

# Cours de Sciences Physiques

## Introduction au Langage Python

14 Décembre 2021

**Matériaux nécessaires: Un ordi avec une connection internet, une compte gmail.**

Python est un langage de programmation informatique. Il est extrêmement puissant comme langage et peut-être utilisé pour construire des algorithmes afin d'automatiser des tâches, exécuter des calculs scientifiques très complexes, modéliser des situations importantes comme la propagation d'un virus, intelligence artificielle, prédictions de climats etc...

Dans notre cours, nous allons apprendre à manipuler ce langage afin de résoudre des problèmes liés à ce que nous étudions en classe.

---

**Un algorithme** est un ensemble de règles ou d'opération à suivre afin d'effectuer une tâche ou de résoudre un problème donné. Par exemple, les étapes que vous suivez en classe de mathématiques pour résoudre une équation constitue un algorithme.

Exemple 1. Pour résoudre l'équation suivante:

$$4x + 6 = 20$$

les étapes:

- 1) On soustrait 6 des deux cotés
- 2) On multiplie par  $\frac{1}{4}$  des deux cotés

Est un algorithme simple qui nous permet de résoudre cette équation pour trouver:

$$x = \frac{7}{2}$$

Exemple 2. Une fonction est aussi un algorithme.

$$f(x) = 6x + 3$$

Cette fonction linéaire nous dit qu'à chaque valeur de  $x$  donné, on la multiplie par 6, et on ajoute 3 au résultat obtenu afin de trouver la valeur de  $f$  correspondante. C'est un algorithme qui nous dit comment trouver  $f$  pour une valeur de  $x$  spécifiée.

Mais nous pouvons aussi considérer un autre exemple:

Considérons une liste de nombres quelconques,  $l = [2,5,4,7,8,9,10,11,24,25,67]$ . Supposons qu'on nous demande de trouver seulement les chiffres/nombres pairs de cette liste.

Pour vous, ceci est simple. Vous n'avez qu'à trouver les chiffres/nombres divisibles par deux et les organiser dans une autre liste.

Exemple: Soit  $l_p$  la nouvelle liste des chiffres et nombres pairs, alors  $l_p = [2,4,8,10,24]$

Mais supposons que cette liste avait 10 millions de nombres. À ce moment, cela serait mieux si on pouvait demander à un ordinateur de le faire.

On peut aussi avoir une réaction chimique, et en connaissant la stœchiométrie de la réaction, nous pouvons demander à l'ordi de prédire quel serait le réactif limitant de la réaction en connaissant les quantités initiales en moles introduits dans le milieu réactionnel?

---

Pour que notre machine soit en mesure de résoudre les problèmes, il faut un langage de programmation où l'on écrit les commandes nécessaires. L'ordi ne comprend pas les commandes écrites. Alors certains langage ont besoin d'un **compilateur ou d'un interprète**, qui jouent le rôle de convertir les commandes écrites en signaux afin que le processeur de l'ordi puisse faire les calculs et qui reconvertissent les signaux à la fin du calcul en résultat que nous puissions comprendre. Python utilise un **interprète** et d'autre langage comme C ou C++ utilise des **compilateurs**. Nous n'établirons pas la différence ici.

---

Google nous permet de d'utiliser python avec un interprète déjà intégré. Nous n'avons donc rien à installer.

- 1) Aller sur [colab.research.google.com](https://colab.research.google.com)
- 2) Vous verrez un nouvel onglet (tab) ouvert. Ceci est un cahier ipython (juste une page où l'on peut programmer du python)
- 3) Cet onglet aura un nom "untitled.ipynb"
- 4) Changer le nom à "introduction.ipynb"

Afin de valider une opération en python il faut appuyer "shift+enter". Si l'on appui seulement "enter" on ira à la ligne.

### Declarations de variables:

- 1) Ecrivez: x et presser "shift+enter". Le programme lancera une erreur

```
-----  
-----  
NameError                                Traceback (most recent  
call last)  
<ipython-input-16-6fcf9dfbd479> in <module>()  
----> 1 x
```

**NameError: name 'x' is not defined**

car nous n'avons jamais dit au programme ce qu'était x.

2) Ecrivez  $x=5$ , ensuite presser "shift+enter". Maintenant le programme affichera "5". Nous avons ici **déclaré la variable x**.

3) Ecrivez `type(x)`. Le programme lancera "int" qui signifie entier (integer). La fonction `type()` nous dit de quelle variable il s'agit.

