

*1<sup>ère</sup> NSI – Thème « #3 »*

# Introduction à la Programmation *(en Python)*

*Marc Biver, Septembre 2023*

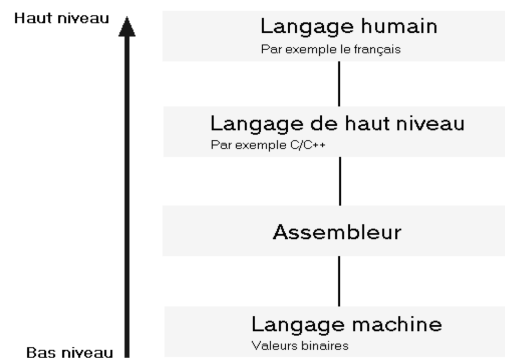
*Lycée Fustel de Coulanges, Massy*

---

## Qu'est-ce qu'un langage de programmation ?

---

- Programmer = demander à un ordinateur d'effectuer des actions.
- Exemple : « calcule-moi la valeur de  $6 + 7$  ».
- Problème : un ordinateur ne « parle » que le binaire ; nous, les humains, pas du tout...
- Solution : on a créé des langages dits « évolués » ou « haut niveau » qui nous permettent d'exprimer de manière naturelle ce que l'on veut que l'ordinateur réalise.



- Un langage de programmation...
  - Comporte une syntaxe, du vocabulaire, une grammaire... comme une langue humaine.
  - A l'inverse d'une langue humaine ils n'ont qu'une vocation très étroite : écrire des algorithmes de manière à ce qu'un ordinateur puisse les exécuter.
  - La « traduction » vers le langage machine que comprend l'ordinateur peut se faire de deux manières :
    - La compilation : l'équivalent de la traduction d'un livre – cela produit un nouveau fichier, appelé « exécutable » qui peut être directement exécuté par l'ordinateur (comme un livre traduit peut être directement lu).

- L'interprétation : comme « dans la vraie vie » c'est une traduction au fur et à mesure que le code est écrit, sans enregistrement du code traduit dans un nouveau fichier.
- Exemples :
  - Le langage C utilise un compilateur – on dit que c'est un langage compilé.
  - Le langage Python utilise un interpréteur – on dit que c'est un langage interprété.

---

## Le langage Python

---

### Pourquoi l'avoir choisi ?

- A notre niveau, pour commencer :
  - Il est « open source » - gratuit et utilisable par tous.
  - Il est « portable » - utilisable sous Linux, MacOS, Windows.
  - Il a une syntaxe simple (si ! si !).
- Il a d'autres caractéristiques dont nous parlerons quand nous aborderons les thèmes qui y correspondent :
  - Il gère lui-même ses ressources mémoire.
  - Il est multi-paradigmes : impératif, fonctionnel, orienté objet.

### Comment l'utiliser ?

On le comprend – taper du code dans un fichier texte ne suffit pas à créer un programme (même dans le cas du HTML qu'on a vu la semaine passée, il y avait un interpréteur – le navigateur).

On va utiliser ce qu'on appelle un IDE (« *integrated development environment* ») – il en existe littéralement des dizaines. On peut citer, pour les plus proches de nous :


- IDLE – qui est installé sur les machines NSI et que l'on va utiliser pour découvrir la console ;
- Basthon – outil en ligne sur lequel s'appuie l'application Capytale que nous allons utiliser bientôt également.
- EduPython – qui est installé sur les machines qui vous ont été fournies par la Région Ile-de-France ;
- Google Decal – outil en ligne que l'on utilisera sans doute plus tard dans l'année ;

---

## Initiation – activités dans la console

---

### Opérations de base

- Lancer IDLE () .
- Testez dans la console les opérations usuelles : (+ - \* /):

```
Python 3.6.1 Shell
Type "help", "copyright", "credits" or "license()" for more information
>>> 3+2
5
>>> 9/5
1.8
>>> |
```

- Testez les opérations : //    %    \*\*    Devinez-vous ce qu'elles font ?

## Entrées – Sorties / Affectations

Une variable est une manière de stocker une information dans la machine en la nommant.

Faire « l'affectation » d'une variable c'est en modifier la valeur – soit en la créant si elle n'existe pas encore, soit en la changeant.

