

L'informatique ne se résume pas seulement à la programmation, loin de là. Personnellement, je la divise en quatre catégories pour une introduction :

1. Science des ordinateurs
2. Programmation
3. Mathématiques et intelligence artificielle
4. Outils informatiques et systèmes d'exploitation

Avant de commencer, un excellent premier exercice serait d'installer VSCode, Python, de configurer VSCode pour Python et d'exécuter le code suivant :

```
print("Hello, world !")
```

L'intérêt de cette démarche est de le faire sans avoir la moindre idée de comment procéder, en recherchant simplement sur Internet et en se débrouillant par soi-même.

Ensuite, un autre exercice intéressant serait d'installer WSL (Ubuntu), de télécharger Python et d'exécuter le même code dans le terminal. Parallèlement, il est judicieux d'apprendre les bases de l'utilisation du terminal. Rien de vraiment compliqué, juste regarder ces commandes, apprendre à les utiliser et en faire une habitude.

Un troisième exercice, toujours dans la même veine, serait d'installer Linux sur son ordinateur. Il est toutefois conseillé de faire au préalable une sauvegarde externe (en ligne ou sur un disque dur externe) de ses fichiers, car l'installation peut parfois mal se dérouler. Voici quelques distributions Linux recommandées pour une première installation :

- Linux Mint
- Ubuntu
- Manjaro (basée sur Arch, à réserver aux utilisateurs un peu plus à l'aise)
- Pop! OS
- ...

Personnellement, je recommande Mint avec l'environnement de bureau Cinnamon.

Il est crucial, en réalisant ces exercices, de comprendre chaque étape et non de simplement copier aveuglément ce que l'on trouve en ligne.

1 Science des ordinateurs

La science des ordinateurs consiste simplement à comprendre le fonctionnement d'un ordinateur. Qu'est-ce que le matériel, comment chaque composant fonctionne, qu'est-ce que la programmation...

Sans aucun doute, la meilleure introduction possible est la série Crash Course Computer Science.

Il est conseillé de prendre des notes, car tous les concepts ne sont pas nécessairement évidents.

Malheureusement, bien que cette série soit une excellente source d'information, elle ne peut pas être considérée comme un *cours* parfait, car elle ne propose pas d'exercices. Je suis d'avis que si l'on ne s'exerce pas, si l'on se contente "d'ingérer" l'information sans la "restituer", on n'apprend pas réellement. Je suggère donc de fournir le texte de la vidéo à ChatGPT, Claude (ou une autre IA) et de lui demander de générer quelques questions.

2 Programmation

La programmation est l'art d'écrire des programmes informatiques dans le but d'accomplir une tâche spécifique.

Le langage généralement recommandé pour débiter est **Python**, et ce pour plusieurs raisons : Python est très polyvalent, c'est l'un des langages les plus populaires, bénéficiant ainsi d'une multitude de ressources en ligne et de communautés pour obtenir de l'aide. De plus, Python possède une syntaxe extrêmement simple, comme on peut le constater dans l'exemple de code ci-dessus. Ce n'est pas le cas de tous les langages. Par exemple, en C, voici comment obtenir le même résultat :

```
#include <stdio.h>

main()
{
    printf("Hello, World !\n");
}
```

Pour Python, je recommande vivement le livre Think Python, qui est excellent et disponible gratuitement (et légalement). Cependant, si tu préfères commencer avec un autre langage, tu peux essayer :

- Java (Mouais)
- JavaScript (Si t'es une femme aesthetic mais alors il faut faire CSS et HTML avec)
- C (pour les bonom)
- Julia (pour les math) et Lua (pour faire le mec différent) ont des syntaxes abordables.
- ... en réalité, tous les langages fonctionnent

La programmation étant une compétence pratique, l'essentiel est de s'exercer. Le livre sur Python propose des exercices, fais-les !

Il est également important de savoir mener un projet de A à Z, ce qui diffère de la simple résolution d'exercices. Je considère qu'on maîtrise un langage de programmation après avoir

réalisé 5 projets (de complexité croissante).

Pour trouver des idées de projets, il suffit de chercher "projets Python" sur YouTube.

Les projets de groupe sont aussi très formateurs. Au début, cela peut sembler ardu, voire insurmontable. On peut alors commencer par regarder et reproduire des projets réalisés par d'autres sur YouTube (cherche "apprendre Python en un projet"), en portant attention à la façon de procéder et de structurer le code. Lire le code d'autres personnes (ayant fait ou non la même chose que nous) sur GitHub par exemple, est aussi très instructif et permet de découvrir différentes approches.

3 **Mathématiques et IA**

Bien que cela ne soit pas indispensable pour une introduction, les notions abordées dans le livre Basic Mathematics de Serge Lang me semblent être des prérequis nécessaires.

4 **Outils informatiques et systèmes d'exploitation**

Il s'agit ici d'apprendre le fonctionnement d'un système d'exploitation (UNIX) ainsi que d'autres outils informatiques (Tous les *gadgets* que tu utilises sur ton ordinateur).

Pour les systèmes d'exploitation, je conseille de : lire le chapitre "The Unix System" du livre *The C Programming Language* ([lien plus haut](#)) ainsi que le livre How Linux Works.

C'est aussi toujours intéressant d'apprendre à utiliser :

- VIM, un éditeur de texte permettant une navigation rapide
- Arch, une distribution Linux plus avancée
- Git et GitHub pour la gestion de versions de code
- Markdown et LaTeX pour la rédaction de documents

5 **Time line**

Proposition, à varier si nécessaire :

Encore une fois, le but est de rester constant, et de faire un peu tous les jours.

- Essaye d'abors d'installer vscode et python (comme expliquer tout au dessus) quand t'es déjà à l'aise avec l'utilisation de vscode, alors tu peux install WSL, puis quand t'es à l'aise avec le terminal alors t'install linux.
- Crash Course : 1 vidéo tous les deux jours, l'autre jours en faire un résumé au propre (profite en pour apprendre MarkDown et faire ton résumé avec, c'est super simple. Utilise mark down dans vscode). Note et resous aussi les quesitons.

- Essaye de Faire 1 chapitre par semaine du livre sur Python, tout en faisant les exercices. Sauvegarde ta resolution des exercices. Une fois que t'as fini le livre, trouve une video projet et fait la. Ensuite, soit t'as pleins d'idées de projet et alors fait toi un plan de comment y arriver, et fait le (très important de finir ses projet, faire que la moitier c'est inutile). Si t'as pas d'idée va voir sur yt
- Je dirais que 1 chapitre de math par semaine me semble faisable, surtout que le premier devraient aller très vite. très important aussi de faire les exercices dans le livre. (peut être pas tous parce que il y'en a vraiment beaucup mais au moins la pluspart.
- pour les autres trucs, c'est vraiment comme tu veux, si tu vois d'autres choses que tu veux apprendre vasy moi j'adore les trucs que je t'ai mis parce que je les utilise tout le temps, mais c'est pas necessaire (peut être latex l'est et github aussi)

6 Médias

- codebh
- AI Explained
- cocadmin
- Code Bullet
- Fireship
- hackintux
- James Bruton
- Micode
- Underscore_
- V2F
- Veritasium