ATOUT MAJEUR
CONSEIL | FORMATION

Prenez de la hauteur

Langage Python

Durée de la formation :
5 jour (35h)

Horaires de formation
08h45 – 12h30 / 13h45 – 17h00

Pré-requis
Connaissances de base en programmation.

Public visé
Développeurs, ingénieurs, chefs de projets proches du développement.

Objectifs et compétences visées
Maîtriser les fondamentaux et les principales règles de syntaxe du langage Python. Comprendre et appliquer les principes de la programmation orientée objet. Exploiter les fonctionnalités et les bibliothèques de l'environnement Python. Développer des interfaces graphiques avec Python. Utiliser les outils de test, de débogage et d'analyse afin d'évaluer et d'améliorer la qualité de ses programmes Python.

Organisation
Formation animée en présentiel. Groupe de 3 à 8 personnes. Emargement signée par ½ journée par les stagiaires et le formateur. A l'issue du stage les apprenants renseignent un questionnaire d'évaluation à chaud qui est ensuite analysé par nos équipes.

Le formateur
La prestation est animée par un consultant formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par Atout Majeur.

Moyens techniques et pédagogiques
Alternance entre théorie et pratique. Mise à disposition d'une salle équipée. Support de cours.

Validation des acquis et attestation
Les exercices réalisés permettent de mesurer le degré de compréhension et d'acquisition des compétences visées. Une attestation de formation comportant les résultats de l'évaluation des acquis est délivrée en fin de formation.

Cette offre de formation est éligible à MON COMPTE FORMATION moncompteformation.gouv.fr

Inscriptions possibles jusqu'à 24 heures ouvrées avant le début de la formation. Si financement CPF, ce délai est porté à 11 jours ouvrés.
Certification RS6962 en option. Le certificateur est Isograd (TOSA). Date de décision de l'enregistrement : 18-12-2024. Date d'échéance de l'enregistrement : 18-12-2027

Module 1 : Maîtriser les fondamentaux du langage Python

Identifier et structurer le code
Comprendre le rôle des identifiants et des références
Appliquer les conventions de programmation et les règles de nommage
Organiser le code en blocs et utiliser les commentaires
Utiliser les types et les variables
Découvrir les types de données disponibles
Déclarer et manipuler des variables
Gérer leur portée locale ou globale et produire des affichages formatés
Manipuler les données
Effectuer des opérations sur les valeurs numériques et les chaînes de caractères
Exploiter les listes, les tuples et les dictionnaires
Lire et écrire des fichiers
Construire le déroulement d'un programme
Mettre en œuvre les conditions if/elif/else, les opérateurs logiques et de comparaison, ainsi que les boucles while et for
Contrôler les itérations avec break et continue et utiliser la fonction range
Créer des fonctions et organiser les applications
Écrire et documenter des fonctions, utiliser les expressions lambda et les générateurs, puis répartir le code dans des modules

Module 2 : Comprendre les principes de la programmation orientée objet

Découvrir le paradigme objet
Assimiler les principes de l'approche orientée objet et les caractéristiques d'un objet : son état, son comportement et son identité
Structurer les objets et les classes
Comprendre la notion de classe ainsi que les attributs et les méthodes
Aborder l'encapsulation des données et les échanges entre objets
Mettre en relation les classes
Étudier l'héritage et la transmission des caractéristiques, le polymorphisme et les associations entre classes
Comprendre le rôle des interfaces
Lire les représentations et modèles de conception
Interpréter les diagrammes de classes, de séquences et d'activités, et découvrir les modèles de conception (Design Patterns)

Module 3 : Développer en programmation objet avec Python

Créer et instancier des classes
Étudier les spécificités de l'approche objet dans Python, écrire des classes et créer leurs instances
Utiliser les constructeurs et les destructeurs
Gérer les attributs et les méthodes
Mettre en place les mécanismes de protection des attributs et des méthodes, comprendre le rôle du paramètre self et les notions de visibilité
Mettre en œuvre les mécanismes objet
Développer des relations d'héritage simple ou multiple, appliquer le polymorphisme, exploiter les méthodes spéciales et pratiquer l'introspection

Appliquer les bonnes pratiques
Implémenter des interfaces, utiliser des modèles de conception courants et gérer les erreurs à l'aide du mécanisme des exceptions

Module 4 : Exploiter la bibliothèque standard de Python (StdLib)

Interagir avec l'environnement d'exécution
Récupérer et exploiter les arguments transmis en ligne de commande
Manipuler les éléments du système de fichiers
Traiter des chaînes avec les expressions régulières
Utiliser le moteur d'expressions régulières de Python et le module re, notamment les caractères spéciaux et les quantificateurs (cardinalités)
Mobiliser les modules de la bibliothèque standard
Découvrir et utiliser les modules essentiels sys, os et os.path
Distribuer et connecter les applications
Préparer l'empaquetage et l'installation d'une bibliothèque Python
Accéder à des bases de données relationnelles et comprendre le fonctionnement de la DB API

Module 5 : Contrôler la qualité et fiabiliser le code Python

Analyser statiquement le code
Découvrir les outils d'analyse statique, notamment Pylint et Pychecker, puis interpréter leurs rapports : messages, avertissements et erreurs
Documenter les développements
Mettre en œuvre l'extraction automatique de la documentation
Déboguer les programmes
Utiliser le débogueur Python pour exécuter le code pas à pas et réaliser une analyse post-mortem
Tester et mesurer la qualité
Appliquer le développement piloté par les tests et les outils de tests unitaires Python, dont Unittest
Automatiser et regrouper les tests
Mesurer la couverture du code et réaliser du profiling

Module 6 : Concevoir des interfaces graphiques (IHM)

Découvrir les environnements graphiques
Assimiler les principes de programmation des interfaces graphiques et connaître plusieurs bibliothèques : TKinter, Qt6, GTK+ et wxWidgets
Construire une interface
Identifier les principaux conteneurs et les widgets disponibles : Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...
Organiser l'affichage
Comprendre le rôle du gestionnaire de fenêtres et positionner les composants à l'aide des différents layouts
Gérer les interactions
Traiter les événements et utiliser l'objet event
Appliquer le modèle Modèle-Vue-Contrôleur dans une interface graphique.

Module 7 : Assurer l'interfaçage entre Python et le langage C

Appeler des bibliothèques C depuis Python
Découvrir le module Ctypes, charger une bibliothèque C et appeler les fonctions qu'elle met à disposition
Compiler et étendre Python avec Cython
Présenter Cython et transformer un script Python en code C afin de créer une extension compilée
Optimiser les performances
Utiliser les annotations pour améliorer les performances de compilation et recourir au profileur de code

Pour vous inscrire

04.78.14.19.19

contact@atoutmajeur-ra.com / www.atoutmajeurlyon.com

(Mise à jour : 11-2026)