Programme de cours
Les fonctions avancées du langage Python

Durée : 3 jours soit 21 heures

Référence : IF-PYAV

Public visé :

Développeur souhaitant approfondir sa maitrise du langage Python.

Pré-requis :

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

Objectifs pédagogiques :

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

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

Modalités d'évaluation et suivi :

Avant
Afin de valider le choix d'un programme de formation, une évaluation des prérequis est réalisée à l'aide d'un questionnaire en ligne ou lors d'un échange avec le formateur(trice) qui validera la base de connaissances nécessaires.
Pendant
Après chaque module théorique, un ou des ateliers pratiques permettent la validation de l'acquisition des connaissances. Un Quizz peut accompagner l'atelier pratique.
Après
Un examen de certification si le programme de formation le prévoit dans les conditions de l'éditeur ou du centre de test (TOSA, Pearson Vue, ENI, PeopleCert)
Enfin
Un questionnaire de satisfaction permet au participant d'évaluer la qualité de la prestation.

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"

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
- Cython
- Le module ctypes