

Python pour les incultes

Bodin Jules

8 avril 2025



Ce document a pour but d'enseigner les rudiments de Python. Python est muni d'une multitude de packages¹ qui ne sont pas à connaître dans le cadre d'une CPGE (cf. prepas.org). Cependant, un de ces packages est très utile dans un cadre d'analyse de données : **matplotlib** et sera surement utilisé en physique. Je recommande donc de s'y intéresser et une page à la fin de ce cours y sera dédiée pour aborder les bases de ce package (il est très conséquent).

Ce cours est construit pour être lu dans l'ordre des parties, mais il est aussi structuré pour permettre au lecteur de naviguer rapidement et simplement dans ce dernier (car le temps c'est des places en prépa). Il abordera toutes les notions nécessaires pour la prépa et plus encore, chaque point important étant accompagné d'un petit exemple. Attention cependant, ces exemples ne suffiront pas à l'apprentissage de Python, seule la pratique régulière permettra de bien comprendre les notions abordées.

Pour finir, le programme d'informatique pour tous aborde les notions de complexité des algorithmes étudiés. Ce cours donnera les complexités associées en les justifiant mais les preuves ne seront pas données, le but étant ici de comprendre la notion de complexité plus que de savoir la démontrer (les profs seront bien meilleurs à ce niveau).

Je rappelle qu'aucune matière ne doit être négligée en prépa, cependant il est plus intéressant d'insister sur celles ayant un plus gros coefficient. Ainsi, il est plus intelligent de bien gérer son temps (encore lui) et de ne pas chercher à maîtriser parfaitement ce cours durant les 2 années de prépa. Comme le disait si bien mon professeur de physique de première année : la progression dans une matière est similaire à celle d'un circuit RC, plus tu y passes de temps et moins l'évolution est grande (Je précise que c'est dans le cas d'une charge hein).

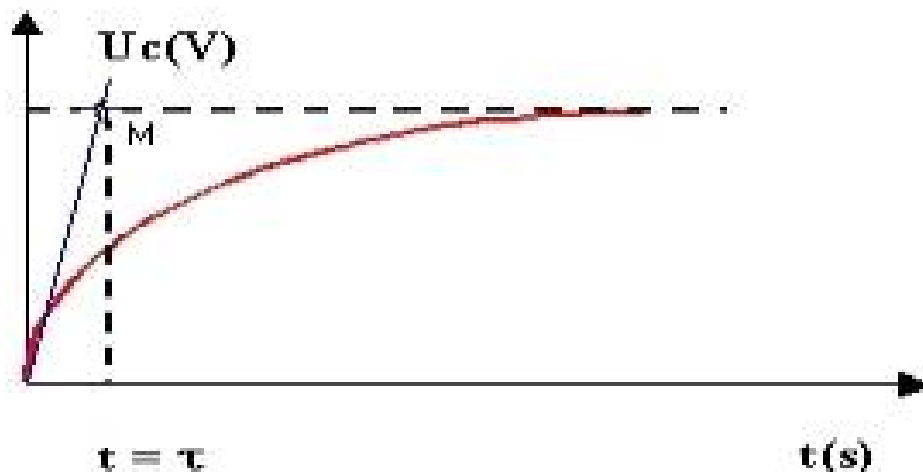


FIGURE 1 – Une charge de circuit RC au cas où le lecteur l'aurait oublié (pas bien ça)

Bonne lecture et bon courage, le chrono est déjà lancé.

1. Un package en Python, c'est juste un dossier qui contient plusieurs fichiers Python (appelés modules) et qui permet d'organiser le code.

Table des matières

1	Introduction	4
2	Les bases	5
2.1	Création et Exécution	5
2.2	Hello world	5
2.3	Les Variables	6
3	Corrections	7

1 Introduction

Python est un des langages de programmation les plus utilisés. C'est un langage de programmation interprété, multiparadigme et multiplateformes. Ses paradigmes sont : impératif, fonctionnel et orienté objet. Oui je sais, ça fait beaucoup de termes nouveaux pour un premier paragraphe d'introduction, mais il n'est pas nécessaire de tous les retenir tout de suite (c'est juste pour la culture).

On va se concentrer sur le terme le plus important : interprété. En informatique, il existe deux catégories principales de langages : les compilés et les interprétés, auxquels s'ajoutent les hybrides (JIT, bytecode + VM, ...). Pour faire court, un langage compilé nécessite une étape de compilation qui traduit le code source en code machine (code illisible à l'oeil nu mais compréhensible par la machine) alors qu'un langage interprété lit le code source et l'exécute. Le premier est plus rapide à l'exécution car directement traduit pour la machine, le second est plus flexible et plus simple à apprendre. Nous allons le voir, Python fait beaucoup (voir **BEAUCOUP**) de chose sans le montrer, dont une gestion automatique de la mémoire par ramasse-miettes. Pour finir, Python est doté d'un typage dynamique fort et d'un système de gestion d'exceptions.

