

Durée : 3 jours soit 21 heures

Référence : IF-PYAV

**Public visé :**

Développeur souhaitant approfondir sa maîtrise du langage Python.

**Pré-requis :**

- Expérience de programmation avec le langage Python
- Avoir suivi la formation "Initiation au langage Python"

**Objectifs pédagogiques :**

- Connaître les concepts avancés du langage Python – Méta-programmation – Performances

**Compétences acquises à l'issue de la formation :**

**Modalités pédagogiques :**

Session dispensée en présentiel ou téléprésentiel, selon la modalité inter-entreprises ou intra-entreprises sur mesure.

La formation est animée par un(e) formateur(trice) durant toute la durée de la session et présentant une suite de modules théoriques clôturés par des ateliers pratiques validant l'acquisition des connaissances. Les ateliers peuvent être accompagnés de Quizz.

L'animateur(trice) présente la partie théorique à l'aide de support de présentation, d'animation réalisée sur un environnement de démonstration.

En présentiel comme en téléprésentiel, l'animateur(trice) accompagne les participants durant la réalisation des ateliers.

**Moyens et supports pédagogiques :**

**Cadre présentiel**

Salles de formation équipées et accessibles aux personnes à mobilité réduite.

- Un poste de travail par participant
- Un support de cours numérique ou papier (au choix)
- Un bloc-notes + stylo
- Vidéoprojection sur tableau blanc
- Connexion Internet
- Accès extranet pour partage de documents et émargement électronique

**Cadre téléprésentiel**

Session dispensée via notre solution iClassroom s'appuyant sur Microsoft Teams.

- Un compte Office 365 par participant
- Un poste virtuel par participant
- Un support numérique (PDF ou Web)
- Accès extranet pour partage de documents et émargement électronique

**Informations sur l'accessibilité :**



## Description / Contenu

### Module 1 : TYPES - OBJETS - POO

- Types de base
- Création de classes
- Héritage, Polymorphisme...
- Traitement des Exceptions
- raise, try, except, finally
- Le "Data-Model"
- Importations "avancées"

- Cython
- Le module ctypes

### Module 2 : SYNTAXE AVANCEE

- Listes en "compréhension"
- Itérateurs et générateurs
- Modules itertools, collections
- Lambda fonctions
- Décorateurs
- Instruction with et Contextlib
- Instruction yield
- Programmation asynchrone et Coroutines

### Module 3 : CLASSES AVANCEES

- Sous-classer les types de base
- Résolution des héritages multiples
- Cas de la méthode "super"
- Descripteurs
- Propriétés (properties)
- \_\_dict\_\_ et \_\_slots\_\_
- Classes abstraites
- Méta-programmation

### Module 4 : ECRITURE DE PACKAGES

- "Meilleures pratiques"
- Règles de nommage
- Choix des arguments
- Méthodes de test
- setup.py et scripts de contrôle
- Installer un package
- Désinstaller un package
- Enregistrer et uploader un package

### Module 5 : QUALITE LOGICIELLE

- Annotations
- Respect de la PEP 8 et normes de codage
- Tests unitaires (doctest et unittest)
- Taux de couverture

### Module 6 : SOLUTIONS D'OPTIMISATION

- Réduction de la complexité
- Bytecode et le module "dis"
- Multithreading
- Multiprocessing
- Gestion des caches
- Profiling
- Analyse de l'occupation mémoire

### Module 7 : DESIGN PATTERNS

- Singleton
- Adapter – Proxy – Facade
- Observer – Visitor – Template

### Module 8 : INTERFACAGE AVEC C/C++

- Objectif et principe
- SWIG