

Programme de formation Langage Python

• Objectifs

Axée principalement sur le développement de l'autonomie, la formation est extrêmement pratique afin de vous permettre de bien assimiler les concepts de ce fabuleux langage. Un support complet avec les principaux points clefs ainsi que les exercices corrigés est fourni. Enfin la dernière demi-journée est consacrée au codage d'un cas pratique proposé (par le client lors d'une formation intra-entreprise) testé et documenté en pair-programming (ou individuellement) avec revue de code collective.

- Connaître les possibilités du langage Python
- Être capable d'aborder des frameworks Python tels que Django, Flask ou Pyramid
- Être capable de réaliser des scripts d'administration système
- Être capable de reprendre et modifier un programme écrit par une autre personne
- Être capable d'installer et d'utiliser des librairies tierces comme PyQt, ElementTree (XML), Numpy, Scipy, Pandas

• Pré requis

Connaissance des bases de la programmation.

• Durée

3 jours

• Public

Développeurs

• Plan de 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

Codage d'une appli testée et documentée

La dernière demi-journée est consacrée au
codage d'un cas pratique testé et documenté
en pair-programming (ou individuellement)
avec revue de code collective.

Les cas pratiques fournis permettent de
reprendre l'ensemble des points clefs évoqués.