4) Ecrivez  $y=5.2$ , puis "shift+enter". Le programme affichera 5.2.

5) `type(y)`. Le programme lancera "float" qui signifie décimale.

6) Faites les opérations  $x+y$ ,  $y-x$ ,  $x**2$  (x au carré),  $y**2$ ,  $x/y$ , en donnant à chacune de ces variables un nom différent. Ensuite vérifier le type de ces variables (floats or integers)

7) Maintenant écrivez `%reset` afin d'effacer toutes les variables de votre cahier

8) Tapez  $x="s"$  (**avec les doubles guillemets**), pressez "shift+enter". Ensuite `print(x)`. Le programme affichera "s"

9) `type(x)`. Le programme affichera "String". En informatique les lettres ou mots sont des **string variables**.

10) Vous pouvez aussi essayer un mot. Ex:  $y="chimie"$ . Ensuite `print(y)`

## Les listes:

Les listes constituent un autre type de variables. Elles sont créées avec des crochets

Exemple:  $l = []$  est une liste vide. Tapez `type(l)` et le programme vous dira qu'il s'agit d'une liste.

Maintenant on peut mettre des éléments à l'intérieur de cette liste de deux manières:

- 1) En redéfinissant la liste:  $l = [5]$
- 2) Avec `.append()`:  $l.append(5)$  et ensuite on vérifie en tapant `print(l)` et le programme affichera [5]
- 3) Faites une liste nommé  $l$  contenant dix nombres quelconques.

## Condition (if statements):

Les conditions existent afin que notre algorithme puisse vérifier une condition, et décider quoi faire si cette condition est vraie ou fausse:

Un exemple: J'initialise une variable  $x=6$  et je veux que mon programme vérifie si ma variable est supérieur à zero, et si oui, de diviser ma variable par 3. Dans ce cas, j'utilise le *if statement*:

$x = 6$

*if*  $x > 0$  :

$y = x/3$

`print(y)`

Le code ci-dessus dit au programme de:

- 1) associer la valeur de 6 à la variable x (comme nous l'avons fait précédemment)
- 2) Vérifier que x soit positif, si oui, créer une nouvelle variable y qui aura comme valeur, x divisé par 3

3) Enfin imprimer la valeur de  $y$ .

**N.B:** Notez bien que le mot “if” est suivi de ce que l’on doit vérifier (si  $x > 0$ ), de deux points, et que ce que l’on doit faire est en dessous de la ligne “if” AVEC un alinéa pour dire que l’opération  $y = x/3$  est à l’intérieur de la condition. Ensuite remarquer aussi que le *print(y)* est en dehors de l’alinéa. Ce qui signifie que cette commande ne fait pas parti de la condition, la condition est rempli après avoir fait la division de  $x$  par 3.

Nous verrons en classe les exemples avec “if and” et “if else”

### Iteration (for loop):

Les iterations sont des processus importants en informatique, et python est particulièrement efficace dans ce type d’opération. Par itérations nous voyons aussi une répétition: Nous pouvons demandé au programme d’effectuer une operation à chaque fois qu’il compte une variable “string” dans le mot “chimie” par exemple. Alors, nous savons que le programme va effectuer cette opération (que l’on spécifiera) six fois, car il y a 6 “string variables” dans *chimie*. Nous appellerons le mot “chimie” une variable *iterable* (en anglais), car on peut compter les constituants de ce mot. Une liste est également une *iterable*.

Exemple

```
s = " chimie"
```

```
for i in s :
```

```
    print(i)
```

```
c
```

```
h
```

```
i
```

```
m
```

```
i
```

```
e
```

Le code ci-dessus dit au programme de:

- 1) Associer la valeur de “chimie” à la variable  $s$  (comme nous l’avons fait précédemment)
- 2) Parcourir la variable *iterable*  $s$ , et associer chaque élément de cette variable à  $i$  et imprimer la valeur de  $i$ . Le programme répétera cette procédure aussi longtemps qu’il y aura des “string variables” dans le mot. C’est pour cela qu’il s’arrête après  $e$ .

**N.B:** Notez bien que le mot “for” est suivi de la variable *iterable* que l’on doit examiner, de deux points, et que ce que l’on doit faire est en dessous de la ligne “for” AVEC un alinéa pour dire que l’opération *print(i)* est à l’intérieur de la boucle.

En classe on fera des exemples sur les listes également.