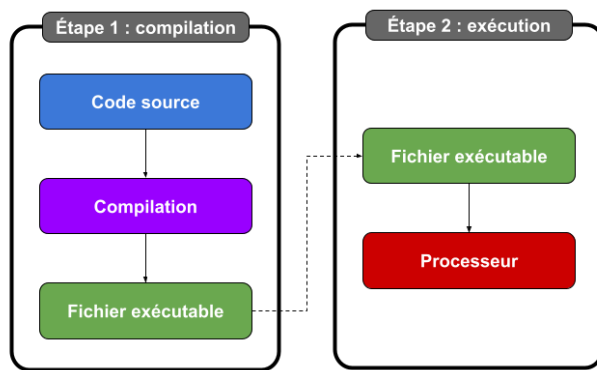


FIGURE 2 – Langage compilé

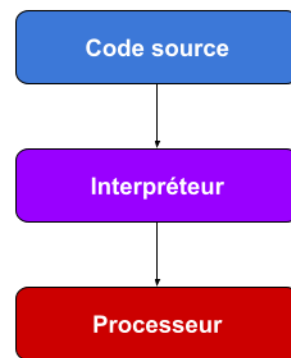


FIGURE 3 – Langage interprété

Cool maintenant on sait qu'on va pas trop galérer à apprendre Python. Il est donc temps de s'y mettre.

2 Les bases

On entre dans le vif du sujet, on va apprendre à coder en Python.

2.1 Création et Exécution

Sous linux, pour créer un programme Python, il faut déjà créer le fichier qui va accueillir son code source.

Pour cela, il suffit de se mettre dans le répertoire souhaité (avec *cd* c'est mieux et ça fait plus pro) et de créer un fichier avec une extension ".py". Voilà la commande à entrer si on veut créer un programme nommé toto :

```
touch toto.py
```

Mini Exercice 2.1 : Créer un programme tata.py dans un sous répertoire "Tutu/".

Indication : Regarder la commande `mkdir`

Correction ici.

On sait maintenant créer le fichier qui va accueillir le code source de notre programme. Cependant, il ne faut pas oublier que python est un langage interprété. Cela signifie qu'un fichier *.py n'est en fait qu'un fichier comprenant pleins de mots qui se suivent. Il faut que l'on indique à notre machine comment lire ce fichier. Pour cela il existe 2 méthodes :

- Le Shebang : Cette méthode consiste à ajouter une en-tête à notre fichier. Il faut écrire sur la **première** ligne.

```
#!/usr/bin/env python3
```

Qu'est ce que ce Shebang fait ? Il indique au système en quelle langue est écrit ce code source. Ici on lui indique que c'est en Python version 3.x. Deux choses à noter cependant :

1. Cette méthode ne fonctionne que sur système Unix
2. On peut seulement écrire python au lieu de python3 sur le Shebang, cependant si plusieurs versions de python sont installées sur votre ordinateur, on en sait pas vers quoi pointera le Shebang.

Avec cette méthode, il ne reste plus qu'à entrer le chemin vers le fichier dans le terminal pour l'exécuter.

- A la main : Cette méthode consiste à préciser directement dans le terminal l'interpréteur à utiliser. Ce qui donne dans le terminal :

```
python3 /chemin/vers/le/fichier.py
```

2.2 Hello world

Le programme le plus classique à faire en informatique est celui qui affiche hello world dans le terminal. Il est grand temps que vous vous lanciez, mais je ne vais pas vous jeter dans la jungle sans rien. En Python, la fonction qui permet d'afficher quelque chose dans le terminal est la fonction `print(*objects)`. Cette fonction est beaucoup plus complexe qu'elle ne le laisse paraître, mais nous verrons ça plus tard. Tout ce que vous devez retenir c'est que print affiche ce que vous mettez dans le champ *objects. De plus, en Python, le texte se met entre guillemets.

Mini Exercice 2.2 : Créer un programme hello.py qui affiche "Hello World!" dans le terminal.

Indication : Sans compter le Shebang, ce programme tiens sur une seule ligne!
Correction [ici](#).

2.3 Les Variables

3 Corrections

Correction 2.1 : Correction 2.2 :