

Le Python expliqué à mon Choupi

Choupi

30 janvier 2018

Introduction

Le pourquoi du comment

Pourquoi s'emmerder à apprendre le Python ?

Je suis sûr que tu te poses la question mon amour. Et même si je te sais naturellement curieuse et que tu aime apprendre de nouvelles chose, pourquoi se faire chier à apprendre un langage de programmation ?

Surtout sachant que ça demande un peu (*beaucoup ?*) de boulot.

Après tout si ça te passionnait tant que ça, tu en aurai fait ton boulot (*Ebawai*), et d'ailleurs y'a des gens qui en ont fait leur boulot et qui seront là pour faire les trucs relou d'informaticien à ta place.

Mais ...(*il y a toujours un mais*)

Je pense quand même qu'il te serai profitable de connaitre les bases de la programmation et de savoir écrire toute seule comme une grande tes propres scripts (voir des petits logiciels, ne t'inquiète pas, on reviendra plus tard sur la terminologie), et ce pour plusieurs raisons :

1. Certes il y a des gens qui sont payé pour faire ça, mais il n'y en a pas forcément des masses, du coup tu deviens vachement tributaire de leur disponibilité pour faire ton boulot.
2. Aujourd'hui très peu de biologistes savent coder, mais rien ne dit que ça va durer (les mÃmes font du python au lycée de nos jours). Disons que ça te permettra d'éviter de passer pour un dinosaure dans quelques années.
3. Comme dis juste au dessus, les biologistes savant coder sont rares. Et tout ce qui est rare est cher ! Disons que ça fait toujours une ligne de plus sur le CV (que les excités de la recherche sauront probablement apprécier).
4. Je pense (et là je donne mon avis d'informaticien, donc ça vaut ce que ça vaut) que de plus en plus de machines, logiciels, ... vont donner la possibilité à leurs utilisateurs de les customiser/augmenter via de la programmation (ex : Excel va inclure Python comme langage pour

faire des macros). Du coup savoir programmer va s'assez rapidement devenir de plus en plus utile.

5. Qui sait, ça se trouve tu va aimer ça ☺
6. Moi c'est sûr, ça va m'amuser

Bon, maintenant que j'ai finis mon oeuvre de propagande voyons un peu à quoi ça peut servir de savoir coder (parce que c'est bien beau de devoir le faire, encore faut il que ça serve à quelque chose).

En gros, coder ça sert à automatiser les tâches relous et à traiter les gros volumes de données (ceux qu'on peut pas se coltiner à la main, à moins d'être prêt à y passer 2 ou 3 siècles).

Quelques exemples en vrac de choses que ça permet de faire :

- Regrouper et renommer automatiquement des fichiers (en faisant plein de tris par type de fichier, par date, etc...)
- Nettoyer des sets de données (parce que soyons franc, y'a souvent un paquet de merde dans nos fichiers excels)
- Croiser des données provenant de plusieurs fichiers différents
- Faire des jolis graphes
- Faire des calculs ou statistiques sur des très gros volume de données (y'a toujours un moment où Excel casse)
- Se la pêter devant ses collègues

Concrètement quand on code, on peut produire plusieurs sortes de choses :

- Des scripts : un script est un programme que tu lance, qui va effectuer une (ou plusieurs) actions définies puis terminer une fois qu'il a fini (un exemple simple de script, ça serait un programme que tu lance, qui supprime les fichiers qui traînent dans la corbeille puis qui se termine)
- Des logiciels : un logiciel, c'est un programme que tu lance, et qui ne termine pas tant que tu ne l'a pas fermé. Un logiciel propose (la plupart du temps) une ou plusieurs actions à l'utilisateur, que l'utilisateur peut effectuer autant de fois qu'il le désire
- Des bibliothèques : une bibliothèque c'est une collection d'actions qui peut être réutilisée dans d'autres programmes. C'est en utilisant des bibliothèques qu'on peut faire des programmes de plus en plus complexes sans avoir à tout écrire depuis le début

Pour faire court, quand on débute en informatique on fait en général des scripts, quand on en fait son métier on fait des logiciels, et quand on est très fort on fait des bibliothèques (en gros ceux qui codent des bibliothèques ce sont les informaticiens des autres informaticiens).

L'objectif de ce petit livre mon choupi n'est pas de faire de toi une ingénieure en développement logiciel (c'est pas ton métier, et à priori tu ne comptes pas

le devenir), mais de te donner assez de billes pour que tu puisse te faire des scripts, voir des petits logiciels. Et si tu te chauffe tu pourrais même te faire une petite librairie perso.

Pour ça je vais essayer de découper ce livre en petits chapitres rapides à lire, et d'éviter au maximum les informations dont tu n'aurais pas concrètement besoin (quitte à y revenir plus tard dans des chapitres plus avancés). En gros tu avoir une formation taillée sur mesure (petite veinarde!).

Sans plus attendre, attaquons dans le vif . . .

Chapitre 1

Théorie

Moi je veux aller vivre en théorie, parcequ'en théorie tout se passe bien

J'imagine que tu meurt d'impatience de te lancer dans le concret, de taper des lignes de codes sur un écran noir comme les hackers dans les films américain. Mais pour que tu comprenne un minimum ce que tu es en train de faire, on va pas y couper il te faut un poil de théorie. Après rassure toi, je vais condenser au maximum et ne garder que ce qui est absolument indispensable. Le but c'est pas d'y passer 3 mois non plus.

Ce chapitre va se découper en trois parties, une partie de maths (je vois d'ici ton visage réjoui à cette perspective), une partie d'informatique théorique et une dernière partie qui est un peu moins théorique et qui portera sur tout ce qui est environnement de développement.

Je préfère te prévenir tout de suite, je mettrai des exercices dans ces chapitres. Et oui, carrément des exercices que je te regarderais faire avec un air sévère (attention à la punition en cas de mauvaise réponse!). Plus sérieusement, ces exercices seront courts et faciles, le but étant que tu pratique les choses 2-3 fois afin de te faire la main. C'est en forgeant qu'on devient forgeron, toussa toussa ...

1.1 Les Maths

Avec un grand 'M'

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

1.1.1 Algèbre de Boole

Tu tires ou tu pointes ?

Algèbre de Boole ou calcul booléen. Introduction (le vrai, le faux, le 0, le 1). Les différents types d'opérateurs (avec des exemples pour chacun) + tables de verités. Quelques exemples plus complexes, comment tester toutes les possibilités. Exercices.

1.1.2 Modulo

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Table des matières

Introduction	i
1 Théorie	1
1.1 Les Maths	2
1.1.1 Algèbre de Boole	2
1.1.2 Modulo	2