

Avant-propos

« *wOOt! I know Python!* »

« Wow ! Je maîtrise Python maintenant ! »

— Neo, retirant son casque

Ce livre traite de Python, un langage de programmation de haut niveau, orienté objet, totalement libre et terriblement efficace, conçu pour produire du code de qualité, portable et facile à intégrer. Ainsi la conception d'un programme Python est très rapide et offre au développeur une bonne productivité. En tant que langage dynamique, il est très souple d'utilisation et constitue un complément idéal à des langages compilés.

Il reste un langage complet et autosuffisant, pour des petits scripts fonctionnels de quelques lignes, comme pour des applicatifs complexes de plusieurs centaines de modules.

Pourquoi ce livre ?

Il existe déjà de nombreux ouvrages excellents traduits de l'anglais qui traitent de Python voire en présentent l'intégralité des modules disponibles. Citons « Python en concentré », le manuel de référence de Mark Lutz et David Ascher, aux éditions O'Reilly, ou encore « Apprendre à programmer avec Python » de Gérard Swinnen, chez le même éditeur, inspiré en partie du texte « *How to think like a computer scientist* » (Downey, Elkner, Meyers) et, comme son titre l'indique, très pédagogique.

Alors pourquoi ce livre ?

Si ce livre présente comme ses prédécesseurs les notions fondamentales du langage, avec bien sûr des exemples originaux, des choix dans la présentation de certains modules, et une approche globale particulière, il tente également d'ajouter à ce socle des éléments qui participent de la philosophie de la programmation en Python, à savoir :

- des conventions de codage ;
- des recommandations pour la programmation dirigée par les tests ;
- des bonnes pratiques de programmation et des techniques d'optimisation ;
- des designs patterns orientés objet.

Même si chacun de ces sujets pourrait à lui seul donner matière à des ouvrages entiers, les réunir dans un seul et même livre contribue à fournir une vue complète de ce qu'un développeur Python averti et son chef de projet mettent en œuvre quotidiennement.

À qui s'adresse l'ouvrage ?

Cet ouvrage s'adresse bien sûr aux développeurs de tous horizons mais également aux chefs de projets.

Les développeurs ne trouveront pas dans ce livre de bases de programmation ; une pratique minimale préalable est indispensable, quel que soit le langage utilisé. Il n'est pour autant pas nécessaire de maîtriser la programmation orientée objet et la connaissance d'un langage impératif est suffisante.

Les développeurs Python débutants – ou les développeurs avertis ne connaissant pas encore ce langage – trouveront dans cet ouvrage des techniques avancées, telles que la programmation dirigée par les tests, les patterns efficaces et l'application de certains designs patterns objet.

Les chefs de projets trouveront des éléments pratiques pour augmenter l'efficacité de leurs équipes, notamment la présentation des principaux modules de la bibliothèque standard – pour lutter contre le syndrome du NIH (*Not Invented Here*) –, des conventions de codage, et un guide explicite des techniques de programmation dirigée par les tests.

Guide de lecture

Le livre est découpé en quatre parties qui peuvent être lues de manière relativement indépendante, en fonction des besoins.

La première partie présente une introduction au langage, décrit les différents domaines d'utilisation de Python, ainsi que la mise en place d'un environnement de développement ; elle s'adresse principalement aux lecteurs qui découvrent Python.

La deuxième partie est consacrée à la présentation du langage, de la syntaxe aux conventions de codage, en passant par les primitives. C'est un référentiel complet utile en toutes circonstances.

La troisième partie présente les modules de la bibliothèque standard les plus fréquemment utilisés, pour ne pas rechercher ailleurs ce qui est déjà disponible. Cette partie s'achève sur une petite série d'exercices.

Enfin, la quatrième partie regroupe les techniques avancées, à savoir la programmation dirigée par les tests, les bonnes pratiques et techniques d'optimisation, et enfin des techniques de programmation orientée objet.

Ce livre s'achève par une série d'annexes qui présentent l'histoire de Python, une liste de bibliothèques tierces, une liste de sites, blogs, et autres sources d'information de la planète Python.

