



Python initiation

Description :

Le langage Python est aujourd'hui omniprésent dans beaucoup de domaines et c'est probablement le langage le plus pratiqué à travers le monde. De l'application web à l'intelligence artificielle en passant par le calcul scientifique, sa polyvalence, sa simplicité et sa communauté ont fait de lui un langage incontournable qui s'invite dans tout type de projet.

Profil de l'intervenant :

Consultant formateur expert développement

Objectifs :

A l'issue de la formation, les stagiaires sauront :

- les possibilités du langage Python
- capable de réaliser des scripts d'administration système
- capable de reprendre et modifier un programme écrit par une autre personne
- capable d'installer et d'utiliser des bibliothèques tierces comme PyQt, ElementTree (XML), Numpy, Scipy, Pandas

Publics :

Développeurs

Durée :

3 jours

Prérequis :

Bonne connaissance en algorithmie.

Méthode pédagogique de cette formation :

La formation est constituée d'apports théoriques, d'exercices pratiques et de réflexions. Remise d'une documentation pédagogique papier ou numérique pendant le stage 6 à 8 personnes maximum par cours.

Méthode d'évaluation des acquis de la formation :

MAJ MARS 2022

•

Auto évaluation des acquis par le stagiaire via une questionnaire. Attestation de fin de stage signée remise au stagiaire en fin de formation

Information handicap :

Cette formation est accessible aux personnes en situation de handicap. Chaque situation étant unique, nous vous demandons de préciser à l'inscription votre handicap. Nous pourrions ainsi confirmer l'ensemble des possibilités d'accueil et vous permettre de suivre la formation dans les meilleures conditions en accord avec votre employeur. Pour toutes informations complémentaires, nous vous conseillons la structure suivante <https://www.agefiph.fr/>

Programme de cette formation :

Python en ligne de commande

- L'interpréteur Python
- Différences avec le mode script
- Les interpréteurs iPython et bPython

Maîtriser les types et objets de bases

- Les chaînes
- Les tableaux indicés (tuples et listes)
- Les tableaux associatifs (dictionnaires)
- Les nombres Python et leurs particularités
- Les booléens
- Les autres types avancés de Python (sets, fractions, ...)
- Introspection (dir, help)
- Gestion de la mémoire, objets muables et immuables : bien comprendre comment cela fonctionne

Structuration du code

- If, for, while, émuler un switch_case
- Les fonctions (paramètres et passage d'arguments)
- Visibilité des variables

Gérer les exceptions et les erreurs

- Try except finally
- Raise
- Exception
- assert
- traceback
- pdb

Mots clefs et fonctions utiles

- with, map, lambda, filter, any, all etc.

Conventions de codage et de nommage

- pep-0008
- pep-0257

-
- pylint

Développer des librairies

- Les packages et modules
- Les techniques et règles d'importation des librairies tierces
- Ecrire ses tests sans gêner les imports
- Installer une librairie avec pip

P.O.O.

- Présentation des concepts de la Programmation Orientée Objets
- Attributs et méthodes de classe/instance
- Les méthodes spéciales (surcharges d'opérateurs, casting, constructeur, destructeur, ...)
- Héritage, surcharge, polymorphisme, la classe object
- L'héritage multiple et le MRO (Method Resolution Order)
- Bonnes pratiques de la programmation objet avec Python

La stdlib

Parcours de quelques modules phares

- Gestion de fichiers
- Argparse
- Logging
- Configparser
- Os
- Subprocess
- re (expressions rationnelles)

Aller plus loin

Présentation des objets et fonctions avancés

- Décorateurs
- Itérateurs et générateurs
- Context managers
- Unittest et docstests
- Documenter avec sphinx