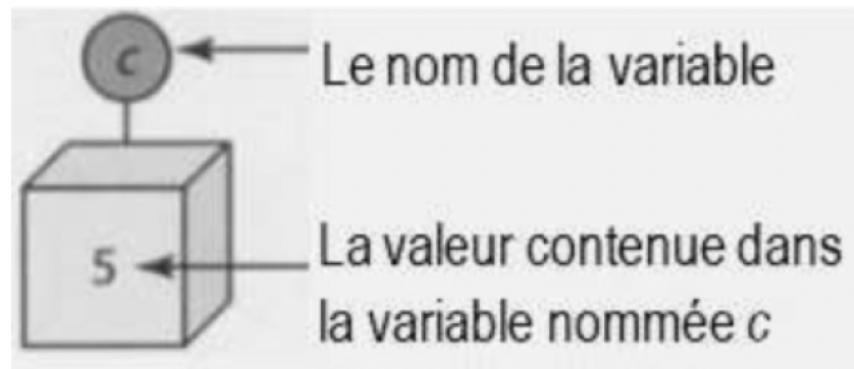
Considérons l'algorithme suivant :

1. **Saisir** a
  2. **Saisir** b
  3.  $c = a + 2 \times b$
  4.  $c = c + 1$
  5. **Ecrire** c
- 

On a ici :

- Trois variables ;
- Deux entrées ;
- Quatre affectations (deux sur des valeurs demandées à l'utilisateur, deux calculées par l'ordinateur) ;
- Une sortie, par écriture à l'écran.

Une variable est caractérisée en premier lieu par un nom (qui permet de la localiser en mémoire) et une valeur ; elle peut être vue comme une boîte étiquetée (son nom) qui peut contenir différentes informations – dans le cas précédent, des nombres entiers. Affecter une variable, c'est modifier ce que contient la boîte.



Une variable a également un type – nombre entier, nombre réel, chaîne de caractères... – mais à l'inverse d'autres langages, il est *implicite* : il n'y a pas besoin de le déclarer.

Pour affecter une valeur à une variable en Python, on utilise le symbole =, le membre de gauche prenant la valeur de celui de droite :

```
# La ligne suivante affecte la valeur 3 à 'a'
a = 3
```

(on notera ici la syntaxe des commentaires en Python)

ATTENTION à ne pas confondre avec les maths :

- « Variable » n'a pas le même sens ;
- Le symbole « = » n'a pas le même sens.

Pour nommer une variable, il faut respecter certaines règles : caractères alphanumériques, ne pas commencer par un chiffre, aucun caractère spécial hormis l'underscore « \_ » (et on évite les accents), et ne doit pas être un mot clé – int, and, break... (on y reviendra)

- Créez plusieurs variables et affectez-y différentes valeurs, cette fois de différents types (entiers, réels, texte...). (vous pouvez utiliser les exemples ci-dessous – mais inventez-en d'autres !)

```
marc = 5
Fustel = 6.578
accueil = "bonjour"
```

- Affichez la valeur de ces variables au moyen de la fonction

```
print()
```

- Demandez à l'utilisateur de renseigner certaines variables au moyen de la fonction input :

```
>>> age_prof = input("Quel age as-tu? ")
Quel age as-tu? |
```

- Vérifiez le type des variables que vous avez créé au moyen de la commande :

```
print(type(NOM_VARIABLE))
```

- On rappelle les opérations usuelles :

Instructions	Signification
$a + b$	addition
$a - b$	soustraction
$a * b$	multiplication
$a ** b$	puissance $a^b$
$a / b$	division
$a // b$	quotient de la division euclidienne de a par b
$a \% b$	reste de la division euclidienne de a par b

- Faites des tests d'opérations usuelles avec ces différentes variables – rencontrez-vous des erreurs ? Lesquelles ?

### Exercices :

On considère le programme Python suivant :

```
a = 7
b = 3
c = a + b
c = c + 1
b = a - c
```

Sur une feuille copiez le tableau suivant et complétez-le :

Contenu des variables	a	b	c
Ligne 1			
Ligne 2			
Ligne 3			
Ligne 4			
Ligne 5			

---

➔ QCM « Affectation de variables » sur ProNote.



---

## Scripts

---

Un script n'est rien de plus qu'une succession d'instructions qui sont exécutées à la suite – dans une console chaque instruction est exécutée lorsque vous passez à la ligne suivante tandis que dans un script elles le sont toutes sans que vous n'ayez rien à faire pour passer de l'une à la suivante.

Un script peut être enregistré pour être exécuté plusieurs fois ou dans plusieurs contextes – c'est, au sens où il contient une suite d'instructions mettant en œuvre un algorithme, réellement un « programme informatique ».

- Depuis votre ENT, lancez l'application « Capytale ».
- Avant de vous lancer n'oubliez pas qu'avec Python on peut convertir (quand c'est possible) des variables d'un type à un autre :

```
# Mettre un nombre au format chaîne
a = 1984
b = str(a)
c = "Roman d'Orwell: " + b
# Ou l'inverse...
a = input("Quel est votre taille? ")
b = float(a)
print(b + 0.2)
```

- Rejoignez l'activité que j'ai préparée pour la séance avec le code :

```
d00f-1846631
```