
- - Techniciens.

Prérequis

- Connaissances de base en Python.

▥Programme détaillé

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 Cython 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.
- Typer 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.

▣Méthodes pédagogiques & évaluation

Méthodes pédagogiques

- ✓ Apports théoriques et retours d'expérience
- ✓ Exercices pratiques et mises en situation
- ✓ Travaux de groupe et échanges entre pairs
- ✓ Études de cas concrets

Modalités d'évaluation

- ✓ Quiz de validation des acquis
- ✓ Mises en situation évaluées
- ✓ Projet fil rouge
- ✓ Feuille de route post-formation

♿Accessibilité

Une formation accessible à tous

Ce programme peut être adapté pour les personnes en situation de handicap. Contactez-nous pour étudier vos besoins spécifiques.