Remerciements

Ce livre n'aurait jamais été possible sans le soutien et l'aide de l'équipe des Éditions Eyrolles et en particulier **Patrick Tonnerre**, qui en plus de son travail correctif, a fourni de précieux conseils pour améliorer le livre, ainsi que **Anne Bognoux**, **Jean-Marie** et **Gaël Thomas**, et **Muriel Shan Sei Fan** pour son soutien et son aide continus (et qui m'a dévoilé quelques-unes de ses bottes secrètes dans l'art de la traduction...).

Merci à **Anahide Tchertchian**, **Olivier Grisel** et **Jean-Philippe Camguilhem**, développeurs Python de choc, qui ont relu des parties du livre et qui m'ont permis d'éviter des incohérences et des erreurs ; **Paul Everitt**, qui a eu la gentillesse de répondre à mon interview pour compléter l'histoire de Python ; **Laurent Godard**, le roi du scripting OpenOffice.org, qui m'a concocté des scripts sur mesure pour la conception de ce livre et qui m'a également distillé ses conseils d'écrivain aguerri (Comment ? Vous n'avez pas encore acheté le livre « Programmation OpenOffice.org 2 » chez le même éditeur ?).

Merci à la communauté Python et à la communauté Zope, dont les membres ont indirectement participé à ce livre en me distillant leurs bons conseils quotidiens sur les canaux de discussion IRC **#zope** et **#python**, sur freenode ; merci à mes amis de l'AFPY (Association Francophone Python) et tout particulièrement à **Bader Ladjemi**, qui a défriché et amorcé le travail de l'annexe B. Merci à mes collègues de Nuxeo, qui sont tous des Pythonneurs convaincus et qui participent eux aussi en force à la cause.

Merci à mes préfaciers de talent, **Stephan Richter** et **Stéfane Fermigier**, et tout particulièrement à Stéfane, qui a contribué à augmenter la qualité de ce livre. Au quotidien, Stéfane sensibilise également son entourage aux vertus de Python et de l'Open Source, et à son contact, toute personne s'élève forcément dans l'art de la programmation.

Merci évidemment à **Guido von Rossum** pour son génie, et à tous les *core developers* de Python, qui prennent soin de l'évolution du bébé.

Enfin et surtout, merci à **Thérèse** et **Milo**, pour avoir supporté un compagnon et un père, penché tous les soirs sur son ordinateur portable pour écrire ce livre, et bien sûr à ma mère, pour m'avoir transmis le virus.

Tarek Ziadé

tarek@ziade.org

www.programmation-python.org

ARTHUR :

Lancelot ! Lancelot ! Lancelot !

[mégaphone de police]

Lanceloooooooooot !

LANCELOT :

Bloody hell, mais que se passe-t-il donc, mon Roi ?

ARTHUR :

Bevedere, explique-lui !

BEVEDERE :

Nous devons te parler d'un nouveau langage de programmation : Python

LANCELOT :

Nouveau ? Cela fait bien dix ans qu'il existe, et je ne vois pas en quoi cela va nous aider à récupérer le Saint-Graal !

BEVEDERE :

Saint-Graal, Saint-Graal...

[soupir]

Tu ne peux pas penser à des activités plus saines que cette quête stupide de temps en temps ?

ARTHUR :

[sort une massue et assomme Bevedere avec]

Son explication était mal partie de toute manière.

GARDES FRANÇAIS :

Est-ce que ces messieurs les Anglais peuvent aller s'entretuer plus loin ?

Ne voyez-vous pas que nous sommes concentrés sur notre jeu en ligne ?

ARTHUR :

Ce tunnel sous la Manche, quelle hérésie !

[racle sa gorge]

Lancelot, assieds-toi, et écoute-moi. (et ferme ce laptop, bloody hell !)

LANCELOT :

[rabat l'écran de son laptop]

ARTHUR :

*La quête a changé. Tu dois maintenant apprendre le langage Python,
et découvrir pourquoi il est de plus en plus prisé par mes sujets.*

LANCELOT :

mais...

ARTHUR :

Il n'y a pas de mais !

[menace Lancelot avec sa massue]

Je suis ton Roi. dot slash.

Prends ce livre, et au travail !

GARDES FRANÇAIS :

Oui, au travail, et en silence !