

Synthèse du cours d'informatique

Nicolas ENGLEBERT

Janvier 2014

Table des matières

1	Introductioné	1
---	---------------	---

1 Introductioné

Cette introduction n'est pas une introduction au cours ni véritablement à la programmation, c'est plus une introduction de la synthèse ainsi que le recadrage de certains points à bien avoir en tête avant de pouvoir coder.

***tl;dr** Si Python ne fait pas ce que vous voulez, c'est que vous lui dites mal comment le faire.*

Tous les langages nécessitent d'être traduits en une série d'instructions directement compréhensibles par la machine. Selon les langages, cette traduction se fait une bonne fois pour toute avant l'exécution (on parle dans ce cas de compilateur comme par exemple pour le C ou C++), ou bien au fur et à mesure pendant l'exécution (on parle alors d'interpréteur, comme c'est le cas pour Python). Vu qu'on ne va parler ici que de Python, je vais désormais utiliser exclusivement le mot "interpréteur".

Si je vous parle de tout ça, ce n'est pas parce que je considère que c'est véritablement important ; c'est pour que vous n'en reteniez qu'une seule chose : il faut voir l'interpréteur comme un esclave très efficace mais qui ne comprend qu'une seule langue. Pourquoi cette comparaison ? Tout d'abord un esclave efficace car il va faire absolument tout ce que vous lui demandez, le seul problème étant qu'il parle une autre langue et qu'il faut donc lui dire correctement. C'est comme si vous avez l'impression de dire à votre esclave "passe l'aspirateur" mais qu'en réalité vous lui avez dit "fais le ménage". La nuance est subtile, parfois très difficile à détecter, mais le résultat peut être tout à fait différent (si par exemple il décide de laver votre beau tapis persan à l'eau de Javel). Avec Python c'est exactement la même chose ; pour qu'il fasse exactement ce que vous voulez, il faut lui dire dans sa langue, avec sa syntaxe, sinon le résultat peut différer du tout au tout.