PROGRAMME DE FORMATION PYTHON

Description

> En réalité, les langages de programmation servent aux programmeurs à communiquer des idées – et ces idées sont destinées à d'autres programmeurs, pas à des ordinateurs.
>
> – Guido van Rossum, créateur de Python

Le langage de programmation Python a été conçu pour être amusant à écrire et facile à lire. C'est un langage qui gagne toujours en popularité; il est employé dans des contextes très exigeants (*youtube* est écrit en Python). Bon à tout faire, il permet aussi bien d'automatiser les tâches administratives que de construire des outils web complexes. Par ailleurs la communauté scientifique et les analyseurs de données ont largement adopté Python pour leur travaux.

Durant ces 3 jours de formation Python, essentiellement pratiques, nous aborderons les principaux concepts du langage, la syntaxe et les bonnes pratiques. Nous verrons où chercher de l'aide, et comment s'appuyer sur la communauté. Nous utiliserons la dernière version stable du langage, Python 3.7, et nous aborderons les stratégies pour moderniser une base de code Python 2.

Rien de mieux que la pratique pour approcher un langage! Nous mettrons en œuvre les notions abordées en créant un site web à l'aide du miroframework flask. On commencera par des tâches de génération automatique de contenu, puis on verra comment construire un site de blog et, pour les plus intrépides, une galerie photo.

Durée

3 jours (21 heures)

Objectifs pédagogiques

- Posséder les bases du langage Python et bien comprendre ses particularités.
- Connaître les bonnes pratiques de développement en Python.
- Debugger et maintenir les scripts développés par d'autres développeurs.
- Réaliser des scripts d'administration système.
- Développer ses propres programmes *from scratch*.

Public visé

Développeurs, administrateurs systèmes, testeurs

Pré-requis

- Connaissances de base en algorithmique.
- Avoir une réelle expérience dans un langage de programmation.
- Ordinateur portable à apporter.

Plan de formation

Jour 1

Les types de données

- strings
- nombres
- tuples
- listes
- dictionnaires
- booléens et savoir quels objets sont vrais
- introspection (id, type, dir, help)
- Les particularités du langage Python : objets muables et immuables.

Les structures de contrôle

- Les boucles for et while, les mots clefs break et continue.
- Les tests if, elif et else, le test else à la sortie d'une boucle.
- Les fonctions spéciales map, lambda, filter.

Les séquences en python

- La notion d'*iterable* en Python.
- zip et enumerate.
- Les constructions par compréhension.
- Embalage et désembalage, notation en étoile.

Gérer les chaînes de caractères

- Problèmes d'encodage, comment rester en unicode.
- Les différentes façons de formater une chaîne de caractères, les `fstrings`.

Jour 2

Les fonctions

- Les paramètres positionnels et paramètres nommés.
- La notation `*args` et `*kwargs`.
- La portée des variables.
- La documentation en ligne et les docstrings.
- Introduction aux décorateurs.

Consolider son code

- Les bonnes pratiques : le zen de python, pep8.
- Les exceptions : `try`, `except`, `raise` et `finally`.
- La gestion de contexte avec `with`.
- Le debugger en ligne de commande avec pdb.
- Les tests unitaires avec pytest et les assertions.
- Les outils de qualité de code pylint, pyflakes, black.

Maîtriser le sys.path

- Comprendre le sys.path, les modules et les packages.
- Les environnements virtuels.
- Installer un module tiers avec pip.

Programmation orientée objet

- Les concepts de la programmation orientée objet.
- Attributs de classe et d'instance, les attributs `property`.
- Les méthodes spéciales (surcharge d'opérateurs,`__iter__`).
- L'héritage et la redéfinition de fonctions.

Jour 3

De Python 2 à Python 3

- Les principales différences.
- Écrire du code compatible pour les deux versions (sixer).
- Traduire du code Python 2 en Python 3: 2to3.

Quelques modules de la bibliothèque standard

- Récupérer les paramètres passés à un script : argparse.
- Exécuter des commandes système depuis Python : subprocess et shutils.
- Les outils de journalisation : logging.
- Itérer différemment: itertools.
- Des objets pratiques: le module collections.

Suivi de formation en option

A l'issue de la formation, nos formateurs peuvent aussi intervenir pour vous accompagner dans la mise en application des compétences acquises :

- en répondant à vos questions lors de rendez-vous téléphoniques réguliers
- en étant présents physiquement à l'amorce du projet
- en réalisant un audit de vos pratiques quelques semaines/mois après la formation

Cette idée vous intéresse ? Faîtes-le nous savoir à l'acceptation du devis pour que nous trouvions la formule adaptée à votre situation.

Méthodes pédagogiques

On visera une alternance de 50% de travaux pratiques et 50% de cours théoriques. Le support de cours sera fourni au format PDF accompagné d'un lien vers les supports numériques (TP & application).

Évaluation des acquis de la formation

Avant le début de la formation, le formateur valide les prérequis et évalue le niveau de départ des participants. L'évaluation des acquis est réalisée tout au long de la formation par des exercices pratiques et des ateliers réalisés par les participants, sous la supervision du formateur. Chaque stagiaire doit faire la démonstration de la maîtrise d'un bloc de connaissance avant de passer au bloc suivant. La démonstration se fait directement dans le shell, puis dans des petits scripts spécialisés de difficulté itérative.