

Python développement - niveau intermédiaire

Référence : **DELY011**

Durée : **3 jours**

Certification : **Aucune**

CONNAISSANCES PREALABLES

- Connaissance de base de Python.

PROFIL DES STAGIAIRES

- Les développeurs en Python.

OBJECTIFS

- Maîtriser les éléments avancés du langage et les design patterns.

CERTIFICATION PREPAREE

Aucune

METHODES PEDAGOGIQUES

- Mise à disposition d'un poste de travail par stagiaire
- Remise d'une documentation pédagogique papier ou numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques, de réflexions et de retours d'expérience
- Le suivi de cette formation donne lieu à la signature d'une feuille d'émargement

FORMATEUR

Consultant-Formateur expert Bigdata

METHODE D'EVALUATION DES ACQUIS

- Auto-évaluation des acquis par le stagiaire via un questionnaire
- Attestation de fin de stage adressée avec la facture

CONTENU DU COURS

Découvrir les éléments avancés du langage

- Expressions Lambda
- Les contextes
- Les itérables
- Les générateurs
- Les décorateurs
- Les métaclasses
- Atelier : Multiples créations : itérable, générateur, itérateur

Gérer des modules Python

- Gestion des modules tiers
- Pypi (Python package index)
- Installation, désinstallation d'un package
- Déploiement avancé (virtualenv, buildout)
- Atelier : création et installation de modules depuis le dépôt central

Découvrir les design patterns

- Origine, catalogue
- Structure d'un patron de conception
- Application d'un pattern

Gérer des instances de classes

- Factory et Abstract Factory pour la création sous condition
- Singleton et dérivé : maîtrise des ressources disponibles
- Atelier : implémentation d'un singleton et d'une factory en Python

Structurer des données

- Le Composite, comment simplifier les listes
- La Facade : clarifier un composant
- Atelier : utilisation d'un composite dans l'application

Maîtrise du comportement

- Strategy : l'usine à méthodes
- L'itérateur et ses implémentation existantes

- Observer : l'événementiel sans événements
 - Chaîne de responsabilité et arbres de responsabilité
 - Visiteur et accès : maîtrise de la collaboration
- Atelier : implémentation d'un observer pour la surveillance de variation sur un objet