Formation Python 3 (expert)

INTERNET FORMATION

Généralités et définitions
- Historique et versions de Python
- Particularités, forces et faiblesses de Python
- Installation et configuration
- Environnement de travail
- Interpréteurs (ligne de commande), console et mode script
- IDE et éditeurs de code adaptés à Python
- Outils d'analyse statique du code (PyChecker, PyLint...)

Bases de développement en Python
- Encodage des caractères (UTF-8)
- Saisie de textes (input)
- Gestion des commentaires de code
- Variables, types de données (chaînes, booléens...) et transtypage
- Chaînes de caractère et spécificités
- Gestion des chiffres et calculs
- Tableaux indicés (tuples et listes)
- Tableaux associatifs (dictionnaires)

Instructions avancées en programmation
- Normes de programmation en Python (PEP)
- Opérateurs (logiques, relationnels...)
- Structures conditionnelles (if... elif... else...)
- Boucles en Python (while, for, range...)
- Gestion des boucles avec break, continue et pass
- Compréhension de liste et generator expression

Fonctions et procédures
- Gestion des fonctions def(), arguments et valeurs à retourner
- Gestion des arguments de fonctions (définis / indéfinis)
- Portée des variables (globales...)
- Fonctions lambda
- Fonctions prédéfinies

Gestion des scripts, packages et modules externes
- À quoi servent les scripts et modules externes ?
- Installation de modules avec pip ou easy_install
- Importation de modules Python
- Stdlib et modules à connaître (os, argparse, re, logging, time...)
- Exécution de programmes externes
- Création de fichiers compilés en Python (.pv, .pvc...)

Gestion des exceptions et erreurs
- Pourquoi faire un contrôle d'erreur ?
- Méthode try... except... (else... finally...)
- Passer les messages d'erreurs avec pass
- Exceptions personnalisées avec raise
- Assertion avec assert et AssertionError
- Localisation d'erreurs avec sys et traceback
- Debugger pdb

Programmation orientée objet (POO) en Python
- Approche et méthodologie objet en Python
- Créations de classes et instanciation
- Création de méthodes et d'attributs
- Self et constructeur __init__
- Notions d'héritage, de surcharge et de polymorphisme
- Héritage multiple et Method Resolution Order (MRO)
- Décorateurs
- Itérateurs et générateurs
- Méthodes spéciales
- Interfaces
- Introspection en Python (dir, inspect, type, id...)
- Composition, metaclasses...

Autres fonctionnalités très avancées
- Gestion des expressions régulières
- Environnements virtuels (virtualenv)
- Coroutines avec yield et send()
- Multithreading et Asyncio

Gestion des fichiers avec Python
- Création et écriture dans un fichier
- Lecture d'un fichier existant
- Suppression d'un fichier
- Modules os, os.path, shutil, zlib, csv...
- Traiter des fichiers XML avec Python et etree

Créer une User Interface (UI) avec Python
- Créer des interfaces graphiques avec Tkinter
- Autres gestionnaires d'UI en Python : Kivy, wxPython, PyForms...
- Créer un jeu vidéo avec PyGame
- Utiliser un controller avec Python (souris, clavier, manette...)

Exercices d'applications en Python

Bilan et questions sur la formation Python

Durée : 4 jours (28 heures)

Prérequis : connaissances en informatique

Public : développeurs, data scientists, chercheurs ou passionnés de programmation...

Objectifs :
* Appréhender le langage Python
* Savoir réaliser des programmes simples en CLI
* Comprendre les forces de Python
* Savoir créer des interfaces utilisateurs (GUI)
* Maîtriser les PEP et coder proprement en Python
* Savoir créer des logiciels informatiques complets
* Mâitrise Python Objet et ses fonctions avancées

Personnalisation du contenu de la formation selon votre niveau et vos besoins

Pour une formation sur mesure ou à dispenser chez vous, n'hésitez pas à contacter Internet-Formation. La solution est à portée de main !

06 76 34 29 91
contact@internet-formation.fr

Internet-Formation (Chartier Mathieu) - SIRET : 512 731 175 00018
Numéro d'enregistrement formation : 54 79 00889 79

www.internet-formation.fr